Optical Topology Programming:
Foundations, Measurements, and Applications

by

Matthew Henry Hall

A dissertation accepted and approved in partial fulfillment of the

requirements for the degree of

Doctor of Philosophy

in Computer Science

Dissertation Committee:

Ramakrishnan Durairajan, Chair

Reza Rejaie, Core Member

Joseph Sventek, Core Member

Brian J. Smith, Institutional Representative

University of Oregon

Winter 2024

© 2024 Matthew Henry Hall
This work is openly licensed via Creative Commons
Attribution-NonCommercial 4.0 International.

DISSERTATION ABSTRACT

Matthew Henry Hall

Doctor of Philosophy in Computer Science

Title: Optical Topology Programming: Foundations, Measurements, and Applications

This thesis advances the state-of-the-art in network management by challenging the prevailing notion that the joint optimization of optical and packet layers is currently impractical. It does so through two key contributions: (1) establishing the theoretical and empirical foundations for programming the optical topology, henceforth referred to as optical topology programming; and (2) demonstrating the advantages of optical topology programming in enhancing network security (e.g., combating network reconnaissance, volumetric DDoS) and network management (e.g., scaling traffic engineering) applications.

We evaluate the performance of optical topology programming for these applications with a custom-built discrete event simulator. We demonstrate the ability of optical topology programming to improve scalability in traffic engineering systems, completely removing all instances of throughput loss for a diverse set of link failure and flash crowd events. We show that it is also capable of subverting attempts at network reconnaissance by dynamically changing the set of active network links and finding hundreds of alternative topology configurations that maintain traffic performance in seconds. Finally, we show that optical topology programming can improve defense capabilities against large-scale link flood attacks, reducing the number of successful link flood attacks from 134 to 9 (94%).

This dissertation includes previously published and unpublished coauthored material.

CURRICULUM VITAE

NAME OF AUTHOR: Matthew Henry Hall

GRADUATE AND UNDERGRADUATE SCHOOLS ATTENDED:

University of Oregon, Eugene, OR, USA
California Polytechnic University Humboldt, CA, USA

DEGREES AWARDED:

Doctor of Philosophy, Computer Science, 2024, University of Oregon
Master of Science, Computer Science, 2022, University of Oregon
Bachelor of Science, Computer Science, 2016, California Polytechnic University Humboldt

AREAS OF SPECIAL INTEREST:

Software Defined Networking
Optics

PROFESSIONAL EXPERIENCE:

Graduate Employee - Researcher, Wrote simulation software and numerical optimization models to design and run experiments prototyping reconfigurable topology applications for DDoS defense and traffic engineering, University of Oregon, June 2019–March 2024

Intern - Smart Optical Fabric & Devices Lab, Contributed to research on stream processing of optical network telemetry data with machine learning. Designed and built an anomaly detection method using statistical methods in Python, Nokia Bell Labs, Summer 2020

Graduate Employee - Teacher, Taught lab sessions in Computer Science courses: Python, Data Structure, Networking Fundamentals, and Operating Systems. Designed hands-on exercises and projects for students in large classes (120 students), University of Oregon, January 2018–June 2019

GRANTS, AWARDS AND HONORS:

University of Oregon Doctoral Research Fellowship, 2022
Bell Labs Summer Research Award for Distinguished Innovation, 2020
Ripple Cyber-security Fellowship, 2019
Erwin & Gertrude Juilfs Scholarship in Computer and Information Science, 2019

PUBLICATIONS:

Note: Publications that are used in this thesis appear in **bold**.

M. Nance-Hall, L. Salamatian, and R. Durairajan, "From Fibers to Fortresses: Combating Modern Reconnaissance via Optical Topology Programming", (In submission) pp. 1–12, 2024.

M. Nance-Hall, Z. Liu, V. Sekar, and R. Durairajan, "Analyzing the benefits of optical topology programming for mitigating link-flood ddos attacks", (To appear) Transactions on Dependable and Secure Computing, pp. 1–17, 2024.

M. Nance-Hall, K.-T. Foerster, P. Barford, and R. Durairajan, "Improving scalability in traffic engineering via optical topology programming", in Transactions on Network and Service Management (TNSM), IEEE, 2023, pp. 1–21.

J. E. Simsarian, G. Hosangadi, W. Van Raemdonck, J. Gripp, M. Nance-Hall, J. Yu, and T. Sizer, "Demonstration of cloud-based streaming telemetry processing for optical network monitoring", in 2021 European Conference on Optical Communication (ECOC), 2021, pp. 1–4.

M. Nance-Hall, P. Barford, K.-T. Foerster, M. Ghobadi, W. Jensen, and R. Durairajan, "Are wans ready for optical topology programming?", in Proceedings of the ACM SIGCOMM 2021 Workshop on Optical Systems, ser. OptSys '21, Virtual Event, USA: Association for Computing Machinery, 2021, pp. 28–33, isbn: 9781450386500.

M. Nance-Hall, K.-T. Foerster, S. Schmid, and R. Durairajan, "A Survey of Reconfigurable Optical Networks", Optical Switching and Networking, vol. 41, 2021.

J. E. Simsarian, M. Nance-Hall, G. Hosangadi, J. Gripp, W. van Raemdonck, J. Yu, and T. Sizer, "Stream Processing for Optical Network Monitoring with Streaming Telemetry and Video Analytics", in European Conference on Optical Communications (ECOC), Virtual Event, Belgium: IEEE, Dec. 2020.

M. Nance-Hall, G. Liu, R. Durairajan, and V. Sekar, "Fighting Fire with Light: Tackling Extreme Terabit DDoS Using Programmable Optics", in Proceedings of the Workshop on Secure Programmable Network Infrastructure (SPIN), Virtual Event, New York, USA: ACM, Aug. 2020.

S. K. Mani, M. Nance-Hall, R. Durairajan, and P. Barford, "Characteristics of Metro Fiber Deployments in the US", in Proceedings of the Network Traffic Measurement and Analysis Conference (TMA), Virtual Event, Germany, Jun. 2020.

M. Nance-Hall and R. Durairajan, "Bridging the optical-packet network chasm via secure enclaves (extended abstract)", in Proceedings of the Workshop on Optical Systems Design, ser. OptSys '20, Virtual Event, USA: Association for Computing Machinery, 2020.

M. Nance-Hall, J. Sommers, and R. Durairajan, "A compressed sensing approach to taming the internet measurement data deluge (poster)", in ACM Internet Measurement Conference, ser. IMC '18, Boston, MA: Association for Computing Machinery, 2018.

M. Nance-Hall, V. Chidambaram, and R. Durairajan, "Vfiber: Virtualizing unused optical fibers (extended abstract)", in USENIX Networked Systems Design and Implementation, ser. NSDI '18, Renton, WA: USENIX, 2018.

M. Nance-Hall, C. Robins, K. Owens, J. Nowatzke, T. Lauck, and L. E. Smith, "High performance supercomputing on a budget", J. Comput. Sci. Coll., vol. 32, no. 4, pp. 86–92, Apr. 2017, issn: 1937-4771.

ACKNOWLEDGEMENTS

I'm fortunate to have many amazing people to whom I'd like to express my gratitude. Without the unwavering support and unconditional love given and expressed by these individuals I would not be writing this text as I write it now.

I thank my wife, Chloe, who has been with me through all of it. Through years without vacations and years with trips to places we wish we'd never had to go. Through tremendous joys and hard times too. Weddings and baby showers—hospitals and funerals. We've been through it all and I have no doubt I'm only here because I had her companionship all along the way.

I thank my mom and dad, who taught me that love is worth more than anything and that it's not what happens to us that must define us, but how we respond to the challenges that life presents.

I thank my siblings, Janel, Madison, and Dallas, Carina and Bianca, who have been a welcome source of levity on this long and arduous road.

I thank my advisor, Ram, who took a chance on me his first year here and went on to teach more than I could have imagined. For teaching me that research is more about asking the right question than having all the answers. More importantly, beyond any professional skills, he taught me about patience and kindness and how to make someone feel like they're a valued member of the team. He taught me how to take rejection in stride and how to use it as fuel to propel myself forward and to stay focused on the big picture. Continuous improvement.

I thank the members of this dissertation advisory council, Reza Rejaie, Joe Sventek, and Brian Smith. I am grateful for their attention, curiosity, and time

throughout my graduate career. The insights I've gathered from them have added quality to this work.

I thank Lei Jiao, who offered constructive feedback throughout my graduate career and helped me formalize the optimization method used in this work.

I would also like to thank the members of the community who collaborated with me on this work, Alan Zaoxing Liu, Bill Jensen, Chris Misa, Guyue (Grace) Liu, Joel Sommers, Klaus-Tycho Foerster, Loqman Salamatian, Manya Ghobadi, Paul Barford, Sathiya Kumaran Mani, Stefan Schmid, Vijay Chidambaram, Vyas Sekar, and Walter Willinger. Although not everything that we ever worked on made it into this thesis or beyond Reviewer 2, they have all been a part of this at one time or another. Our discussions, whether they were early on in the process or much later, helped inevitably shape the form of this document.

I thank Hank Childs, whom I emailed in 2017 to ask about pursuing a PhD at the UO after another appointment didn't work out. He invited me to graduate orientation that fall and introduced me to Ram. The rest is ~~history~~ this thesis.

Finally, I would like to thank the funding agencies that contributed financially to my education and allowed me the opportunity to pursue this line of research. The findings, views, and opinions in this document are mine alone, and do not reflect any official endorsement, view, or opinion of these agencies: The National Science Foundation (CNS-2212590, CNS-2145813, CNS-1703592, CNS-2039146 and SaTC-2132651), The University of Oregon (Doctoral Research Fellowship), Ripple Labs, Inc. (Cyber Security Fellowship).

To my dad, Kenneth Alan Hall, who always encouraged me to believe in myself and to never give up. July 1, 1963 — October 11, 2022

TABLE OF CONTENTS

Chapter Page

Chapter Page

LIST OF FIGURES

Figure Page

Figure Page

LIST OF TABLES

Table Page

CHAPTER I

INTRODUCTION

Wide-area networks in the Internet are more integrated with our lives today than ever before. Applications such as web search, GPS navigation, video streaming, ride-hailing, food delivery, telehealth, and conference video calls for remote work and learning, among many others, are familiar and used by billions of people a day. All of these services generate massive amounts of data constantly (2.5 quintillion bytes daily as of 2020 [70]), thus putting strain on the wide-area networks underneath it all. Special attention has been given to the task of optimizing wide-area network performance in order to make all of these applications run smoothly and seamlessly for people every day [2, 130, 166].

While traffic demands on wide-area networks from everyday user applications have risen, cybercriminals have found it lucrative to disrupt networked services through various attacks. For example, Microsoft's private cloud network, Azure, was attacked with a 3.25 terabit per second attack in May 2022. This was the largest recorded attack against Azure's infrastructure to date [266]. In June 2022 Cloudflare, a content delivery network hosting hundreds of thousands of websites on the Internet was hit with its largest attack recorded—26 million HTTPS requests per second [311]. These recent attacks target infrastructure, overwhelming network capacity and causing disruption to services for all users whose applications use the targeted network [202].

Recently, programmable networks have emerged in response to our insatiable hunger for bandwidth and to mitigate the rising security threats that networks face [203]. Programmability in the network has been applied to load balancers [219], intrusion detection systems [173], network interface cards [95], and software-defined networking controllers [2, 130, 166]. Some programmable network applications,

such as traffic-engineering, have primarily been applied to private content provider networks [2,130,166] (e.g., Google, Amazon, Azure, etc.) while others, such as defense against distributed denial of service attacks have been applied to both private [86], and public [181], networks (where public networks are those which users connect to via their Internet service provider). Programmability has been necessary to scale networks into the behemoths that underlie all of the familiar applications people can enjoy today while adding flexibility to services and securing them from malicious actors. Yet, as demand continues to grow, the programmable solutions that research and industry have given us will only enable networks to scale so far.

At the bottom of everything related to wide-area networks is the physical (optical) layer and this layer is unfortunately not programmable. The optical layer of the networking stack has evolved independently of the innovations made at the higher layers [212]. This domain has been the subject of research efforts that have largely revolved around the task of sending the largest quantity of data possible into one end of an optical fiber as fast as possible and recovering and decoding the transmission error-free at the other end just as fast as it can be transmitted [143]. A detailed look at optical layer advancements and their limited application to higher-layer protocols is given in our survey on reconfigurable optical networks [215].

1.1 Challenges in Optical Topology Programming (OTP)

Optical topology programming has been shown to increase the efficiency of intra-datacenter networks [100] but has not yet been used to great effect in wide-area networks. Its application to wide-area networks has been prolonged due to three core challenges outlined below. We aim to address these three challenges in this thesis.

1.1.1 Foundations. Jointly programming the optical and networking layers is NP-hard [160] and this fact introduces a fundamental challenge to realizing

programmability for the optical layer at an enterprise network scale. We find that a key driver for the computational complexity is enumerating all possible network paths in the presence or absence of any combination of network links and calculating the flow distribution among those paths. We present models to address this challenge heuristically in Chapter IV, where a key insight is that we can dramatically reduce the number of potential paths while still finding feasible solutions for topology and routing. This approach is applied in application-specific scenarios, where the set of paths input to the framework reflects the needs of the application.

Evaluating optical topology programming requires access to a wide-area network backbone which we do not have. To address this challenge, we have constructed a Python-based discrete event simulator. While network simulators for traffic engineering and security applications in recent work [2, 165, 301] have taken topology as a fixed input to show how routing decisions affect performance as a function of traffic, our simulator aims to show how topology *and* routing decisions affect performance as a function of traffic.

1.1.2 Measurements. While we can model the benefits of optical topology programming numerically, it is just as important to understand the physical requirements for adding and removing optical signals in a span of optical fiber. To this end, we take a measurement approach to quantify the reconfiguration time to establish an optical circuit between two ends of a long-haul link traversing several amplifiers. Noting that this operation has historically taken several to tens of minutes, we dig into the cause of the delay and demonstrate that the same feat can be achieved in under one second. This study, presented in Chapter V has been featured in two of our publications [210, 211].

1.1.3 Applications. The foundations established and measurements gathered motivate the development of three novel network applications to harness optical topology programming. In this work, we describe such applications: traffic engineering, network reconnaissance defense, and defense against link-flood distributed denial of service attacks (link-flood DDoS attack or LFA).

1.1.3.1 Traffic Engineering. Chapter VI presents a novel framework, GreyLambda, to improve the scalability of traffic engineering systems. Traffic engineering systems continuously monitor traffic and allocate network resources based on observed demands. The temporal requirement for these systems is to have a time-to-solution in 5 minutes or less. Additionally, traffic allocations have a spatial requirement, which is to enable all traffic flows to traverse the network without encountering an over-subscribed link. However, the multi-commodity flow-based traffic engineering formulation cannot scale with increasing network sizes. Recent approaches have relaxed multi-commodity flow constraints to meet the temporal requirement but fail to satisfy the spatial requirement of traffic engineering systems due to changing traffic demands, resulting in oversubscribed links or infeasible solutions [2, 166].

To satisfy both these requirements, we utilize optical topology programming to rapidly reconfigure optical wavelengths in critical network paths and provide localized bandwidth scaling and new paths for traffic forwarding. GreyLambda integrates optical topology programming into traffic engineering systems by introducing a heuristic algorithm that capitalizes on latent hardware resources at high-degree nodes to offer bandwidth scaling, and a method to reduce optical path reconfiguration latencies. Our experiments show that GreyLambda enhances the performance of two state-of-the-art traffic engineering systems, SMORE [166] and NCFlow [2] in

real-world topologies with challenging traffic and link failure scenarios. This work was published in [210].

1.1.3.2 Network Reconnaissance. Successful network reconnaissance is a prerequisite task for deploying an LFA [149, 260]. Recent efforts [153, 196] have suggested that network obfuscation may be a viable technique to thwart would-be attackers from discovering the network topology, thereby preventing link-flood attacks. Equalnet [153] creates virtual link interfaces for nodes and links that do not physically exist; these virtual nodes and links are made to appear as if they are in the network when an attacker launches active measurement probes into the network. However, the obfuscation techniques shown in [153] may be reversible via its method for generating randomized IP addresses for virtual nodes and link interfaces.

In Chapter VII we show how internet tomography and out-of-band measurements, can glean topology information for several real-world enterprise networks. We then propose a *topology jitter* method, which we call Doppler, to deter attackers. This work is currently in submission.

1.1.3.3 Link Flood Attacks. An LFA overwhelms bandwidth on links in a network using traffic from many sources, indistinguishable from benign traffic. Unfortunately, traditional DDoS defenses [86] are incapable of stopping such attacks and recently proposed software-defined solutions [148, 301] are shown in this work to be situationally ineffective.

We observe a new opportunity for mitigating link-flood attacks using optical topology programming. In essence, we envision new capabilities to scale capacity on-demand to avoid congestion and add new links to the network to create new paths for traffic during link-flood attack incidents. Realizing the benefits of optical topology programming raises unique challenges; the search space for candidate topology

configurations is very large and joint optimization of topology and routing is NP-hard [160].

Chapter VIII presents ONSET—a framework that tackles these challenges to lay a practical foundation for topology programming-based defenses against link-flood attacks. We show that ONSET complements existing programmable network defenses and amplifies their benefits. We perform a *what-if* style analysis of ONSET by simulating a wide-ranging set of attacks, including terabit-scale attacks, against every single link on five networks with two different routing capabilities and observe that ONSET provides the means to mitigate congestion loss in more than 90% of the hundreds of diverse attack scenarios considered.

1.2 Scope and Contribution

> This thesis advances the state-of-the-art in network management by challenging the prevailing notion that the joint optimization of optical and packet layers is currently impractical. It does so through two key contributions: (1) establishing the theoretical and empirical foundations for programming the optical topology, henceforth referred to as optical topology programming; and (2) demonstrating the advantages of optical topology programming in enhancing network security (e.g., combating network reconnaissance, volumetric DDoS) and network management (e.g., scaling traffic engineering) applications.

The methods and applications developed in this thesis are intended for private enterprise backbone network environments and have not been evaluated for inter-domain routing settings. The scale of networks is not a significantly limiting factor regarding the performance benefits of our OTP applications. In total, these

applications have been evaluated on 14 networks, with up to 149 nodes and distances (e.g., fiber miles) between network endpoints up to 3840 km (2386 mi) long.

1.3 Attribution of Coauthored Material

Ramakrishnan Durairajan contributed to all chapters of this dissertation in an advisory capacity and is a coauthor on all previously published and unpublished works referenced herein.

Chapters II and III contain previously published coauthored material from [214] and coauthored by Klaus-Tycho Foerster and Stefan Schmid.

Chapter IV, § 4.2 contains previously published coauthored material from [210], with coauthors Paul Barford and Klaus-Tycho Foerster. Chapter IV, § 4.3 contains previously unpublished work that is scheduled to appear in [217] and coauthored with Zaoxing (Alan) Liu and Vyas Sekar.

Chapter V contains previously published coauthored material from [211], coauthored with Paul Barford, Klaus-Tycho Foerster, Manya Ghobadi, and William Jensen.

Chapter VI contains previously published coauthored material from [210], with coauthors Paul Barford and Klaus-Tycho Foerster.

Chapter VII contains previously unpublished coauthored material this is in submission [218]. This work is coauthored with Loqman Salamatian.

Chapter VIII contains previously unpublished coauthored material that is scheduled to appear in [217], with coauthors Zaoxing (Alan) Liu and Vyas Sekar.

Chapter IX does not contain any previously published or unpublished work.

CHAPTER II

PRIMER ON OPTICAL NETWORKS

This chapter contains previously published coauthored material from [214] and coauthored by Klaus-Tycho Foerster, Stefan Schmid, and Ramakrishnan Durairajan. The sections of [214] that appear here were written entirely by the dissertation author. The coauthors assisted in editing these sections.

This chapter introduces fundamental concepts regarding optical networking. We introduce the network architecture models, or design patterns, that are repeated across different optical networks. We then describe the optical networking hardware commonly deployed today.

2.1 Network Architectures

In this section, we briefly discuss two network architecture models that can leverage reconfigurable optics, IP-over-OTN networks and hybrid electric-optical data center networks. Our focus in this survey is to highlight and categorize reconfigurable optical networks in enterprise networks, and therefore leave last-mile optical networks, such as passive-optical networks and fiber-to-the-home networks beyond the scope of our discussion.

We also briefly outline principles leveraged in different contexts by reconfigurable optical networks, software defined networking and elastic optical networking. This discussion introduces key aspects for network designers to consider when building a reconfigurable optical network. This discussion reinforces our illustration of how full-stack perspective aids in the network design process. In sections 3.2 and 3.3, we look at specific implementations of reconfigurable optical networks in more detail.

2.1.1 IP-over-Optical Transport Networks. IP-over-Optical Transport Networks (IP-over-OTN), defined in ITU-T G.709, is the standardized protocol that

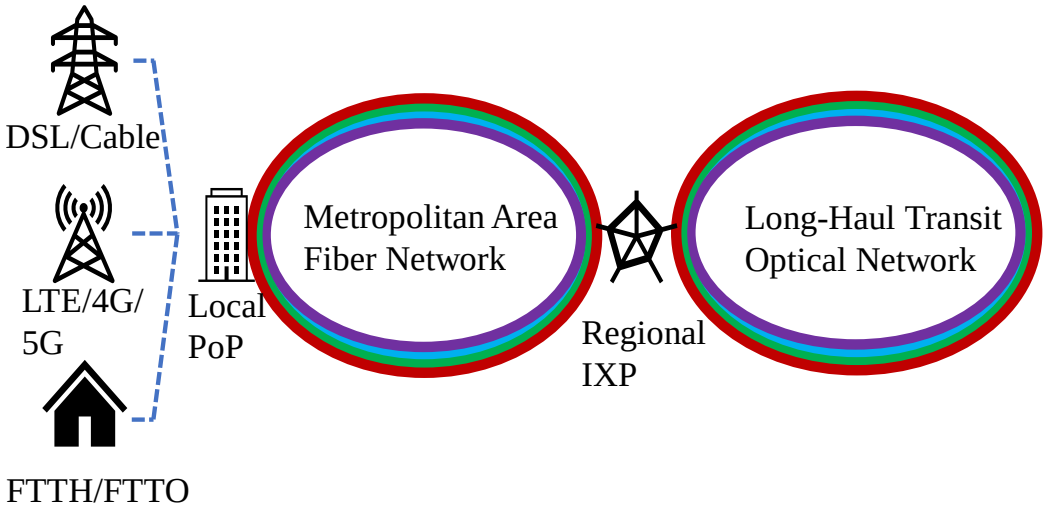


Figure 1. Metro, regional, and long-haul networks are connected by the IP-over-OTN standard.

links metro, regional, and long-haul networks, as illustrated in Figure 1. Thus, we discuss IP-over-OTN when referring to the network's IP and the optical layers. In IP-over-OTN, hosts (e.g.data centers, points-of-presence or PoPs, servers, etc.) connect to routers, and these routers are connected through the optical transport network (OTN). A node in the optical layer is an Optical Cross-Connect (OXC). An OXC transmits data on modulated light through the optical fiber. The modulated light is called a lambda, wavelength, or circuit. The OXC can also act as a *relay* for other OXC nodes to transparently route wavelengths. When acting as a relay for remote hosts, an OXC provides optical switching capabilities, thus giving the network flexibility in choosing where to send transmitting lambdas over the OXC node.

Figure 2 illustrates the connectivity at different layers of the IP-over-OTN model. The physical network connects points-of-presence (PoPs) with optical fiber spans. OXC nodes connects these PoPs with optical paths or circuits. The physical routes of the paths are abstracted away, and shown in color for reference. In the IP topology, the colors of light are also abstraced away, and we see a mesh IP network connected

by routers and switches. Hosts connect to nodes at this layer, and their traffic travels down the optical paths in the physical network to reach its destination.

IP-over-OTN networks are not new. However, they are built at a great cost. Historically network planners have engineered them to accommodate the worst-case expected demand by (1) over-provisioning of dense wavelength division multiplexing (DWDM) optical channels and (2) laying redundant fiber spans as a fail-safe for unexpected traffic surges. These surges could come from user behavior changes or failures elsewhere in the network that forces traffic onto a given path. Only recently have reconfigurable optical systems begun to gain attention in the data center and wide-area network settings. For more information about early IP-over-OTN, we defer to Bannister et al. [24] and references therein, where the authors present work on optimizing WDM networks for node placement, fiber placement, and wavelength allocation.

2.1.2 Data Center Architectures. Historically, data centers relied on packet-switched networks to connect their servers; however, as scale and demand increased, the cost to build and manage these packet-switched networks became too large. As a result of this change, new reconfigurable network topologies gained more attention from researchers and large cloud providers. Many novel data center architectures with reconfigurable optical topologies have been proposed over the last decade. These architectures have in common that they reduce the static network provisioning requirements, thereby reducing the network's cost by presenting a means for bandwidth between hosts to change periodically. Figure 3 shows one such example of a hybrid electrical-optical data center architecture. These architectures reduce cost and complexity via scheduling methods, which change bandwidth on optical paths in the data center. Various approaches have been demonstrated.

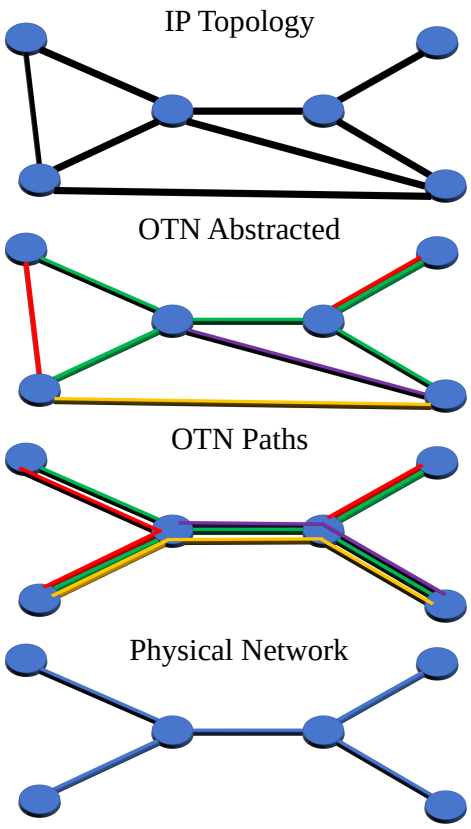


Figure 2. IP-over-OTN network architecture model, showing the connection between IP and optical layers.

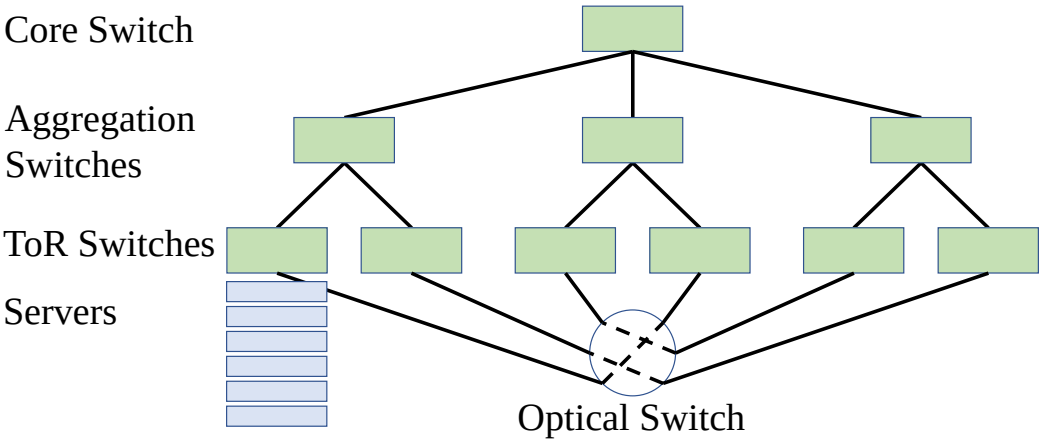


Figure 3. Data center architecture proposed in c-Through [288]

Notable architectures employ fixed, and deterministic scheduling approaches [23, 198] or demand-aware changes that prioritize establishing optical paths between servers with mutual connectivity requests [286, 292]. Switching fabrics are also diverse for data center optical systems. These include fabrics based on nanosecond tunable lasers [168], digital micromirror devices (DMD) [108], and liquid crystal on silicon (LCOS) wavelength selective switches (WSS) [245].

2.1.3 Software Defined Networking. Modern data center, metro, and wide-area networks have been substantially influenced by developments in Software Defined Networking (SDN) [298], and this trend has also been making its way to optical networks [273]. The SDN paradigm decouples the control and data plane in network hardware, giving operators greater control and flexibility for controlling traffic within their network. Without this decoupling, it is more difficult to make lock-step changes to network functions, such as routing. SDN offers a logically centralized point of control for implementing policies across the network, thus enabling better network utilization for bandwidth, latency, security policies, etc. These concepts can also map further down the network stack to manage optical infrastructure, thereby 1) improving optical layer performance with technology, which we describe in Section 2.2,

and 2) allowing management algorithms to adapt the optical paths in a demand-aware fashion, which we describe in Section 3.2 for data center networks and in Section 3.3 for metro and wide-area networks.

Notwithstanding, providing a standardized stable and reliable programmable optical physical layer control plane for SDNs is still an ongoing effort, as recently outlined by the *TURBO* project [151]. One important step in this direction is the development of virtual testbeds to evaluate the cross-layer operation of SDN control planes [169].

2.1.4 Elastic Optical Networks. A span of optical fiber enables transmission of data over a *spectrum* or set of wavelengths. These wavelengths can be allocated in a fixed or flexible (flex) grid. Networks that allow flex grid allocations are also called Elastic Optical Networks (EONs). For example, according to the ITU-T G.694.1 fixed grid standard, frequencies must be 12.5, 25, 50, or 100 GHz apart [273]. However, in elastic optical networks (EONs), also known as flex-grid networks, the frequency of a channel can be any multiple of 6.25 GHz away from the central frequency (193.1 THz) and have a width that is a multiple of 12.5 GHz. Figure 4 illustrates the difference between a flex-grid and fixed-grid allocation.

Flex grid networks can greatly improve the spectral efficiency of IP-over-OTN, allowing the network to pack data channels more densely within a span of optical fiber. However, they can also lead to unique challenges, particularly fragmentation. Fragmentation occurs when spectrum allocated on a fiber has gaps in it that are too narrow to be filled. Novel approaches to managing EONs with fragmentation-aware algorithms are covered in depth by Chatterjee et al. [47].

2.1.5 Summary. Our survey relates the latest developments in reconfigurable networks for data centers and WANs. The IP-over-OTN model is

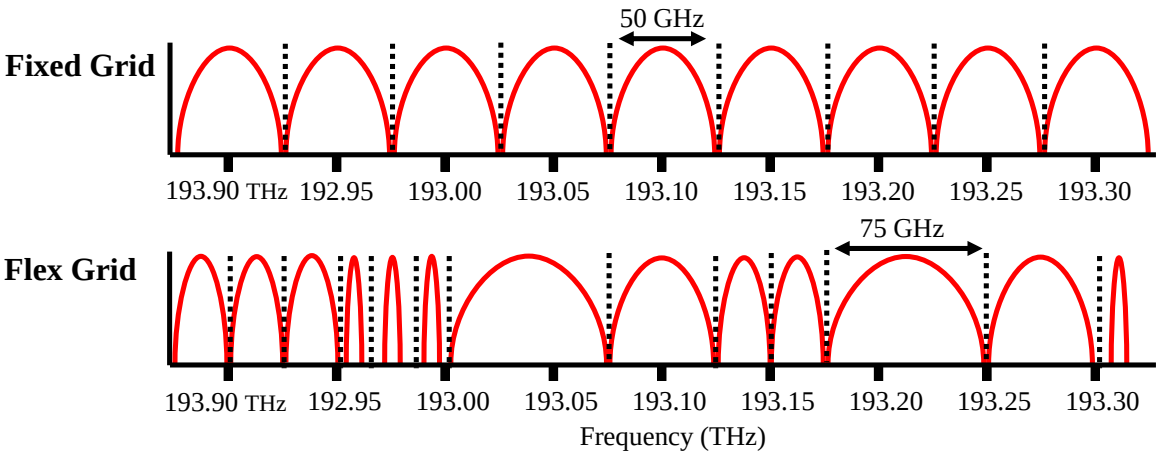


Figure 4. Example of fixed grid and flex grid spectrum allocation.

a useful framework for reasoning about and managing optical metropolitan, regional, and wide-area networks. Similarly, we are seeing data center architectures become more reconfigurable and demand-aware with optical circuit switching. SDN is poised to bring substantial changes to the operation of optical networks in both domains by offering a centralized point for management and control for more network infrastructure, from routing of packets to routing of optical paths. Moreover, EONs are also enabling better spectral efficiency.

2.2 Enabling Hardware Technologies

In this section, we discuss hardware technologies that enable reconfigurable optical networks. In our end-to-end discussion on reconfigurable optical networks, the hardware is the foundational layer from which systems are built. Understanding these devices and their capabilities is crucial for designing and building real-world reconfigurable optical networking systems. We show examples of different optical technologies, including optical switches and transponders, and examples of systems that use them. We also highlight recent advances in silicon photonics, and the implications this may have for reconfigurable optical networks in the near future.

Finally, we discuss open challenges in reconfigurable optical networks that might be solved with next-generation hardware.

2.2.1 Wavelength Selective Switching. In contrast to packet-switched networks, optically circuit-switched systems operate at a more coarse granularity. The transmission of information over a circuit requires an end-to-end path for the communicating parties. Although packet switching has generally prevailed in today's Internet, recent research has revitalized the prospect of circuit switching for data centers and wide-area networks by illuminating areas in which flexible bandwidth benefits outweigh the start-up cost of circuit building.

Technological advancements for optical hardware, primarily driven by physics and electrical engineering research, have been instrumental in making circuit-switched networks a viable model for data center networks. Among these technologies are low-cost/low-loss hardware architectures. Here we give a brief overview of technological advancements in this domain that have had the most significant impact on networked systems.

Kachris et al. [144] have an in-depth look at optical switching architectures in data centers from 2012. In their survey, they primarily look at competing *data center* architectures and switch models. In this section, we choose to focus instead on those architectures' physical manifestations (i.e.the base components that make them up). Furthermore, exciting new developments have occurred since then, which we highlight in this section.

Polymer waveguides are a low-cost architecture for optical circuit switches. These have been fabricated and studied in depth over the last 20 years, including work by Taboada et al. [265] in 1999, Yeniay et al. [309] in 2004, and Felipe et al. [69] in 2018. Early implementations such as Taboada et al. [265] showed fabrication

techniques for simple polymer waveguide taps. Multiple waveguide taps can be combined to form an Array Waveguide Grating (AWG), and the signals traversing the AWGs can then be blocked or unblocked to create an optical circuit switch. A major inhibitor of the polymer waveguide architecture was signal-loss, which was as high as 0.2 dB/cm until Yeniay et al. [309] discovered an improvement on the state-of-the-art with ultra low-loss waveguides in 2004. Their waveguides, made with fluorocarbons, have 4× less loss (0.05 dB/cm) than the next best waveguides at the time, made from hydrocarbons. Felipe et al. [69] demonstrate the effectiveness of a polymer waveguide-based switching architecture for reconfiguring groups of optical flows of up to 1 Tbps, proving that that AWG is a viable and competitive switching architecture for data centers. More recently, in 2020, AWGs were demonstrated to work in conjunction with sub-nanosecond tunable transmitters to create flat topologies, significantly reducing power consumption for data center networks due to the passive—no power required—nature AWGs [62]. Switching speeds below 820 ps have been demonstrated using a 1×60 AWG and tunable laser [168]. AWGs with as many as 512 ports have been demonstrated [55].

Microelectromechanical Systems (MEMS), introduced by Toshiyoshi et al. [275] in 1996, offered a lower-loss and more flexible alternative to polymer waveguide systems of the day. MEMS devices are made up of small mirrors, which can be triggered between states (i.e.*on* and *off*). Therefore, in a MEMS system light is *reflected* rather than *guided* (as in the polymer waveguide systems). This distinction between reflection and guiding implies generally slower switching speeds for MEMS based systems, as the mirror must be physically turned to steer light out of the desired switch-port. Despite this limitation, MEMS systems evolved to be competitive with polymer waveguides in modern systems. Advances in MEMS technology have

yielded wavelength selective switches (WSS) scalable to 32 ports with switching speeds under 0.5 ms [278]. Data center solutions leveraging MEMS based switches include Helios [85].

Liquid Crystal on Silicon (LCOS) was demonstrated as another viable optical switching architecture by Baxter et al. [26] in 2006. An LCOS switch is depicted in Figure 5. Multiplexed optical signals enter the system from a fiber array. These signals are directed to a conventional diffraction grating where the different colors of light are spatially separated from each signal. These colors are then projected onto a unique position in the LCOS switching element. This element is divided into pixels or cells, and charged with an electrical current. The voltage applied to any cell in the switching element determines which output fiber a given channel will leave through. From there, the signal travels back through the system and into a different fiber in the array.

Switches based on this technology have a response time of $10 - 100$ μs [293]. Recent work by Yang et al. [306] demonstrates the construction of a 12×12 and 1×144 port WSS based on a 1×12 LCOS architecture. Chen et al. [48] developed an improved LCOS architecture with which they demonstrated a 16×16 optical switch. LCOS switches are commercially available and are recognized as a key enabler for reconfigurable optical networks [245].

Summary. Table 1 summarizes optical switch performance metrics. Each architecture comes with advantages under distinct circumstances. Highly scalable data center architectures have been developed with sub-nanosecond tunable lasers and AWGs [23, 62, 168]. MEMS have generally better scalability, lower insertion loss, and less crosstalk over LCOS systems [320] but also demand higher precision manufacturing to ensure that all $N \times M$ mirrors configurations are accurately aligned.

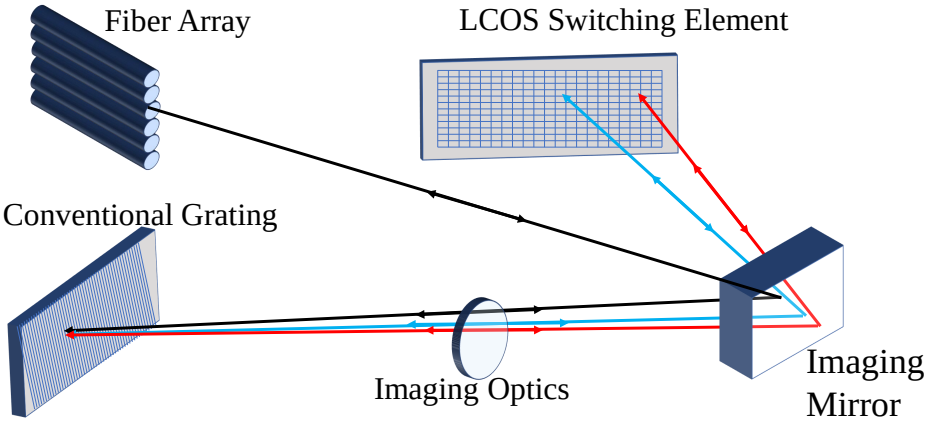


Figure 5. Liquid crystal on silicon wavelength selective switch.

	Port Scalability	**Switching Speed**	
AWG	512×512	< 820 ps	Highly scalable with unsurpassed demonstrations for short-reach applications with tunable lasers.
MEMS	32×32	< 0.5 ms	Higher scalability and lower insertion loss, less crosstalk.
LCOS	16×16	$10 - 100$ μs	Lower scalability and optical performance, but more modular design than MEMS.

Table 1. Summary of systems implementations of reconfigurable wide area networks

LCOS elements can also be packed more compactly into a modular unit due to the absence of moving parts that are present in MEMS.

2.2.2 ROADMs. Reconfigurable add-drop multiplexers, or ROADMs, are an integral component of IP-over-OTN networks. These devices have evolved over the years to provide greater functionality and flexibility to optical transport network operators. We briefly describe the evolution of ROADM architectures. Figure 6 shows a broadcast and select ROADM architecture. Please refer to [192] for more information about ROADM architectures.

Colorless (C). Early ROADMs were effectively programmable wavelength *splitter-and-blockers*, or *broadcast-and-select* devices. A wavelength splitter-and-blocker can be placed before an IP-layer switch. If the switch is intended to *add/drop* a

wavelength (i.e.transceive data on it), then the blocker prohibits light on the upstream path and enables light on the path to the switch. These splitter-and-blocker systems are better known as Colorless, or C-ROADMs, as the *splitter-and-blocker* architecture is independent of any specific frequency of light. To receive the maximum benefit from C-ROADMs, operators should deploy their networks with tunable transceivers as they allow more flexibility for the end hosts when connecting to remote hosts.

Colorless, Directionless (CD). The CD-ROADMs extend the architecture of C-ROADMs by pairing multiple C-ROADMs together in the same unit to allow for a wave to travel in one of many directions. One shortfall of this architecture is that the drop ports from each direction are fixed, and therefore if all of the drop ports are used from one direction, the remaining points from other directions cannot be used. Due to the limitation of drop ports in different directions, the CD architecture is not *contentionless*.

Colorless, Directionless, Contentionless (CDC). The CDC-ROADM solves the contention problem by providing a shared add/drop port for each direction of the ROADM. This allows contentionless reconfiguration of the ROADM as any drop-signal is routed to a common port regardless of the direction from which the wave begins/terminates.

Colorless, Directionless, Contentionless w. Flexible Grid (CDC-F). Flexi-grid, or elastic optical networks, are networks carrying optical channels with non-uniform grid alignment. This contrasts with a fixed-grid network, where different wavelengths are spaced with a fixed distance (e.g.50 GHz spacing). Wideband spacing allows signals to travel farther before becoming incoherent due to chromatic dispersion. Thus, CDC-Flex or CDC-F ROADMs enable the reconfiguration of

wavelengths with heterogeneous grid alignments. These are most useful for wide area networks, with combinations of sub-sea and terrestrial circuits.

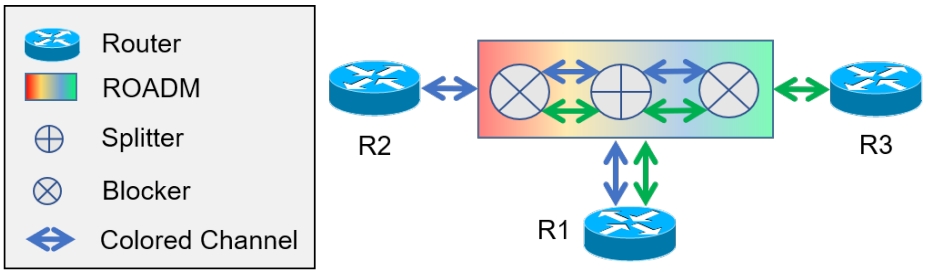


Figure 6. Broadcast and Select colorful ROADM. The add/drop node, R1, has ports for two optical channels. These channels are directed at the ROADM. The ROADM uses a splitter to *broadcast* the channels onto two outbound ports, where a wavelength blocker *selects* the appropriate channel for the next router.

2.2.3 Bandwidth-variable Transponders. Before we discuss bandwidth-variable transponders, we must first take a moment to illuminate a common concept to all physical communications systems, not only optical fiber. This concept is modulation formats. Modulation formats determine the number of binary bits that a signal carries in one *symbol*. Two parties, a sender and receiver, agree on a symbol rate (baud), which determines a clock-speed to which the receiver is tuned when it interprets a symbol from the sender. The simplest modulation format is on-off keying (OOK), which transmits one bit per symbol. In OOK, the symbol is sent via a high or low power level, as shown in Figure 7A. A higher-order modulation technique is Quadrature Phase Shift Keying (QPSK), in which the symbol is a sinusoidal wave whose phase-offset indicates the symbol. In QPSK, there are four phase shifts agreed upon by the communicating parties, and therefore the system achieves two bits per symbol, or two baud, seen in Figure 7B. A constellation diagram for QPSK is shown in Figure 7B. As modulations become more complex, it is more useful to visualize them in the phase plane shown by their constellation diagram. Higher-order modulation formats are of the type, N-Quadrature Amplitude Modulation (QAM)

techniques (Figure 7D), and these permit $log_2(N)$ bits per symbol[1]. In QAM, the symbol is denoted by phase and amplitude changes. Figure 7D shows an example of a constellation diagram for 16-QAM modulation, which offers 4 bits per symbol, or twice the baud of QPSK.

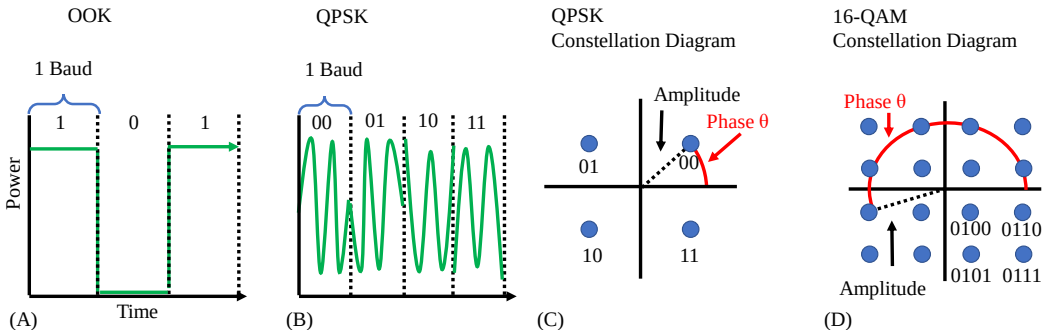


Figure 7. Modulation examples of on-off keying, quadrature phase shift keying (QPSK), quadrature amplitude modulation (QAM), and constellation diagrams for QPSK and 16-QAM.

Fiber optic communications are subject to noise. The noise level is termed *Signal to Noise Ratio* (SNR), and this metric determines the highest possible modulation format. In turn, the modulation format yields a potential capacity (Gbps) for an optical channel. For example, in [93], the authors claim that SNR of just 6 dB is sufficient to carry a 100 Gbps signal, while a circuit with an SNR of 13 dB can transmit 200 Gbps.

Bandwidth Variable Transponders (BVTs) [142] have recently proven to have significant applications for wide-area networks. These devices are programmable, allowing for the operator to choose from two or more different modulation formats, baud rates, and the number of subcarriers when operating an optical circuit. For example, the same transponder may be used for high-capacity/short-reach transmission (16-QAM or greater) or lower-capacity/longer-reach transmission

[1]where N is generally a power of 2

(e.g.QPSK). Higher modulation formats offer higher data rates. They are also more sensitive to the optical SNR, which decreases in a step-wise manner with distance, as illustrated in Figure 8. We note that BVTs enable network operators to meet the ever-growing demand in backbone traffic by increasing optical circuits' spectral efficiency.

Low spectrum utilization, or waste, can be an issue for BVT circuits. For example, a BVT configured for a low-modulation circuit such as QPSK instead of 16-QAM has a potential for untapped bandwidth. Sambo et al. [241] introduced an improvement to the BVT architecture, known as Sliceable-BVT (S-BVT), which addresses this issue. They describe an architecture that allows a transponder to propagate numerous BVT channels simultaneously. Channels in the S-BVT architecture are sliceable in that they can adapt to offer higher or lower modulation in any number of the given subchannels.

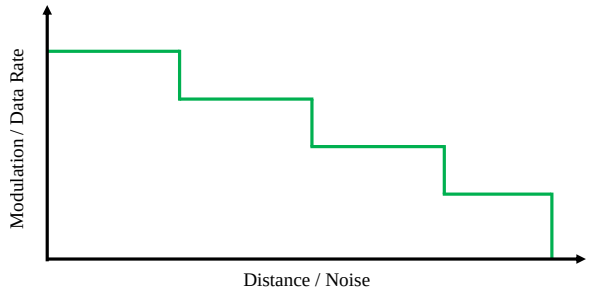


Figure 8. Conceptualization of the trade off between modulation/data rate and distance/noise with BVT. Noise, which can be measured with bit error rate, Q factor, or SNR, increases with the distance covered by an optical circuit. As more noise is accumulated over greater distance, the highest-order modulation that the circuit can support, and thereby the data rate on that circuit, falls in a piece-wise manner.

2.2.4 Silicon Photonics. Various materials (e.g.GaAs, Si, SiGe) can be used to make photonics hardware required for data transmission. These devices include photodetectors, modulators, amplifiers, waveguides, and others. Silicon (Si) is the preferred material for these devices due to its low cost. However, there are

challenges to manufacturing these silicon devices, such as optical power loss and free carrier absorption. Other materials, notably GaAs, have better properties for propagating light; however, GaAs is more costly to manufacture. Despite these challenges, research into efficient and quality transmission using silicon-based photonic devices has boomed in the last decade. Early advances were made towards silicon photonics (SiP) in the 80s, particularly for waveguides, which are the basis for circuit switches and multiplexers. Today, SiP is an integral part of almost all optical hardware, including lasers, modulators, and amplifiers.

A significant challenge for power-efficient SiP transceivers is coupling loss between the laser source and passive waveguide on Si integrated circuit waveguides, which can be as high as 2.3 dB, or 25% power loss [124]. Recent work by Billah et al. [32] explores the integration of indium pihosphide (InP) lasers on chips, demonstrating a coupling with only 0.4 dB of loss, or roughly 10%. InP appears to be a promising compound for other SiP technology too, as evident by demonstrations of InP in-line amplification for WSS [194]. Costs are falling for optical hardware as more efficient and scalable manufacturing techniques are enabled by SiP [296], thus allowing network operators to deploy newer technology into their systems at a more advanced pace as the devices' quality and guarantees have continued to improve. For more information on silicon photonics, see the survey by Thomson et al. [272].

2.2.5 Summary. Hardware for reconfigurable optical networks is improving at rapid scales, where researchers are developing more scalable optical switches with faster response times year after year. These WSS architectures are quickly being integrated with ROADMs to offer CDC-F flexibility for networks. Meanwhile, improvements to transponder technology are also paving the way for reconfigurable optics at network endpoints. In particular, S-BVTs offer dramatic CAPEX savings

as one transponder can deliver multiple modulated signals in parallel. These improvements are accelerated by silicon photonics, bringing CMOS manufacturing to optical hardware and greatly reducing the cost to deploy optical switches and upgraded transponders in networks.

CHAPTER III

RELATED WORK

This chapter contains previously published coauthored material from [214] and coauthored by Klaus-Tycho Foerster, Stefan Schmid, and Ramakrishnan Durairajan. Klaus-Tycho Foerster and Stefan Schmid helped with the classification of related works particularly for reconfigurable data center networks. The dissertation author primarily classified related works for reconfigurable wide-area networks. The coauthors assisted with editing.

3.1 Introduction

In this chapter, we present an *end-to-end perspective* on reconfigurable optical networks by (a) emphasizing the interdependence of optical technologies with algorithms and systems and (b) identifying the open challenges and future work at the intersection of optics, theory, algorithms, and systems communities. Our survey is tutorial in nature and focuses on concepts rather than exhaustive related work, concentrating on selected articles. Hence, our paper targets students, researchers, experts, and decision-makers in the networking industry who would like to obtain an overview of the critical concepts and state-of-the-art results in reconfigurable optical networks. We start with an overview of the enabling optical hardware technologies. We explore where data center and WAN systems have integrated this hardware. We review cost models, discuss the novel algorithmic challenges and solutions in the literature, and elaborate on systems and implementation aspects. We also identify the major open issues which require further exploration and research to design the next generation reconfigurable optical networks.

3.2 Optically Reconfigurable Data Centers

In this section, we illuminate efforts to improve DCNs with reconfigurable optics. Related surveys on this subject include Foerster et al. [100] and Lu et al. [183]. We divide the state of reconfigurable optical DCNs into technology, cost modeling, and algorithms. In technology, we supplement the discussion from § 2.2 with hardware capabilities that currently exist only for DCNs. Such features include free-space optics and sub-second switching. Next, we highlight cost modeling research, whose goal is to derive formal estimates or guarantees on the benefit of reconfigurable optical networks over static topologies for DCNs. Finally, we survey the relevant algorithms for managing and optimizing reconfigurable optical networks in the data center. Many of these algorithms focus on the interdependencies between optical path set-up and routing and optimize them across layers. Notwithstanding, there is also work that optimizes the physical layer simultaneously as well, respectively focuses on the interplay between software defined networking (SDN) and the physical layer, as illustrated in Figure 9. We discuss these examples in more detail and also survey further related work across the next subsections.

A key challenge for data centers is to optimize the utilization of the data center network (DCN). In a DCN, many different services are running and competing for shared bandwidth. Communication patterns between top-of-rack (ToR) switches vary with the underlying applications that are running (e.g.map-reduce, video stream processing, physics simulations, etc.). Thus, as future applications and user's needs change, it is challenging to predict where bandwidth will be needed.

Static and reconfigurable network solutions have been posed by research and industry to address this challenge. There is an assumption that the connectivity graph of the network cannot change in static network solutions. These solutions also

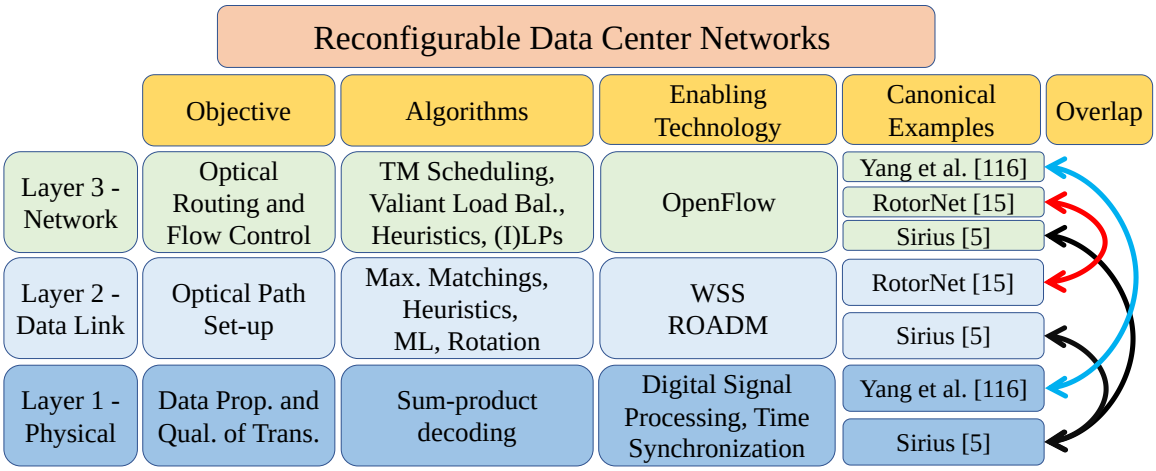


Figure 9. Solving the challenges involved in reconfigurable optics for data center networks requires bridging the gap between different technologies and goals for different layers of the network protocol stack.

assume fixed capacity (or bandwidth) on links. In reconfigurable network solutions, by contrast, these assumptions regarding connectivity and bandwidth are relaxed. Servers and switches (collectively referred to as nodes) may connect some subset of the other nodes in the network, and the nodes to which they are adjacent may change over time. Further, the bandwidth of a connection may also change over time.

Under the assumption of a static physical topology, different network architectures and best practices have been established. Some of these architectures include Clos, fat-tree, and torus topologies. Best practices include (over-)provisioning all links such that the expected utilization is a small fraction of the total bandwidth for all connections. These solutions can incur high cabling costs and are inefficient.

Reconfigurable network solutions circumvent the limitations of the static network solutions by reducing cabling costs or reducing the need to over-provision links. The flexibility of light primarily empowers these reconfigurable solutions. Some of these flexibilities include the steering of light (e.g.with MEMs or polymer waveguides)

and the high capacity of fiber-optics as a medium (e.g.dense wavelength division multiplexing, or DWDM, enables transmitting $\mathcal{O}(Tb/s)$ on a single fiber).

3.2.1 DCN-specific Technologies. Innovations in reconfigurable optical networks are enabled by hardware's evolution, as discussed in Section § 2.1. There is a subset of innovations that are well-suited for data centers only. These are *free-space optics* and *sub-second switching*. Although we have separated these below, there may be overlaps between free-space optics and sub-second switching systems as well.

Free-space Optics. In free-space optics systems, light propagates through the air from one transceiver to another. Free-space optics enables operators to reduce their network's complexity (a function of cabling cost). These closed environments and their highly variable nature of intra-data center traffic make such solutions appealing; we refer to the overview by Hamza et al. [121] for further application scenarios. Recent works such as Firefly [19] have demonstrated that free space optics are capable of reducing latency for time-sensitive applications by routing high-volume/low-priority traffic over the wireless optical network while persistently serving low-volume/high-priority traffic on a packet-switched network. High fan-out (1-to-thousands) for free-space optics is enabled with DMDs, or Dense Micro-mirror Devices, as shown by ProjecToR [109]. The DMDs are placed near Top-of-Rack (ToR) switches and pair with disco-balls, fixed to the ceiling above the racks. The DMD is programmed to target a specific mirror on the disco-ball, guiding the light to another ToR in the data center. Figure 10 illustrates the main properties of the free space optics deployment proposed in [109]. The deployment and operation of a free-space optics data center are fraught with unique challenges, e.g.geometrical placement as investigated in 2D in OWCell [122] and in 3D in Diamond [66], but also particularly for keeping the air clear between transceivers and DMDs. Any particulate matter that the

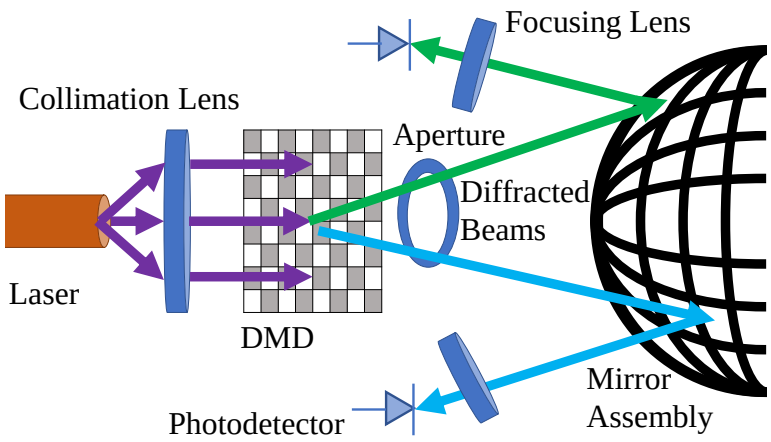


Figure 10. Free-space optics switching architecture for data centers [109]

light comes into contact with can severely degrade performance and cause link failures should they persist. This phenomenon is known as atmospheric attenuation [34]. Another aspect is misalignment due to, e.g.vibrations, requiring active alignment systems [276] respectively a tradeoff between beamwidth and received power density, depending on the distance covered [19]. In summary, even though free-space optics is an attractive alternative for many scenarios [121], and can be seen as "*fiber without the fiber*" [276], these technologies "*are not used in commercial data centers yet*" [271], and hence the main challenge is working towards their practical deployment. We refer to two recent specialized surveys for more details [120, 271].

Sub-second Switching. In data centers, distances are short between hosts, and therefore they do not lose their strength to such a degree that mid-line devices such as amplifiers are necessary. Therefore, applications can benefit from all of the agility of optical layer devices without accounting for physical-layer impairments, which can slow down reconfiguration times in wide-area networks. Research has shown that micro-second switching of application traffic is possible in data center environments [83, 84, 234]. The ability to conduct circuit switching at microsecond timescales has illuminated further intrigue, particularly for transport protocols

running on top of these networks. In c-Through [288], the authors observed that throughput for TCP applications dropped when their traffic migrated to the optical network. They showed how to mitigate this by increasing the queue size for optical circuit switches and adjusting the host behaviors. Mukerjee et al. [207] augmented their solution by expanding TCP for reconfigurable data center networks. Another method to deal with rapid reconfiguration times at a micro-second level is using traffic matrix scheduling, as we will further elaborate in Section § 3.2.3.

However already e.g. Alistarh et al. [6] showcased the possibility of switching in the order of nano-seconds in a thousand port 25 Gbps+ optical switch design. Notwithstanding, a challenging question is how to make use of such fast reconfiguration times, when accounting for computation and routing update delays. Mellette et al. follow an intriguing design choice with their rotor switches [199], by creating demand-oblivious connections that change in the order of micro-seconds, in turn pre-configuring the routing in RotorNet [198] and Opera [197]. Project Sirius expands such ideas to the sub-nano-second level [62, 168], resulting in a demand-oblivious design that can perform end-to-end reconfigurations in less than 4 nano-seconds at 50 Gbps [23]. We further discuss these strategies in Section § 3.2.3.

Summary. Unlike in the WAN, data center technologies allow extremely fast switching times and high fan-out across the whole network, the latter in particular in the case of free-space optics. Hence especially the algorithmic design ideas allow substantially more flexibility and often differ fundamentally, as we will see in Section § 3.2.3.

3.2.2 Cost Modeling. Momentum has been building for data centers to move to optically switched and electrical/optical hybrid networks. However, there is a general reluctance to walk away from the old paradigm of a packet-switched-only

network (PSO) due to the additional complexity of optical circuit switching (e.g.the control plane management of optical circuits with shifting demand, and the variety of optical switching architectures available). Further, without a quantitative measure of *value-added* by optical switching over PSO, DCN operators are understandably reluctant to spend capital on an unvetted system. A discussion on the cost differences between optically and electrically switched data center networks can be found in the work of Kassing et al. [150], with an analysis for non-wired topologies in the works of Shin et al. [249] and Terzi and Korpeoglu [271].

To address the concerns surrounding complexity and value while raising awareness for the necessity of optically switched interconnects, researchers have constructed cost models to demonstrate the benefit of optical switching and hybrid architectures. Wang et al. [287] developed one such model. They conducted intra-DC traffic measurements, which consisted of mixed workloads (e.g.Map-Reduce, MPI, and scientific applications). They then played the traces back in simulation, assuming that three optical circuits could be created and reconfigured between racks every 30 seconds. Their data center with seven racks showed that rack-to-rack traffic could be reduced by 50% with circuit switching.

The following sections present more cost modeling work in the context of algorithmic simulations and systems implementations.

3.2.3 Algorithms. The capability of optical circuit switching for data center networks comes with the need to define new algorithms for optimizing utilization, bandwidth, fairness, latency, or any other metric of interest. Research has presented many different approaches for optimizing the metric relevant to the network operator in static networks. Traffic Engineering (TE) generally refers to the determination of paths for flows through the network, and the proportion of bandwidth levied for

any particular flow. If the data center has a static network topology (e.g.fat-tree), then TE is simple enough that switches can conclude how to route flows. However, introducing reconfigurable paths complicates the process of TE significantly: network elements (e.g.switches) must now also determine with whom and when to establish optical paths, and when to change them.

Overview. The current algorithmic ideas to establish such optical paths can be classified into roughly five different areas, which we will discuss next. Due to the inherent hardware constraints (forming circuits), all of them rely on 1) matchings, where on its own the main idea is to maximize matching's weight, e.g.representing throughput, latency, etc. However maximum matchings can be slow to compute, and hence there has been interest in 2) demand-oblivious approaches, cycling through different network designs, 3) traffic matrix scheduling, to batch-compute a whole set of matchings ahead of time, and also leveraging the speed-up of 4) machine learning algorithms. Lastly, another way of quickly reacting to demand changes is by borrowing ideas from 5) self-adjusting data structures, in particular adapting the aspect of purely local circuit changes.

Matchings can be computed quickly [76] and often provide a good approximation, especially in settings where the goal is to maximize single-hop throughput along with reconfigurable links. Matching algorithms hence frequently form the basis of reconfigurable optical networks, e.g. Helios [85], c-Through [288] and [68] rely on maximum matching algorithms. If there exist multiple reconfigurable links (say b many), it can be useful to directly work with a generalization of matching called b-matching [208]: b-matchings are for example used in Proteus [255] and its extension OSA [50], as well as in BMA [31] which relies on an online b-matching algorithm; BMA also establishes a connection to online (link) caching problems. In some scenarios,

for example, when minimizing the average weighted path length under segregated routing, maximum b-matching algorithms even provide optimal results [97, 99]. This however is not always true, e.g.when considering non-segregated routing policies [97, 99], which require heuristics [19, §5.1], [88].

Oblivious Approaches. Matchings also play a role in reconfigurable networks which do not account for the traffic they serve, i.e.in *demand-oblivious* networks. The prime example here is RotorNet [198] which relies on a small set of matchings through which the network cycles endlessly: since these reconfigurations are "dumb", they are fast (compared to demand-aware networks) and provide frequent and periodic direct connections between nodes, which can significantly reduce infrastructure cost (also known as "bandwidth tax") compared to multihop routing, see also Teh et al. [267]. In case of uniform (delay-tolerant) traffic, such single-hop forwarding can saturate the network's bisection bandwidth [198]; for skewed traffic matrices, it can be useful to employ Valiant load balancing [285] to avoid underutilized direct connections, an idea recently also leveraged in Sirius [23] via Chang et al. [45]. Opera [197] extends RotorNet [198] by maintaining expander graphs in its periodic reconfigurations. Even though the reconfiguration scheduling of Opera is deterministic and oblivious, the precomputation of the topology layouts in their current form is still randomized. Expander graphs (and their variants, such as random graphs [254]) are generally considered very powerful in data center contexts. An example of a demand-aware expander topology was proposed in Tale of Two Topologies [300], where the topology locally converts between Clos and random graphs.

Traffic Matrix Scheduling. Another general algorithmic approach is known as *traffic matrix scheduling*: the algorithmic optimizations are performed based on a snapshot of the demand, i.e.based on a traffic matrix. For example, Mordia [234] is

based on an algorithm that reconfigures the network multiple times for a single (traffic demand) snapshot. To this end, the traffic demand matrix is scaled into a bandwidth allocation matrix, which represents the fraction of bandwidth every possible matching edge should be allocated in an ideal schedule. Next, the allocation matrix is decomposed into a schedule, employing a computationally efficient [112] Birkhoff-von-Neumann decomposition, resulting in $O(n^2)$ reconfigurations and durations. This technique also applies to scheduling in hybrid data center networks which combine optical components with electrical ones, see e.g.the heuristic used by Solstice [177]. Eclipse [286] uses traffic matrix scheduling to achieve a $(1 - 1/e^{(1-\varepsilon)})$-approximation for throughput in the hybrid switch architecture with reconfiguration delay, but only for direct routing along with single-hop reconfigurable connections. Recently Gupta *et al.* [119] expanded similar approximation guarantees to multi-hop reconfigurable connections, for an objective function closely related to throughput.

While Eclipse is an offline algorithm, Schwartz et al. [243] presented online greedy algorithms for this problem, achieving a provable competitive ratio over time; both algorithms allow to account for reconfiguration costs. Another example of traffic matrix scheduling is DANs [11, 12, 13, 16] (short for demand-aware networks, which are optimized toward a given snapshot of the demand). DANs rely on concepts of demand-optimized data structures (such as biased binary search trees) and coding (such as Huffman coding) and typically aim to minimize the expected path length [11, 12, 13, 16], or congestion [13]. In general, the problem features intriguing connections to the scheduling literature, e.g., the work by Anand et al. [8], and more recently, Dinitz et al. [71] and Kulkarni et al. [162]; the latter two works however are not based on matchings or bipartite graphs. In Dinitz et al. [71], the demands are the edges of a general graph, and a vertex cover can be communicated in each round.

Each node can only send a certain number of packets in one round. The approach by Kulkarni et al. [162] considers a model where communication requests arrive online over time and uses an analysis based on on LP relaxation and dual fitting.

Self-Adjusting Datastructures. A potential drawback of traffic matrix scheduling algorithms is that without countermeasures, the optimal topology may change significantly from one traffic matrix snapshot to the next, even though the matrix is similar. There is a series of algorithms for reconfigurable networks that account for reconfiguration costs, by making a connection to self-adjusting data structures (such as splay trees) and coding (such as dynamic Huffman coding) [10, 14, 15, 16, 17, 232, 233, 242]. These networks react *quickly and locally* to new communication requests, aiming to strike an optimal trade-off between the benefits of reconfigurations (e.g.shorter routes) and their costs (e.g.reconfiguration latency, energy, packet reorderings, etc.).

To be more specific, the idea of the self-adjusting data structure based algorithms is to organize the communication partners (i.e.the destinations) of a given communication source in either a static binary search or Huffman tree (if the demand is known), or in a dynamic tree (if the demand is not known or if the distribution changes over time). The tree optimized for a single source is sometimes called the *ego-tree*, and the approach relies on combining these ego-trees of the different sources into a network while keeping the resulting node degree constant and preserving distances (i.e.low distortion). The demand-aware topology resulting from taking the union of these ego-trees may also be complemented with a demand-oblivious topology, e.g., to serve low-latency flows or control traffic; see the ReNet architecture for an example [17].

Machine Learning. Another natural approach to devise algorithms for reconfigurable optical networks is to use machine learning. To just give two examples, xWeaver [292] and DeepConf [240] use neural networks to provide traffic-driven topology adaptation. Another approach is taken by Kalmbach et al. [145], who aim to strike a balance between topology optimization and "keeping flexibilities", leveraging self-driving networks. Finally, Truong-Huu et al. [277] proposed an algorithm that uses a probabilistic, Markov-chain based model to rank ToR nodes in data centers as candidates for light-path creation.

Accounting for Additional Aspects. Last but not least, several algorithms account for additional and practical aspects. In the context of shared mediums (e.g.non-beamformed wireless broadcast, fiber[1] (rings)), contention and interference of signals can be avoided by using different channels and wavelengths. The algorithmic challenge is then to find (optimal) edge-colorings on multi-graphs, an NP-hard problem for which fast heuristics exist [201]. However, on specialized topologies, optimal solutions can be found in polynomial time, e.g.in WaveCube [51]. Shared mediums also have the benefit that it is easier to distribute data in a one-to-many setting [290]. For example, on fiber rings, all nodes on the ring can intercept the signal [52, §3.1]. One-to-many paradigms[2] such as multicast can also be implemented in other technologies, using e.g.optical splitters for optical circuit switches or half-reflection mirrors for free-space optics [25, 186, 262, 263, 299].

3.2.4 Systems Implementations.

There have been many demonstrations of systems for reconfigurable optics in data centers. Many of the papers that we discuss in Section 3.2.3 are fully operational systems. Another notable research

[1] In the context of data center proposals, shared fiber is the more popular medium, e.g.in [50, 52, 234].

[2] Conceptually similar challenges arise for coflows [132, 291].

development that does not fit into algorithms is the work by Mukerjee et al. [207]. They describe amendments to the TCP protocol to increase the efficiency of reconfigurable data center networks. These amendments include dynamic buffer resizing for switches and sharing explicit network feedback with hosts. Moreover, Yang et al. [307] showcase an interesting cross-layer aspect where the physical layer itself is controlled by SDN, in the sense that they allow for transceiver tuning in real-time. Their main contributions relate to new SDN control modules and interfaces, being orthogonal to (scheduling) algorithms. Much of the other work on reconfigurable DCNs are summarized in Table 2. Finally, recent publications by Google [178, 235] have disclosed that they have been using reconfigurable optics in their data centers as far back as 2013.

We see two main conceptual differences in current reconfigurable data center network designs, namely concerning 1) the demand-aware or -oblivious circuit control plane and the 2) all-optical or hybrid fabric. Sirius [23], Opera [197], and RotorNet [198] all propose a demand-oblivious optical layer, in essence rotating through a set of topologies, letting the higher layers take advantage of the changing optical connections. To this end, there is no computational delay, but on the other hand, specifically skewed demands can suffer from performance degradation. Demand-aware control planes can adapt to any demands but need careful tuning to avoid scaling and prediction issues, which then again can be inferior to demand-oblivious network designs, depending on the scenario. Notwithstanding, the three listed demand-oblivious designs currently rely on specialized and experimental hardware. Regarding the choice of fabric, hybrid designs are highly beneficial for small and short-lived flows, and hence a combination of packet and circuit switching, such as in RotorNet [198] or Eclipse [286], can combine the best of both worlds.

	Fabric	Demand-Aware	Novelty
Helios [85]	Hybrid	✓	First hybrid system using WDM for busty low-latency traffic
c-Through [288]	Hybrid	✓	Enlarged buffers for optical ports increases utilization
ProjecToR [109]	Hybrid/FSO	✓	Introduces DMDs for free-space switching thus enabling a fan-out potential to thousands of nodes
Proteus [255]	All-optical	✓	Design of an all-optical and reconfigurable DCN.
OSA [50]	All-optical	✓	Demonstrates greater reconfiguration flexibility and bisection bandwidth than hybrid architectures
RotorNet [198]	Hybrid	×	An all-optical demand-oblivious DCN architecture for simplified network management
Opera [197]	All-optical	×	Extends Rotornet to include expander graphs rotations
Flat-tree [300]	Hybrid	✓	A hybrid of random graphs and Clos topologies brings reconfigurable optics closer to existing DCNs.
Solstice [177]	Hybrid	✓	Exploits sparse traffic patterns in DCNs to achieve fast scheduling of reconfigurable networks.
Eclipse [286]	Hybrid	✓	Outperforms Solstice by applying sub-modular optimization theory to hybrid network scheduling.
xWeaver [292]	Hybrid	✓	Trains neural networks to construct performant topologies based on training data from historic traffic traces.
DeepConf [240]	Hybrid	✓	Presents a generic model for constructing learning systems of dynamic optical networks
WaveCube [51]	Hybrid	✓	A modular network architecture for supporting diverse traffic patterns.
Sirius [23]	All-optical	×	Achieves nanosecond-granularity reconfiguration for thousands of nodes

Table 2. Summary of systems implementations of reconfigurable data center networks

Notwithstanding, provisioning for both types of networks leads to overheads in terms of cost and cross-fabric efficiency, and thus are not a silver bullet solution. An intriguing design in this context is Opera [197], as it always provisions a small

diameter network with optical links, emulating classic DCN properties inside their circuit choices. However, as mentioned above, this design choice comes with the price of demand-obliviousness, and it would be interesting to see how other all-optical demand-aware systems, such as e.g.OSA [50], can implement such properties as well

3.2.5 Summary. There is a wide range of data center specific technology and algorithmic ideas that enable efficient circuit switching in data center networks, with newer developments focusing on leveraging the benefits of faster circuit reconfigurations. In contrast, there has also been some recent work [269] that discusses the idea of robust topology engineering, e.g.adapting the circuits only every few minutes or even days [268]. Notwithstanding, scaling current system designs can be problematic, in particular, due to the speed of the control plane and fan-out restrictions. Whereas one solution for the latter is free-space optics, those still face significant practical deployment issues in data center contexts. On the other hand, demand-oblivious system designs inherently overcome such control plane delays, but cannot adapt well to skewed demands. In their current form, they are not available as off-the-shelf hardware. Designing scalable demand-aware reconfigurable data centers is hence one of the main next challenges.

3.3 Reconfigurable Optical Metro and Wide-area Networks

In this section, we survey recent research in reconfigurable optics in metropolitan (metro) and wide-area networks (WAN). Reconfigurable optics refers to dynamism in the physical-layer technology that enables high-speed and high-throughput WAN communications, fiber optics. We divide reconfigurable optical innovations into two sub-categories, rate-adaptive transceivers, and dynamic optical paths. Rate adaptive transceivers, or bandwidth-variable transceivers (introduced in section 2.2.3) are optical transceivers that can change their modulation format to adapt to physical

layer impairments such as span-loss and noise. Dynamic optical paths refer to the ability to *steer light*, thus allowing the edges of the network graph to change (e.g.to avoid a link that has failed).

Many groups have studied the programmability and autonomy of optical networks. Gringeri et al. [115] wrote a concise and illuminating introduction to the topic. In it, the authors propose extending Software Defined Network (SDN) principles to optical transport networks. They highlight challenges, such as reconfiguration latency in long-haul networks, and provide a trade-off characterization of distributed vs. centralized control for an optical SDN system. They claim that a tiered hierarchy of control for a multi-regional network (e.g., segregated optical and network control loops) will offer the best quality solution. Further, they argue that centralized control should work best to optimize competing demands across the network, but that the controller's latency will be too slow to react to network events, e.g.link outages quickly. Therefore, the network devices should keep some functionality in their control plane to respond to link failures in a decentralized manner, e.g.reallocating the lost wavelengths by negotiating an alternative path between the endpoints.

The question of centralized vs. distributed network control is just one example of the many interesting questions that arise when considering reconfigurable optical networks for metro and wide areas. This space is unique because many of the solutions here require understanding and sharing of information across layers of the network stack. For example, figure 11 illustrates interdependence between the objectives for communication across different layers of the stack; these features include algorithms, enabling technologies. We highlight several canonical examples of systems that exist in those domains and across different layers. In this section, we will explore these examples more deeply along with other related efforts.

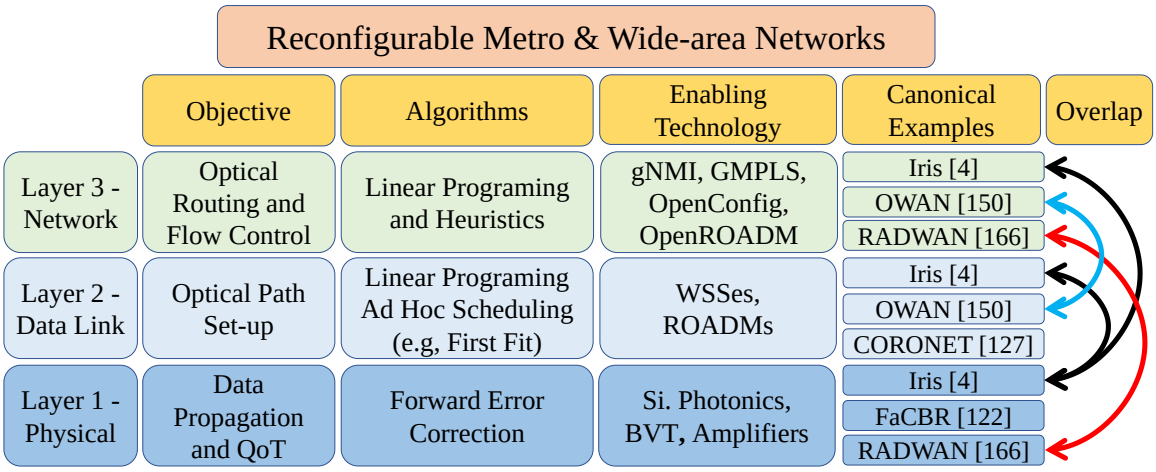


Figure 11. To deploy and operate reconfigurable optical networks in metro and wide-area networks require expertise spanning the bottom three layers of the network stack, including algorithms and enabling technology. We highlight several canonical examples of systems that exist in this space and explore other related works along with these systems more deeply in this section.

3.3.1 Metro/WAN-Specific Challenges and Solutions. There are many reasons for the prevalence of optical fiber as the de-facto leader for long-distance communications. First, it has incredible reach compared to copper—optical signals can propagate 80 to 100 km before being amplified. Second, it has an incredibly high bandwidth compared to the radio spectrum. Third, optical fiber itself has proved to be a robust medium over decades, as improvements to the transponders at the ends of the fiber have enabled operators to gain better value out of the same fiber year after year.

To design a WAN, the network architect must solve several difficult challenges, such as estimating the demand on the network now and into the future, optimal placement of routers and quantity of ports on those routers within the network, and optimal placement of amplifiers in the network.

Many design challenges solve more easily in a static WAN, where optical channels are initialized once and maintained for the network's life. For example, amplifiers

carrying the channel must have their gain set in such a way that the signal is transmitted while maximizing the signal-to-noise ratio (SNR). This calculation can take minutes or hours depending on the network's characteristics (e.g.the number of indeterminate hosts and the number of distinct channels on shared amplifiers).

Dynamic optical networks must rapidly address these challenges (in sub-second time frames) to achieve the highest possible utilization, posing a significant challenge. For example, it requires multiple orders of magnitude increases in the provisioning time for optical circuits beyond what is typically offered by hardware vendors. Therefore, several research efforts have explored ways to automate WAN network elements' configuration concerning physical layer impairments in a robust and time-efficient manner.

Chromatic Dispersion. DWDM makes efficient use of optical fiber by putting as many distinct optical channels, each identified by a frequency (or lambda λ) onto the shared fiber. Each of these lambdas travels at a different speed relative to the speed of light. Therefore, two bits of information transmitted simultaneously via two different lambdas will arrive at the destination at two different times. Further, chromatic dispersion is also responsible for pulse-broadening, which reduces channel spacing between WDM channels and can cause FEC errors. Therefore, DWDM systems must handle this physical impairment.

Amplified Spontaneous Emission (ASE) Noise. A significant limitation of circuit switching is the latency of establishing the circuit due to ASE noise constraints [58]. Although SDN principles can apply to ROADMs and WSSs (to automate the control plan of these devices), physical layer properties, such as Noise Figure (NF) and Gain Flatness (GF) complicate the picture. When adding or removing optical channels to or from a long-haul span of fiber, traversing multiple

amplifiers, the amplifiers on that path must adjust their gain settings to accommodate the new set of channels. To this end, researchers have worked to address the challenge of dynamically configuring amplifiers. Oliveira et al. [224] demonstrated how to control gain on EDFAs using GMPLS. They evaluated their solution on heterogeneous optical connections (10, 100, 200, and 400 Gbps) and modulations (OOK, QPSK, and 16-QAM). They used attenuators to disturb connections and allow their GMPLS control loop to adjust the amplifier's gains. They show that their control loop helps amplifiers to adjust while transmitting bits with BER below the FEC threshold for up to 6 dB of added attenuation.

Moura et al. [205] present a machine learning approach for configuring amplifier gain on optical circuits. Their approach uses case-based reasoning (CBR) as a foundation. The intuition behind CBR is that the gain setting for a set of circuits will be similar if similar circuits are present on a shared fiber. They present a genetic algorithm for configuring amplifiers based on their case-based reasoning assumption. They show that their methodology is suitable for configuring multiple amplifiers on a span with multiple optical channels. In a follow-up study, they present FAcCBR [206], an optimization of their genetic algorithm, which yields gain recommendations more quickly by limiting the number of data-points recorded by their algorithm.

Synchronization. Managing a WAN requires coordinating services (e.g.end-to-end connections) among diverse sets of hardware appliances (transponders, amplifiers, routers), logically and consistently. The Internet Engineering Task Force (IETF) has defined protocols and standards for configuring WAN networks. As the needs and capabilities of networks have evolved, so have the protocols. Over the years, new protocols have been defined to bring more control and automation to the network operator's domain. These protocols are Simple Network Management

Protocol (SNMP) [87] and Network Configuration Protocol (NETCONF) [79]. Additionally, network operators and hardware vendors have been working to define a set of generalized data models and configuration practices for automating WAN networks under the name OpenConfig [226]. Although OpenConfig is not currently standardized with the IETF, it is deployed and has demonstrated its value in several unique settings.

In addition to the standardized and proposed protocols for general-purpose WAN (re)configuration, there has been a push by various independent research groups to design and test protocols specifically for reserving and allocating optical channels in WAN networks.

One protocol was developed in conjunction with the CORONET [171] program, whose body of research has led to several other developments in reconfigurable optical WANs. The proposal, by *Skoog et al.* [256], describes a three-way handshake (3WHS) for reserving and establishing optical paths in single and multi-domain networks. In the 3WHS, messages are exchanged over an optical supervisory channel (OSC)—an out-of-band connection between devices isolated from user traffic. The transaction is initiated by one Optical Cross-Connect (OXC^A) and directed at a remote OXC, OXC^Z. At each hop along the way, the intermediate nodes append the available channels to the message. Then, OXC^Z chooses a channel via the first-fit strategy [315] and sends a message to OXC^A describing the chosen channel. Finally, OXC^A activates the chosen channel and beings sending data over it to OXC^Z. This protocol is claimed to meet the CORONET project standard for a setup time of 50 ms + RTT between nodes. Bit arrays are used to communicate the various potential channels between nodes and are processed in hardware. The blocking probability is 10^{-3} if there is one

channel reserved between any two OXC elements so long as there are at least 28 total channels possible between OXCs [256].

3.3.2 Cost Modeling. Fiber infrastructure for wide-area networks is incredibly costly. Provisioning of fiber in the ground requires legal permitting processes through various governing bodies. As the length of the span grows beyond metropolitan areas, to connect cities or continents, the number of governing bodies with whom to acquire the legal rights to lay the fiber grows [74]. Then, keeping the fiber lit also incurs high cost; power requirements are a vital consideration for wide-area network provisioning [130]. Therefore, reliable cost models are necessary for deploying and managing wide-area networks. In this section, we look at cost modeling efforts particularly suited for reconfigurable optical networks.

An early study on the cost comparison of IP/WDM vs IP/OTN networks (in particular: European backbone networks) was conducted by Tsirilakis et al. in [281]. The IP/WDM network consists of core routers connected directly over point-to-point WDM links in their study. In contrast, the IP/OTN network connects the core routers through a reconfigurable optical backbone consisting of electro-optical cross-connects (OXCs) interconnected in a mesh WDM network.

Capacity planning is a core responsibility of a network operator in which they assess the needs of a backbone network based on the projected growth of network usage. Gerstel et al. [106] relates the capacity planning process in detail, which includes finding links that require more transponders and finding shared-risk-link-groups that need to be broken-up, among other things. They note that in this process, the IP and Optical network topologies are historically optimized separately. They propose an improvement to the process via multi-layer optimization, considering the connection between IP and optical layers. They save 40 to 60% of the required

transponders in the network with this multi-layer approach. The networks they looked at were Deutsche Telekom [117] and Telefonica Spain core networks. These authors' work provides a strong motivation for jointly optimizing IP and Optical network layers and sharing of information between the two.

Papanikolaou *et. at.* [229] propose a cost model for joint multi-layer planning for optical networks. Their paper presents three network planning solutions; dual-plane network design, failure-driven network design, and integrated multi-layer survivable network design. They show that dual-plane and failure-driven designs over-provision the IP layer, leaving resources on the table that are only used if link failures occur. They show that integrated multi-layer survivable network design enables a significant reduction in CapEx and that the cost savings increases beyond dual plane and failure driven designs.

Cost models for evaluating C-ROADM vs. CDC-ROADM network architectures are described by Kozdrowski et al. [159]. They show that for three regional optical networks (Germany, Poland, USA), CDC-ROADM based networks can offer 2 to 3× more aggregate capacity over C-ROADM based networks. They evaluate their model with uniform traffic matrices (TMs) and apply various scalar multipliers to the TM. Their model accounts for many optical hardware related constraints, including the number of available wavelengths and cost factors associated with manual-(re)configuration of C-ROADM elements. However, their model does not include an optical-reach constraint. They limit solver computation time to 20 hours and present the best feasible solution determined in that amount of time.

Service velocity refers to the speed with which operators may grow their network as demand for capacity grows. Woodward et al. [297] tackles the problem of increasing service velocity for WANs. In this context, they assume a network of colorless non-

directional ROADMS (CN-ROADMs)[3], in which any incoming wavelength can be routed on any outgoing fiber. They claim that one of the largest impedances for network growth in these networks is the availability of *regenerators*. To solve this problem, they present three algorithms for determining regenerators' placement in a network as service demand grows. The algorithms are: locally aware, neighbor aware, and globally aware. Each algorithm essentially considers a broader scope of the network, which a node uses to determine if an additional regenerator is needed at the site at a particular time. They show, via Monte Carlo simulations, varying optical reach and traffic matrices. The broadest scope algorithm performs the best and allocates enough regenerators at the relevant sites without over-provisioning. This work shows that service velocity is improved with demand forecasting, enabling infrastructure to be placed to meet those projected demands.

Programmable and elastic optical networks can also work together with Network Function Virtualization (NFV) to offer lower-cost service-chaining to users. Optimal strategies have been demonstrated, with heuristic algorithms, to quickly find near-optimal solutions for users and service brokers by Chen et al. [54]. In their work, they take a game-theoretic approach to modeling the competition among service brokers—who compete to offer the lowest cost optical routes and service chains, and between users—who compete to find the lowest cost and highest utility service chains among the brokers. They demonstrate both parties' strategies, which converge on low-latency service chain solutions with low blocking probability for optical paths.

Modeling *opportunity cost* of optically switched paths is explored by Zhang et al. [319]. In their work, they present an algorithm for quickly evaluating the opportunity cost of a wavelength-switched path. Given a request and a set of future requests,

[3]CN-ROADMs are also called CD-ROADMs in other papers. These both refer to the same ROADM architecture.

the opportunity cost for accommodating the initial request is the number of future requests blocked as a result of the accommodation. Thus, the network operator's goal is to minimize opportunity cost by permitting connections that interfere with the fewest future requests.

3.3.3 Algorithms. Jointly optimizing both the optical and the network layer in wide-area networks leads to new opportunities to improve performance and efficiency, while introducing new algorithmic challenges. In contrast to the previously discussed data center networks, it is impossible to create new topological connections in a wide-area network (without deploying more fiber. Free space optics solutions do not apply here). Instead, reconfigurability is possible by adjusting and shifting bandwidth capacities along the fiber edges, possibly over multiple hops. Hence, we need a different set of algorithmic ideas that optimize standard metrics such as throughput, completion time, blocking probability, and resilience. In this section, we discuss recent papers that tackle these issues, starting with some earlier ones. Moreover, there is the need for some central control to apply the routing, policy, lightpath etc. changes, for which we refer to recent surveys [273].

Routing aspects are explored intensively in this context. Algorithmic approaches to managing reconfigurable optical topologies have been studied for a decade, but are recently gaining new attention. Early work by Kodialam et al. [156] explores IP and optical wavelength routing for a series of connection requests. Their algorithm determines whether a request should be routed over the existing IP topology, or if a new optical path should be provisioned for it. Brzezinski and Modiano [39] leverage matching algorithms and Birkhoff–von Neumann matrix decompositions and evaluate multi- versus single-hop routing[4] in WDM networks under stochastic traffic. However,

[4]See also the idea of lightpath splitting in Elastic Optical Networks [323].

the authors mostly consider relatively small networks, e.g.with three to six nodes. For larger networks, shortest lightpath routing is a popular choice [231]. Another fundamental aspect frequently considered in the literature regards resilience [49, 193, 303]. For example, Xu et al. [303] investigate resilience in the context of shared risk link groups (SLRGs) and propose a method on how to provision the circuits in a WAN. To this end, they construct Integer Linear Programs to obtain maximally SLRG-diverse routes, which they then augment with post-processing for DWDM system selection and network design issues. We now introduce further selected algorithmic works, starting with the topic of bulk transfers [187].

In *OWAN* [140], Jin et al. optimize bulk transfers in a cross-layer approach, which leverages both the optical and the network layer. Their main objective is to improve completion time; while an integer linear program formulation would be too slow, the authors rely on a simulated annealing approach. A local search shifts wavelength allocations, allowing heuristic improvements to be computed at a sub-second scale. The scheduling of the bulk transfer then follows the standard shortest job first approaches. When updating the network state, if desired, *OWAN* can extend prior consistent network update solutions [141] by introducing circuit nodes in the corresponding dependency graphs. *OWAN* also considers deadline constrained traffic, implementing the earliest deadline first policy. Follow-up work extended *OWAN* in two directions, via theoretic scheduling results and for improvements on deadline-constrained transfers.

In *DaRTree* [185], Luo et al. develop an appropriate relaxation of the cross-layer optimization problem for bulk transfers under deadlines. Their approach relies on a non-greedy allocation in an online setting, which allows future transfers to be scheduled efficiently without needing to reallocate currently utilized wavelengths. To

enhance multicast transfers (e.g.for replication), they develop load-adaptive Steiner Tree heuristics.

Jia et al. [138] design various online scheduling algorithms and prove their competitiveness in the setting of *OWAN* [140]. The authors consider the minimum makespan and sum completion time, analyzing and extending greedy cross-layer scheduling algorithms, achieving small competitive ratios. Dinitz and Moseley [71] extend the work of Jia et al. by considering a different objective, the sum of flow times in an online setting. They show that resource augmentation is necessary for acceptable competitive bounds in this setting, leading to nearly (offline) optimal competitive ratios. While their algorithms are easy to implement (e.g.relying on ordering by release time or by job density), the analysis is complicated and relies on linear program relaxations. Moreover, their algorithm also allows for constant approximations in the weighted completion time setting, without augmentations.

Another (algorithmic) challenge is the integration of cross-layer algorithms into current traffic engineering systems. Such TEs are tried and tested, and hence service providers are reluctant to adapt their designs. To this end, Singh et al. [251] propose an abstraction on how dynamic link capacities (e.g.via bandwidth variable transceivers) can be inserted into classic TEs. Even though the TE is oblivious to the optical layer, an augmentation of the IP layer with fake links enables cross-layer optimization via the TE. A proposal [252] for a new TE for such dynamic link capacities is discussed in the next Section 3.3.4. Singh et al. [251] also discuss consistent update methods [96] for dynamic link capacities, which Tseng [280] formalizes into a rate adaption planning problem, providing intractability results and an LP-based heuristic.

OptFlow [98] proposes a cross-layer abstraction for programmable topologies as well, but focuses on shifting wavelengths between neighboring fibers. Here, the abstraction concept is extended by not only creating fake links but also augmenting the traffic matrix with additional flows. As both links and flows are part of the input for TEs, *OptFlow* enables the compilation of optical components into the IP layer for various traffic engineering objectives and constraints. Concerning consistent updates, classic flow-based techniques [96] carry over, enabling consistent cross-layer network updates too.

Optimizing reconfigurable optical networks for circuit provisioning and per-flow rate allocation is a complex and challenging endeavor; the static routing and wavelength allocation problem is NP-complete [57]. Recent work by Guo et al. [118] explores the potential for an artificial intelligence (AI) implementation of a network controller using deep-learning. They describe a network control agent based on deep-learning which determines where and when to activate and deactivate a limited set of circuits given a snapshot of demand between hosts in the network. They also explore inherent drawbacks and precautions to consider settings in which such an agent is deployed. Their study offers insights for the potential benefit of an AI-assisted optical network controller, and novel challenges to consider for their given model.

Algorithms that optimize optical network topology for higher-layer applications, such as virtual network functions (VNF) have recently gained attention. In particular, VNF network embedding (VNF-NE) has been studied by various groups [258, 294]. VNFs are an abstraction of resources in networks that have traditionally been deployed as hardware devices (e.g.intrusion detection systems, firewalls, load-balances, etc.). Now, instead of monolithic hardware appliances many of these devices are deployed as software on commodity servers, giving more flexibility

to add and remove them at will and yielding cost-savings for network operators. Network embedding is a physical layer abstraction for creating end-to-end paths for network applications or network function virtualization (NFV) service chains. Paths have requirements for both bandwidth and CPU resources along the service chain. Wang et al. [294] proves this problem to be NP-complete for elastic optical networks. Soto et al. [258] provides an integer linear program (ILP) to solve the VNF-NE problem. The ILP solution is intractable for large networks. Thus they provide a heuristic that uses a ranking-system for optical paths. Their heuristic ranks optical paths by considering a set of end-to-end connection requests. Paths with higher rank satisfy a more significant proportion of the demand for bandwidth and CPU among all of the requests.

Optical layer routing with traffic and application constraints is a difficult problem. The running theme has been that linear programming solutions can find provably optimal solutions [227], but take too long to converge for most use cases. However, network traffic is not entirely random and therefore has an underlying structure that may be exploited by offline linear program solvers, as shown by Kokkinos et al. [157]. They use a two-stage approach for routing optical paths in an online manner. Their technique finds periodic patterns over an epoch (e.g., daily, weekly, or monthly) and solves the demand characterized within the epoch with an offline linear program. Then, their online heuristic makes changes to the topology to accommodate random changes in demand within the epoch.

3.3.4 Systems Implementations. The integration of reconfigurable optics with WAN systems has been impracticable due to its cost and a lack of convergence on cross-layer APIs for managing the WAN optical layer with popular SDN controllers. However, some exciting work has demonstrated the promise for reconfigurable optics

	BVT	Network Design	Amps.	Algorithms
CORONET [171]	×	×	×	ROLEX protocol
OWAN [140]	×	×	×	Simulated Annealing
FACcBR [205]	×	×	✓	Case Based Reasoning
RADWAN [252]	✓	×	×	Linear Program
DDN [27]	×	✓	✓	Time-slotted packet scheduling
Iris [72]	×	✓	✓	Shortest path for any failure scenario
Shoofly [250]	✓	✓	✓	Linear programming

Table 3. Summary of systems implementations of reconfigurable wide area networks

in closed settings. Notably, RADWAN [252] and CORONET [171] for bandwidth-variable WAN systems and systems with dynamic optical paths, respectively. In this section, we explore reconfigurable optical WAN systems more deeply in these two contexts. Table 3 summarizes these systems.

Bandwidth Variable Transceivers. A team of researchers at Microsoft evaluates bandwidth variable transponders' applicability for increased throughput in Azure's backbone in North America [93]. They find that throughput for the WAN can increase if they replace the fixed-rate transponders in their backbone network with three-way sliceable transponders. They also show that for higher-order slices, bandwidth gran increases at diminishing returns.

Traffic Engineering with rate-adaptive transceivers was recently proposed by Singh et al. [252]. The authors are motivated by a data-set of Microsoft's WAN backbone Signal-to-Noise ratio from all transceivers in the North-American backbone, over two and a half years. They note that over 60% of links in the network could operate at 0.75× higher capacity and that 25% of observed outages due to SNR drops could be mitigated by reducing the modulation of the affected transceivers. They evaluate the reconfigurability of Bandwidth-Variable Transponders, showing that reconfiguration time for the transceivers could be reduced

from minutes to milliseconds by *not* turning the transceivers off. Then, they propose a TE objective function via linear-programming, to minimize churn, or impact due to SNR fluctuations, in a WAN. Finally, they evaluate their TE controller on a testbed WAN and show that they improve network throughput by 40% over a competitive software-defined networking controller, SWAN [129]. In 2021, Singh et al. [250] proposed Shoofly, for dynamic capacity provisioning in wide-area networks. Their system uses a linear programming optimization solver to find 'shortcut' tunnels through a WAN and makes these tunnels available to a central traffic engineering controller. They efficiently solve the mixed integer linear programming optimization by allowing a 0.1% gap for an optimal solution, and find that this relaxation allows them to consistently find a feasible solution in 10 seconds or less. They find that Shoofly can save WAN hardware costs by 40% without impacting network traffic performance.

Dynamic Optical Paths. In the early 2000s, researchers explored the benefit of dynamic optical paths for networks in the context of *grid-computing*. Early efforts by Figueira et al. [92] addressed how a system might manage dynamic optical paths in networks. In this work, the authors propose a web-based interface for submitting optical reconfiguration requests and a controller for optimizing the requests' fulfillment. They evaluate their system on OMNInet [29], a metropolitan area network with 10 Gbps interconnects between 4 nodes and Wavelength Selective Switches between them. They claim that they can construct optical circuits between the OMNInet nodes in 48 seconds. Further, they show that amortized setup time and transfer is faster than packet-switching for files 2.5 Gb or larger (assuming 1 Gbps or greater optical interconnect and 300 Mbps packet switching throughput). They go on to evaluate file transfer speeds using the optical interconnect and show that they can

achieve average transfer speeds of 680 Gbps. Iovanna et al. [134] address practical aspects of managing multilayer packet-optical systems. They present a set of useful abstractions for operating reconfigurable optical paths in traffic engineering using an existing management protocol, GMPLS.

Stability is an important feature of any network. An interesting question about reconfigurable optical networked systems arises regarding the stability of optically switched paths. That is if the topology can continuously change to accommodate random requests, what service guarantees can the network make? Can the fluctuation of the optical layer be detrimental to IP layer services? Chamania et al. [44] explore this issue in detail, providing an optimal solution to keep quality of service guarantees for IP traffic while also improving performance beyond static optical layer systems.

Blocking probability is a crucial metric for assessing the flexibility of an optical network. It is the probability that a request for an end-to-end lightpath in the network cannot be provisioned. Turkcu et al. [283] provides analytical probability models to predict the blocking probability in ROADM based networks with tunable transceivers and validate their models with simulation considering two types of ROADM architecture in their analysis, namely *share-per-node* and *share-per-link*. In share-per-link, each end of a link has a fixed number of transponders that can use it. In share-per-node, a node has a fixed set of transponders that may use any incident links. The authors show that a low tunable range (4 to 8 channels, out of 32 possible) is sufficient for reducing blocking probability in two topologies, NSF Net (14 Nodes), and a ring topology with 14 nodes. As the tunable range moves beyond 8 and up to 32, there is little to no benefit for split-per-node and share-per-link architectures. As the load on the network increases, blocking probability increases, as well as the gap between blocking probability of split-per-node and split-per-link decreases.

Bandwidth-on-demand (BoD) is an exciting application of reconfigurable networks. Von Lehmen et al. [171] describe their experience in deploying BoD services on CORONET, DARPA's WAN backbone. They implement protocols for add/dropping wavelengths in their WAN with a novel 3-way-handshake protocol. They demonstrate how their system can utilize SWAN [129] Traffic Engineering Controller as one such application that benefits from the BoD service.

More recently, there has been a resurgence of academic work highlighting the potential benefit of dynamic optical paths in the WAN. One such system, called OWAN (Optical Wide-Area Network) [140], proposes how to use dynamic optical paths to improve the delivery time for bulk transfers between data centers. They build a testbed network with home-built ROADMs and implement a TE controller to orchestrate bulk transfers between hosts in a mesh optical network of nine nodes. They compare their results with other state-of-the-art TE systems, emphasizing that OWAN delivers more transfers *on time* than any other competing methods.

Dynamic optical paths increase the complexity of networks and capacity planning tasks because any optical fiber may need to accommodate diverse and variable channels. However, this complexity is rewarded with robustness or tolerance to fiber link outages. Gossels et al. [113] propose dynamic optical paths to make long-haul networks more robust and resilient to node and link failures by presenting algorithms for allocating bandwidth on optical paths dynamically in a mesh network. Their objective is to protect networks from any single node or link failure event. To this end, they present an optimization framework for network planners, which determines where to deploy transponders to minimize costs while running a network over dynamic optical paths.

Another effort in reducing the complexity of dynamic optical path WAN systems was presented by Dukic et al. [72]. Their system, *Iris*, exploits a unique property of regional connectivity, i.e.the vast abundance of optical fiber in dense metropolitan areas [190]. They find that the complexity of managing dynamic optical paths is greatly reduced when switching at the fiber-strand level versus the (sub-fiber) wavelength level. To this end, they detail their design trade-off space for inter-data center connectivity across metropolitan areas. They deploy their system in a hardware testbed to emulate connectivity between three data centers, verifying that optical switching can be done in 50 to 70 ms over three amplifiers. They obviate amplifier reconfiguration delays by conducting fiber-level switching rather than wavelength-level. Thus, the amplifiers on a fiber path are configured once for the channel that traverses it. When a circuit changes its path, away from one data center and towards another, it uses a series of amplifiers that have been pre-configured to accommodate the loss of that given circuit.

Inter-data center network connectivity over a regional optical backbone was also investigated by Benzaoui et al. [27]. Their system, *Deterministic Dynamic Network (DDN)*, imposes strict constraints for application layer latency and jitter. They show that they can reconfigure optical links in under 2 ms, and guarantee consistent latency and jitter through their time-slotted scheduling approach.

3.3.5 Summary. Reconfigurable optics for metro and wide-area networks have gained substantial attention in the last decade. This push requires cross-domain collaboration as demand aware changes at the optical layer are influenced by physical-layer impairments (signal-loss, chromatic dispersion, noise, etc.), in addition to higher-layer performance metrics (latency, demand, congestion, etc.). There are various novel works that have addressed several fundamental questions in

reconfigurable optical networks. Cost-modeling efforts predict network performance with various classes of reconfigurable hardware. Algorithmic work suggests efficient methods for efficiently managing network layer and optical layer elements in the face of shifting traffic demands. Researchers have proposed and prototyped several systems for reconfigurable optical networks in recent years, but much of this work is still in the design and proof-of-concept phase. All in all, there are still many open challenges ahead to widely deploy and efficiently utilize reconfigurable optics in production networks, as we discuss next.

3.4 Open Challenges in Reconfigurable Optical Networks

Hardware technologies. The development of hardware for reconfigurable optical networking is a burgeoning field in engineering and research. While CDC-F ROADMs exist today, they are costly to produce, and their capabilities are found lacking. In particular, the benefit of integrating CDC-F ROADMs with optical transport networks is limited by cascading fiber impairments, signal loss at WSS modules, and wavelength and fiber collision [174]. We expect silicon photonics to bring down the cost of transport hardware, thereby increasing access to such devices and lowering entry barriers for research and development.

Data center networks. Our understanding of algorithms and topologies in reconfigurable networks is still early, but first insights into efficient designs are being published. One front where much more research is required concerns the modeling (and dealing with) reconfiguration costs. Indeed, existing works differ significantly in their assumptions, even for the same technology, making it challenging to compare algorithms. Related to this is also the question of how reconfigurations affect other layers in the networking stack, and how to design (distributed) controllers. In terms of algorithms, even though a majority of problems are intractable to solve

optimally, due to integral connection constraints, the question of approximation guarantees is mostly open. For example, consider designing a data center with minimum average weighted path length. A logarithmic approximation is easy to achieve by simply minimizing the diameter of a (constant-degree) static topology. However, computing an optimal solution is NP-hard. So, can we obtain polynomial approximation algorithms with constant performance trade-offs? Similarly, do good (fixed) parameter characterizations enable efficient run times, and what can we expect from e.g.linear time and distributed algorithms? Moreover, beyond general settings, how do specific network designs enable better algorithms, and how does their design interplay with topologies of the same equipment cost?

Next, going beyond scheduling, how can the framework of online algorithms be leveraged in this context? Ideally, we want a reconfigurable link to exist *before* the traffic appears. How can we balance this from a worst-case perspective? In this context, traffic prediction techniques might reduce the possible solution space massively, but we will still need extremely rapid reaction times to new traffic information.

Another open challenge is the efficient interplay between reconfigurable and non-reconfigurable network parts. Theory for specific reconfigurable topologies (e.g.traffic matrix scheduling for a single optical switch) has seen much progress. However, more general settings, particularly non-segregated routing onto both network parts, are still an open issue, beyond an abstract view of the combination with a single packet switch.

Metro and Wide-area Networks. Metro and wide-area optical networks are rich with open challenges. The works presented in this section highlight significant developments that have been made towards reconfigurable WAN systems

and illuminate great benefits for such systems. However, programmability, cross-layer information sharing, and physical properties of light still must be solved. On the programmability front, efforts such as OpenConfig [226], OpenROADM [223], and ONOS [28] are working to provide white-box system stacks for optical layer equipment. If these are widely adopted and standardized, this will open the door for agile and efficient use of wide-area networks for a variety of applications (e.g.new tools to combat DDoS [213]). Other challenges include wrangling with the physical constraints of efficient and rapidly reconfigurable WANs, for example, coordination of power adjustments across amplifiers for long-haul circuits.

CHAPTER IV

FOUNDATIONS FOR OPTICAL TOPOLOGY PROGRAMMING (OTP)

This chapter, particularly § 4.2, contains previously published coauthored material from [210], with coauthors Paul Barford, Klaus-Tycho Foerster, and Ramakrishnan Durairajan. Klaus-Tycho Foerster and the dissertation author wrote Theorem 1 together. Paul Barford and Ramakrishnan Durairajan assisted with editing. § 4.3 contains previously unpublished work that is scheduled to appear in [217] and coauthored with Zaoxing (Alan) Liu, Vyas Sekar and Ramakrishnan Durairajan. The dissertation authors wrote the linear programming model defined in this section. The coauthors assisted with editing. The dissertation author designed and implemented the simulator described in § 4.4.

4.1 Introduction

Historically, enterprise network operation has progressed on two divergent trajectories simultaneously. On one track, the optical layer of the network has evolved to enable high throughput links between network endpoints that have expanded well beyond the terabit range. On the other track, the IP network layer has evolved with the advent of software defined networking, which offers flexibility and fine-grained forwarding behavior of traffic on these optical links. However, emerging applications such as machine learning and content streaming are pushing networks to run closer to their limits [78, 167], and as a result, we are amidst a paradigm shift in networking. The data and control planes have been decoupled. White box, programmable switches are replacing one-size-fits-all proprietary black-box models. Software-defined networking principles have matured, and networks have largely benefited as a result. This has enabled operators to scale capacity in their networks, for both wide-area settings and data centers [130, 248]. As demand on networks

continues to grow, it has become apparent that a jointly optimized optical-packet architecture can give networks more power to serve their demand better [115].

Joint optimization of the packet and optical networks offers greater performance (e.g., throughput, latency, utilization), however, there is still no unified system for operating both networks simultaneously. This is primarily because the optical layer remains invisible to higher-layer network management systems. Another reason for this shortcoming is that there are fundamental questions open at the intersection of the packet and optical networks, such as how fast optical network switching can take place in the presence of amplifiers, and how does optical switching affect traffic, both on a switched lambda and *witness waves* (the set of lambdas that were actively transmitting data on the fiber when the switched lambda was introduced or removed). Moreover, jointly optimizing the network topology with routing is an NP-Hard problem [160] and therefore finding scalable solutions has remained an elusive challenge. As a result, the research efforts driven by optical networking and packet/networked systems communities are in disagreement about what services the optical layer can offer, and how these services can be utilized by higher layers of the protocol stack [102]. This optical-packet *chasm* is no accident, as the Internet was designed to be built of independent and logically separated abstract layers.

Uniquely, this thesis looks at jointly programming the network topology in conjunction with routing. We present an optimization method for joint optimization of topology and routing that is made scalable by aggressively limiting the search space for potential solutions based on the set of potential forwarding links and the set of forwarding paths among those links. More details about this approach are to be found in the application-centric chapters that leverage this technique,

namely chapters VII and VIII. The general constraints common to the different implementations are discussed in this chapter, § 4.3.

We evaluate three network applications as they stand to benefit from OTP, namely traffic engineering, network reconnaissance subversion, and DDoS attack mitigation. We evaluate these models with a custom-built OTP simulator, introduced in § 4.4.

4.2 Formal Model and Theoretical Guarantees of OTP

Due to the complexity that optical topology programming introduces to network operation and management, and the uncertain benefits it enables, the problem space is lacking in formal, theoretical guarantees regarding its application. Therefore, in this section we aim to formally prove tight bounds on the gains of optical topology programming.

As a preliminary step, we first give a precise model setting for the case of regeneration and wavelength conversion at each node.

Physical host graph: We model the physical host graph as $G = (\mathcal{V}, \mathcal{L}, \sigma)$, with nodes $\mathcal{V}$ and (multi-)edges $\mathcal{L}$. We assume that G is connected, that nodes correspond to ROADMs and edges correspond to physical fibers. Each node $v \in \mathcal{V}$ has two attributes: (i) v) is the degree of node v, i.e., the number of incident edges (fibers), (ii) $\sigma(v)$, $\sigma : \mathcal{V} \to \mathbb{N}$ is the total number of transponders in v that can be allocated to v's edges. Depending on the modulation technology, each fiber edge has an attribute, $\mu(e)$, that corresponds to the maximum possible wavelengths on the fiber.[1]

Wavelength allocation: In order to route traffic on an edge $e = (v, w)$ in G, we need to assign wavelengths to e. In optical communications, a transponder is a device that sends and receives the optical signal on a fiber. Each wavelength requires a transponder on the sending node and receiving node. Although fiber is

[1]E.g., $\mu(e) = 120$ wavelengths for QPSK modulation with 37.5 GHz spacing.

unidirectional, today's transponders enforce bidirectionality. Hence, reconfiguring a wavelength between v and w requires the reverse path to be reconfigured from w to v.[2] Given this, we define a wavelength allocation Λ of a graph G by $\Lambda : \mathcal{V} \times \mathcal{L} \rightarrow \mathbb{N}$ representing the number of wavelengths allocated on each edge by the nodes.

In finding a new wavelength allocation, nodes are limited by the pool of transponders they have ($\sigma(v)$). Hence, the total number of wavelengths on each edge, $c(e)$, cannot exceed the number of available transponders on its neighboring nodes; i.e., $\sum_{e=(v,w)} \Lambda(v,e) \leq \sigma(v),\ \forall v \in \mathcal{V}$ and $c(e) = \max(\Lambda(v,e), \Lambda(w,e), \mu(e))$. We denote G_Λ when wavelength allocation Λ is applied to G.

Static WAN: The state-of-the-art in binding wavelengths to fibers is a static allocation based on the history of traffic demand, growth prediction, and failure resiliency. Once a wavelength allocation is set up, it does not change for months. We assume the static topology is an optimal wavelength allocation able to route traffic demands under failure resiliency, while minimizing the maximum utilization, and that each fiber can be populated with wavelengths. Without these assumptions, it is easy to fabricate unreasonably large savings factors, e.g., by comparing with allocations that are ineffective on purpose or cannot survive fiber cuts.

Utilization and throughput gain factor: We define the gains Y of moving from a static to a dynamic capacity WAN by $Y = \max_{D \in \mathcal{D}, F \in \mathcal{F}} \frac{T(G_\Lambda, \mathcal{D}, F)}{T(G_*, D, F)}$, where $T(G_*, D, F)$ is the maximum link utilization in the static topology G_*, for demand matrices $D \in \mathcal{D}$ and for failure scenarios $F \in \mathcal{F}$. Similarly, $T(G_\Lambda, D, F)$ is the maximum link utilization in the dynamic wavelength allocation Λ obtained by reprogramming the wavelengths with respect to the demand matrix D and failure scenario F. Note that G_* and G_Λ share the same physical graph G; the difference

[2]Recent work shows this assumption is not optimal; there are gains in building a unidirectional WAN using unidirectional transponders [318], but we leave this discussion for future work.

lies in the ability to reallocate wavelengths after failures. We define the gains for total throughput analogously. Lastly, in this setting, we only allow survivable failure scenarios F where the network is not physically disconnected.

We provide bounds on the throughput and utilization gains of OTP in the setting where each node in the physical network graph employs transponders that terminate the optical link and regenerate the signal.

Theorem 1. *Given a physical graph G, the utilization (and throughput) gain factor Y is bounded by $1 \leq Y \leq O(\Delta)$, where Δ is the maximum degree in G. This bound holds for any topology, under any survivable edge failure scenario.*

Proof of Theorem 1. Recall that an optimal *static* wavelength allocation (our "competition") minimizes the gain factor of reconfiguration. Consider a wavelength allocation Λ_A, where each node v distributes its $\sigma(v)$ transponders evenly over all neighbors, possibly wasting up to $d(v) - 1 \leq \Delta - 1$ transponders to obtain identical numbers for all neighbors. Furthermore, even more transponders can be wasted due to μ restricting the number of wavelengths possible. Λ_A cannot be better than an optimal static wavelength allocation as Λ_A is feasible, i.e., we have a lower bound on static performance. Next, consider a wavelength allocation Λ_B, which we obtain as follows: We begin with Λ_A, but multiply every transponder assignment of a node to a neighbor by $2\Delta - 1$. Observe that Λ_B does not have to be feasible, but it clearly can satisfy any flows feasible in Λ_A. Furthermore, assume there is a feasible Λ_C which results in a better output for the objective function than on Λ_B: this leads to a contradiction as any flow routing or utilization feasible in Λ_C is also feasible in Λ_B. Lastly, assume Λ_B has a gain of $X > 2\Delta - 1$ compared to Λ_A for some demand and some failure scenario (possibly empty). Then, we take all flows Λ_B, dividing their

size by $2\Delta - 1$, meaning they are feasible in Λ_A, but due to $X > 2\Delta - 1$ we obtain a contradiction, analogously for the utilization. □

We observe that the result of Theorem 1 cannot be improved with respect to its dependency on Δ, the maximum degree of a single node in the network, and briefly sketch the reasoning next. Consider traffic matrices that change between the outgoing links of a central node. A static allocation has to distribute its wavelengths along all neighbors, whereas a dynamic allocation can shift all allowed wavelengths to just one neighbor at a time. The argument can made made analogously for fiber cuts. Hence, there are cases where the gain $\Omega(\Delta)$ matches the upper bound of $O(\Delta)$ and note that this example can be generalized beyond a single central node.

Theorem 1 tells us intuitively that OTP has greater benefits for networks with higher degree nodes. A corollary to this is that within a single network, the benefits of enabling OTP are greatest at nodes in the network whose degrees are greatest.

	Constants
G_0	Initial topology
E_0	Initial set of (directional) links
$\mathcal{E}$	Set of potential (directional) links (includes E_0)
V	Set of all nodes
D	Set of all demands
$\texttt{txp}(v)$	Transponders at v
	Variables
b_e or $b_{(u,v)}$	Binary link-status variable
$\mathcal{F}^{s \to t}$	The set of links that are available to any potential path $s \to t$
$\texttt{flow}^{s \to t}_{(u,v)}$	Flow allocated from $D(s,t)$ onto edge (u,v) s.t. $(u,v) \in \mathcal{E}$
$\texttt{in}(n)$	Total flow from all demands going into node n
$\texttt{out}(n)$	Total flow from all demands departing node n
$\texttt{cap}(u,v)$	Capacity of edge (u,v)

Table 4. Reference for notation, variables, and constants in equations 4.1–4.7.

4.3 Optimization

In this section, we present general constraints that we apply to networks when developing OTP applications. This set of linear programming constraints comprises topology connectivity rules (4.1–4.4) which we introduce to the typical network flow conservation rules (4.5–4.7) that are observed in many modern traffic engineering systems [2, 129, 137, 166, 179]. Table 4 shows the descriptions of variables used in this model.

$$\forall (u, v) \in E, \texttt{cap}(u, v) = \texttt{cap}(v, u) \tag{4.1}$$

The capacity of each directional link, (u, v), is symmetrical. This ensures that a link is only active if it can be activated in each direction.

$$\forall n \in V, \texttt{txp}(n) \geq \sum_{u \in b_{(u,v)}} u \tag{4.2}$$

The total number of fallow transponders at a node, $\texttt{txp}(n)$ limits the total number of links in the topology that can start from n.

$$\forall e \in \mathcal{E}, \texttt{cap}(e) = b_e C_e \tag{4.3}$$

An edge's capacity is C_e or 0, where C_e is capacity of a network link when edge e is active in the network.

$$\forall (s, t) \in D, \sum_{e \in \mathcal{F}^{s \to t}} \texttt{flow}_e^{s \to t} \leq \texttt{cap}(e) \tag{4.4}$$

The sum of flows allocated from all demands allocated onto an edge must be bound by the capacity of that edge. Note that in the constraint, the only edges considered for a demand, $s \rightarrow t$, are limited to $\mathcal{F}^{s \to t}$ rather than the entire set of links $\mathcal{E}$. In practice, we employ the link selection and path finding strategies described in Section 8.4.1 to

find the appropriate set of candidate links for each pair of nodes in the graph induced by all possible edges, $\mathcal{E}$.

$$\forall n \in V, \forall (s,t) \in D,\ \texttt{in}(n) = \sum_{(u,v) \in \mathcal{E} | n=u} \texttt{flow}_{(u,v)}^{s \to t} \tag{4.5}$$

$$\forall n \in V, \forall (s,t) \in D,\ \texttt{out}(n) = \sum_{(u,v) \in \mathcal{E} | n=v} \texttt{flow}_{(u,v)}^{s \to t} \tag{4.6}$$

$$\forall n \in V, \forall (s,t) \in D, \begin{cases} d(s,t) + \texttt{in}(n) = \texttt{out}(n) & \text{if } n = s \\ \texttt{in}(n) = d(s,t) + \texttt{out}(n) & \text{if } n = t \\ \texttt{in}(n) = \texttt{out}(n) & \text{otherwise.} \end{cases} \tag{4.7}$$

Constraints (4.5–4.7) are general multi-commodity flow optimization constraints [81] and ensure conservation of flow along paths through the network.

4.4 OTP Simulator

Evaluating OTP requires access to a WAN backbone which we do not have. To address this challenge, we have constructed a Python-based discrete event simulator, the *OTP simulator*. While TE simulators in recent work [2,165] have taken topology as a fixed input to show how routing decisions affect performance as a function of the traffic, our OTP simulator aims to show how topology *and* routing decisions affect performance as a function of traffic.

Our OTP simulator is designed with the following goals in mind. The first is to parameterize low-level network topology features, e.g., the number of transponders at network nodes and the pairing of transponders between nodes. The second is to integrate OTP into the network operator's control loop. The third is to enable the prototyping of different OTP methods in conjunction with different high level network applications. The simulator is written with $\sim 21k$ lines of Python 3 code[3].

[3]OTP Simulator is openly accessible at `https://github.com/mattall/topology-programming`

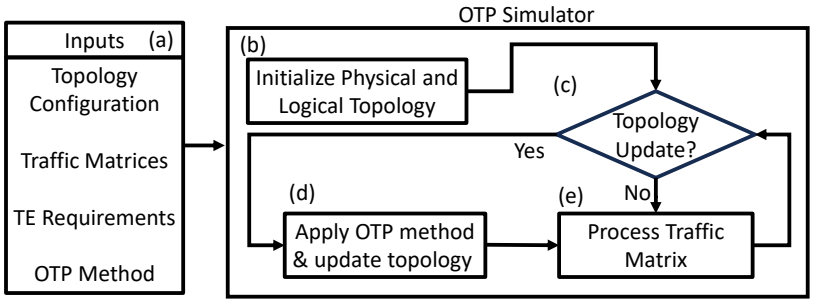


Figure 12. OTP Simulator

The high level architecture of the OTP simulator is shown in Figure 12. The simulator receives the following inputs: topology configuration, traffic matrices, TE requirements, and an OTP Method (a). The topology configuration contains the low level details about the network that are required to invoke any OTP method or adhere to any TE requirements. These include the set of nodes in the network, the set of physical links between nodes, the number of transponders at each node, and the capacity of links that are activated by a corresponding pair of transponders. The traffic matrices give the demand between any two nodes in the network over a fixed interval of time. This is naturally a time-series of one or more matrices. The TE requirements describe how traffic is forwarded across the active links in the network. For example, this can be an optimization model with an explicit set of constraints and objective function such as minimizing max link utilization with multi-commodity flow optimization, or a simpler requirement such as equal-cost multi-path routing. The OTP method describes how the topology is reconfigured before and after the simulator defines the set of forwarding paths and rates among paths for a traffic matrix from the time-series of traffic matrices. The OTP can invoke its own unique set of TE requirements.

The initialization of the physical and logical topology (b) uses the initial topology configuration to instantiate a set of logical forwarding links in the network that are

consistent with the physical resources described in the configuration. Each node in the network has a hash map with an entry for every transponder at that node. The hash map either holds the ID of the remote node to which that transponder is assigned or `-1` if the transponder has no assignment[4].

Depending on the OTP method, the topology can be updated before or after a traffic matrix is processed. For example, a topology update might be triggered by the intrinsic properties of the logical graph of the network (e.g., edge betweenness of a network link is too high or too few edge-disjoint paths between a single pair of nodes), or by a network event (e.g., max link utilization is too high, or traffic loss has occurred).

If a topology update is triggered the OTP method is invoked and the topology is updated (d). The OTP method uses the information from the topology configuration to determine how to reconfigure the topology. This is a user designed method that is application specific. It could involve solving a linear programming optimization problem, or applying a simple set of heuristics based on the resources available and active network conditions.

After a topology update decision is made and whether or not the OTP method is invoked, the traffic matrix is processed according to the TE requirements (e). In this step the set of forwarding paths are defined for every pair of nodes, and in turn, the forwarding rate for flows among those paths. The demand forwarded across ever link is added together to determine the aggregate utilization for each link, as well as the latency, throughput, and loss for all of the flow demands in the traffic matrix.

The loop starts again at the topology update decision as long as there are still traffic matrices to be processed. When the last traffic matrix is processed the

[4]Valid node IDs are integers that are zero or greater.

simulation terminates and a summary of the network state (latency, throughput, loss, congestion) for each traffic matrix is saved to a file.

CHAPTER V

MEASUREMENTS

This chapter contains previously published coauthored material from [211], coauthored with Paul Barford, Klaus-Tycho Foerster, Manya Ghobadi, William Jensen, and Ramakrishnan Durairajan. The dissertation configured the lab testbed, and designed and ran all the experiments on the testbed. Paul Barford, Manya Ghobadi, and Ramakrishnan Durairajan contributed to discussions around the goals of the measurement analysis and assisted in editing. William Jensen assisted in facilitating physical access to the testbed location and in the configuration of the lab equipment.

5.1 Introduction

As the world's online services (e.g., AI, ML) migrate onto the cloud, demands on optical layer will continue to grow. In response, inspired by reconfigurable topologies in data centers [19,52,85,100,109,147,176,177,180,198,234,324], concerted efforts have been pursued to reconfigure the optical layer [140, 204, 215, 229]. Further, the recent development of OpenConfig [226] will enable a more flexible and programmable optical layer. Such an environment serves as a starting point for physical-to-network layer coordination via *Optical Topology Programming* (OTP), i.e., the ability to quickly and flexibly reconfigure wavelengths between endpoints in an optical network.

While a programmable optical layer is poised to benefit the higher layers of the network stack, the jury is out regarding whether wide-area networks (WANs) are ready. On the one hand, some believe that the optical layer is OTP-ready and point to the theoretical efforts and optimization techniques for a programmable physical layer [56,138,205,229]. On the other hand, others argue that OTP cannot be achieved

in today's WANs due to pragmatic issues (e.g., reconfiguration delay imposed by amplifiers) at the optical layer.

To shed light on the pragmatic issues, we empirically measure the reconfiguration delays imposed by optical equipment and automated test schemes in WANs. To this end, we conducted experiments using standard optical gear (including optical amplifiers connected via spools of single-mode fiber) deployed in today's operational backbones to (a) highlight the technical challenges associated with practically realizing OTP and (b) establish a baseline for the time required for light paths to stabilize (i.e., to be ready for sending data after wavelengths are added or removed from an optical path). Our experiments show that 2–6 *minutes* are typically required for light paths to stabilize when equipment is operated with *standard automated test and adjustment* features. Most importantly, our experiments highlight the fact that many of the features *unnecessarily stretch* the reconfiguration time. This leads to our conclusion that the WANs—operated based on standard best practices—are *not* ready for OTP.

We find that automated test and adjustment features impose a significant delay that suggests OTP is impractical. We suspect that studying the behavior of these automated features may uncover outdated built-in assumptions, and provide an opportunity to make OTP feasible. Based on this intuition, we explore those features in detail and find that disabling a select few (effectively operating the amplifiers in manual mode) dramatically decreases the reconfiguration delay to 13–27 seconds. We verify that operating the devices in manual mode has no impact on the IP-layer traffic.

Finally, we use a lookup table to reduce the reconfiguration time. As wavelengths are added or removed, amplifiers adjust their gain to maximize the optical signal-to-noise ratio. This process happens each time the set of wavelengths changes, but

the results from the computation are the same for a similar set of wavelengths and amplifiers. Therefore, we store these parameters in a lookup table and show that wavelengths can be added in approximately 500 milliseconds.

In summary, we make the following contributions. (1) We measure reconfiguration delay on a long-haul fiber span. (2) We show how to reduce the time to provision a circuit by an order of magnitude—from minutes to seconds. (3) We propose a method to quickly store and load optical network equipment settings, reducing the time to less than 1 second.

5.2 Motivation: Is OTP Feasible Now?

Perspective of the Optics Community. There is a prevailing sentiment that OTP is possible in today's WANs, pointing to efforts on a diverse set of fronts towards a programmable physical layer. Notable categories and examples include protocol descriptions for dynamic path provisioning [56], lab-based evaluations of multi-layer control [138, 140], amplifier modeling [205], and operations research [229]. However, these efforts are not enough to enable a highly programmable optical WAN. What is lacking here is a pragmatic evaluation of optical layer components and their readiness for providing dynamic wavelength services in response to changing network and application layer demands.

Perspective of the Networking Community. To the best of our knowledge, we are not aware of practical OTP-ready WANs.[1] We posit that this is primarily due to the pragmatic issues in realizing OTP; this is also the widely accepted perspective of networking community [102]. More concretely, the efforts in the optics community [56, 138, 140, 205, 229] hardly begin to close the book on practical applications of OTP.

[1]We note that for data center networks (DCNs), the networked systems communities have proposed a variety of programmable topologies [100]. However, our focus is on OTP-ready WANs and hence we defer our discussion on DCNs.

For example, CORONET [56] presents protocols and abstractions for operating a WAN with OTP but falls short to demonstrate methods for quickly turning up waves, and settles for add-times on the order of minutes. Similarly, OWAN [140] demonstrates benefits for multi-layer control, but their testbed trivializes amplifier control by considering one amplifier per link; long-haul links typically have half a dozen or more amplifiers. AcCBR [205] is an ML framework for configuring amplifiers in a WAN, but requires additional hardware at each amplifier in the network to collect sufficient data to build its model. Finally, theoretical efforts such as those done by Papanikolaou et al. [229] show that multi-layer control clearly offers better performance and survivability in the case of outages, but only via numerical models, not practical implementations.

These contradictory perspectives indicate a chasm between the communities on the practicality and feasibility of OTP. To bridge this ongoing divide between the two communities, this work seeks to shed light on the pragmatic issues in making optical layer OTP-ready using lab-based measurements.

5.3 Laboratory-based Experiments

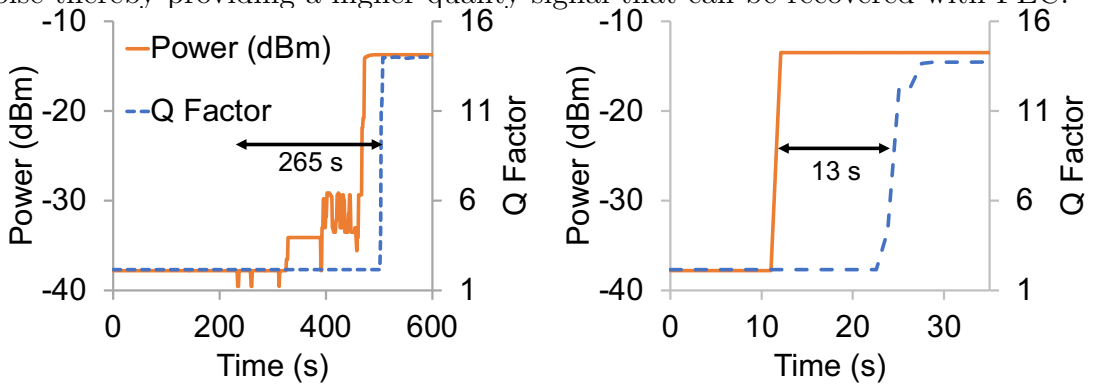
5.3.1 Objectives and Testbed. The main goal of this work is to investigate the feasibility of OTP by measuring the time taken by an optical path to stabilize to the point where it can be used to transport data after adding or removing wavelengths from higher layers. To this end, our testbed includes equipment found in points of presence (transponders, multiplexers) and on long-haul paths (amplifiers). Specifically, we employ three pairs of transponders, each of which transmits 10 × 10 Gbps *bands*. Band Multiplexing Modules (BMMs) receive these three 100 Gbps bands, and multiplex them onto a single fiber. The BMMs are equipped with Erbium Doped Fiber Amplifiers (EDFAs) which support variable gain

from 19 to 26.5 dB. These EDFAs can boost a signal for approximately 80 km before another amplifier is needed. Our testbed has seven amplifiers in total. For specific details on the testbed, see Appendix A.1.

Metrics. The key metrics for our tests are the level of total optical power (dBm—decibel relative to 1 milliwatt of power) into and out of each band multiplexer and amplifier, and Q or quality factor at the receive-end transponders where wavelengths are added or removed. We measure add-time for a circuit as the time that it takes for power and Q factor to stabilize after a wavelength change is made. We take measurements using an Optical Spectrum Analyzer (OSA) to measure power levels directly on the fiber, as well as SNMP Management Information Base (MIB) values available from the administrator interface.

5.3.2 Standard Reconfiguration Delay. Standard best practice in network operations assumes a stable and reliable physical layer topology. Due to this assumption, optical equipment vendors have implemented a host of automated tests and adjustment features—which we refer to as the *automatic mode*—to ensure that devices return to a stable/predictable state after certain events (e.g. adding/dropping wavelengths). This mode works as follows: a transponder *tests* a sending power level and receives feedback from the amplifier. The feedback instructs the transponders to increase or decrease (i.e., *adjust*) its power level. This process continues in a loop until the first hop amplifier is satisfied with the power level for the channel it receives. After the channel's power is accepted by the first amplifier, each successive amplifier on the path repeats a variation of this process with the amplifier before it. Upon reaching the transponder at the receiving end, the signal is decoded back into the electrical domain. Forward Error Correction (FEC) is implemented in hardware to correct any bits that are flipped due to noise on the channel. If any bits are uncorrectable, an

alarm is raised. Subsequently, a signal is sent to the amplifiers to repeat their tests and adjustments to find gain settings that reduce Amplified Spontaneous Emission (ASE) noise thereby providing a higher-quality signal that can be recovered with FEC.



(a) Automatic mode: add-time is 4 min and 25 s. *Hence, today's WANs are not OTP-ready.*

(b) Manual mode: add-time is 13 s—over 19× faster than automatic mode (Figure 13a).

Figure 13. Comparison of automatic & manual modes.

Using our testbed, we evaluate the add/drop-time that can be reasonably expected by hardware operating in automatic mode. Figure 13a shows the ingress power to the first amplifier hop plotted with the Q factor[2] of a corresponding wave within the band at the receiver. We evaluate the *add-time* as the difference between the first change in receiver power at the amplifiers and the stabilization of Q factor above 11 at the receiver. In this instance, the add-time for this wave is 265 seconds. After running this experiment 8 times, we find that add-times vary from 2 to 6 minutes. We note that these estimates are conservative, underestimating add-times for longer spans with more amplifiers.

Main findings and implications. The add-time for long-haul optical circuits, in practice, is on the order of minutes. This implies that *today's WANs are not OTP-ready.* This is primarily due to two standard features from the telephony era: *(i)*

[2]Q, or *quality* factor is a numeric representation of the signal quality. The minimum Q-factor required for error-free transmission is system-dependant [133]. In our evaluation, a Q factor of 7 was sufficient for complete error-free transmission.

transponders incrementally and conservatively increasing their sending power level until it reaches the target level for the first hop, and *(ii)* the Automatic Gain Control (AGC) loop, which sets the gain at each amplifier on the path.[3] The main implication of this finding is that these features, if manipulated appropriately, can provide an opportunity to make OTP feasible. Intuitively, for feature *(i)*, if the appropriate power level is known a priori for a transponder on an optical path, then the 4 minutes spent ramping up power can be saved by automatically applying that power. We focus on *(i)* next and address factor *(ii)* in § 5.4.

5.3.3 Reconfiguration Delay From *min* to *s*. Next, we investigate a method for reducing add-time via intervention in the protocol between the transponders and their ingress BMM. Typically, in automatic mode, the launch power for a wave is determined by a protocol between the transponder and the ingress BMM's amplifier. However, there is a configuration parameter on the BMM and transponder which enables us to side-step this negotiation process and set the launch power explicitly. This feature is available across devices from different vendors, thus, we take the transponder and BMM out of automatic mode and put them into "manual mode". In manual mode, the wave's launch power must be set such that the ingress BMM's amplifier receives it within a hardware-specific target range. In our case, the BMM's amplifier expects to receive signals of -14 to -12.5 dBm from any band port. Thus, we set the transponder's sending power such that it hits the target. This value only needs to be determined once for any transponder/ingress amplifier pair.

Figure 13b shows the add-time for a circuit across 7 amplifiers with transponders operating in manual mode. We set the launch power to 0.5 dBm, and used a variable optical attenuator (VOA) to add/drop the signal. When attenuation is set to zero,

[3]We focus our attention only on add-time because dropping optical circuits is trivial; our evaluations on the effect of drop on other waves were negligible.

power at the ingress BMM jumps to -13.5 in one time-step (1 second). 13 seconds later, the Q factor for the received signal increase beyond 11, then settles to 13.73. We also conducted an extensive analysis on the impacts of OTP on existing wavelengths (see Appendix A.2) and found that it is safe to add/drop waves in manual mode to increase the agility of the physical layer via OTP.

Main finding and implication. Based on this experiment, we find that optical circuits can be provisioned over 19× faster by setting the sender's power level manually. Moreover, in light of OTP, the warm-up time can be obviated without impact. This result suggests a way forward toward achieving OTP in today's WANs.

5.4 Toward *ms* Reconfiguration Delays

Our measurements in § 5.3.2 and § 5.3.3 lead us to conclude that amplifiers operate with no knowledge of their past configurations. That is, they can find an appropriate gain level for a set of signals. But if you take away one signal and add it again, they start from scratch to find how to efficiently boost it. This is understandable if fast reconfiguration is an objective (which it was not in the telephony era).

To address this issue, we propose a new mechanism that uses a lookup table to choose gain values at each amplifier, to further reduce reconfiguration delays and make OTP feasible in today's WANs. First, we describe how to construct the amplifier table, and then present latency measurements collected in building the table. Then, we use these measurements to predict the performance for add-times with a system that can access an amplifier table. For a series of amplifiers in the path, we also compare the reconfiguration delays resulting from the *automatic* and *manual* modes with the ones obtained using our proposed lookup mechanism. One might argue that a lookup table is too simple of an application. However to the best of our knowledge, this has not been developed before. We argue that this first OTP utility should be as

simple as possible. Only after it is demonstrated can we develop more intelligent and efficient methods (e.g. machine learning), and perhaps drive down circuit add-time even further.

Amplifier Table. We start by building a simple local controller (LC), which will be the key point of coordination for various optical components. An LC resides on a VM near transponders for an optical path (OP) and maintains a table that relates an optical configuration (OC) (i.e., set of active wavelengths) to amplifier's gain and Quality of Transmission (QoT). OCs in the table are aggregated by power level to keep the size of the table manageable by a VM.

The LC has two components: a management engine and an amplifier table. The management engine receives requests and sets/gets values to/from optical path hardware (transponders, amplifiers, etc.). The amplifier table[4] is a data structure maintained by the management engine for rapidly provisioning optical circuits. When the LC receives a Configuration Change Request (CCR) (e.g., activate band n on OP x), it checks the amplifier table table to see if there is a configuration stored for the path where the present waves and the requested waves are all active. If it finds that configuration, it applies the gains corresponding to that table entry on all of the amplifiers of the path in parallel; commands are issued over the optical supervisory channel. If no such entry exists, the LC activates the requested circuit(s) and waits for AGC to set the appropriate gain on each amplifier. Then, it stores the stabilized gains for the CCR in the amplifier table and sends a response back to the requesting agent.

Measurements. We investigate two methods for constructing the amplifier table, namely TL1 [60] and SNMP [87]. These are the two APIs available for querying

[4]There are several systems issues including how many tables a network should maintain, how to populate the tables at scale, slow local vs. fast remote and their impacts on table lookup, etc. These issues are beyond the scope of this work and will be considered in future work, see [321].

amplifiers pragmatically in today's WANs. We use both for polling the gain value from each amplifier along the path in parallel, and report the time for the operation over 100 iterations. We find that TL1's median gain access time is ∼3 seconds, 6× faster than the time to activate a light path in manual mode. We also find that with SNMP, we can reduce this latency to about half of a second. Therefore, we suggest that manufacturers enable an SNMP-like interface for configuring gain on amplifiers of long-haul paths. With this capability, we see the potential for speedup greater than 200× over the expected configuration time for light-paths in automatic mode (see Figure 14).

Performance. As shown in Figure 14, the expected time for adding a wavelength in manual mode, with no gain information, is about 20 seconds. Therefore any new configuration added to the path will be installed, on average, in 20 seconds. After the configuration metrics are stored in the amplifier table, any future request for that configuration can be added, on average, in 0.56 seconds (as indicated with SNMP).

Validation. We collected Q factor and latency data on a 100 Gbps circuit. We

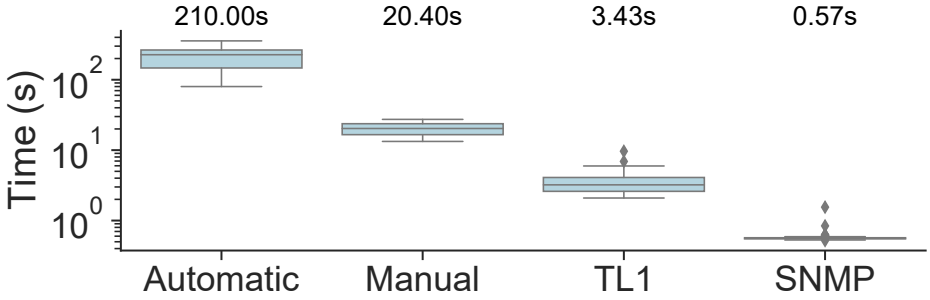
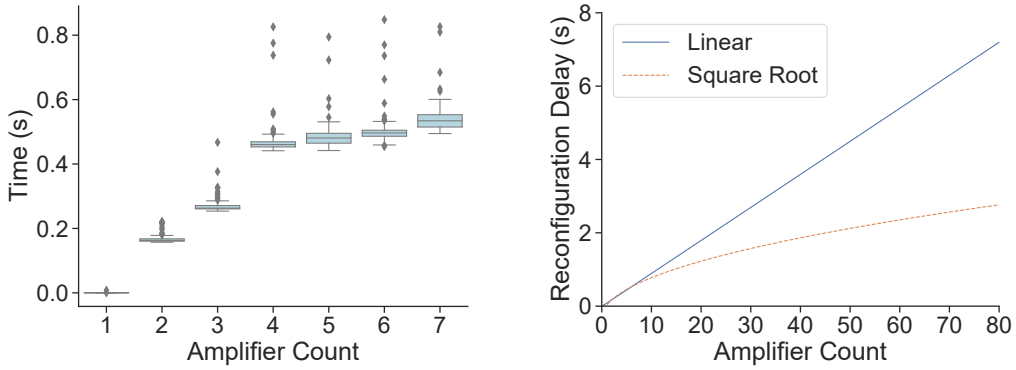


Figure 14. Reconfiguration delays for various modes (mean value shown above).

found that adding noise to the channel, thereby triggering AGC changes, does not have any impact on the latency of Ethernet packets mapped into the ODU frames. We used a layer-3 traffic generator to produce packets of various sizes (95, 1500, and 9216 bytes) and found that RTT stayed constant, plus or minus 0.1 microsecond.

The average jitter was constantly 0.0 microseconds. This implies that any noise that is added to an optical circuit by changing gain at amplifiers will not impact layer-3 performance. Therefore, it is safe to use the gain values stored in the amplifier table.

5.4.1 A Performance Model for Long-haul Paths and Submarine Cables. Optical paths often traverse thousands to tens-of-thousands of kilometers. To predict the expected performance of an amplifier lookup table-based controller on these paths, we use a least-squares regression model trained with the seven amplifiers in our lab. We collected data by polling different subsets of amplifiers with parallel SNMP queries (the same method used in Figure 14). For each set of amplifiers tested, we repeated our measurement for the gain retrieval time 100 times. Figure 15 shows the data we collected (15a), and the model (15b). According to the model, an optical path with 25 amplifiers can be reconfigured in 1.5 to 2.3 s. This is much faster than the automatic mode. That is, the amplifiers in automatic mode can be expected to take more than 9 minutes (assuming a linear model, where 7 amplifiers take 155 s to reconfigure). In manual mode, we estimate the reconfiguration delay to be about 46 s, based on similar analysis.



(a) We collect 100 measurements for each set of amplifiers using SNMP.

(b) Linear (solid) and square-root (dotted) regression models.

Figure 15. Gain retrieval time for a path of seven amplifiers (15a), and projected reconfiguration time for longer paths (15b).

We apply this model to longer paths such as inter-continental submarine cable deployments. As an example, for links that are 6,600 km long [270] with $\tilde{8}0$ amplifiers, we can expect reconfiguration times between 3 and 8 seconds. However, this model is missing critical features that complicate submarine deployments. Environmental settings such as water pressure and temperatures may affect the power budget. Furthermore, infrastructure risk from human activity (e.g., anchors, fishing nets) and marine life (e.g., shark bites) should inform the prospect of OTP in submarine settings in addition to the reconfiguration delays that we consider. Therefore, more measurement work and experiments are required to critically evaluate the prospects for OTP with submarine cable deployments.

5.5 Discussion

We believe that empirical measurement efforts like ours can identify and inform several scientific gaps between the optical and networking communities. In what follows, we describe two such gaps, outline how the measurements can help by designing useful tools, and elucidate the key challenges in building those tools. We leave the implementation and evaluation details for future work.

For one, the assumption of the "stable physical layer" model is at odds with the "dynamic physical layer" model of OTP. Understanding this dynamism calls for (a) creation of an end-to-end *optical layer traceroute* tool that can offer visibility (e.g., via TL1 or SNMP) into several optical devices in a network path, (b) unified interfaces to expose measurements from the optical layer to higher layers of the network stack, and (c) an adaptation framework to seamlessly adapt protocols at the higher layers in response to the dynamism of the optical layer (e.g., change IS-IS or OSPF link weights in the face of Q-drop at the optical layer).

Another important question raised and addressed by this work is the perceived risk of disabling "automatic" mode. Clearly, there are opportunities for developing new capabilities for optical hardware that serve the same purposes in addition to supporting OTP. Measurement efforts offer the objective basis to evaluate the safety of these capabilities.

Building an end-to-end optical layer traceroute tool requires participation from network operators from several constituents (e.g. enterprises, transit providers, etc.). Second, designing cross-layer interfaces and exposing optical measurements from those interfaces call for expertise from and collaboration among optics, measurements, and networked systems researchers. Third, we posit that the fate of the envisioned measurement tools will be similar to layer-3 traceroute due to privacy and security reasons (e.g. blocking/dropping measurements, malicious intent to map the wavelength allocation in a network, etc.). Assuming participation from network operators, one way to address this challenge is to build an enclave (similar to secure containers in Intel SGX) in optical devices where SNMP or TL1 could be used to query the devices and provide responses *without* violating privacy and security restrictions [212].

5.6 Summary

In this chapter we conducted experiments that benchmark the time to activate an optical signal carrying data on a shared optical fiber with other, on-going, signals present. We found that we are able to reduce the time to introduce new circuits to a DWDM fiber span from minutes to seconds in our lab environment. We further propose changes to optical amplifier interfaces that could reduce this reconfiguration delay further, into the sub-second domain.

CHAPTER VI

GREYLAMBDA: A FRAMEWORK TO SCALE TRAFFIC ENGINEERING USING OTP

This chapter contains previously published coauthored material from [210], with coauthors Paul Barford, Klaus-Tycho Foerster, and Ramakrishnan Durairajan. The dissertation author designed the experiments in consultation with the coauthors. The dissertation author implemented and ran all of the experiments.

6.1 Introduction

Internet service and cloud providers have been working to *scale* their network performance by making various parts of the network *programmable*, from load balancers [77, 230] to switch stacks [36, 182, 195] to network interface cards [95]. This has led to the replacement of ad hoc traffic engineering (TE) in wide-area networks (WANs) with software-defined systems [2, 35, 129, 137, 161, 166, 179, 189, 322], to better manage WAN resources, respond to dynamic traffic shifts and unforeseen events, and provide custom services to customers.

TE systems aim to continuously monitor traffic demand and utilization across the entire network using a range of measurement tools, allocate network resources based on the observed demands, and update the traffic forwarding behavior of network resources accordingly. The success of these systems is contingent upon the optimization step being completed within a defined time frame, such as a time-to-solution of five minutes or less [35, 129, 322]. This time constraint is referred to as the "temporal requirement." Furthermore, flow allocations onto links must not over-subscribe those links within a geographical scope of the network; this is referred to as the "spatial requirement" of the TE system.

The multi-commodity flow (MCF) formulation used in the TE optimization step cannot keep up with the increasing size of network backbones and changing traffic

demands, as seen in unforeseen events such as sudden flash crowds or fiber cuts. To address this challenge, recent approaches such as SMORE [166] and NCFlow [2] have relaxed MCF constraints in order to meet the temporal requirement. However, our evaluation shows that these relaxed constraints can lead to either oversubscribed traffic (and, consequently, throughput drop) or infeasible solutions in critical network paths during unforeseen events. Furthermore, we note that considering the entire network infrastructure in the optimization step is not always necessary, as these unforeseen events are often localized to specific critical network paths. This warrants the right scoping of those critical paths as part of the spatial requirement. Improving the scalability of TE systems by satisfying both the temporal and spatial requirements simultaneously is an open problem.

In this work, we identify a novel solution for improving the scalability of TE systems by utilizing the recent development in optical networking known as *optical topology programming* (OTP). OTP enables the reconfiguration of existing optical wavelengths and the creation of new ones in critical network paths, providing two key advantages. First, OTP allows for localized link bandwidth scaling to reduce congestion in the oversubscribed network links. Second, OTP provides new paths for forwarding traffic and absorbing dynamic traffic shifts caused by unexpected events.

Harnessing these benefits in practice to satisfy the two requirements of TE systems, however, requires addressing three key challenges. First, implementing OTP on large networks requires a significant investment in optical equipment, such as transponders and amplifiers, to establish new traffic forwarding paths. Second, the current optical equipment deployed in WANs often experiences substantial reconfiguration delays due to optical path-protection mechanisms, such as amplifier gain control and transponder power adjustments. These mechanisms are at odds with the temporal requirement.

Finally, there is no unified formulation to evaluate the effectiveness of OTP versus static allocation and determine the optimal routing of traffic flows through the network by considering the benefits of OTP compared to static backup paths.

To address these challenges, we present GreyLambda, a framework that enhances current TE systems by integrating OTP. GreyLambda comprises two innovative components. Firstly, a heuristic algorithm that capitalizes on the presence of latent hardware resources, e.g. optical transponders, at high-degree nodes to offer bandwidth scaling on up to two links simultaneously. At the core of this algorithm is a theorem that demonstrates the benefits of these resources increase with the degree of the node in which they are placed. This directly addresses the spatial requirement by mitigating losses locally through simple optical layer bandwidth adjustments, rather than performing a global computation of all paths and flows. Secondly, we conduct lab-based experiments on commercial long-haul optical fiber hardware to delve into the reasons for optical path reconfiguration latencies and present a method to reduce these latencies to milliseconds for paths with several optical amplifiers. Finally, we demonstrate the potential of GreyLambda to enhance the performance of two state-of-the-art TE systems, SMORE [166] and NCFlow [1], by integrating the two components of GreyLambda and evaluating the results in real-world topologies with challenging traffic and link failure scenarios.

6.2 Background and Motivation

6.2.1 Traffic Engineering. WAN infrastructures are costly investments, and the routing systems adopted by the public Internet, e.g., OSPF and IS-IS, are prone to suffer high performance impacts from node or link failures unless the infrastructure is highly over-provisioned, e.g., with links typically using 40%-60% of their available capacity [129] at any given time. To maximize the return on

investment from WAN infrastructure and to achieve higher utilization of the links that connect end infrastructures (e.g., data centers), TE has been widely used by large content and Internet service providers (e.g., Verizon, Microsoft, Google, etc.) [35, 129, 130, 146, 163, 166, 179, 189, 252]. At a high level, the TE formulation consists of three steps: (1) observe network demand and link utilization, (2) optimize traffic allocations (including path selection and flow allocations per path) according to the observed demands using numerical optimization solvers, and (3) update the forwarding state of network routers and switches using the optimization result [308, 322]. In this work, we are primarily concerned with step 2 of TE and contribute a framework that enables TE to solve this step quickly when bandwidth demand on links in the network is greatest (e.g., from flash crowd events or from fiber cuts).

TE optimization has been the subject of numerous recent studies [35, 129, 146, 166, 179, 189, 252]. These efforts have been prompted by the shortfalls of greedy, shortest path routing, for managing inter-datacenter traffic at scale [129] and advances in programmable network monitoring and control software [28,195,279]. TE optimization solvers are expected to compute as well as provision traffic paths and flow allocations on those paths approximately once every five minutes [35,129,322]. We call this the *temporal* requirement of TE optimization solvers. Although MCF is the most optimal way to route network traffic, solving MCF-based TE optimization is infeasible for large networks [129]. In light of this, a host of TE systems have been proposed to address the temporal challenge while maintaining high throughput throughout the network [2, 166, 189]. We note that any solution to scale the performance of TE systems should satisfy the temporal requirement.

6.2.2 State-of-the-art and their Limitations.

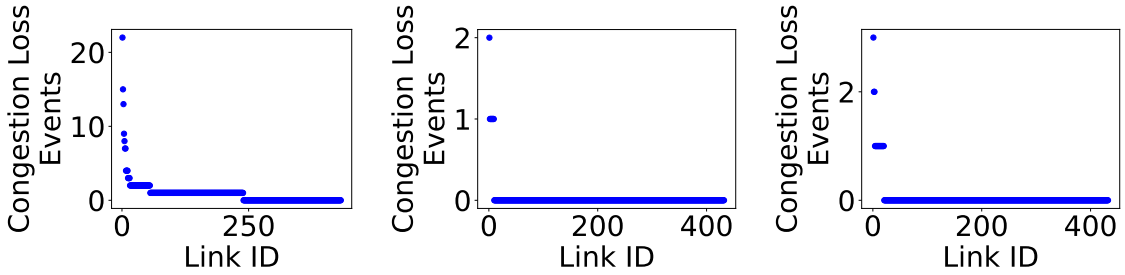
Our work is motivated by the following two key limitations of state-of-the-art TE systems:

Limitation 1: Falling Short of the Spatial Requirement.

Complementary to the temporal requirement is the spatial requirement of TE systems. The spatial requirement pertains to the geographic scope of the network infrastructure considered (e.g., all links vs. top k links) by TE optimization solvers in the face of unforeseen events with dynamic shift traffic, such as flash crowds and fiber cuts. Typically, TE optimization runs globally, addressing the spatial requirement in a roundabout way, i.e., by provisioning flow tunnels along edge-disjoint paths [166] or by reserving *headroom* on all network links in case of an unforeseen event [179]. Prior efforts have pointed out that events typically have a local spatial scope [65,252], potentially affecting only a handful of links in the network. Thus, reducing the spatial scope to the affected links is key to accelerating the TE optimization step and scaling network performance.

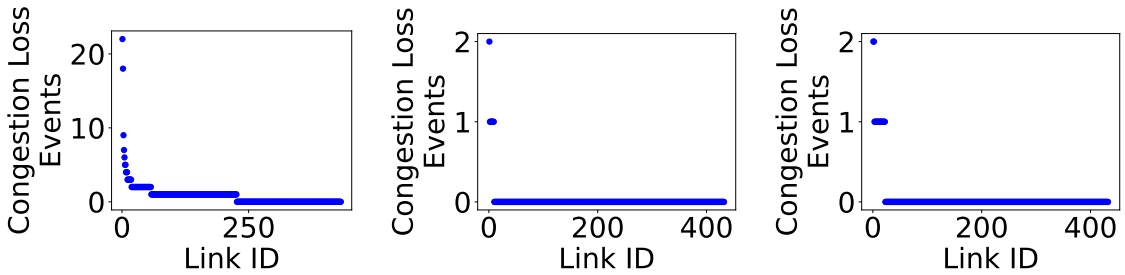
To illustrate, we investigate how frequently a given link in Microsoft Azure's global WAN [20] experiences congestion loss—defined as bandwidth demand greater than link capacity—during a diverse set of flash crowd and fiber cut events. To this end, we generate 432 traffic matrices (see § 7.5 for details), where each matrix targets one direction of a single link in Azure's network. We plot the number of times that each link in the network sees congestion loss given different TE routing strategies, including equal-cost multi-path (ECMP) routing, semi-oblivious path selection[1] with MCF (SMORE [166]), and MCF (without path-based restrictions) in Figures 16, 17, and 18.

[1] *Oblivious* with respect to traffic demand. These paths are pre-computed for the network without considering demand, similar to ECMP. Unlike ECMP, they are chosen to effectively minimize the sharing of edges between flows.



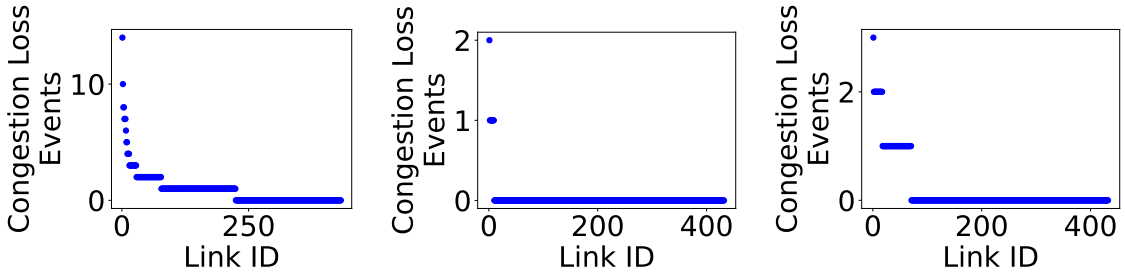
(a) ECMP (b) MCF (c) SMORE

Figure 16. Total Congestion Loss events per link in Azure with flash crowds with various TE schemes.



(a) ECMP (b) MCF (c) SMORE

Figure 17. Total Congestion Loss events per link in Azure with flash crowds and one link failure with various TE schemes.



(a) ECMP (b) MCF (c) SMORE

Figure 18. Total Congestion Loss events per link in Azure with flash crowds and two link failures with various TE schemes.

In this analysis we show ECMP because of its historical significance and because it is still used in networks today. ECMP forwards traffic along the shortest paths between hosts. Therefore links that are central to the topology end up being bottlenecks as they are on the greatest number of shortest paths. Thus, relying on them to forward the bulk of traffic leads to a small number of links being congested

by many different traffic events. This is clearly visible in Figures 16a, 17a, and 18a, where some links are congested by upwards of 20 different traffic matrices.

At the other end of the spectrum, MCF makes the optimal choice for routing paths considering traffic. This routing strategy is as close to as perfect as we can get concerning TE, but is not scalable; solving MCF for Azure in our experiments took more than an hour to solve for each traffic matrix. Even with this optimal path selection and forwarding strategy, Figures 16b, 17b, and 18b, show a small number of links (fewer than other routing strategies) that are congested by more than one event.

SMORE is more scalable than MCF, but also has more links that are critically impacted by multiple flash-crowd/fiber cut scenarios. Figures 16c, 17c, and 18c show that the result for congestion loss events per link in SMORE is also a long-tail distribution that falls somewhere between those observed in ECMP and MCF. SMORE is the latest of these three TE strategies and therefore we exclude ECMP and MCF from the rest of the paper.

Takeaway: There is a small set of critical network paths that are affected by a diverse set of congestion-causing events (including flash crowds and multi-link failures). Unfortunately, many of the TE solvers run globally (i.e. without considering the right scope of the critical paths as part of the spatial requirement).

Limitation 2: Not Considering the Temporal and Spatial Requirements of TE Simultaneously.

NCFlow [2] partitions the network topology into a small number of clusters, which they refer to as contractions, and solves a TE optimization within each network contraction in parallel, while also optimizing inter-contraction traffic. BlastShield [161] also partitions the network into clusters, but uses distributed

controllers to route traffic through each cluster (rather than having the optimization coordinated by one central server as in NCFlow). While these solutions scale to global content provider networks and satisfy the temporal requirement of TE, they still fall short regarding the spatial requirement by maintaining a simplified view of network topology that lacks geographic considerations such as the impact of fiber cuts on shared links.

Researchers have proposed systems for WAN operation considering the spatial requirements of TE. For example, SMORE [166] makes use of oblivious path selection to route traffic so that shortest-path links are not oversubscribed. Unfortunately, SMORE is still unable to meet the spatial requirement for some flash-crowd and link-failure scenarios; the fixed bandwidth available on links leads to infeasible solutions where bandwidth allocated to critical links exceeds capacity. Figures 16c, 17c, and 18c illustrate this observation, where the same critical links are oversubscribed by various flash crowd and link-flood scenarios. This is similarly the case for other TE systems that have fault tolerance as a core design constraint, such as FFC [179] and TeaVar [35]. These systems attempt to reduce the impact of spatial events like flash crowds and link failures by under-subscribing network links such that there is additional room on alternate path links when the primary path fails or is over-subscribed.

Recently, there has been a promising line of work highlighting packet-optical network co-optimization and topology reconfiguration in response to events such as link failures. For example, Arrow [322] enables partial restoration of lost link capacity by using transponders at the ends of a failed link to activate a new optical circuit on an alternate physical path. The system relies on amplified spontaneous emission (ASE) noise generators to occupy spectral bandwidth on redundant paths until a traffic-

carrying signal replaces the noise channel on a backup fiber. Arrow uses a system of linear programming optimization functions to choose restoration paths from a set of candidates and maximize throughput for end-to-end traffic on the (partially) restored path. The optimization runs globally across the entire network and depends on ASE channels to meet the temporal requirement of TE. If these ASE noise channels are not available the reconfiguration latency increases from seconds to tens of minutes [322], which is at odds with the temporal requirement. This is also a limitation because there is no oracle to tell which link will fail *a priori* and thus the ASE channels cannot be maintained globally for every link in the network. This limitation notwithstanding, Arrow is a key inspiration for this work and points us to a novel opportunity that we leverage in this work.

Takeaway: A body of solutions consider the temporal requirement but fall short of the spatial requirement. The solutions that prioritize the spatial requirement do not get the right geographic scoping of critical network paths. What is critically lacking is a solution that satisfies both the temporal and spatial requirements of TE simultaneously.

6.3 Opportunity

We observe a new opportunity to address both these requirements simultaneously in the TE optimization step by leveraging OTP, a recent advancement in optical networking. Using OTP, an operator can affect a network's topological structure via optical wavelength reconfiguration in addition to the traffic forwarding behavior.

OTP leads to two new opportunities for accelerating TE optimization and satisfying both requirements. First, it allows a network's underlying topology to scale capacity on demand in a fine-grained, localized fashion to avoid congestion resulting from a fiber cut or flash crowd. Second, OTP enables an operator to amplify the

benefits of traditional TE mechanisms. Improved general network performance is possible because changes made at the optical layer give us increased possibilities for forwarding traffic on new paths in the face of network events.

To illustrate these opportunities, Figure 19a shows a simple graph/network with two nodes v, w connected via edges/fiber, with the number of wavelengths per edge indicated. Figure 19b shows an *optimally resilient* static allocation of three wavelengths in the sense that for any two fiber cuts, as in 19c, at least one wavelength remains between v, w. With OTP, all wavelengths can be steered onto the surviving fiber, restoring the original throughput for the network 19d.

Figure 20a illustrates a traffic shift without failures. In this case, previous traffic required bandwidth of 2 between s, t and v, w. However, if traffic shifts to flow only between s and t, any TE is limited to a throughput of 2 as shown in 20a, whereas TE+OTP can adapt to the situation and obtain a throughput of 4 as shown in 20b.

6.4 Challenges

Leveraging OTP to scale TE systems entails three unique challenges:

(**C1**) *Is it possible to identify and run OTP on certain critical paths to satisfy the spatial requirement?* Large WAN networks have hundreds of nodes and many more edges, e.g., the Azure network discussed in § 6.2 has 113 nodes and 216 edges. Enabling OTP on every one of these links globally would require significant investments in equipment to guarantee that a backup path could be provisioned for every possible link failure event. In addition to the hardware support required, there are also practical concerns for the reliability and efficiency of a software system trusted to orchestrate dynamic physical connections between all of the network nodes across all of the potential paths. Such an investment is not realistic and therefore, to reap

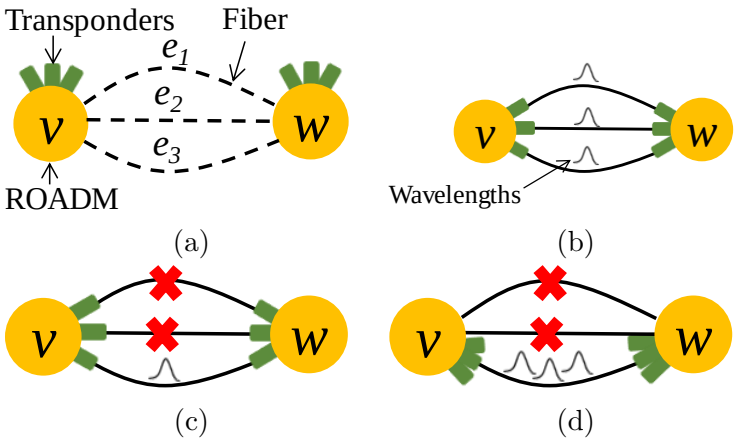
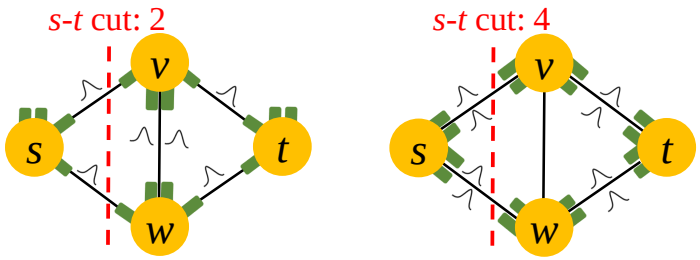


Figure 19. A physical graph with three transponders at every node in (a). The most resilient way to *statically* allocate wavelengths is shown in (b), as two fiber cuts are survivable, as in (c). With OTP, however, we can recover from these two fiber cuts and retain three wavelengths between v, w as in (d).



(a) TE on the static graph (b) TE+OTP

Figure 20. A physical graph with four transponders at each node in (a). Adapting the static wavelength allocation in (b) yields a gain factor of 2 for the throughput from s to t in (b). Conceptually, the minimum cut between s and t limited the performance of TE in (a). OTP on the other hand increased the minimum cut to 4, by moving wavelengths away from the middle fiber.

the benefits of OTP, we must be strategic concerning which links in the network would benefit the most with reconfigurability.

(**C2**) *How can wavelength reconfiguration latencies be reduced to satisfy the temporal requirement in the absence of amplified spontaneous emission (ASE) noise generators?* Historically, OTP has not been widely used in WAN networks due to the reconfiguration latency that occurs when activating a new circuit on a shared fiber span [211]. Moreover, careful measures must be taken to ensure that the introduction

of a new circuit to a fiber span does not degrade the optical layer performance (as shown in § 5.3.2). In light of these limitations, packet and optical network innovations have generally occurred independently of each other [212], and bridging the chasm between the communities require revisiting some of the base assumptions (e.g., TE assuming that there is a stable and static topology).

(**C3**) *What are the benefits of OTP for existing TE systems?* In the absence of large-scale optical testbeds to pragmatically investigate the benefits of OTP, it is important to understand how existing TE systems pair with OTP. To do this, we require answers to several *what-if* questions regarding network performance (e.g., throughput, latency, utilization) under a diverse set of operational configurations, including TE system, demand profiles, and OTP capabilities.

6.5 Design Approach and Roadmap

We propose a framework called GreyLambda that seeks to scale the performance of current TE systems by integrating OTP to accommodate dynamic traffic shifts and unforeseen events such as fiber cuts. Concretely, GreyLambda augments existing TE systems with OTP at the right scope concerning an area impacted by congestion or failure, enabling it to react quickly with a locally optimal solution that has global benefits for network performance. For example, the system could be configured to respond with topology adaptation (adding links or bandwidth to specific pairs of nodes) only in the event of link failure if desired or deployed more liberally to change the topology with traffic if there is a high likelihood of performance benefit.

At the core of GreyLambda are the three novel insights:

(**I1**) To address **C1**, GreyLambda leverages insight from the formal model with theoretical guarantees (presented in § 4.2) to reduce the scope of TE optimization and identifies certain critical optical layer links (§ 6.6).

(**I2**) To address **C2**, GreyLambda employs a fast topology programming mechanism (described in § 5.4) to reduce the wavelength reconfiguration latencies of links identified in (I1).

(**I3**) To address **C3**, GreyLambda informs TE (at the network layer) about those identified links, thus amplifying the TE benefit and accelerating its solution process (§ 6.7).

6.6 Reducing the Scope of TE Optimization

To satisfy spatial requirements we address two goals: (**G1**) Identify critical links in a topology, e.g., such as those that underlie WANs for cloud and Internet service provider backbones, where GreyLambda will have the greatest benefit. (**G2**) Reduce the spatial scope of TE optimization to the critical links.

We show how to achieve **G1** considering the physical topology alone before traffic is running through the network. To do so, we leverage an intuitive feature of mesh topologies, namely that they contain high-degree nodes where bandwidth scaling can be achieved for any two adjacent edges with as few as two extra transponders at the incident nodes. Leveraging this feature, we hone into the links that are being affected by high demand and temporarily increase their capacity at the optical layer, thus reducing the scope of TE (**G2**), and saving traffic loss that would occur while an optimization solver recomputes and allocates flows onto new paths.

Concretely, Theorem 1 proves that the throughput and utilization gain factor of enabling optical topology programming is between one and $O(\Delta)$, for wavelengths between neighboring nodes, where $\Delta = max_{v \in \mathcal{V}} v)$ is the maximum node degree v) in the physical graph G: i.e., a low node degree implies low potential benefits, whereas a high node degree signals large potential benefits. We further prove that these bounds hold for any graph, under any edge (i.e., fiber) failure and demand scenario.

Intuitively, this result quantifies how much (over)utilization can be reduced, or throughput increased, under changing traffic demands and edge failures, by adapting the wavelengths dynamically.

The theorem informs our heuristic algorithm (in § 6.6.1) and reduces the scope of the TE objective function by limiting the number of flows that are considered for forwarding path adjustments. For example, when traffic shifts dramatically in a typical network, the TE controller recomputes flow allocations globally for all network paths. However, when we scale bandwidth at the optical layer we only need to consider the paths that are contending for bandwidth on the critically affected link and scale bandwidth on it accordingly.

6.6.1 Model-based Bandwidth Scaling Algorithm. The theoretical result from Theorem 1 suggests that the benefit of a reconfigurable topology vs. a statically configured one is signaled by the maximum node degree in the network. We leverage this finding to strategically place two additional transponders at every node in the network, knowing that their benefit will be the greatest at nodes with high degree. We call the transponders that are provisioned for the express purpose of reconfigurability *fallow*, which refers to an agricultural practice in which fertile land is plowed but not seeded, and is instead left idle until better growing conditions are present. The intuition of this practice for WAN operation, as motivated by Theorem 1, is that the best link for the fallow transponders to activate upon will be determined by the changing operating conditions of the network, for example in response to flash crowds or link failures.

Provisioning fallow transponders in the network has multiple benefits that we explore in this work. In addition to allowing for *bandwidth on demand* at key moments and places in the network, it is consistent with WAN operator goals

regarding high-utilization and lower capital expenses. In § 7.5 we give a detailed analysis of the cost and benefit of static topologies compared to dynamic topologies. Finally, strategically provisioning fallow transponders dramatically aids in reducing computational complexity for optimally choosing where and when to activate reconfigurable links.

Algorithm 1 shows the conditions for activating bandwidth on demand for links where traffic is lost from congestion (e.g., from flash crowds or link failures). The first condition that we check is whether the nodes incident to the traffic loss event (congestion or failure) have fallow transponders. If they do, we activate these transponders to establish a higher-bandwidth link between the two nodes. The condition looks for opportunities to scale bandwidth on pre-existing IP links, and therefore the additional bandwidth on these links can be instantiated without re-computing TE flow allocations and forwarding paths. The second condition fires when there are no fallow transponders at the nodes incident to the loss event. In such cases, it searches the topology for links in which to increase bandwidth such that loss will be averted. These mechanisms simply offer higher bandwidth to the existing TE controller and enable the system to be integrated without constructing a new TE optimization algorithm.

The "Activate_Link" method should be rapid to minimize traffic loss before the network paths are updated. In the following section, we explore the capabilities of modern optical networking hardware and benchmark the time for activating a long-haul circuit.

6.7 Evaluation

We demonstrate the benefits of OTP in practice through simulations by augmenting IP-layer TE schemes with OTP. Our goal is to quantify the improvements

Algorithm 1 Bandwidth Expansion Algorithm

Require: u, v ▷ nodes incident to a congestion or loss event
  if u has fallow transponder & v has fallow transponder **then**
    Activate_Link(u, v)
  else
    Find nodes $(\hat{u}, \hat{v})$ with fallow transponders near event.
    Activate_Link$(\hat{u}, \hat{v})$
  end if

that existing TE schemes can achieve by using OTP vs. static backup paths. We analyze the performance impact from (one or two simultaneous) fiber cuts and different traffic demands on flows routed through networks. Since a fiber cut in the physical layer may result in the loss of several IP connections, we posit that the rapid reconfiguration of wavelengths enabled by OTP is key to boosting the efficiency of TE schemes.

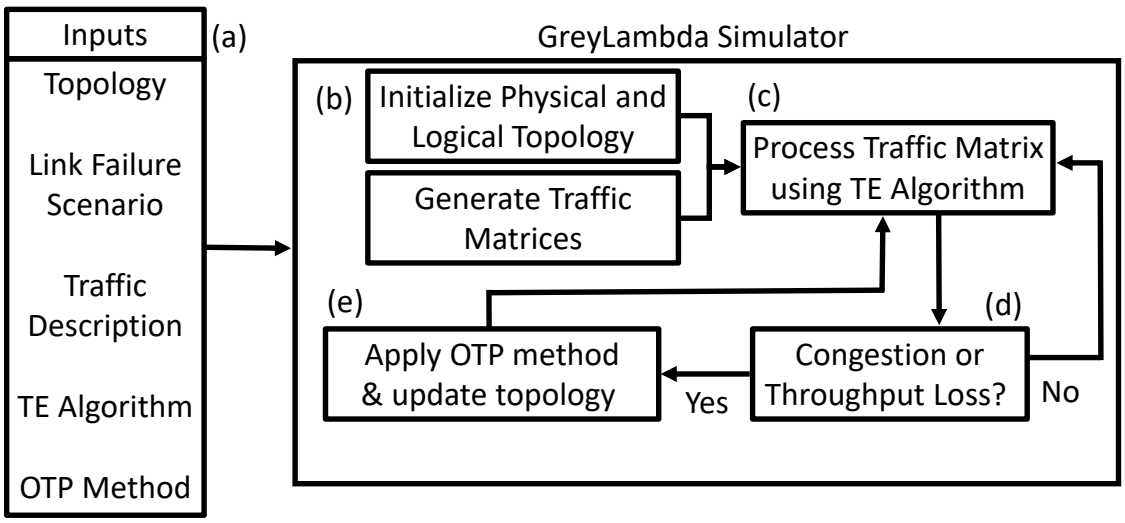


Figure 21. Architecture of the GreyLambda simulator.

6.7.1 Simulator Parameterization. The inputs to the GreyLambda simulator are shown in figure 21 (a). Parameterized topology settings include the

bandwidth of optical links, the initial quantity of links between each pair of nodes, and the number of fallow transponders allocated to each node. The user can define a set of physical layer link failure scenarios and a high-level description of traffic in terms of aggregate volume and type. For example, in our analysis that follows, we use the link failure and traffic description parameters to simulate link failure scenarios on each link with traffic matrices whose demand would be concentrated on the failed link. The TE algorithm parameter defines how traffic is routed in the network, and can be plugged in with any existing TE scheme (e.g., SMORE, NCFlow, etc.). Finally, the OTP method defines how the topology changes to mitigate loss in instances where the TE algorithm can not.

The generic execution of a typical GreyLambda simulation follows the following discrete steps, which start at Figure 21 (b). First, the topology is initialized and traffic matrices are generated according to the high-level description. The simulator maintains complementary IP and Physical layer views of the network; resource allocations at the optical layer are kept in the *physical layer view*, and their culmination in terms of connectivity and bandwidth is reflected in the *IP layer view*. A series of traffic matrices are constructed according to the description passed in. These can be made with generic traffic matrix generation scripts (e.g., [126]) or custom traffic generation methods. We use a custom method, described later in this section to generate traffic matrices for flash crowd scenarios.

In Figure 21 (c), the GreyLambda simulator processes a traffic matrix with the TE algorithm chosen by the user. Subsequently, it checks whether any links in the network were congested or if aggregate throughput was below a desired threshold (e.g., 100%), as shown in Figure 21 (d). In the case that no traffic loss or congestion occurs, the GreyLambda simulator is functionally equivalent to a TE simulator for

the given TE algorithm. That is, it processes the next traffic matrix in the series until there are no matrices left.

The Greylambda simulator employs an OTP method (Figure 21 (e)) when it detects link congestion or a throughput drop below the desired threshold. In this work, we evaluate the OTP method defined by Algorithm 1 in § 6.6.1. This method queries the transponders available at each end of the congested link(s) and activates a pair of complimentary transponders across those links when possible. Our experiments from 5.4 serve as a baseline for this step as the GreyLambda simulator estimates link reconfiguration times using experimental data and the model given in Figure 15b. After this process is complete, the GreyLambda simulator returns to step Figure 21 (c) and processes the next traffic matrix with the updated topology.

The simulator's methods for finding transponders at each network node and activating an optical signal between pairs of complimentary transponders serve as templates that can be used to define look-up and control messages to hardware in a real-world topology. To move GreyLambda from the simulated environment to a real-world deployment one would extend their SDN controller by adding the amplifier gain lookup table described in 5.4 and implementing the hardware querying and control messages templated in the simulator's code.

Topologies: We include topologies from five large content and Internet service providers in our evaluation. These topologies come from Internet Atlas [75] and manual transcription of publicly available network infrastructure maps [20, 21]. A summary of the topology information is given in table 5.

Wavelength blocking: In wavelength division multiplexed (WDM) networks, an optical signal can use a link only if there is spectral bandwidth available for that signal. We construct our wavelength topology such that the wavelength blocking

Network	Nodes	Edges
B4 [21]	54	118
Zayo [75]	96	110
Verizon [75]	116	151
Azure [20]	113	216
Comcast [75]	149	195

Table 5. Network topologies used in this study.

constraint is satisfied by leaving spectrum available for a single optical wavelength on each fiber. We also assume that the two fallow transponders at each node are tunable, i.e., that their frequency can be adjusted to match the available spectrum on an adjacent fiber. We note that wavelength tunable transceivers for long-haul paths are commercially available [282].

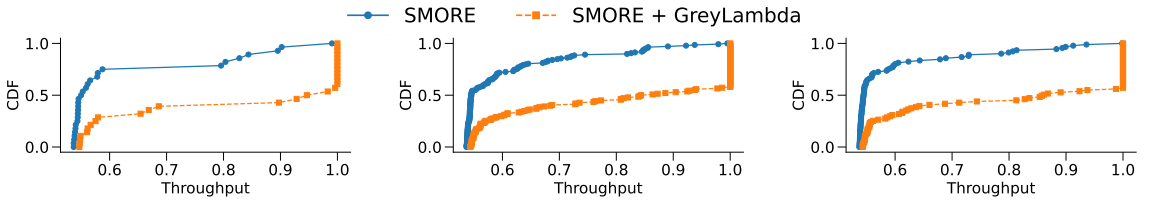
IP path selection and flow allocation: We compare the performance of two recent state-of-the-art TE algorithms, namely SMORE [166] and NCFlow [1]. Given an IP topology and traffic matrix, we simulate the traffic on the network with both TE systems and compare their performance with and without GreyLambda.

Traffic matrix generator: We constructed traffic matrices to emulate flash crowd events targeting each individual link in each network topology. To construct these matrices, we find the set of shortest paths for all pairs of nodes in the network, then for each link in the network, add flow demand in a traffic matrix for all flows that share the given link in their set of shortest paths. Algorithm 2 shows our flash crowd generation method explicitly.

6.7.2 SMORE Comparison. We emulate flash crowd events, each with an aggregate strength of 2x link capacity against every link in the five large CDN and ISP topologies while removing up to two links. We then compare the performance of SMORE vs. SMORE+GreyLambda in these scenarios. Figures 22– 26 show the

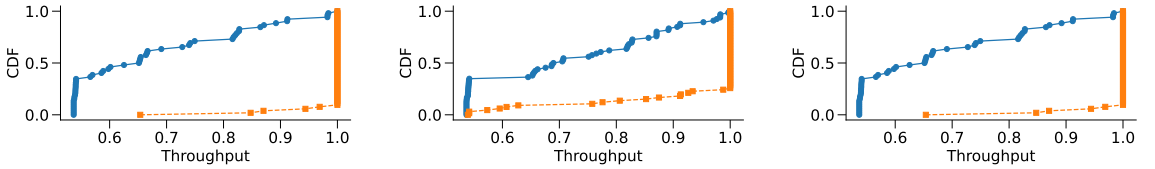
Algorithm 2 Flash Crowd Traffic Matrix Generation

Require: $G = (V, E)$ ▷ Network topology G of vertices V and edges E.
Require: $f : (u, v) \rightarrow list : paths$ ▷ map of links in topology to paths using that link
Require: $aggregate\_strength$ ▷ Volume of flash crowd traffic desired in each matrix
Require: $list : D$ ▷ list of $|E|$ demand traffic matrices. Each $n \times n$ zeros where $n = |V|$
  for $d, (u, v) \in D, f$ **do**
    $n\_paths = f[(u, v)].length$ ▷ —paths— containing (u, v)
    $flow\_strength \leftarrow aggregate\_strength/n\_paths$
    for $p \in f[(u, v)]$ **do**
      $s = p.head$
      $t = p.tail$
      $d[(s, t)] += flow\_strength$
    end for
  end for



(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failures

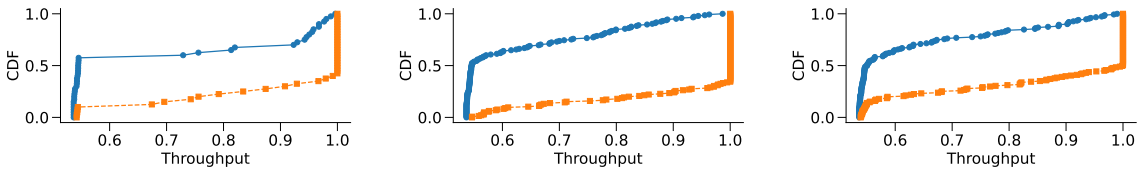
Figure 22. Throughput in Zayo under flash crowds combined with one and two link failures.



(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failures

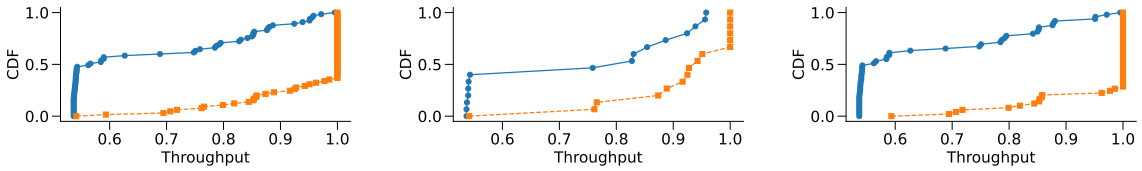
Figure 23. Throughput in B4 under flash crowds combined with one and two link failures.

results for aggregate network throughput. Overall, we find that GreyLambda can increase the throughput of SMORE for all traffic and link-failure scenarios.



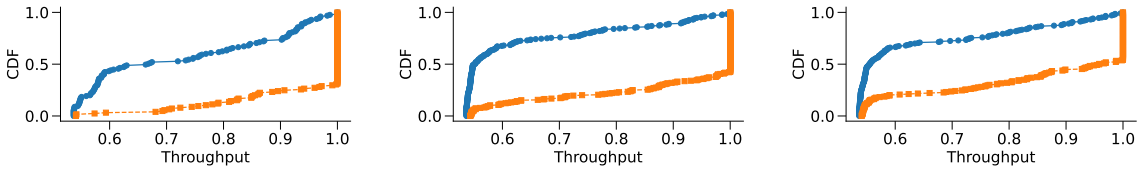
(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failures

Figure 24. Throughput in Verizon under flash crowds combined with one and two link failures.



(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failures

Figure 25. Throughput in Azure under flash crowds combined with one and two link failures.



(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failures

Figure 26. Throughput in Comcast under flash crowds combined with one and two link failures.

Spatial requirement: In all figures 22– 26, we see a gap in the CDF for throughput between SMORE and SMORE+GreyLambda. This gap indicates that the spatial requirement of TE is not being met in many scenarios by SMORE alone. It shows us that GreyLambda enables SMORE to improve aggregate network throughput in all traffic and fiber cut scenarios. This performance boost is attainable because GreyLambda considers the spatial requirement as a primary objective. In other words, GreyLambda hones into points of the network where traffic is being

dropped and increases the capacity at those locations. This capability reduces network bottlenecks that SMORE considers as a fixed constraint, thereby allowing more traffic to flow through the network.

Temporal requirement: Among the networks and failure scenarios Comcast and Verizon are the only two networks for which SMORE's TE execution time exceeds 1 min. The max/median/min for Comcast's optimization times are 69.8 s / 53.0 s / 39.3 s while Verizon's are 51.5 s / 15.0 s / 13.4s. Generally, SMORE can compute the TE optimization within the 5 min interval required by network operators. The max link length in Comcast is 3840 km and the 90^{th} percentile is 640 km; therefore, from our analysis in § 5.4 the estimated reconfiguration time for the longest link, with 47 amplifiers, is 2 to 4 s, and 0.8 s or less for 90% of the links (where the number of amplifiers is 7 or fewer). Therefore, GreyLambda meets the temporal requirement for TE. The time gap between GreyLambda and SMORE shows an opportunity to enhance the performance of network traffic with SMORE by quickly allocating bandwidth on congested links or around failed links more quickly than the time taken to recompute network flow allocations.

6.7.3 NCFlow Comparison. We compare the performance of NCFlow [2] by itself versus NCFlow+GreyLambda. This analysis uses the latest available version of the NCFlow simulator [1]. We make minor changes to the simulator to support GreyLambda by adding ∼700 lines of Python code. These code changes support the GreyLambda analysis by (1) allowing us to process NCFLow traffic matrices with different topology configurations (i.e., to support variable capacity edges) and streamlining the reporting of performance data from experiments, such as total link utilization on each network link after an experiment.

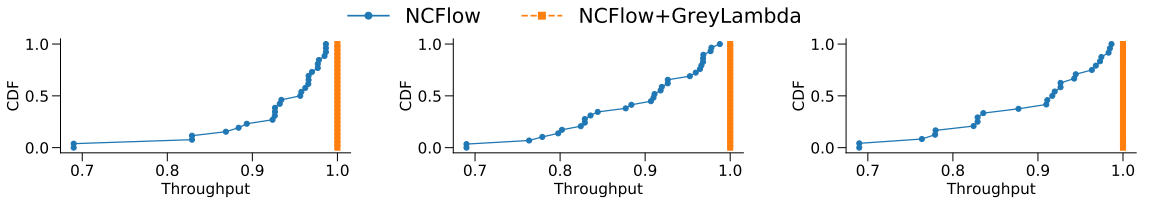
Similarly to the SMORE analysis previously reported, we test the performance of NCFLow and NCFlow+GreyLambda during flash crowd events as well as single and double link failure events. Our findings show that NCFlow+GreyLambda can fully satisfy all demands during flash crowd and fiber cut events in all five networks studied.

Spatial requirement: We find that in many cases, where there is a flash crowd or link failure in networks running NCFlow, throughput is severely impacted. As was the case with SMORE, there exist scenarios in a fixed network topology and among potential link failure scenarios where mitigating traffic loss is simply infeasible. However, NCFlow+GreyLambda can completely mitigate all traffic loss that occurs among the set of fiber cut and flash crowd scenarios.

Temporal requirement: In every experiment with NCFlow (on every network, traffic, and link failure scenario) the maximum time to solve the TE objective function is less than 3 s and the average TE computation time is 0.03 s. NCFlow satisfies the temporal requirement of TE. In cases where NCFlow is not able to completely fulfill the throughput demands, NCFlow+GreyLambda can bring a new link online in as few as 3 s, potentially stymieing losses at 300 Gb total for a 100 Gb link. Note that with TE alone the loss would endure so long as the traffic demand continues or until a physical link restoration is made.

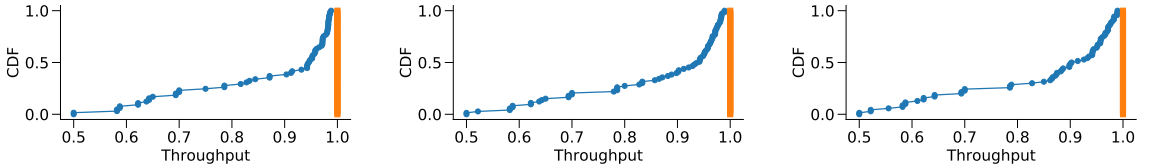
Extended discussion on NCFlow: It may be surprising that NCFlow+GreyLambda results are flawless concerning throughput. The reason we can guarantee such performance comes down to the speediness with which NCFlow solves its optimization function; when a link failure or traffic surge occurs, we can simulate the network throughput assuming every link has 2x bandwidth offline, then find which links in the network were utilized above 50%. When we go to implement the bandwidth

expansion along the constrained path with GreyLambda, we only need to activate links along the path where throughput was above 50% in the prior simulation. We can also find the critical path for expanding bandwidth in 0.03 s after the first traffic loss event is detected.



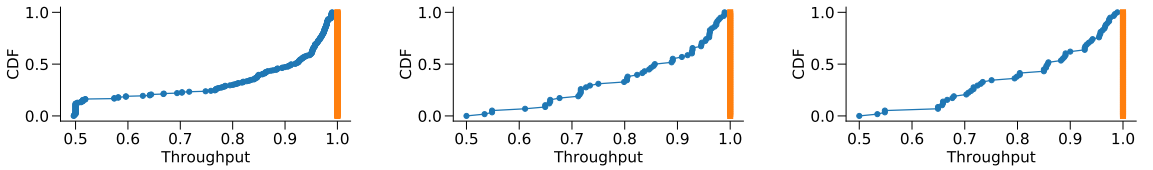
(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failures

Figure 27. Throughput in Zayo with NCFlow and NCFlow+GreyLambda.



(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failures

Figure 28. Throughput in B4 with NCFlow and NCFlow+GreyLambda.

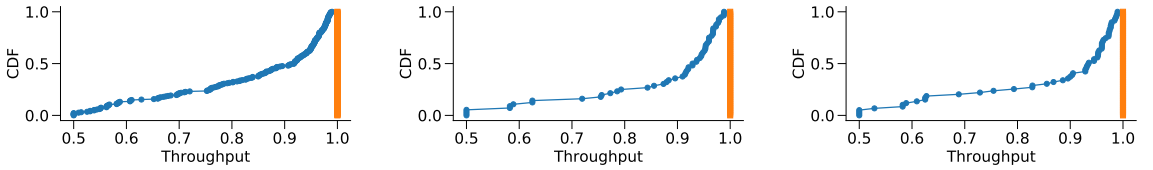


(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failures

Figure 29. Throughput in Verizon with NCFlow and NCFlow+GreyLambda.

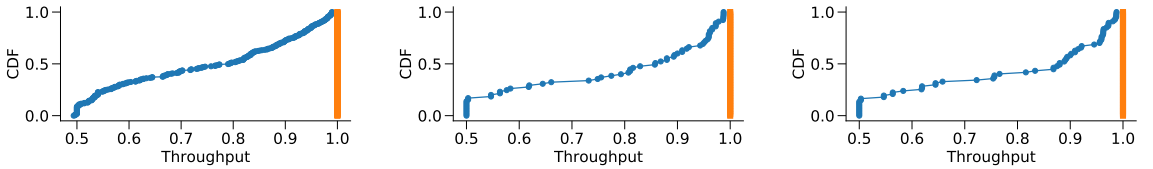
6.8 Related Work

Optimizing WAN network performance via TE has been of interest to both industrial and academic communities [2, 129, 137, 140, 170]. Approaches include B4 [137], SWAN [129], Owan [140], SMORE [166], and others [46, 61, 93, 110, 127,



(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failures

Figure 30. Throughput in Azure with NCFlow and NCFlow+GreyLambda.



(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failures

Figure 31. Throughput in Comcast with NCFlow and NCFlow+GreyLambda.

136, 138, 152, 222, 252, 253, 302, 325], each of which aims at improving the utilization of inter-datacenter WANs. A survey of related efforts is available here [7, 100, 273]. We posit that deployment of the techniques described in these studies along with OTP has the potential to improve performance results. Efforts complementary to ours include [46, 110, 152, 302, 322]. These share our goal of introducing programmability to the optical layer. However, with the exception of [322], these efforts do not address the performance penalties incurred by reconfiguring optical components in WANs. While [322] uses ASE noise channels to *prime* amplifiers for the addition of new wavelengths, we track and set amplifier gain explicitly, thus enabling new wavelengths to be provisioned between nodes where ASE noise channels are not present.

Provisioning infrastructure to enhance the robustness of networked systems has been a focus of many prior studies including backup routing [105], preventive routing via risk analysis [80], management system for provisioning [37], and backup paths via IGP link weight optimization [103] or RSVP-TE's *fast reroute* mechanism [228].

In the absence of dynamic allocations enabled by OTP, prior work has focused on fast failure recovery techniques at the higher layers of the network (e.g., [9,128,246]). Our work can be used in conjunction with these efforts since it augments the network capacity by dynamically allocating wavelengths to the recovery paths.

OpenConfig [226] provides the optical networking community with an "open" system [64] designed to connect the optical and IP layers. While OpenConfig is a compelling effort, given the scope of the technical challenges, we posit that the current level of attention in the networking community is not nearly enough. Specifically, to enable programmability at the optical layer, we must understand the potential gains and challenges in realizing OTP—the main focus of this chapter. Currently, there is no unified formulation of how much value OTP-like methods can bring to currently deployed TE schemes. Without such understanding, providers will be reluctant to adopt OTP, since it implies a radical change in a network's control system.

The concept of OTP has similarities with prior work on providing cross-domain light path provisioning for multi broker-based multi-domain software-defined elastic optical networks (SD-EONs); e.g., see [41,42,53,191,261,313] and references therein. These broker-based approaches realize cross-domain light path provisioning with Nash bargaining-type cooperative games [53, 261]. Whereas the experiments described in this chapter focus on demonstrating and quantifying the performance of OTP. In contrast, the work on a distributed multi-continental infrastructure reported in [41] is concerned with assessing the feasibility and validity of managing the workflow of a broker-based architecture.

Mahimkar et al. designed a bandwidth on demand service, and benchmarked link activation times between ROADMs [188]. Their system was described as a service that a tier-1 ISP might provide for large clients (e.g, cloud providers). We differentiate our

work from theirs in that we study the benefit of OTP with TE in a more limited scope by addressing the performance penalties imposed by amplifiers and transponders.

A method for optimizing bandwidth globally using bandwidth variable transceivers (BVTs) is considered by Ives et al. in [136]. A followup effort [135] varies the length of fiber spans and quantization steps for BVTs to analyze the throughput gains in a point-to-point (and not transparent) network. Both efforts seek to tackle the problem of reconfiguring optical transponder's modulation formats, while only the first one considers wavelength routing. We note that these efforts produce static allocations for optical paths and do not consider rapid reconfiguration or recovery in the face of unforeseen events like fiber cuts or flash crowds.

Similar to our effort, stabilizing optical paths via predetermined amplifier gains are explored in [205, 224]. In particular, Oliveira et al. [224] show that they can use a cognitive approach to select amplifier gain. Building on top of [224], Moura et al. [205] present a case-based-reasoning solution for stabilizing circuits in OTP. These efforts require extensive offline measurements of the amplifiers in the network. In contrast to these efforts, we do not require such measurements; we build our knowledge of the amplifier's optimal gain by directly applying wavelengths to the optical path and saving the resulting configuration settings in a lookup table for future reference. Moreover, unlike these efforts, we also capture and explicitly set the power levels between the transponders and ingress amplifiers.

6.9 Summary

In this work, we present GreyLambda, a framework for augmenting traffic engineering with optical topology programming. We present theoretical models to quantify the potential benefit of topology programming. We then conduct lab-based experiments on long-haul optical fiber to quantify, dissect, and reduce the link

reconfiguration time from minutes to milliseconds. Finally, we bring the theoretical model and data from our lab experiments together with a cross-layer optical and traffic engineering network performance simulator. We use the simulator to analyze the benefit of topology programming for five real-world network topologies under diverse traffic and link failure scenarios using two state-of-the-art traffic engineering systems. We find that optical topology programming offers a significant benefit to network performance during high traffic and adverse link failure scenarios.

INTERLUDE: LINK-FLOOD ATTACKS

A link-flood attack (LFA) is a DDoS attack that overwhelms the network bandwidth for a victim [202]. This attack can be broken down into a sequence of steps that the attack may repeat indefinitely. Figure 32 shows a high level overview of the stages of an LFA. First, the attacker gathers information about the network from any available sources. Then, they aggregate this data to construct a map of the network. The map is used to identify weak links in the network.We refer to these two steps as the *reconnaissance phase* of the LFA. After the reconnaissance is complete, the attacker deploys bots strategically within the network. Then, the bots being sending packets through the network that appear as legitimate messages and web queries but that flood the link and thereby make communications across that link severely limited for all other legitimate users. Chapter VII shows how OTP can subvert network reconnaissance efforts and Chapter VIII shows how OTP can be used to reconfigure the topology of a network during an ongoing LFA and thereby minimize the attack's impact on network availability.

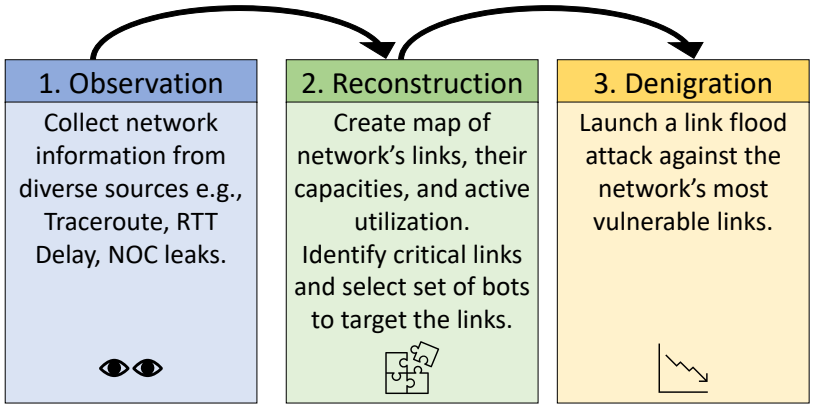


Figure 32. Attack stages for an LFA. Chapter VII presents an applications to mitigate the reconnaissance steps (1 and 2). Chapter VIII presents an application to address the active denigration stage of the attack.

CHAPTER VII

DOPPLER: A FRAMEWORK TO DEFEND NETWORK RECONNAISSANCE ATTACKS

This chapter contains previously unpublished coauthored material currently in submission [218]. This work is coauthored with Loqman Salamatian and Ramakrishnan Durairajan. The dissertation author collected the data used in this section. Loqman Salamatian used the data to implement the attack presented. The dissertation author designed and implemented the defense mechanism for the attack and ran the experiments assessing the efficacy of the defense. The coauthors assisted with editing.

7.1 Introduction

The prevalence of Distributed Denial-of-Service (DDoS) attacks is increasing, posing a significant threat to today's Internet [40, 67, 104, 237]. The escalating attack volumes, diverse attack strategies, and the low cost to launching such attacks make DDoS a critical issue. A key trend in this context is the growing focus of attackers on infrastructure through *reconnaissance*. This involves fingerprinting the network (e.g., mapping link bottlenecks using `traceroute`) and sending targeted traffic to flood those identified bottlenecks [149, 260]. The ultimate objective for attackers is to execute adaptive DDoS attacks, characterized by a cycle of *attack, reconnaissance, adaptation, and repetition*.

To combat network reconnaissance, several efforts have investigated IP-based topology obfuscation techniques. Notable efforts including NetHide [196], EqualNet [153], ProTo [131], among others, employ virtual IP addresses and obscure the topology by adding virtual links. However, using virtual addresses in a network is costly and not scalable, as they consume allocable space and disrupt the typical

routing-tree model. This reduction in allocable addresses, combined with the scarcity of IPv4 addresses, has led to the squatting of addresses in networks—a trend that may be incentivized by obfuscation techniques inflating address numbers. Additionally, the claim that virtual IP addresses are unnoticed by adversaries due to randomization within a subnet is questionable. Well-known techniques [30, 101] exist today to detect IP subnets, making a network's subnet plan reversible and noticeable to outsiders.

Recent developments in the network measurement community highlight a novel approach to extract topological information through delay measurements between network hosts [238]. Originally intended for unraveling the topological characteristics of private backbones owned by global cloud service providers, we show this method poses a genuine threat to network security and privacy communities. Concretely, our work introduces a first-of-its-kind network reconnaissance called the Ricci attack, leveraging this approach. Unlike the traceroute-based reconnaissance, the Ricci attack exclusively depends on end-to-end latency measurements, which are considerably more challenging to disguise without potentially disrupting service for everyday users. We demonstrate with four examples that network operators often publicly share tools (e.g., via open network operations centers or NOCs) that can be exploited to obtain the necessary latency information. Our approach is capable of identifying on average 50% of the network connections (i.e., edges) that would be detected using non-obfuscated traceroute probes, while relying solely on end-to-end latency data.

Taking a step back, an observation we make is that existing defenses to combat network reconnaissance treat network topology as a static entity, obscuring only the packet processing/forwarding logic. What is critically lacking are new dimensions of defense agility that can programmatically control topological characteristics to combat advanced network reconnaissance such as Ricci attacks.

This work explores one such promising avenue for countering advanced network reconnaissance by leveraging advances in optical networking called *Optical Topology Programming* (OTP). OTP allows dynamic reconfiguration of the optical layer by programming wavelengths. Our key insight is that dynamic reconfiguration at the optical layer is oblivious to `traceroute`-based reconnaissance. Building on this insight, in this work, we introduce the concept of "topology jitter" to thwart advance network reconnaissance. Thus far, OTP have been adopted for classical networking tasks (e.g., traffic engineering [73, 109, 210, 234]). To the best of our knowledge, the benefits of OTP have not been explored in depth for defending advanced network reconnaissance.

Bringing OTP to defend reconnaissance in practice, however, is fraught with three challenges. The first challenge is to ensure that the defense is robust against multiple-vector reconnaissance (e.g., traceroute, Ricci, etc.), even considering attackers' awareness of our defenses and ability to adapt probing methods to changing network topology. The second challenge is to prevent the inadvertent disruption of the network's functionality when employing OTP. Specifically, care must be to taken to ensure that endpoints, which were identified as targets in a previous reconnaissance mission, do not affect the performance of benign users or lead to a new DDoS once the optical layer has been reconfigured. The third challenge is to make the output of the reconnaissance on the updated network topology significantly different than from the previous one in a cost-effective manner. This forces attackers to initiate a completely new reconnaissance phase, as their prior insights are ineffective in launching an attack.

We introduce Doppler, an innovative OTP-based defense to fortify networks against reconnaissance. Unlike conventional IP-based methods, Doppler induces "topology jitter" by dynamically altering network links through manipulation of

optical wavelengths, thereby rendering traditional reconnaissance tactics ineffective. Doppler not only thwarts reconnaissance attempts but also maintains optimal network performance. Our approach hinges on the fundamental premise that the adversary seeks to identify and exploit specific source-destination pairs and their corresponding traffic flows, rather than merely map the network topology. Thus, our objective is to increase the cost for attackers to discern these flows. To this end, Doppler uses the following key insights. First, Doppler ensures that the network is in a constant state of flux, making reconnaissance attempts based on assumption of static topology ineffective. Next, by employing an optimization model, Doppler rapidly adapts the network topology without affecting the performance of benign users, significantly outpacing the time it takes for adversaries to fingerprint the network. Finally, the defense strategy also considers performance simulation for multiple topology solutions and adapts to scenarios where spare transponders are unavailable, showcasing its versatility in operational networks.

We evaluate the efficacy of Doppler under a diverse set of operating parameters emulating potential deployments, varying the number of optical transponders present at different network endpoints and the number of endpoints at which they are present. Within each set of physical transponder deployment scenarios that the system is evaluated under, we also look at two selection strategies for point-to-point network layer links among them—a conservative and a liberal strategy where the conservative strategy is 3 to 10x smaller than the conservative one depending on the specific topology (generally the difference in size between them is greater for larger networks). In summary, this work makes the following key contributions:

(1) We demonstrate an advanced network reconnaissance attack called the Ricci attack that only relies on end-to-end latency measurements.

(2) We show that Ricci attack is possible at scale by running latency measurements on open network operations centers (NOCs) of four networks.

(3) We propose a first-of-its-kind OTP-based defense called Doppler against advanced network reconnaissance.

(4) We demonstrate the efficacy of Doppler using simulations. Our results show that Doppler is fast, capable of solving more than 90% of problem instances in 30 seconds or less while ensuring network throughput for 100% of traffic.

7.2 Background and Motivation

To carry out a reconnaissance, the attacker must identify their target network e.g., a small or regional backbone network. The attacker must have a set of source probing nodes, e.g., bots that run `traceroute` in a loop. The bots should be inside and outside the network to maximize the coverage of the scanning effort, and should send probes to addresses both inside and outside the network as well. The result of the scans is then aggregated into a collection of paths, and the collection of paths is then aggregated into a collection of network links by finding paths that share one or more mutual nodes. The attacker can use advanced network measurement techniques to improve their resultant map, for example tools like Scamper [184] can aid the attacker with alias resolution by clustering sets of nodes from the `traceroute` dataset, each representing a specific router interface, into a single node to represent the physical router itself. With their map of the network's topology in hand, they can strategically place their bots to maximize traffic demand for the link that they wish to target.

7.2.1 Threat model.. In this work, we are primarily concerned with the attacker's ability to gather information with sufficient accuracy to enable a LFA. The network they target is a small-to-medium sized network (e.g., campus, enterprise, or local Internet service provider). The attacker has the following tools at their disposal

which they execute from a set of compromised hosts in the target network. Firstly, they can run `traceroute` between compromised hosts; the `traceroute` probes may or may not be successful in identifying router paths in the network (e.g., possible responses to probes may be stars (`***`)). Secondly, they can collect round-trip time *delay* measurements between compromised hosts. The attacker can always collect delay measurements so long as the two hosts from which the measurements are taken are reachable through the network (the network is not partitioned). Thirdly, they can collect insider information about a network via network operation center (NOC) servers when such servers are openly accessible via the Internet. Information that the network may potentially leak in their NOC includes intra and inter-network links, operator descriptions of the links, their geography, bandwidth, and utilization. The attacker synthesizes the information available from these diverse sources to find a set of vulnerable network links, i.e., links that may be bottlenecks, heavily utilized, or low capacity.

7.2.2 Prior Efforts. There have been various efforts to prevent network reconnaissance in prior work. These range from IP-based topology obfuscation techniques (e.g., NetHide [196], EqualNet [153]) to the wholesale practice of anonymizing routers on the Internet by disallowing them to respond to `traceroute` probes.

IP-based obfuscation is a reconnaissance defense which causes an observer to see a distorted picture of a network's connectivity from the outside. To present this distorted view, the defense introduces virtual nodes and links in the network. One such work by Meier et al., NetHide [196], aims to generate a network that is *secure*, and *debuggable*; in this context, secure means that it, "prevents the attacker(s) from determining the set of flows to congest any link," and debuggable means, "still

allowing non-malicious users to perform network diagnosis." In other words, their goal is to distort `traceroute` messages for outsiders while preserving just enough real-world information in them for an operator to be able to map them back to a physical location.

A more recent state of the art defense, EqualNet [153], highlights some of NetHide's weaknesses and improves on them. In particular, EqualNet argues that the method for generating random topologies in NetHide is insufficient and that these topologies "leak" information about the underlying physical connectivity. To address this, EqualNet proposes a solution that minimizes "leakage", defined as the greatest difference in *flow density* between any two nodes in the network. Here, flow densitiy is the number of times that a node is seen in a set of `traceroute` responses. To minimize leakage, they present an algorithm that deploys virtual routers and links for *every hop* along a given path until leakage is below an operator-specified level.

7.2.3 Limitations of Prior Efforts. Using virtual addresses is costly and, therefore not scalable. Virtual addresses take away allocable space in the network's address range. Furthermore, by creating a virtual interface for *every hop* through the network, the practice turns the typical routing-tree model of a scalable network upside down. Figure 33 shows a simple example of a topology where seven virtual routers are added to obscure the paths between three hosts.

Introducing virtual addresses to a network reduces the number of allocable addresses within a specific AS's pool. IPv4 addresses are a scarce commodity [236] and as a consequence of this scarcity some networks have begun to assign addresses that do not belong to them to infrastructure and hosts in their network. According to the study [239] the practice of squatting has increased greatly since 2020. Seeing this trend we posit that obfuscation techniques which inflates the number

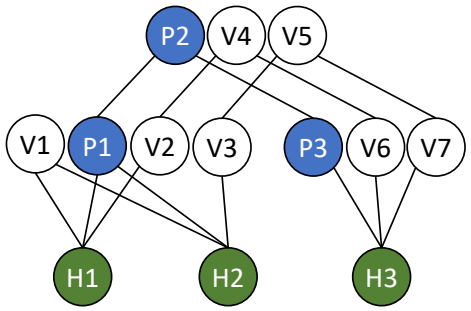


Figure 33. Routing tree for a network with three physical core routers (P1, P2, P3), three hosts (H1, H2, H3), and seven virtual routers (V1, ..., V7). The virtual routers provide the hosts with the illusion of two disjoint paths between each other.

of addresses in the network will inevitably incentivize further squatting. While IPv6 presents an alternative, its adoption faces several challenges, including complexities in transitioning from IPv4 due to compatibility issues [114], the need for updated networking equipment and software [172], security concerns unique to IPv6's architecture [314], and a lack of familiarity and training among IT professionals. Together, these factors contribute to the slow uptake of IPv6, despite its potential to solve the address scarcity problem.

Even in cases where address space is not a concern, there are still other drawbacks associated with the IP-based obfuscation approach. In [153], the authors claim that EqualNet's virtual IP addresses are not noticeable to an adversary because they are randomized within a subnet, and the subnet structure is assumed to be private and unlearnable knowledge from the attacker's perspective. However, there are techniques to detect subnets in a network [101, 116] and these efforts, demonstrated by prior network measurement work, directly counter this assumption. That is to say, a network's sub-net plan is conceivably reversible and noticeable to an outside observer [30, 101]. If an attacker observes hotspots in a network's address space then this is evidence enough to differentiate between virtual and physical interfaces.

Summary. There have been various efforts to prevent reconnaissance, ranging from obfuscation techniques to completely discarding traceroute probes. All of these previous attempts were focused on disturbing the mapping of a network's topology by blurring traceroute probes. In § 7.3, we demonstrate that even in the extreme case, where the traceroute only returns end-to-end information, an attacker can still identify bottlenecks with high accuracy and launch impactful attacks.

7.3 Modern Network Reconnaissance: Beyond `traceroute`

In this section, we use a set of tools that do not include `traceroute` to map the link-layer topology of four public networks. Specifically, we show that we can use *Ricci curvature* to find highly-depended-on links in these networks, culminating in a new network reconnaissance attack that we call the *Ricci Attack*. In some cases, we can leverage *router proxies* from these networks over the open web to collect the vital RTT measurements needed for the Ricci analysis. Furthermore, we can obtain highly specific link-layer maps of some networks from router proxies without any active measurements, simply by using a command that we found available on every router proxy that we accessed. A summary of the networks we investigate is shown in Table 6.

Network	L2+L3 Switches	Open Router Proxy	Mappable?
$Network_1$	14	×	✓
$Network_2$	12	×	✓
$Network_3$	17	✓	✓
$Network_4$	10	✓	✓

Table 6. Networks investigated in this study. Names are anonymized.

7.3.1 The Ricci Attack.

As `traceroute` has lost its utility in the network measurements community, researchers have discovered a new method to uncover topological information using only delay measurements between hosts in a

network [238]. Although this method was initially proposed to uncover details about private backbone topologies owned by global cloud service providers, we recognize that the method poses a legitimate threat to network security and privacy communities. As a motivating example, we demonstrate how we extend the delay-based measurement technique to find critical links in the four backbone networks (from Table 6) operated by publicly owned entities and non-profit organizations.

A primer on Ricci Curvatures. The crucial insight of this approach lies in integrating round trip-time (RTT) measurements taken from a series of geographically distributed measurement devices combined with ideas originating from Riemannian geometry. Much like the original work, our approach begins by collecting latency measurements from various points within the network. The distance between measurement vantage points ranges from as low as a few hundred feet for some networks, to a few hundred miles miles for other. Given our lack of access to precise geolocation information, our study utilized the measured latency instead of the residual latency considered in the context of large cloud providers' backbones. We then create a graph for varying performance thresholds (ϵ), where a link is constructed between two points only if their latency is less than the said threshold. For each instance of ϵ, we form a graph of node pairs with "nearly straight" (up to ϵ) links. Given the significantly smaller granularity of our study compared to the original work, we re-scale the thresholds to be 100 $\times$ smaller than the one they had considered. In lower thresholds, the resulting graph informs on the localized structures; as we increase the threshold, we start to spot connectivity bridges where the physical and logical connectivity coincide.

To discern these pivotal bridges, we use the concept of Ollivier-Ricci curvature. Intuitively, the curvature of a link can be thought of as a local "betweenness" measure.

Negatively curved links are frequently part of local shortest paths. Considering reconnaissance contexts, we assign each link its curvature the first time it appears in the graph and hone our focus on those with pronounced negative curvature. Such links are particularly susceptible to attacks and thus emerge as primary targets for attackers planning to disrupt network activities, given their potential to affect a significant number of connected links.

7.3.2 Ricci Attack Workflow.

From curved edges to understanding vulnerability in the network. Using latency information along with the Ricci curvature enables a study of the topology that does not rely on traceroutes at all. We discuss how one can translate the insights obtained from the Ricci curvature study to launching a specific attack. The most important insight from that study is the presence of negatively curved edges that bridges connectivity between points in the network. By selecting the most negative edges, we identify end-points in the network whose link capacity we want to overload. It is crucial to recognize that these negatively curved edges might not represent *actual* links in the real underlying network graph. They can be an amalgamation of multiple edges. In this case, we assume that the links across the shortest paths are the ones that the attacker overloads at the end of reconnaissance. This phenomenon bears a resemblance with the properties of invisible, unresponsive hops that dramatically reduce the relevance of traceroute-based discovery, but does not impact the relevance of the inferred negatively curved edges as illustrated in Figure 34.

7.3.3 Ricci Attacker's Metric of Success. An attacker's metrics of success are:

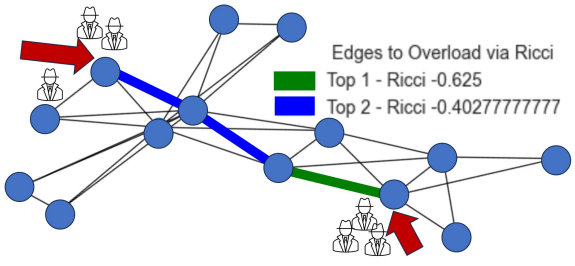


Figure 34. By finding edges with the most negative curvature, the attacker can strategically place their bots in the network to launch an attack with maximal impact.

Overlap coefficient. We consider the Overlap Coefficient [264] between a network's physical topology and the topology that we infer with respect to the set of links in each. For two sets of undirected links $\mathcal{E}$ and $\hat{\mathcal{E}}$, the overlap similarity, $\mathcal{O}$ is the ratio of the size of the set union to the size of the smaller set (Eq. 7.3.3).

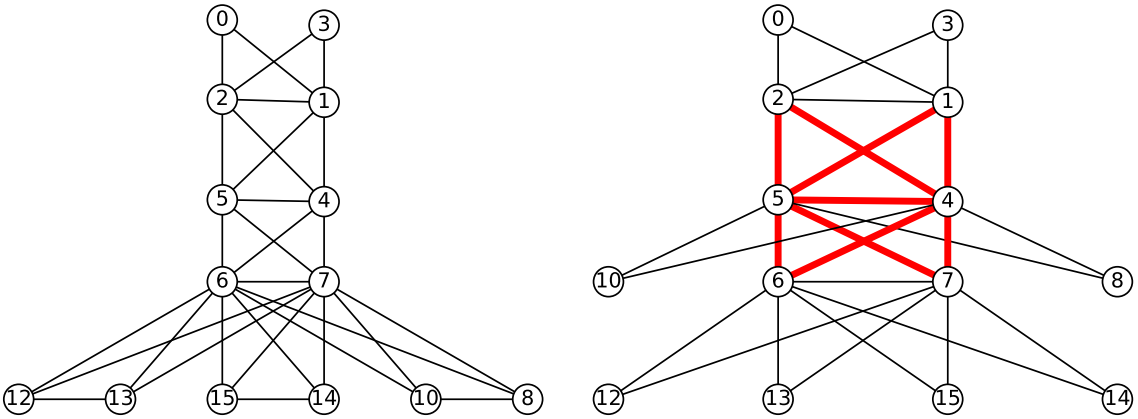
$$\mathcal{O} = \frac{|\ \bigcup(\mathcal{E},\ \hat{\mathcal{E}})\ |}{\min(\ |\mathcal{E}|,\ |\hat{\mathcal{E}}|\ )} \tag{7.1}$$

There are many ways for to infer topology, and to measure the efficacy of a given technique one can compare the overlaps between the real topology and the one that you has been inferred from said technique.

Curvature. Combining both notions of overlap and curvature, the overlap of negatively curved links for a reconstruction within the true network topology is an especially potent bit of information. This tells us how useful the reconstruction is to the attacker because if they can find a set of negatively curved links in the network then they have the information required to launch a devastating link flood attack on the most vulnerable and depended-on links in the network

7.3.4 Demonstration of Ricci Attack.

Network$_1$ **Case Study:** We applied the technique from Salamatian et al. [238] to a campus backbone network (hereafter referred to as *Network*$_1$) show in Figure 35. *Network*$_1$'s backbone, shown in Figure 35a, has 14 nodes and 30 links. The reconstruction, shown in Figure 35b, has 14 nodes and 27 links. The reconstruction correctly identified 24 of the 30 ground truth links, giving it an accuracy of 80%. Moreover, the delay based measurements were able to detect the critical links in the network as shown with bold red lines in figure 35b. These links were shown to have a negative Ricci curvature [225] based on the delay measurement technique, which indicates their importance for forwarding traffic from nodes on one network's end to the other.



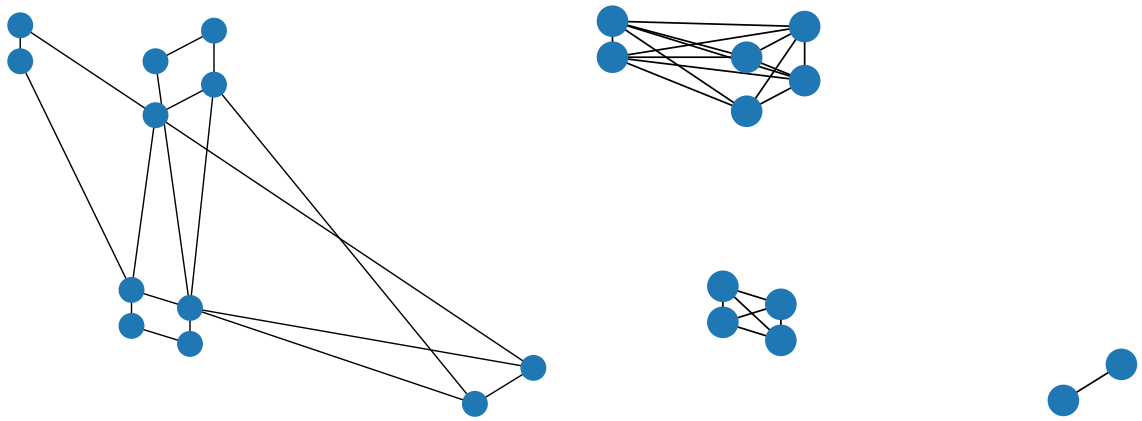
(a) Ground Truth topology from network operator.

(b) Topology inferred with Ricci Attack

Figure 35. Comparison of ground truth topology of *Network*$_1$ (a) vs. *Network*$_1$'s topology inferred with Ricci attack (b).

Network$_2$ **Case Study:** We worked with a regional research & education (R&E) network to assess whether it was vulnerable to the Ricci attack. This network's abstract topology is shown in figure 36. To this end, we collected Min RTT delay measurements from all pairs of routers in the network and applied the Ricci curvature

analysis to the data. The reconstruction accurately captured 50% of the links in the network. None of the links came up with a negative value for Ricci curvature.



(a) Ground Truth topology from network operator.

(b) Topology inferred with Ricci Attack.

Figure 36. Comparison of ground truth topology of $Network_2$ (a) vs. $Network_2$'s topology inferred with Ricci attack (b).

7.3.5 Open NOC Vulnerability. Having demonstrated the Ricci attack, we now show how the attack is possible by leveraging public information gleaned by running measurements on open network operations centers (NOCs) [111]. We call this the *open NOC vulnerability*.

While we find it possible to infer a network's vulnerable links via active measurements from outside the network, we also find some networks provide information about their infrastructure via a public facing, web-accessible, NOC. Some network operators host a Router Proxy in a public NOC, where anyone can open the page and scrape information from the routers. An anonymous user may run commands, e.g., `show interface`, `show route`, `show bgp neighbor`, and many more, to gain valuable information about the network's connectivity. For example, `show interface` lists all of the physical and logical interfaces on the device, and many of these have plain text descriptions to describe their connectivity with other internal

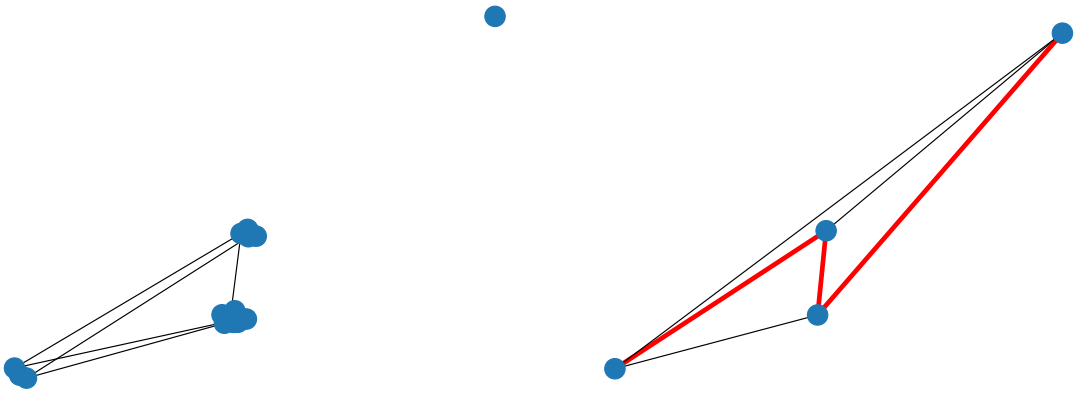
network routers, gateways to cloud datacenters, and the public Internet. We were able to use data collected from `show route` and `show bgp neighbor` to accurately map the infrastructure of three public networks with 100% accuracy.

These router proxies also lend themselves to be the perfect place to launch a Ricci topological inference attack, as you can directly run the commands `ping` and `traceroute` from each router to each other router on the NOC. When you launch a command from a router on the proxy server, you are given the IPv4 address of that router. From there, it is as simple as writing a script to collect the IP address of every router, and then to have the script send a ping command from every router to every one of the IP addresses that was collected.

Network$_3$ **Case Study:** Figure 37a shows the ground truth topology reconstructed by parsing `show route exact`. To do this, we ran `show route exact` between every pair of routers to get their routing table entry associated with the remote entity. We added a node to the graph for every router that issued commands and an edge connecting it to the address given by `next-hop`. If two routers share a `next-hop`, then we add a link between them. If the routers are more than two hops from each other we lack sufficient information to add an edge connecting them. We note that for this network, the only commands available were `show route exact`, `ping`, and `traceroute`. This process yielded six edges between 19 nodes in four distinct geographic clusters.

We collected min RTT data via `ping` from all pairs of routers in the router proxy, sending 30 ICMP echo requests between every pair of routers over one hour and thirty-six minutes. From here, we aggregated the nodes into a single point based on its geographic cluster. Figure 37b show the network's three most critical links as bold red edges connecting the clusters. The overlap from the Ricci attack topology with

the one inferred from `show route exact` is 3 of 3 (100%), and the overlap of Ricci edges to the Open NOC attack is 2 of 3 (66%).



(a) Topology Inferred by Routers

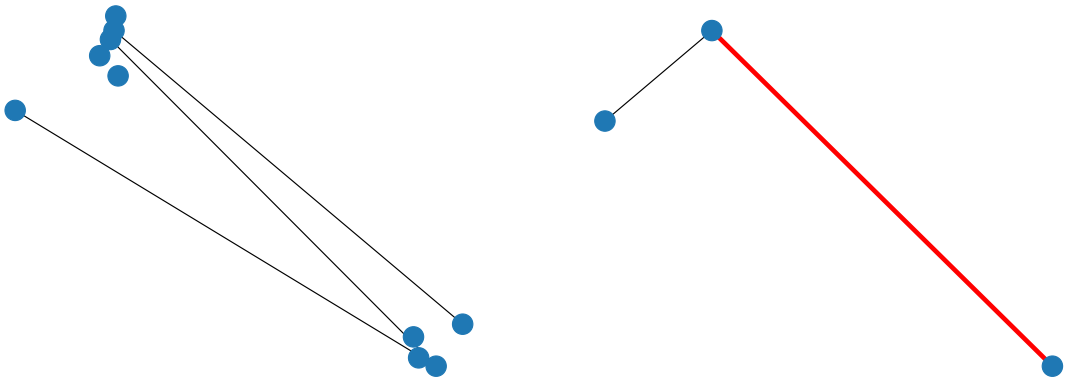
(b) Critical Links Inferred with Ricci Attack

Figure 37. Comparison of ground truth topology of $Network_3$ (a) vs. $Network_3$'s critical links inferred with Ricci attack (b).

$Network_4$ **Case Study:** Figures 38a shows the ground truth topology inferred by parsing `show route` and `show bgp neighbor`. The map was created in the same fashion as $Network_3$ Case Study, but this time we had access to `show bgp neighbor` to corroborate our findings. Specifically, we ran `show bgp neighbor` from every router. For each router, we first ran through the table and collected the Local Address and Local ID of the router to populate a list of aliases for that router. Then, we looked through the table again to find a Peer Address and Peer ID for every neighbor entry. We added these two addresses to a list of alias for that peer, then checked to see if either alias been added to a node in the graph yet. If not, then we created a new node for that peer and a link connecting it to the source router. The graphs produced through this method and via `show rout` were 100% identical, having no dissimilar nodes or edges.

We collected min RTT data via `ping` from all pairs of routers in the router proxy, sending 30 ICMP echo requests between every pair of routers over 40 minutes. From here, we aggregated the nodes into a single point based on its geographic cluster. Figure 38b shows the network's a single critical link as bold red edge connecting the clusters.

Figure 38b shows the inference gleaned from collecting min RTT data via `ping` from all pairs of routers in the router proxy. The overlap from the Ricci attack topology with the one inferred from `show route` and `show bgp neighbor` is 1 of 2 (50%), and the overlap of the Ricci edge to the Open NOC attack is 1 of 1 (100%).



(a) Topology Inferred by Routers

(b) Critical Links Inferred with Ricci Attack

Figure 38. Comparison of ground truth topology of $Network_4$ (a) vs. $Network_4$'s critical links inferred with Ricci attack (b).

7.4 Defending Ricci Attacks using Doppler

We propose an Optical Topology Programming (OTP)-based defense called Doppler against network reconnaissance. The key insight of Doppler is that changing the set of links by programming the optical wavelengths in a network fundamentally changes all indirect inferences that can be made by an outsider. This insight induces "topology jitter" and enables Doppler to overcome the pitfalls of pure IP-

based topology obfuscation (§ 7.2.3). Collecting topology information can be a long and methodical process, and the known methods assume that the topology under investigation is relatively static, only changing every few months or years. However, we show that alternative ways to (re)organize the links in a network frequently while (1) thwarting network reconnaissance attacks and (2) maintaining performance for ongoing network traffic are possible.

7.4.1 Challenges.

C1: Defending against multi-vector smart reconnaissance. An attacker can conduct reconnaissance campaigns that are *adaptive* to network topology changes and are *diverse* with respect to the methods employed to conduct the reconnaissance (e.g., `traceroute`, Ricci Attack, and exploiting Open NOC vulnerabilities). As we have shown, `traceroute` is far less effective today than in prior years and attackers have new ways to glean topological insight. Therefore, we must address this trend with an advanced defense that is effective against more reconnaissance vectors than `traceroute` alone. Furthermore, we recognize that the attacker studying the network may be aware of our defense. They can even solve our optimization model and gain a solution from it. Therefore, they can adapt their probing methods as the topology changes. In light of this capability our defense must be effective despite the adversary's awareness of the defense.

C2: Ensure benign users are not hurt by the defense. We are proposing to change the network's physical topology with OTP. If the topology changes are carelessly rendered, then this could result in reduced performance for benign traffic. Therefore, we must provide an optimization for OTP that jointly increases the difficulty of useful reconnaissance for the attacker while maintaining high quality

of service for benign network users. This is especially challenging because joint optimization of topology and routing is NP-Hard [160].

C3: A cost-effective defense. Our defense must be cost effective, and further introduce a *cost asymmetry* between the defending network and the attacker. It must therefore maximally increase the attacker's cost to learn a set of *critical flows* while minimally increasing the cost for the network to employ the defense.

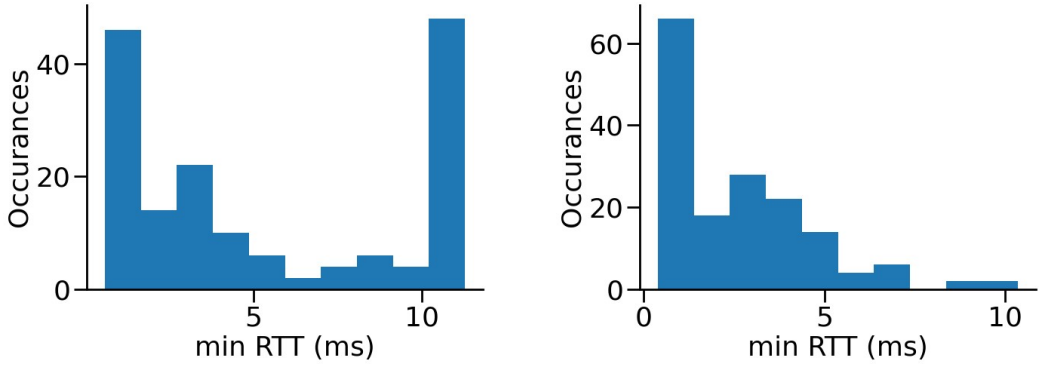
7.4.2 Our Approach. In this work, we take a first principles approach to network reconnaissance defense by addressing the aforementioned challenges. The basic and irreducible assumption that guides our approach is that the attacker's ultimate goal is to find a set of source and destination pairs and sending rates (i.e., *flows*), $F = \{ (s, d, c) \}$, such that sending data from each source, s_i, to its destination, d_i, at rate, c_i, bits per second (bps) leads to a decline in network performance for all users of a given network. In other words, the attacker's goal is not to *map* the network with `traceroute` or to find all of the links and the underlying routing system that connects end-hosts through them. Rather, mapping the network is a just a means to this end. Therefore, *our goal is to increase the cost, K, for an attacker to find this set of flows, F.*

To this end, we propose an OTP-based defense called Doppler that uses the following key insights:

I1: A network always in flux. To address (C1), Doppler uses an optimization model (described in § 7.4.3) that is designed to be fast, i.e., solvable in 30 seconds to 5 minutes, depending on the needs of the network. This is well below the order of time than that taken for an adversary to fully map the network (with either delay

tomography or `traceroute`) because delay measurements are inherently jittery. It can take many attempts to get a min RTT that is usable for reconnaissance efforts.

For example, Figure 39 shows the distribution of min RTT that was collected from $Network_3$ with two different probing campaigns. First (Figure 39a), the command `ping count 3` was sent once from every pair of routers, with requests staggered in 5 second intervals (so as not to have the sender flagged by the receiver for sending too many requests). Overall this took about 25 minutes and the distribution was multi-modal, with a significant number of min RTT values above 10 ms and two other modes at 1 ms and 3 ms. Upon retrying the campaign, this time sending ten `ping count 3` commands from each pair, the distribution settled into just two modes, around 1 ms and 3 ms.



(a) (b)

Figure 39. min RTT frequency distribution for $Network_3$ with two timing intervals. (*a*) 3 pings per router pair, 25 minutes. (*b*) 30 pings per router pair, 1 hour and 36 minutes.

We solve the optimization fast by enumerating a limited set of alternative paths that are possible through the network when different sets of nodes are active. This set can be very large, but we do not need *every* possible path; just enough to enable some topological variance.

If the topology is changed at less frequent intervals, a smart attacker might still enumerate the different possible map the routing for each possible topology as it occurs. Therefore, the topology change must be non-deterministic so that that attack cannot simply cycle to their next map when they notice a change. We enumerate 100 potential solutions to our objective function and rather than taking the optimal objective, we take a random solution that satisfies all of our constraints. The number 100 is not prescriptive; we could enumerate more or fewer topologies, but we find that we are able to consistently find 100 solutions that satisfy our constraints with varying input parameters, and the set gives an attacker a mere 1% chance of accurately guessing the topology if they are solving the same objective function to aid their reconnaissance.

I2: Performance simulation for many solutions. When we find a candidate set of solution topologies, we compare the expected network performance of each solution against each other. We find the expected network link utilization for links in the prospective topology and the latency distribution between all pairs of nodes. If any of the solutions fails to meet one of the network operator's desired criteria and/or affect the performance of benign users in the network, then that solution is discarded.

I3: No spare transponders? No problem. While it is advantageous to have extra transponders throughput the network, it is not always possible due to cost constraints within the network's operating budget. Therefore, we have designed a solution that can re-assign network links between existing pairs of transponders in the network. This limits the number of potential solutions, but when on-going network traffic demand is light we show that there are still a diverse set of topology configuration solutions available without adding more transponders to the network.

Regarding the attacker's cost, we show that there are sets of topology configurations that are more difficult for the Ricci Attack to gain inference from. This set of solutions can be useful for long-term rout planning in networks that do not want to enable OTP while also considering the Ricci Attack as a threat to their network operation.

7.4.3 Doppler Optimization Model. The objective of the optimization model in Doppler is to produce a topology with the feasible flow forwarding property that has minimal edge overlap with the original topology.

The Doppler optimization model uses the constraints defined in Chapter IV, § 4.3. It's objective function is given by equation 7.2.

$$\texttt{minimize}(|E_0 \cap E'|) \tag{7.2}$$

Where E_0 is the initial set of (directional) links active in the network and E' is the new set of links.

The this model is passed as an OTP method to our OTP simulator introduced in Chapter IV, § 4.4. The first step in leveraging the model is to enumerate the set of paths through the network that traffic can use when different sets of links are active. To this end, we find a set of paths for each source and destination using an iterative depth first search on the graph with all of the links active. In the model implementation, we limit the set of edges available to any flow to only those edges that appear on a path that was found a priori. The model yields a set of solutions whose quantity can be selected by the network operator.

7.5 Evaluation

We evaluate the efficacy of Doppler to quickly find a set of solution topologies that nullify an attacker's previous reconnaissance efforts while maintaining steady

performance for ongoing network traffic. Our evaluation uses the four networks shown in Table 6 and seeks to answer the following research questions.

- Can Doppler adapt topology even if there are no fallow transponders? (§ 7.5.2)
- Are there unintended consequences of Doppler? (§ 7.5.3)
- How do reconnaissance outputs compare before and after Doppler? (§ 7.5.4)
- How does Doppler perform with low time constraints? (§ 7.5.5)

7.5.1 Simulator Parameterization. Before diving into the results, we briefly describe the simulator parameters and their ranges used in our analysis.

Transponders. We vary the number of transponders allocated to the different sets of nodes within the network. In the analysis to follow "Top K" represents the top K% of nodes in the network based on betweenness centrality and varies from 0 to 100% in steps of 10. The parameter, *n_ftx*, refers to the number of fallow transponders (ftx) each node in the Top K set has available. We vary n_ftx from 1 to 3. Note that at K=0, there are no fallow transponders at any of the nodes in the network. The topology can still change in this scenario, but any links that are in the updated topology must use transponders re-allocated onto the new link.

Path finding constraint. The size of the set of paths that the model finds before can be configured with a setting we call "candidate link selection". It is essentially an informal constraint that limits the set of paths derived by Doppler by limiting the set of links that are available to the paths. We have implemented two strategies, which we evaluate head-to-head. In the "conservative" strategy, we enable links to be added if they can short-cut one of the core links of the graph, i.e., a link with a low curvature

value. In the other strategy we call "max" there is potential for a link between any pair of nodes that are two hops away from each other in the physical topology. The candidate link selection strategy will ultimately depend on the network's available hardware e.g., wavelength switches or ROADMs, at nodes to facilitate bringing any of the candidate links physically online.

Runtime constraint. Doppler can be configured to yield a set of solutions which satisfy the constraints but which might not be strictly optimal by setting a solution time limit. This parameter can be set by a network administrator according to the needs of their network and their preference for optimality vs. speed. In our evaluation, we show Doppler's performance when given a time limit of 30 seconds, 1 minute, and 5 minutes.

7.5.2 Can Doppler Adapt Topology Even if There Are No Fallow Transponders?.

We first explore the cost-effectiveness of Doppler to adapt the topology when there are no fallow (i.e., extra) transponders available. Figure 40 shows a the number of solutions found by Doppler with no fallow transponders available at any nodes. $Network_1$ and $Network_2$ struggled to find more than one solution for some of the operating parameter settings. Specifically, when given 30 seconds to solve the optimization and when using the "max" candidate link choice strategy. These parameters left the model with too many variables to solve for and not enough time. Interestingly, when the "conservative" candidate link choice strategy was employed both of these networks completed their solution pool. The set of paths in the conservative pool was 65 to 70% smaller than "max", and offered enough potential path diversity for a solution to be found quickly. $Network_3$ and $Network_4$ and virtually no trouble completing the solution pool irrespective of the operating parameters.

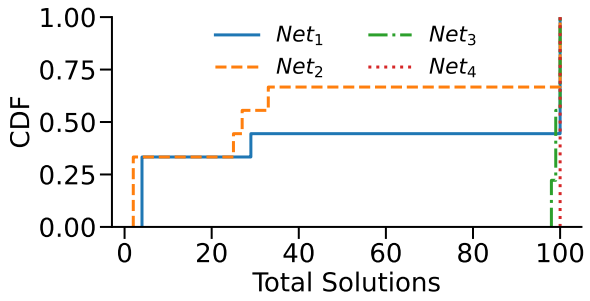


Figure 40. CDF of total solutions found for different networks with no fallow transponders.

Figure 41 shows the change in max link utilization for the entire solution pool across all parameters as a letter-value box plot [125]; in each plot, the largest box shows where 50% of the data points are, and the next captures 25%, and each successive box is another half of the remaining data. In this experiment, there are still no fallow transponders at any of the nodes, and links must be reconfigured using existing transponders at the various nodes. We see that $Network_1$, $Network_2$, and $Network_4$ have similar distributions. That is, they display modest increases in max link utilization, with 75% of the results having yielding a 10% increase or less. In $Network_2$ 79% of the results actually yielded a decrease in max link utilization. $Network_3$, however, struggled more to maintain low max link utilization across the board. Only 2% of the solutions have a max link utilization change of 30% or less.

These results show us that the impact of topology programming on existing traffic can be minimized even when the network does not have extra transponders. However, as show by the stark difference in our results for $Network_3$ compared to the other networks, the extent to which network performance impact can be minimized his highly specific to the underlying topology and traffic matrix.

7.5.3 Are There Unintended Consequences of Doppler?. We study whether the endpoints identified for flooding in the original graph's reconnaissance

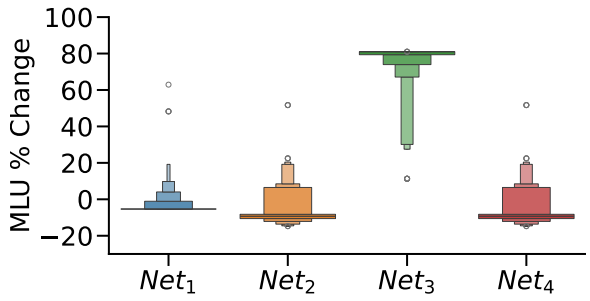


Figure 41. Letter value box for the change in Max Link Utilization from the starting topology to the solution topologies.

do not inadvertently flood the newly formed topology. This step is crucial to prevent any unintended disruption of the network and/or consequences on the performance of benign users. To study this question, we delve into the consequences of carrying-out an attack on the revised network topology using "prior" reconnaissance. In particular, we identify which end-points are picked to be overloaded as a result of the initial reconnaissance on the initial graph and then assess if these end points, post-update, traverse any bottlenecks links as identified by running another reconnaissance on the updated graph. The CDF plot in figure 42 highlights the most negatively curved edge encountered along the attack path. A shift towards more positive curvature values indicates improved effectiveness in mitigating attacks that are launched with out-dated intelligence. Conversely, if the attack path in the updated network still encounters negatively curved edges, it indicates that the attack might still be successful despite our change in the resulting topology.

7.5.4 How Do Reconnaissance Outputs Compare Before and After Applying Doppler?. Our next focus is to guarantee that the reconnaissance findings on the network, after it has been updated, differ significantly from those of the original reconnaissance. This distinction is crucial to prevent the application of insights gained from the original topology to the updated one. In Figure 43, we

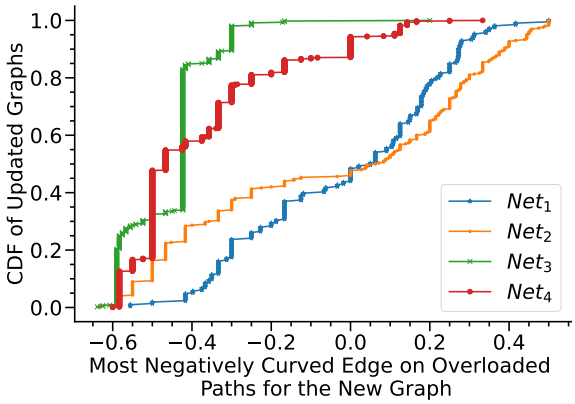


Figure 42. CDF showing the curvature of the most negatively impacted edges along attack paths post-Doppler updates. More positively curved edges are less likely to be bottlenecks and, therefore, their overload is less likely to impact the network at large, whereas negatively curved edges are still bottlenecks.

showcase the overlap of the top 5 most negatively curved edges between the original and post-update topologies. The choice of 5 edges is based on the observation that more than 99% of the graph considered in our study admitted a maximum of five negatively curved edges. Ideally, a successful topology update should result in no common negative edges between the two states (i.e., before and after Doppler), whereas a less efficient strategy would result in a larger overlap of these critical edges. In particular, we see that the new topologies are highly diverse with respect to negative edges across solutions, where more than 50% of all solutions in all networks tested share no negative edges at all.

7.5.5 How Does Doppler Perform with Low Time Constraints?. We turn our attention to the runtime performance of Doppler. Specifically, we want to know how Doppler performs with different time constraints, and how the time constraint affects the quality and quantity of solutions. The the group of figures below, we show Total Solutions and Minimum Max Link Utilization across a set of time intervals. We show only the results for the "conservative" below, as it represents

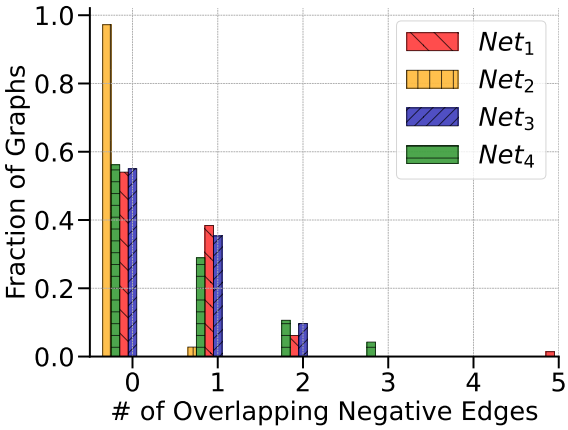
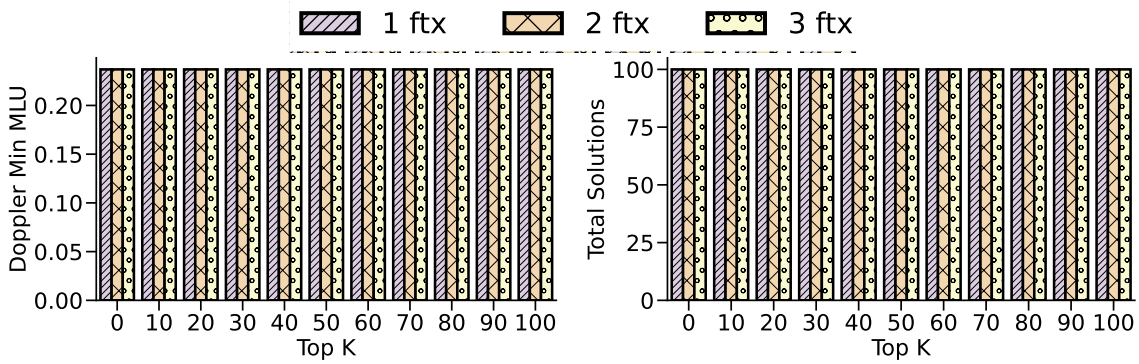


Figure 43. Histogram of the number of overlaps amongst the top 5 most negatively curved edges before and after applying Doppler. We see that, for more than $59-90\%$ of the instances, there is no overlap, highlighting the efficacy of Doppler.

the more sensible choice between the two from an operating perspective. We find that the "max" strategy gives an excessively large set of paths.

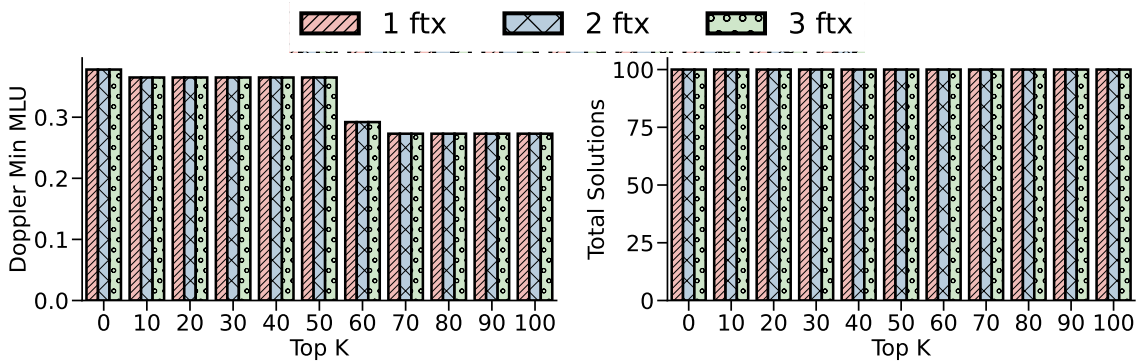
Figure 44a shows the minimum max link utilization in the solution set (Figure 44a) and the total number of solutions found 44b for $Network_1$. Doppler was able to meet the expectation for all sets of transponder allotments and distributions.



(a) Minimum Max Link Utilization (b) Total Solutions Found

Figure 44. In $Network_1$ with a 30 second optimization time limit, and all allocations of fallow transponders to network nodes, Doppler (a) maintains a minimum max link-utilization below 25% (b) finds 100 distinct OTP solutions.

Figure 45a shows the minimum max link utilization in the solution set (Figure 45a) and the total number of solutions found 45b for $Network_2$. Doppler was able to meet the expectation for all sets of transponder allotments and distributions. In Figure 45a you will notice that max link utilization starts to taper down as the set of Top K nodes gradually increases. It is no surprise that adding more resources to the network comes with an additional benefit in capacity savings.



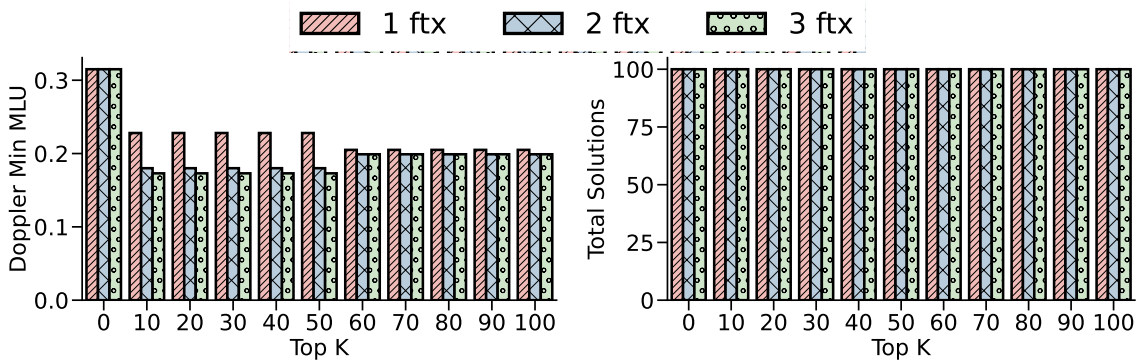
(a) Minimum Max Link Utilization (b) Total Solutions Found

Figure 45. In $Network_2$ with a 30 second optimization time limit, and all allocations of fallow transponders to network nodes, Doppler (a) maintains a minimum max link-utilization below 40% (b) finds 100 distinct OTP solutions.

Figures 46 and 47 show similar results for $Network_3$ and $Network_4$. We notice that $Network_4$ had a slight dip in solutions found, particularly when Top 40% of nodes had been allocated with 3 fallow transponders. This is due to the heuristics implemented by Gurobi's optimization engine.

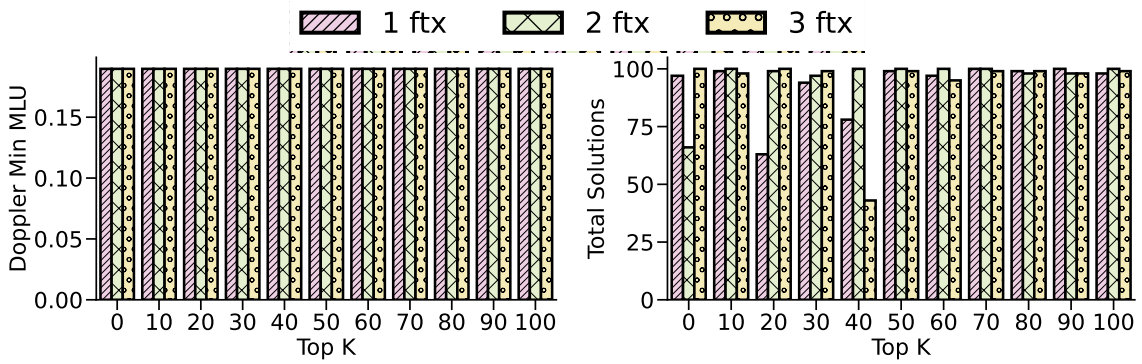
7.6 Summary

Our work unveils a new advanced network reconnaissance called the Ricci attack, proposes a novel optical topology programming (OTP)-based defense mechanism called Doppler, and presents evidence of its effectiveness in countering advanced network reconnaissance. Our work underscores the importance of incorporating



(a) Minimum Max Link Utilization (b) Total Solutions Found

Figure 46. In $Network_3$ with a 30 second optimization time limit, and all allocations of fallow transponders to network nodes, Doppler (a) maintains a minimum max link-utilization below 32% (b) finds 100 distinct OTP solutions.



(a) Minimum Max Link Utilization (b) Total Solutions Found

Figure 47. In $Network_4$ with a 30 second optimization time limit, and all allocations of fallow transponders to network nodes, Doppler (a) maintains a minimum max link-utilization below 20% (b) finds 43 to 100 distinct OTP solutions.

dynamic defenses, such as OTP-based Doppler, to enhance network security in the face of evolving cyber threats.

CHAPTER VIII

ONSET: A FRAMEWORK TO COMBAT TERABIT LINK FLOOD ATTACKS

This chapter VIII contains previously unpublished coauthored material that is scheduled to appear in [217], with coauthors Zaoxing (Alan) Liu, Vyas Sekar and Ramakrishnan Durairajan. The coauthors assisted in discussions about the motivation for the defense presented and in discussions to define the goals for the defense. The dissertation author implemented the defense's optimization method and designed and ran all of the experiments. The coauthors assisted in editing.

8.1 Introduction

Distributed denial-of-service attacks (DDoS) that overwhelm a network's bandwidth are on the rise [202, 295]. The immense attack volumes, attack diversity, sophisticated attack strategies, and low cost to launch attacks make them long-term cybersecurity issues. In 2021, 9.7 million DDoS attacks occurred. This number marked a 14% increase over 2019 [220].

Of particular concern within this broader class of threats are *link-flood attacks* (LFAs) which are also known as network-layer DDoS attacks. While this attack variant has been a scholarly curiosity in years past, it is now a legitimate threat to networks; according to a CloudFlare report, the number of LFAs recorded in their network increased by 109% in the second quarter of 2022 year-on-year. They also recorded an 8% increase in the number of LFAs with 100 Gbps of attack traffic during the previous quarter [312]. LFAs are more effective than conventional volumetric attacks as they are targeting on shared links instead of victim hosts. In this context, we observe a few key trends in the LFA landscape. First, the attacks are adapting to defenses by changing their traffic characteristics frequently. Static mechanisms to defend them become ineffective and treat attack and benign flows equally, affecting

the performance of benign flows. Second, the attackers are generating terabits per second (Tbps) of malicious traffic [237, 274, 295, 310].

In light of this increasing LFA sophistication, existing defense capabilities (e.g., packet scrubbing [3, 18, 43, 63, 164], in-network filtering [22, 90, 139, 305], routing around congestion [257], overlays for tracking [259], and more recent software-defined defenses [86,301,317]) can be improved. For example, since simple network layer filters are ineffective (the indistinguishable nature of the benign and malicious traffic), we need to reroute traffic to sophisticated packet scrubbers for deeper inspection. This inevitably impacts benign traffic and/or worsens network congestion. Similarly, even programmable defenses (e.g., [86, 317]) are ineffective. As we show empirically in our results, LFAs can induce a substantial penalty for legitimate traffic as programmable defenses simply shift the attack-induced congestion elsewhere in the network.

In this work, we observe a new opportunity to bolster LFA defenses in wide area networks (WANs) by leveraging recent advances in optical networking called *topology programming*. Topology programming scales/augments existing LFA defenses by dynamically adding new optical wavelengths to scale the network capacity and alleviate network congestion [59]. Similarly, using reconfigurable add-drop multiplexers (ROADMs) [175], topology programming allows steering of wavelengths at finer granularities and enables fast traffic rerouting [234] in the face of congestion. Optical topology programming has been adopted for classical networking tasks (e.g., traffic engineering [109, 234]) and is increasingly being commoditized [73, 211, 212], but its benefits have not been explored in depth for LFA defenses.

Leveraging optical topology programming leads to two novel opportunities in combating terabit LFAs. (1) Scaling capacity on demand to avoid congestion. By dynamically scaling the capacity of optical paths on-demand, we can potentially

reduce network congestion in targeted links. (2) New capabilities for advanced/future LFAs. With novel optical features such as rapid wavelength reconfiguration, we can improve defenses for LFAs by providing new *links and paths* to route around congested links.

Realizing these benefits in practice, however, requires addressing two key challenges. The first challenge is to dynamically identify the optimal topology, out of $\mathcal{O}(2^{n^2})$ possibilities for a network with n nodes, achievable using topology programming[1]. We refer to this as topology enumeration. Here, optimal is with respect to the maximum reduction in attack-induced congestion. This is key because a sub-optimal topology configuration can shift attack-induced congestion to a different link. Second, there is a challenge for managing network performance with dynamic topology changes; i.e., given a set of links and paths we can activate, we need to choose a routing configuration that is optimal with respect to network demand from both legitimate and attack traffic. In other words, we must jointly optimize routing and topology to effectively address attack-induced congestion. Joint optimization of topology and routing is an NP-hard problem [160].

To address these challenges, we propose ONSET (**O**ptics-enabled **N**etwork defen**S**e for **E**xtreme **T**erabit LFAs). ONSET is a framework for augmenting existing link-flood defenses with topology programming and consists of two components. First is the topology pruning algorithm that addresses the enumeration challenge. This algorithm computes a subset of the potential network topology link sets considered by introducing k new links. Subsequently, the algorithm computes a set of shortest paths based on the augmented topology with k new links. Second is the optimization component that tackles the NP-hard problem by formulating a mixed-integer linear

[1]Because a complete graph of n nodes has $\mathcal{O}(n^2)$ links, the number of topologies possible is the power set of these links $|\mathcal{P}(n^2)| = 2^{n^2}$

program to optimally map and forward traffic atop the augmented links, minimizing attack-induced congestion. This formulation is agnostic to any packet-processing logic or mitigation methods and can be augmented to any existing defenses and network controllers.

Given the limitations of existing network simulation and emulation tools to study topology programming, we use a custom discrete-event network simulator to analyze the benefits of ONSET.[2] Our simulator models how different topologies forward the same traffic, allowing us see how this change affects link utilization across the network. We use it to test ONSET using different terabit LFA scenarios for a diverse set of networks.

Using the simulator, we explore *what-if* questions regarding the value added by our framework for topology programming with analysis on the merit of defenses that can employ it. We approach these questions by processing traffic on simulated networks for which the topology can change subject to a set of real-world limitations. To this end, we simulated a wide variety of LFAs against several networks, where each attack targets a specific link or set of links. Full details about our attack generation and assumptions are discussed in § 8.5. We observe that defenses *with topology programming* offer traffic performance that is always at least as good as a defense without it. Concretely, in 93% of the hundreds of attacks simulated, topology programming helped mitigate all congestion loss from the attack. In every case, ONSET yields its solution in under 1 minute.

In summary, we make the following contributions:

- The first optical topology programming-based defense framework for terabit LFAs. In our prior work [213], we introduced the idea of a topology programming-based

[2]Source code of the simulator is at github.com/mattall/topology-programming and datasets will be released to the community upon publication.

defense for DDoS and presented simple numerical models to demonstrate its benefit for simple topologies and attacks. In this work we go much further in our analysis, presenting a complete framework for applying an optical topology defense for LFAs for any topology, considering various instances of attacks.

- A topology pruning algorithm to tame the complexity associated with modeling the exponential number of possible network topologies and paths on each of them.
- A formal mixed-integer linear programming model for the optimal mapping of traffic to new optical links.
- A simulator for what-if analysis, demonstrating our approach under diverse attack scenarios and for a diverse set of networks.
- A decision-support capability for incremental deployment of ONSET and measure the cost-benefit trade-off for enabling the optical topology programming-based defense at different locations in a network.

8.2 Background and Related Work

8.2.1 Threat Model. The attacker has access to a *botnet* or large number of compromised hosts and services. The attacker uses the botnet to send terabytes of traffic to a network. We are primarily concerned with sustained attacks where the duration is at least five minutes or longer. The attacker can flood either or both directions of a bidirectional network link that is critical for the service of targeted hosts. The attacker measures their success based on whether they can induce network congestion and thereby degrade the performance of legitimate traffic intended for the network. We assume that all attack traffic comes from legitimate non-spoofed senders, is protocol-conforming, and is indistinguishable from benign network traffic. We assume that an attacker has obtained an accurate map of a network's link-layer topology, with which it determines which bots to activate and what destinations they

will send traffic to. How the attacker acquires the network topology information is beyond the scope of this paper. In this work we do not consider a "smart attacker" who updates their network reconnaissance—the attacker has a one-time snapshot of the network topology and deploys bot traffic strategically based on that snapshot. This assumption keeps the work grounded in LFA defense specifically, rather than entering the network reconnaissance and obfuscation space which is out of scope for this paper. Note that this assumption in our threat model is consistent with prior efforts (e.g. Ripple [301]). Finally, we assume there exists a mechanism to detect an LFA. This is reasonable because the bandwidth utilization of links is easy to monitor. We do not assume that detecting an attack implies easily flagging/dropping attack traffic with high accuracy.

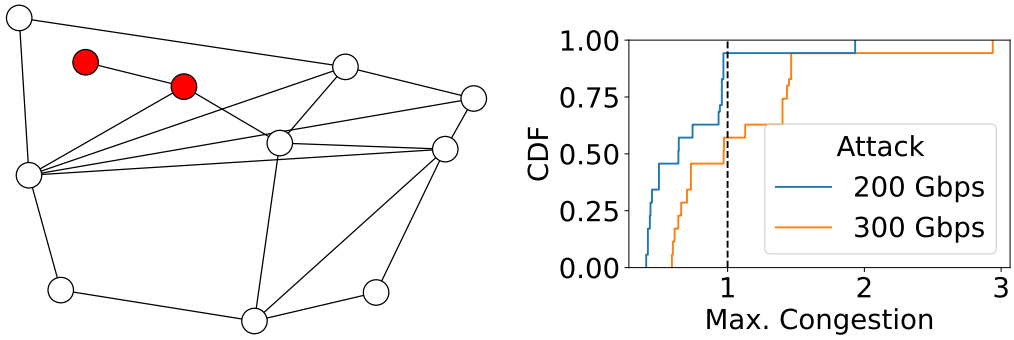
8.2.2 Prior Efforts and Their Limitations. State-of-the-art defenses against LFAs use software-defined networking (SDN), thus altering only the network's *forwarding behavior* to mitigate attacks [148, 301]. The SDN-based approaches for LFA defense can broadly fit into three categories: programmable control plane, programmable data plane, or hybrid. Control plane programmable defenses (e.g., Spiffy [148]) use a central network controller to monitor traffic. The controller issues commands to network forwarding devices and updates their forwarding paths when an attack is detected. Data plane programmable defenses (e.g., Ripple [301]) cut out the centralized aspect from prior work, and implement monitoring and mitigation within the network switches themselves. Hybrid approaches (e.g., Jaqen [181]) use a mix of both data plane and control plane programmability; they can adapt the forwarding paths for suspicious traffic, thereby sending it to specialized switches that will run defense programs and drop malicious traffic or allow benign traffic to traverse the network further. We note that Jaqen has not been applied to LFAs, but include it

in our taxonomy of prior work to show that the hybrid approach has been applied to DDoS mitigation in a general sense. Another recent defense, ACC-Turbo [5], employs a fast clustering technique to flag and deprioritize suspicious traffic at line rate.

State-of-the-art LFA defenses treat topology as a static entity and thus overlook an opportunity to remove the network bottlenecks created by LFAs. There are no defenses that optimize or change the underlying topology, and therefore prior efforts are forced to filter and drop traffic during an attack on a highly congested link shared by multiple hosts. As we will see in the empirical example below (in § 8.2.2), the lack of topology flexibility implies inevitable congestion loss for high-volume LFAs. We discuss related work in more detail in § 8.7.

Empirical Example: Figure 48 illustrates the limitations inherent in adapting forwarding behavior to defend against LFAs for a real-world network, Sprint, from the Internet Topology Zoo [155]. In this quantitative demonstration, we created a set of Coremelt attack traffic matrices, each targeting an individual link in the network; see § 8.5 for details. We also varied the attack strength from 200 Gbps to 300 Gbps. We gave the network SDN routing defense capabilities, whereby the network optimally routes traffic and minimizes max link congestion. The routing strategy used in this example is more optimal than the current generation of traffic engineering systems (e.g., B4 [137], Orion [89]) because those systems rely on heuristics to scale and compute allocations quickly. In our example, this routing system has unlimited time to find the optimal set of paths for each flow. Figure 48b shows CDFs of max link congestion for both of these attacks. We observe congestion loss with 200 Gbps of attack traffic in only one instance, where a link to a leaf node was flooded (this link is identified with the highlighted nodes in Figure 48a). When we increase the attack to 300 Gbps we see that SDN-based rerouting has significantly greater difficulty

mitigating loss. In fact, about 50% of all links targeted with this attack incurred a loss.



(a) Sprint Network.

(b) CDF of Max Congestion. Dashed line shows congestion-loss threshold.

Figure 48. Every link in the network was targeted individually with a 200 and 300 Gbps Coremelt attack. At 200 Gbps, it was impossible to guard one link from congestion loss. At 300 Gbps, ∼50% of links experienced loss.

This result illustrates that defenses that only adapt forwarding behavior have a finite breaking point at which congestion loss is unavoidable, even when routing choices are optimal. Observing these factors, we raise our motivating question: how can we enable capacity on demand and topology flexibility without the attendant problem of collateral loss?

Summary: *We observe that state-of-the-art LFA defenses suffer from a key limitation that network topology is treated as a static entity. It is often impossible to reroute malicious traffic and insulate benign traffic from loss in the face of attacks that can overwhelm a link's bandwidth multiple times over.*

8.3 Approach: Optical Topology Programming for LFA Defenses

We observe a new opportunity to bolster LFA defenses by leveraging a recent advancement in optical networking called *topology programming* to achieve topology adaptation. Using topology programming, an operator can affect a network's

topological structure via optical wavelength reconfiguration in addition to the traffic forwarding behavior.

Combining topology programming, provided by optics, and adaptive forwarding behavior, provided by SDN and programmable switches, leads to two new opportunities in combating terabit LFAs. First, it allows a network's underlying topology to scale capacity on demand to avoid congestion. Second, topology programming enables a defender to amplify the benefits of traditional programmable defenses. Improved general network performance is possible because changes made at the optical layer give us increased possibilities to forward traffic on new paths in the face of attacks. Note that we do not claim that topology programming offers a panacea for all DDoS-related concerns—we claim that it provides a novel means to bolster existing programmable defenses for LFAs as described above, and investigate that means more deeply than any prior work to date.

While topology programming is compelling, it has not received a great deal of attention for DDoS. In § 8.3.1, we outline the challenges of using topology programming for LFA defenses.

8.3.1 Challenges. To use topology programming for LFA defense, we need to solve two unique challenges (**C**s).

C1: Topology Enumeration. When we open the door to topology programming, we are immediately confronted with an exponential number of network link-layer configurations to choose from. This size is further compounded with every path on each of those topologies and the number of ways to split traffic among a set of paths. The state space for network topologies wherein the set of active links can change is $\mathcal{O}(2^{n^2})$ where n is the number of nodes. Considering these topologies and

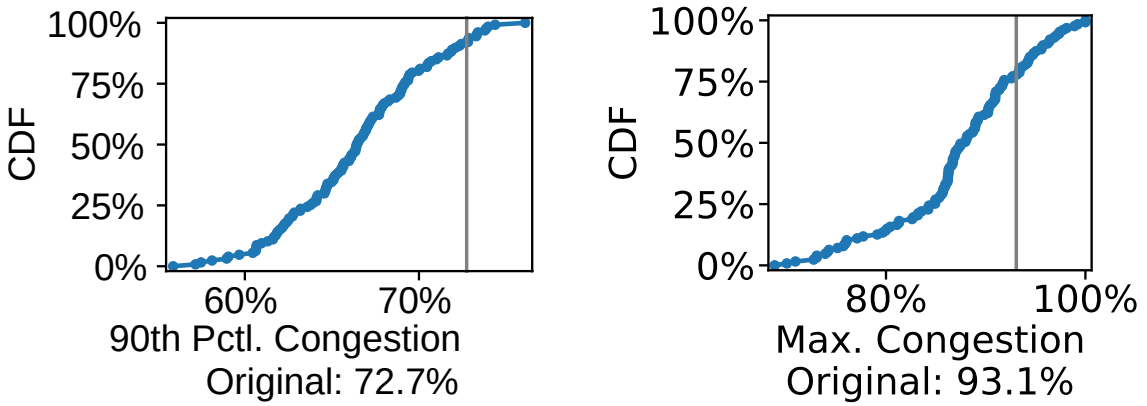
their relative benefit for attack-specific demand introduces the challenge of topology enumeration.

To illustrate, consider a network with 30 nodes has 435 possible links $(30 * 29/2)$. The set of all combinations of these links is 2^{435}. Therefore the number of different topology instances that we might create is $\sim$50 orders of magnitude larger than the number of atoms in the known universe. The runtime complexity for enumerating all shortest paths on every instance of the network topology, therefore, is $\mathcal{O}(2^{n^2}n^3)$[3]. Clearly, enumerating each potential network state (the paired sets of active links and available paths on those links) and storing these states is a daunting task, but it will enable us to quickly instantiate the most opportune configuration of links given the shifting behavior of an attacker.

C2: Managing Network Performance. Any addition or removal of a link from the network can have a unique effect on traffic performance across the entire network as seen in Figure 49; these plots show that arbitrarily adding a new link on the ANS network (from the network topology zoo [155]), while employing ECMP routing, can occasionally increase link utilization and induce network congestion. Figure 49a shows that the original 90^{th} percentile congestion was 72.7%, indicated by the vertical bar, and roughly 15% of all links that were added increased the 90^{th} percentile congestion. Similarly, Figure 49b shows an increase in maximum link congestion for roughly 25% of all possible single-link additions. This observation holds on any network that uses a link-state routing strategy such as ECMP because when we add a link we change the set of paths favored for routing traffic between some pairs of hosts. If a new link creates a new shortest path for every pair of nodes, then that new link will quickly become congested. This is an instance of Braess's paradox [38]. Therefore,

[3]The Floyd-Warshall all-pairs shortest path algorithm is $\mathcal{O}(n^3)$

we must have an optimization method to ensure that the changes that we introduce to a network by adding or removing links has a net-positive benefit for all traffic across the network.



(a) CDF of 90th percentile congestion after adding different links.

(b) CDF of maximum congestion after adding different links.

Figure 49. Effect on network congestion in ANS from adding different links with ECMP routing.

In SDN-based networks wherein routing paths can be centrally defined and controlled, we must also choose to tread carefully between the trade-off of congestion avoidance and topology adaptation. We seek to optimally choose a topology and the set of routes based on it, but choosing a set of new links to activate in a network while considering the different routing choices available is NP-Hard [160].

As adding and removing links affects traffic paths, it may be that the frequency with which those paths are changed can lead to performance impacts. Therefore, we must verify that the frequency with which optical topology changes are made is not a cause for a performance error in § 8.5.4.

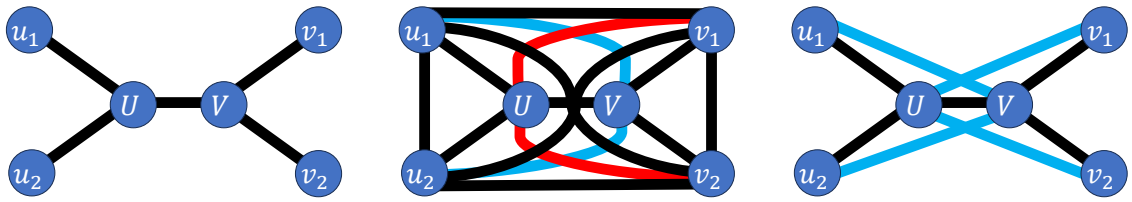


Figure 50. (a) Nodes U and V represent a bottleneck link between their neighbors, u_1, u_2, v_1, and v_2. (b) Set off all possible candidate links around U and V. (c) Illustration of the topology programming idea. ONSET considers a pruned down set of candidate links, containing. For each (U, V) link in the top 10% of ranked links, it chooses $(U, v*)$ and $(V, u*)$ where $v*$ and $u*$ are mutually exclusive neighbors of U and V respectively.

8.4 ONSET: An LFA Defense Framework Using Optical Topology Programming

We present ONSET (**O**ptics-enabled **N**etwork defen**S**e for **E**xtreme **T**erabit LFAs)—a defense framework for augmenting existing programmable defenses with optical topology programming to defend terabit LFAs. ONSET consists of a model and algorithm that address the major challenges for an LFA defense. Figure 51 outlines the two major algorithmic and modeling components of our framework. The first component, *Topology Pruning*, is an algorithmic step that (1) catalogues the different topologies that we may instance by activating a set of links and paths and (2) finds the set of shortest paths available under these topologies. The second component, *Joint Topology and Routing Optimization*, is an optimization model that runs when an ongoing LFA is detected (the instrument for detection is beyond the scope of this work). This component accommodates the demand present in the network using the topologies found during Topology Pruning. The result of this optimization is a set of links to add to the network that will alleviate congestion loss from the ongoing LFA.

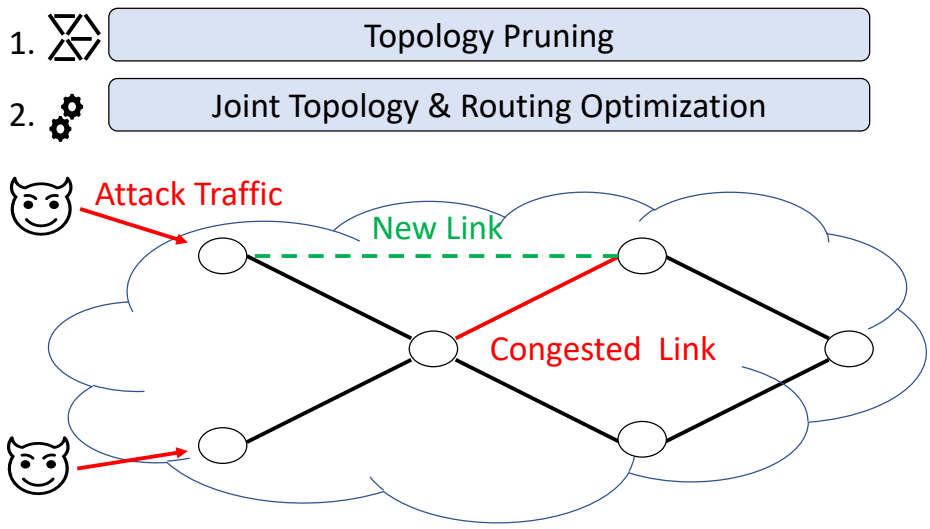


Figure 51. Overview of the ONSET defense framework.

Hardware Requirements: The hardware requirements for ONSET under this framework are (1) optical fiber and transponders for establishing new links, and (2) ROADMs at nodes where new links originate, terminate, and bypass other nodes. ONSET can be incrementally deployed with these resources deployed at a subset of the network.

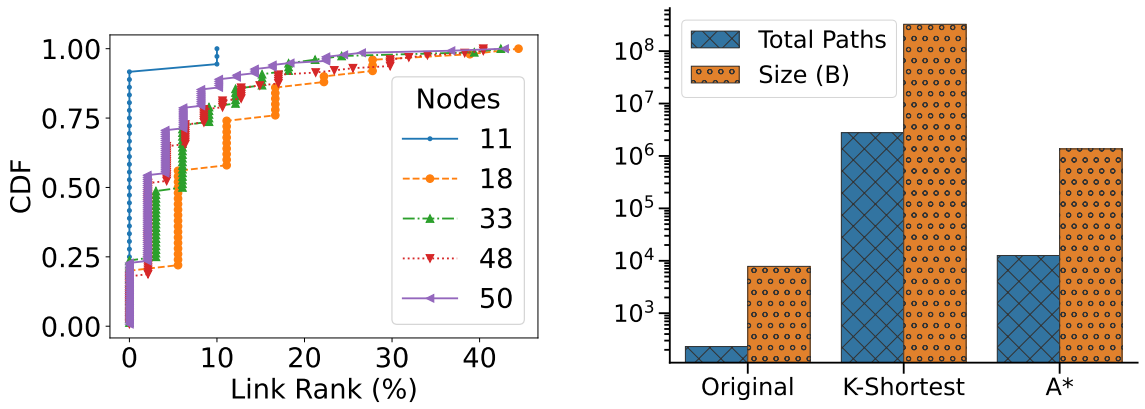
8.4.1 Topology Pruning. We address **C1** as follows. Given topology, T, and budget, B, we first find a set of links, L, and the network paths on these links, P_L. The set is large but can be pruned down considerably. As a first-order pruning step, we eliminate the possibility for links that are longer than the maximum transmission distance supported by the transponders (e.g., 5,000 km for 100 Gbps circuits [250]). This pruning removes infeasible links in large networks but does not help reduce the number of topologies in networks for which all of the nodes are closer than the max transmission distance.

Link Rank: A striking observation allows us to trim the candidate set further and consider a smaller set of topologies. We observe that for a diverse set of attacks on

a network, each targeting a different subset of links, there is a consistent set of links that are disproportionately affected. We introduce a metric, *link rank*, which captures this phenomena. Consider a set of possible LFAs on a network, each of which targets a different set of links. The link rank is the percentage of attacks in which a given link is congested. For example, when 100 attacks are considered on a network, and a given link experiences congestion loss in 12 of those cases, the link rank for that link is 12%.

Figure 52a shows the CDF of link rank for networks of different sizes. From this result, we observe that a majority of links (for all networks considered) have a small link rank (i.e., less than 10%). Only a minority of links experience congestion during a relatively high proportion of the total attacks. Concretely, in the network with 50 nodes, only two links are congested in 43% and 37% of the attacks, respectively. At the tail end of the distribution, 74 of the links were *only* congested for one attack, or not congested at all. This observation suggests that only a handful of vulnerable links are severely affected by LFAs. We leverage this insight to prune candidate topologies: more redundancy is granted to links with high rank by prioritizing the k highest ranked links when enumerating potential candidate links and topologies.

In practice, we choose the top 10% of links and consider the potential to add new, *candidate links* links dynamically that bypass these bottlenecks. Figure 50 illustrates how the pruning process drastically decreases the search space for the reconfigurable topologies by honing in on bottleneck links and considering candidate links as those the provide potential for new paths that do not traverse the bottleneck. The subgraph (50a) has a bottleneck (U, V). The complete graph induced by connecting all nodes in the neighborhood of (U, V) (50b) has 15 edges, 10 of which are not in the



(a) (b)

Figure 52. (a) Link Rank for attacks on networks with different sizes, noted by the number of nodes. (b) Space complexity for comparison for path finding methods. "Original" represents the set of paths that would be stored in a traditional SDN system. "K-shortest" is the set of "K-Shortest" paths among the ranked links. "A*" is the pruned down selection of those paths.

original topology. ONSET considers just 4 of the 10 links (50c) when enumerating topology and routing solutions.

Path Selection: While a general K-shortest paths search gives a small number of paths for a fixed topology, we must include paths with links that may or may not be members of the physical topology at any given time. Therefore, we must broaden the search. We might enumerate a set of paths for each pair of nodes that is exponentially greater in magnitude than the original set of K shortest paths. Therefore, we add a heuristic function to our graph searching process to mitigate the explosion in space complexity for our paths. Our path finding method is implementation of the A* algorithm [123]. Figure 52b shows the number of paths found with a A* versus a general all-pairs k-shortest paths. To ensure that the set of paths includes enough diversity with respect to candidate links, we populate that paths until the length of the path is greater than the length of the original path. Furthermore, to account for the potential of a link from the original graph to be removed, the original set of

"shortest paths" for single-hop paths is expanded to include paths that are at least 3-hops long.

8.4.2 Joint Topology & Routing Optimization. We address challenge **C2** with a mixed-integer linear program (MILP), presented in Chapter IV, § 4.3. The optimization in this application is tailored to find the set of links that will reduce network congestion by the greatest amount. The objective is to minimize the max link utilization ($\mathcal{U}$).

$$\texttt{minimize}\ \mathcal{U} \tag{8.1}$$

The MILP is hosted by the ONSET controller, as shown in Figure 53. The model uses the enumerated topology and path data set to yield a set of optimal links to activate in a network in light of an ongoing LFA. These optimally chosen links come from the set of *candidate links* which are input to the system. Candidate links are links that exist in the network at the time the optimization solver is invoked but that can be quickly activated to augment the topology and allow data to travel directly between two nodes.

The controller periodically receives a traffic matrix and link utilization data with flow demands aggregated over a series of epochs. We assume an oracle for detecting the presence of an attack Note that this oracle does not identify attack traffic. It merely answers the yes or no question, "Is the network under attack?"; when an attack is detected the controller runs the optimization model and yields a set of links to add to the network. These links persist until network congestion falls below the levels that were seen before the attack.

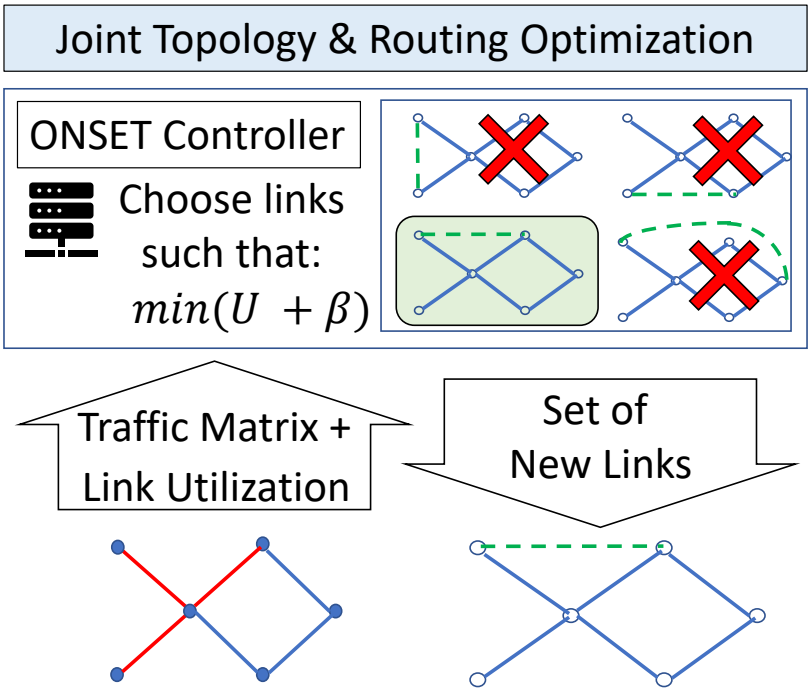


Figure 53. Topology optimization process for ONSET.

Having described the topology pruning and joint optimization components, we next focus on meaningfully assessing the efficacy of ONSET in the face of diverse terabit LFAs.

8.5 Evaluation

In this section, we evaluate the ONSET framework for defending LFAs. The key metric of success is link congestion (the aggregate traffic demand for the most-utilized link in the network). Traffic loss and reduced throughput occur when link congestion is greater than one. We compare ONSET against a baseline ECMP-routed network and an SDN-enabled network that optimizes traffic allocations to minimize max utilization across all network links. We denote the SDN defense as Ripple* throughout this section. We show that ONSET is an additive capability for network defense that can be applied to ECMP-routed networks or SDN-controlled networks employing the Ripple defense. We, therefore, compare network performance for both

strategies with and without ONSET. To this end, we demonstrate the following key results.

(1) ONSET improves the capabilities of the Ripple defense for multi-target high volume attacks such as Coremelt (§ 8.5.2). We show ONSET can complement Ripple to mitigate terabit LFAs.

(2) Regional attacks, such as Crossfire, that target all links adjacent to a node, can be mitigated with the ONSET framework (§ 8.5.3). We show ONSET improves crossfire attack mitigation in over 90% of simulated attacks on 5 networks.

(3) ONSET can respond and mitigate rolling attacks, where a series of attacks with different volumes, numbers of links target targeted, and attack styles, vary in succession (§ 8.5.4). ONSET was effective at mitigating congestion loss in 64 out of 70 rolling attacks.

We conclude this section by presenting a cost-benefit analysis for ONSET vs. statically over-provisioning network links (§ 8.5.5), and taking a deep-dive into cost optimization with variable fallow transponder allocations where ONSET is enabled only for a subset of network links (§ 8.5.5).

8.5.1 Simulator Parameterization. We use the OTP simulator described in Chapter IV, § 4.4 to evaluate ONSET. Specifically, we use it to evaluate how different topological link configurations perform when they are forwarding the same attack traffic. The goal is to see how *topology* changes *link utilization* across the network in the face of terabit LFAs.

This simulator enables us to ask valuable *what-if* questions about topology programming and its applicability for defending LFAs without access to a wide-area backbone optical network. Pertinent questions include how can the ONSET framework augment existing defenses to combat different types of LFAs, against

attacks on different sets of links? What quantity of fallow transponders is required at the network's nodes to support the flexibility required to mitigate those threats using ONSET? How does the distribution of fallow transponders among nodes affect the ability of ONSET to mitigate traffic loss for a set of attacks?

Figure 54 shows a block diagram of the simulator's control loop as used to evaluate ONSET. This loop models the way we envision ONSET to be used in a live deployment. The network operator defines optical constraints and traffic engineering system. Optical constraints include the number of simulated links available for adding, the max. link utilization thresholds which will trigger a topology-update event, and a target link utilization threshold which is used by the optimization method to find the best set of links to add. The ONSET controller, which controls SDN and optical components of the network, receives these input parameters from the operator and uses them along with the link utilization data to decide on a runtime defense strategy, whereby it adds links to the network and monitors their utilization. The traffic matrix processed by ONSET is a mixed bag of attack and benign traffic, the two of which are indistinguishable.

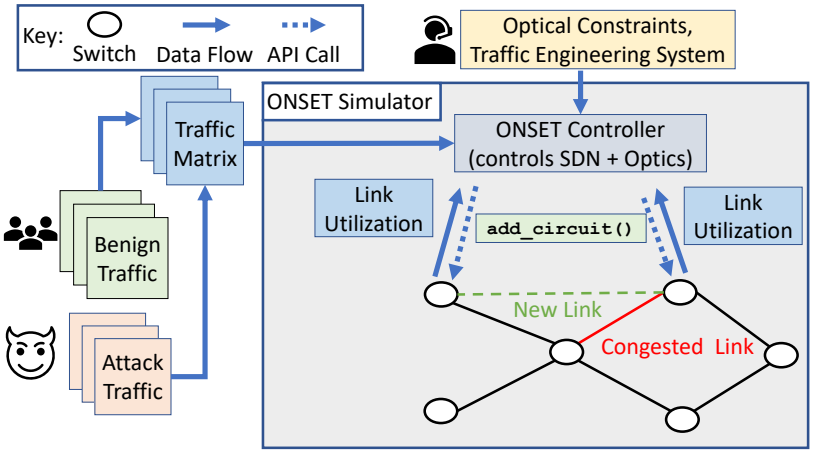


Figure 54. Overview of ONSET simulator.

In this application, the OTP simulator uses Yates [165] to implement traffic engineering and routing requirements with two methods: ECMP and multi-commodity flow (MCF); ECMP routing is commonly implemented in enterprise networks, as it is supported out-of-the-box by commodity switches and routers while MCF is seeing adoption in emerging SDN deployments [200] and is used in our analysis to emulate Ripple [301].

Attack Traffic Matrices We generate attack matrices using a custom tool written in Python to emulate three attacks: Coremelt [260], Crossfire [149], and Spiffy [148], which we refer to as $TM_{Coremelt}$, $TM_{Crossfire}$, and TM_{Spiffy} respectively. The TM tool takes in the topology of the network as an input, then finds the shortest paths between pairs of nodes, and creates demand between hosts that share a common link. $TM_{Coremelt}$ is made by choosing a random link (or links) in the network, and then choosing pairs of hosts for which their shortest paths use the chosen link(s). The Crossfire attack targets a region of the network. In our evaluation, we restrict a region to a single node. $TM_{Crossfire}$ floods all of the adjacent links to the target node. TM_{Spiffy} is constructed by finding that most-shared link(s) or node(s) and flooding them.

We emphasize that the attacker does not have control over the network routing. Our attacker assumes traffic is routed via the shortest path. The assumption does not hold for Ripple's defense due to its ability to optimize path and flow allocations, and we will see that Ripple, therefore, performs well enough for mild attacks. However, as an attacker increases their power with more traffic, Ripple's defense has a breaking point where ONSET improves the capability to defend.

An attack traffic matrix encapsulates two important attributes of the botnets, namely the size of the botnet (by proxy of its aggregate bandwidth), and the locations of the bots in the network (explicitly by the nodes from which their traffic originates).

Benign Traffic Matrices Unless otherwise stated, we used TMGen [126] to create random gravity model traffic matrices for benign traffic in our experiments.

Routing Our evaluations address two routing strategies, ECMP and Ripple*. ECMP is commonly implemented in service-provider networks. We pair it with ONSET to observe how legacy networks might benefit from the ONSET framework. On the other hand, modern enterprise and cloud backbone networks are increasingly looking to SDN to address network resource (e.g., bandwidth) management. Recent proposals for LFA defenses use SDN as a primary tool to insulate legitimate traffic from the effects of malicious traffic [148, 301]. SDN-based networks can use a central network controller to update forwarding paths and flow rates applied to these paths. Ripple attempts to drop malicious traffic before forwarding it, but when attack traffic cannot be detected, the Ripple defense reroutes traffic to avoid congestion on links. We emulate this capability by using a multi-commodity flow optimization to route traffic during attacks and denote this as Ripple*. The implementation of the ECMP+ONSET defense cannot tune traffic forwarding rates among paths by definition, and to model ECMP routing with binary links would introduce quadratic constraints to the model. However, to compute an ECMP routing assignment for a single topology is quick and efficient. Therefore, we elect to generate 100 sub-optimal solutions from our model and simulate the ECMP link utilization for all network links in all of the model's solution topologies in parallel. The network's topology is then configured based on the solution with the best ECMP congestion result.

Networks Our evaluations consider five real-world network topologies, shown in Table 7. These networks are representative of enterprise optical backbone networks and have been used to investigate other LFA defenses in prior work [301]. These networks range in size from 18 to 68 links. For each of these networks, we apply a similar series of tests where we vary the strength of an attack and the number of links targeted. In our experiments, every link in the network has a bandwidth of 100 Gbps unless otherwise stated. We gave every node in the network 10×100 Gbps fallow transponders; we revisit this allocation in § 8.5.6. Therefore, each node is capable of establishing a 100 Gbps link between itself and up to 10 other remote nodes. This bandwidth constraint per transponder is emulative of a 100 Gbps polarization multiplexed quadrature phase shift keying (PM-QPSK) transponder [33]; this type of transponder has been widely deployed in backbone networks for decades, and can reliably transmit 100 Gbps data channels approximately 5,000 km [94]. While higher-bandwidth transponders are also widely deployed, we only consider 100 Gbps QPSK transponders in this study. This is a conservative assumption for a lowest-common-denominator evaluation of the ONSET defense—we expect higher power/bandwidth transponders will improve the network performance further.

Network	Nodes	Links
Sprint	11	18
ANS	18	25
CRL	33	38
Bell Canada	48	65
SurfNet	50	68

Table 7. Networks used in our study.

Optimization Time: Our model implementation has a 1 minute cut-off window. Said differently, if the model does not find an optimal solution by then, it returns the best feasible solution. In cases where the solver finds a solution early, it may

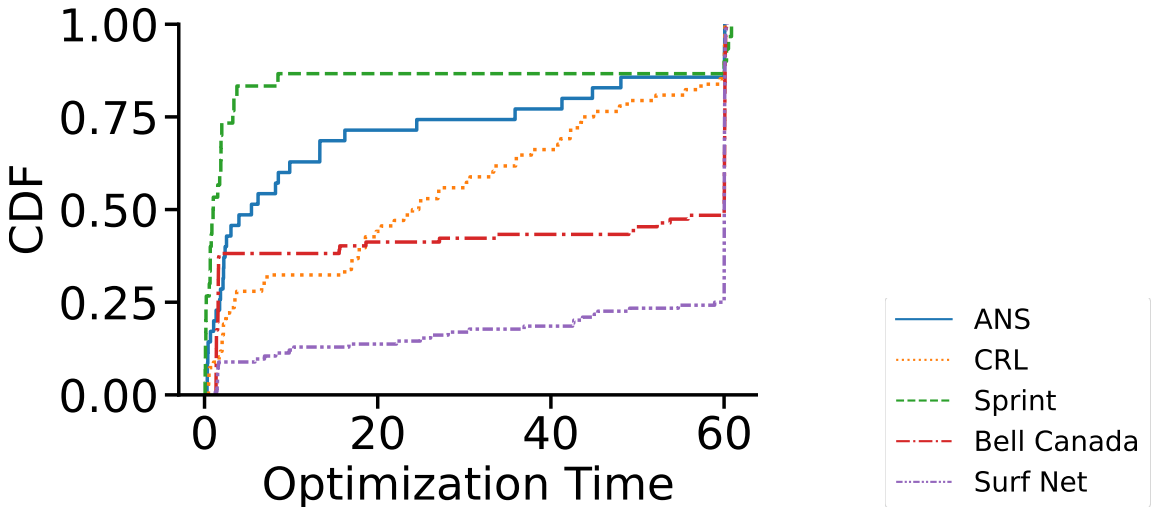


Figure 55. CDF of optimization time for ONSET experiments by network.

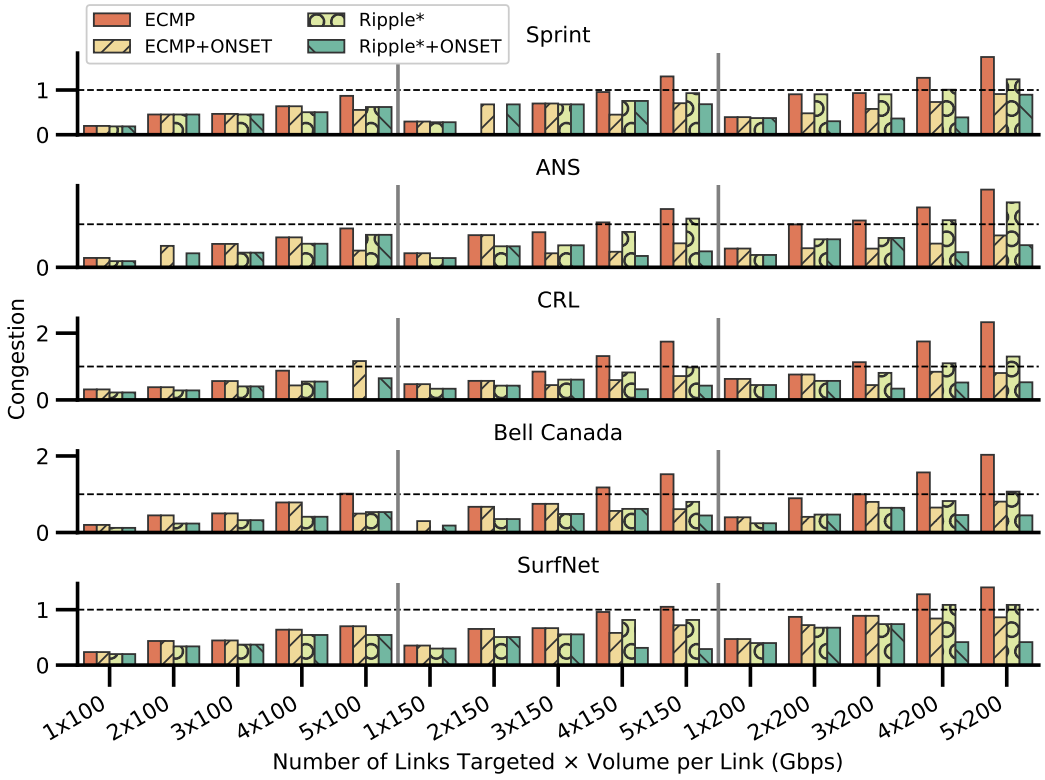


Figure 56. Network congestion induced by coremelt attacks varying in strength and total targets on networks with different routing strategies and optical topology programming capabilities. The x-axis is encoded (links targeted × attack strength per link).

populate a set of alternative feasible solutions with the remaining time. We find that ONSET is able to dynamically derive topology configuration and routing settings for many attack scenarios presented to it. Figure 55 shows the time distribution for all of the ONSET models evaluated in this section. We observe that for the graphs Sprint, CRL, and ANS, all have a strong majority of evaluations where an optimal solution is found before the cut-off period at one minute. Bell Canada has 15 more nodes than CRL and nearly twice as many edges. ONSET found an optimal solution for attack on Bell Canada within the prescribed time in 38% of experiments. Surfnet, with only marginally more nodes and edges than Bell Canada, found optimal solutions in the allotted time in 25% experiments.

Presentation of Results: Due to the different nature of the coremelt, Crossfire, and Rolling attacks, we plot the results for each test differently. Coremelt attacks target one or more network links, and the targets can be arbitrary and random. Therefore, we plot these results as grouped bar charts, where a group corresponds to a specific attack, and each bar represents the network performance of the different mitigation strategies (ECMP, ECMP+ONSET, Ripple*, and Ripple*+ONSET). The attack in each group of bars targets the same exact set of links with the same volume of traffic.

In many of the results we show network performance as it relates to maximum link congestion because LFAs, by purpose, attempt to maximally congest a link or set of network links. Therefore, we use this metric to determine the success or failure of an attack for each experiment. Related studies, e.g., Ripple [301], also narrow the scope of their evaluation specifically to links that are targeted by an attack. However, we include the complimentary results for total network throughput in the evaluation of the coremelt attacks in § 8.5.2, as it illustrates the relationship between maximum congestion and the overall performance of the network traffic. After establishing this

relationship we omit throughput from the results as we are primarily interested in keeping maximum link utilization below 100% and keeping throughput at 100%.

The Crossfire attack targets a region of the network. To see how the different mitigation strategies perform, we launch crossfire attacks against each node in the network independently by targeting each link incident to each node targeted with an LFA. To view the performance of multiple attacks for each network, we present the results as CDF, where the X-axis shows max network congestion for each attack in the distribution and the Y-axis shows the CDF function for a given value of congestion.

The rolling attacks can be composed of crossfire and coremelt attacks, and we are interested in seeing how the network performance changes as the attacks change over time. Therefore, we plot network performance as a time series, where the X-value is a point in time, and the Y-value is the relative network performance metric at that time.

8.5.2 Coremelt Attack. To evaluate the performance of our framework against the coremelt attack, we consider a variety of attack strengths and attacks against a varying number of total links. In particular, we generate matrices composed of attack traffic with volumes of 100, 150, and 200 Gbps, each targeting 1 to 5 links simultaneously. These parameters are chosen in an attempt to get a broad-scope view of the impact of ONSET for a range of (multi)-attacks, each of which is capable of flooding a link with 1x to 2x its maximum capacity. We settled on these settings after discovering that they are severe enough to demonstrate a breaking point for Ripple*. Figure 56 shows the effect on network congestion from this suit of attacks for all of the networks in this study. In this figure, the x-axis is encoded as (number of links targeted $\times$ attack strength per link). For example, a 5×200 Gbps traffic matrix has a total volume of 1 Tbps; this volume is spread between 5 attacks targeting different

links with 200 Gbps of traffic each. The matrices that we use in this test are made up completely of attack traffic. The y-axis shows the maximum link congestion (max congestion) in the network. When max congestion is greater than 1 the attacker successfully induces traffic loss.

We separate these results based on routing strategy (ECMP or Ripple*) and whether or not the network employed an ONSET topology programming defense. We see that SDN-based routing with Ripple's defense can offer notable savings up to a point. For example, in the CRL network, when 5 links are targeted with a 100 Gbps attack each, this network experiences congestion loss. However, *Ripple*+ONSET is able to prevent congestion loss in every 100 Gbps attack against 5 or fewer links in every network.*

We also observe that link-state routing with ECMP has greater difficulty mitigating loss from adversarial traffic. A 100 Gbps attack is able to induce congestion when only two links are targeted in ANS. As the number of targets increases to three, all of the networks experienced congestion loss. In every attack shown, ONSET is able to find a topology and routing solution in under 1 minute that completely mitigates all congestion loss.

Summary*: Out of 94 crossfire attacks against 5 networks, only 15 attacks resulted in congestion loss with ONSET. Of the routing-based defense without ONSET (plain ECMP or Ripple*) 68 of the attacks resulted in traffic loss. ONSET reduced loss rates in the limited cases where it faced loss.*

8.5.3 Crossfire Attack. We evaluate the resilience of ECMP and the Ripple* defense against crossfire attacks, where each node in each network is targeted with a 100 Gbps attack on all incident links and a 200 Gbps attack on all incident links. Similar to our evaluation of ONSET's added benefit for Coremelt attacks, these

parameters are chosen because they represent a range of moderate to strong attacks that are capable of inducing traffic loss under ECMP and Ripple* respectively. In this section, we highlight the results for both these attacks on Sprint, ANS, and CRL, Bell Canada and SurfNet. Figures 57–61 show the results for each network. Subfigures, (a) and (b), show the effect on max. congestion when the network uses ECMP routing with and without ONSET for a 100 Gbps attack (a) and a 200 Gbps attack (b). Subfigures (c) and (d) show the effect when the network uses the Ripple* defense.

We find that the link-state routing protocol, ECMP, is highly vulnerable to crossfire attacks. An attack of 100 Gbps is enough to cause congestion for approximately 20% of the 100 Gbps attacks to induce traffic loss. In comparison, ONSET had congestion loss in less than 5% of all events at this volume.

When networks use the Ripple* defense, they can aptly mitigate the lower-rate, 100 Gbps attacks (Figures 57c, 58c, and 59c). However, for larger scale attacks, at the 200 Gbps level (Figures 57d, 58d, and 59d) 68 of the attacks are successful at inducing congestion. Ripple*+ONSET had 37 congestion loss events for the same set of attacks.

The reason that performance in Sprint is not perfect for every attack is due to the size of the network. It is the smallest network in the evaluation and therefore has the fewest possibilities for adding links dynamically with ONSET. Similarly, the aggregate bandwidth from the attacks is concentrated on fewer total links, magnifying their impact. This underscores the notion that bandwidth *is* limited—even if you can establish new links opportunistically. However, ONSET increases the amount of traffic needed by an attacker to induce congestion loss with an LFA.

Summary: *Defending Crossfire attacks with ONSET can greatly improve the defensive posture of a network and is complementary to SDN defenses that adapt the forwarding behavior of network traffic. In total, we simulated ONSET against 222 attacks on five networks. ECMP (without ONSET) led to traffic loss in 84 attacks (67%). With ONSET, ECMP led to traffic loss in 6 attacks (4%). The Ripple* defense without ONSET resulted in traffic loss for 50 of the 124 attacks (40%). With ONSET, the number of attacks that resulted in traffic loss fell to 3 (2%).*

8.5.4 Rolling Attack. Next, we evaluate the ability of ONSET to adapt to an ongoing/rolling attack. We evaluate a series of traffic matrices constructed to model several attacks. We model the attack traffic matrices for Crossfire and Coremelt attacks as described above. We also included a Spiffy attack, where the attacker gradually increases their demand until a cost threshold and targets a link that is expected to be shared by the greatest number of paths.

We simulated seven attacks, sampling traffic metrics (throughput/loss/congestion) at 5-second intervals over a 60-minute period. The time between attacks varies from 5 seconds to 5 minutes. Figure 62 shows the network performance with respect to congestion during these attacks for a Ripple* routed network. Figure 63 shows simulated network congestion over an hour, sampled at 5-minute intervals for rolling attacks in an ECMP-routed network with and without ONSET. The black dashed line at Congestion = 1.0 marks the loss threshold; any congestion beyond that point results in traffic loss. These results show that the ONSET framework can quickly adapt to dynamic attacks. In more than 90 percent of instances, ONSET completely mitigates attack induced congestion loss..

Figure 64 shows the total number of active network links during the rolling attacks. Our optimization is triggered whenever congestion is above the loss threshold. If

congestion remains above that threshold, then we invoke the optimization again to find more links to add to the network. In every event, the optimizer yields a solution that the network can instantiate in under sixty seconds. When congestion reduces back to a level seen before the attack started, the flux links are released. Therefore, in the last two attacks which happen in quick succession, the number of flux links drops to zero as the attack ends, and then quickly jumps up again after the next attack begins.

Summary*: ONSET can be used with SDN and link-state routing to react and adapt to rolling attacks. When traffic demand falls after an attack is over, ONSET is able to detect the change in utilization and deactivate links that it had activated. These fallow transponders can then be used to respond to new attacks that target different sets of links.*

8.5.5 Cost Benefit Analysis. We now assess how the cost of provisioning ONSET (i.e., the capital expense for hardware required to realize optical topology programming) compares with defenses on a static topologies. To this end, we count the number of transponders required to insulate legitimate traffic from attack induced-congestion when an attack occurs leveraging 2x and 3x the bandwidth of one transponder. Table 8 shows that the cost-benefit of ONSET comes from scaling our defense with the number of nodes in the network, rather than links; to defend an arbitrary attack in a static topology, you must over-provision all of the links by a factor, e.g., 2 or 3x, depending on what the volume of the attacks you want to be protected from is. In ONSET, if you simply provision 1 or 2 fallow transponders per node, you can provide the same bandwidth guarantee for any link without over-provisioning them all. To see this intuitively, consider star graph with 5 spokes. To guarantee an attack threatening 2x bandwidth utilization on any link, you will need

20 transponders (4 per edge given by 2 per each end of each link). If you wanted to provide the same benefit with ONSET, you just need 16 (the original 10 and one more per each of the six nodes). This benefit is modest for the simple example but translates to hundreds of transponders in savings for real-world networks as seen in Table 8.

Network	2x Static	2x ONSET	3x Static	3x ONSET
Sprint	72	47	108	58
ANS	100	68	150	86
CRL	152	109	228	142
Bell Canada	256	176	384	224
SurfNet	272	186	408	236

Table 8. Cost to defend an attack threatening 2 or 3x Max Link Utilization on an arbitrary link with a Static Topology vs. ONSET.

8.5.6 Cost Reduction via Variable Fallow Transponder Allocation.

Cost numbers in Table 8 and link ranks from Figure 52a together suggest that we may be able to further reduce the cost of provisioning ONSET by deploying more fallow transponders around critical links and fewer fallow transponders at other nodes in the network. We evaluate this prospect by starting with a naïve approach wherein we provision 10 fallow transponders to the top 10% ranked links (given by link rank metric defined in § 8.4.1) and then provision half as many fallow transponders at every other node in the network. We then simulate coremelt attacks on each single network link and compare the performance of ONSET with the static and variable fallow transponder allocations. We reproduce this experiment for all of the networks considered in this study, both using ECMP routing and the Ripple* defense. Our results conclusively show that reducing the number of fallow transponders we provision for the bottom 90% of nodes does not reduce the performance of ONSET in defending single-link coremelt attacks—the results were identical to those seen in Figure 56.

Motivated by this result, we explore the effect of variable fallow transponder allocations on ONSET's performance more deeply. In this pursuit, we seek for a *decision-support capability* to determine the appropriate fallow transponder allocation strategy based on an operator's budget and the magnitude of loss they are willing to tolerate. We now allocate only *two* fallow transponders to each node if the node's *rank* is greater than or equal to a given rank, n. We vary n over all of the numeric *rank* values for nodes in the given network. We identify the *cost* of an allocation n as the total number of fallow transponders provisioned under that allocation. In practice, this cost can be swapped with the dollar value of that same number of transponders. Figure 65 shows the cost of each allocation strategy in ANS (65b) and CRL (65a). The most costly solution is to deploy the fallow transponders at every node (where $n \geq 1$). As n grows, we restrict fallow transponders to more highly-ranked nodes. If we limit these to nodes with a rank of 3 or higher, we reduce the cost from 36 to 24 in ANS, and from 66 to 36 in CRL.

To gauge the relative *value* of each of these allocations, we enumerate a series of stressful attacks against every link in each network, repeating this series of attacks on the networks under each fallow transponder allocation, n. We plot the total number of *loss events* for this set of attacks against the cost of a given fallow transponder allocation. We conducted this experiment for both ECMP-based routing and Ripple*. The results, shown in Figure 66, show a Pareto front cost and loss events under each allocation n. In these graphs, better quality solutions fall closest to the origin of the graph, where Loss Events and Cost are both minimized.

Summary*: We provide a decision-support capability in ONSET with which operators can choose how to deploy fallow transponders based on their needs and budget. In practice, an operator can use this capability to deploy ONSET by leveraging data*

from historical attacks they have been exposed to and the existing routing and defense strategy they employ.

8.6 Future Work

In this section, we discuss three opportunities that we plan to explore as part of future work.

(1) Topology Programming API: As ongoing work, we are considering methods to construct a high-level API that can be leveraged to programmatically control network topology and routing. Concretely, our envisioned list of API calls includes:

1. `get_available_transponder(`*node*`)` which returns an index to a fallow transponder at *node*
2. `add_circuit(`$node_u, node_v$`)` which queries fallow transponders at both nodes and pairs them.
3. `get_peer_transponder(`$node_u, node_v$`)` which returns an index to a $node_u$ transponder peered with $node_v$
4. `drop_circuit(`$node_u, node_v$`)` which queries peered transponders at both nodes and de-allocates them.

An example of how these API calls can be employed in coordination with the optimization model described in § 8.4.2 is shown in Algorithm 3. In this example, `SIG_LFA_DETECTED` and `SIG_LFA_OVER` are flags that are set by a network monitor. We assume that this signal is generated by a mechanism outside the scope of this work, e.g., from a programmable switch. This program is agnostic to the type of LFA occurring (e.g., crossfire or coremelt). It uses link utilization data to choose where to add one or more flux links to the network using the available fallow transponders.

Algorithm 3 Topology Programming API for LFAs

```
1: flux_links ← [ ]
2: if SIG_LFA_DETECTED then
3:     flux_links ← optimize_topology()
4: end if
5: for (u, v) ∈ flux_links do
6:     add_circuit(u, v)
7: end for
8: if SIG_LFA_OVER then
9:     for (u, v) ∈ flux_links do
10:        drop_circuit(u, v)
11:    end for
12: end if
```

Line 2 states the triggering condition for activating the optimization step. Line 3 invokes the optimized method from § 8.4.2. The solver returns a set of links that will minimize max link utilization in the network, and in lines 5–6 the links are added to the network with the link provisioning API call. When LFA is over, lines 9–10 remove the flux links from the network.

Other low-level optical hardware configuration requirements must be met to support this high-level API, e.g., configuring transponder power, amplifier gain adjustments on the optical path, and configuring paths with ROADMs. In this work, we are most interested in defining the requirements of our framework at a high level and evaluating the potential benefit of it for LFAs.

Figure 67 illustrates the controller's view of transponder allocations while using the API. The API enables the network operator programmatically query the set of *allocated* and *fallow* transponders at each node in the network. The API also has methods to pair fallow transponders together, thereby establishing a new link in the network. When a link is added to the topology, a pair of fallow transponders between the nodes is activated and those transponders become unavailable for future links until the pair is deactivated.

(2) Topology Jitter: Before launching infrastructure-centric attacks targeting specific network links such as Crossfire [149], attackers must obtain sufficient network topology information, usually through network reconnaissance. This is effective if the network topology is stable/static and attackers use path probing tools such as `traceroute`. Existing countermeasures [148, 301] on infrastructure attacks tend to distinguish between legitimate and attack traffic without handling network reconnaissance. However, these solutions make an unrealistic assumption that link flooding attack traffic is distinguishable from legitimate traffic while reconnaissance tools let attackers easily probe the network paths around the target link and access public services with "indistinguishable" traffic. Ideally, we should thwart attackers' reconnaissance to effectively mitigate the attacks from the root.

To tackle network reconnaissance, we plan to investigate "topology jitter" using ONSET. The idea is to employ a *moving-target defense* by dynamically changing the optical topology to combat network reconnaissance in two steps. (1) In the first step, we will enable *dynamic capacities* by invoking ONSET to allocate new wavelengths on-demand to physically isolate suspicious and malicious flows and steer away from the attack-induced congestion on a targeted link. (2) Second, we will write a defense application using the API calls described above to periodically reallocate wavelengths for suspicious traffic in the optical layer.

(3) Stress Testing and Adversarial Considerations: Our simulation-based analysis of ONSET only scratches the surface for evaluating a topology-programming defense against LFAs. In the previous section, we have attempted to deeply explore basic questions regarding the potential benefit of ONSET with some generous assumptions regarding the availability of optical resources while looking deeply at the effect of network throughput in the face of high-volume attacks. More work is yet to be

done in expanding this analysis; for example, we have yet to consider the cross-traffic dynamics for legitimate and benign traffic as they compete with network services on a dynamic topology. Low-level implementation of the physical links concerning optical-grid spacing and the impact of bandwidth-variable transceivers on the defense framework is also a ripe area of exploration for future work. We hope that our open-source implementation of the framework aids researchers in exploring this area more deeply.

Inspired by [4], adversarial considerations including potential attacks against the ONSET system, overwhelming the compute capability of the network controller that runs the optimization to configure the network topology, among others, are also needed. We plan to consider these as part of future work.

8.7 Related Work

In addition to the work described in § 8.2.2, we refer the readers to recent surveys [284, 316] about LFAs and other DDoS attacks. We cover a few other related efforts here.

Software-based DDoS Defense: SDN and network function virtualization (NFV) enable a wide range of software solutions to detect and mitigate DDoS attacks. For instance, Bohatei [86] orchestrates available NFV resources dynamically to allocate sufficient defense capabilities towards various volumetric attack vectors. SPIFFY [148] leverages SDN capabilities to temporarily increase the bandwidth on a congested link by rerouting around the link and identify the potential attackers via sudden bandwidth augmentation. While software-based defenses bring highest flexibility, they do not scale to terabit LFAs. ACC-Turbo [5] presents a programmable switch based defense for pulse-wave DDoS attacks without dropping suspicious traffic,

but rater, prioritizing it. However, ACC-Turbo is not suited towards sustained LFAs, and when congested, will drop traffic.

Switch-based DDoS Defense: Programmable switches have emerged as a promising platform to perform DDoS detection and mitigation. Unlike traditional switches that focus only on packet forwarding, programmable switches adopt a new type of programmable ASICs and can support additional computation (e.g., DDoS related computation like packet filtering, rate limiting, and hash tables) at a per-packet basis while retaining high line rate guarantees. For instance, Poseidon [317] uses programmable switches as a first-line defender to augment a DDoS scrubbing cluster. Jaqen [181] introduces a switch-native approach to detect and mitigate volumetric attacks. Their design includes a range of probabilistic data structures to efficiently utilize the switch resources for DDoS defense. However, switch-based DDoS defenses highly rely on accurate identification of malicious and benign traffic, which is fundamentally challenging in LFA scenarios where attack traffic may appear as legitimate.

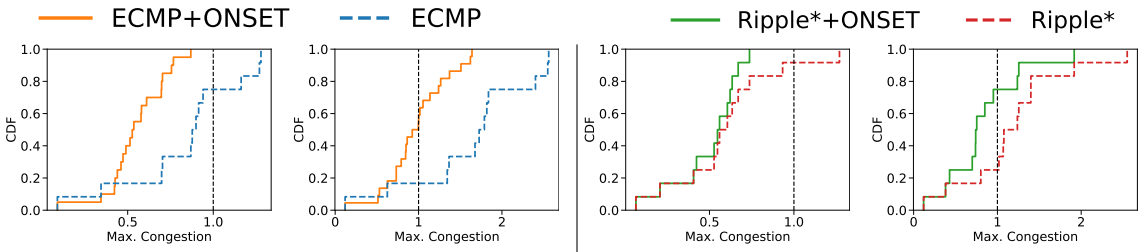
Topology Obfuscation Techniques: There has been a concerted effort to stop attackers from gaining the information about topology required to launch an LFA. These efforts revolve around topology obfuscation, or techniques to hide topological information from an adversary. Efforts include NetHide [196], BottleNet [154], EqualNet [153] and references therein. Topology obfuscation is an orthogonal goal to LFA mitigation. In this work we assume that the attacker has gained knowledge of the topology, and is able to use that knowledge to launch their attacks. We are concerned with finding ways to mitigate loss that may occur during such an attack.

Topology Reconfiguration Techniques: Optical layer topology programming has recently gained attention in several networking contexts. Its benefits have been

demonstrated in the context of traffic engineering in WANs [73, 140, 322] and data centers [109, 234]. Prior work has posed topology reconfiguration to augment DDoS defense [216, 247]. Our paper moves beyond prior work by providing the first general framework for an optical defenses against LFAs and demonstrating its applicability to various networks.

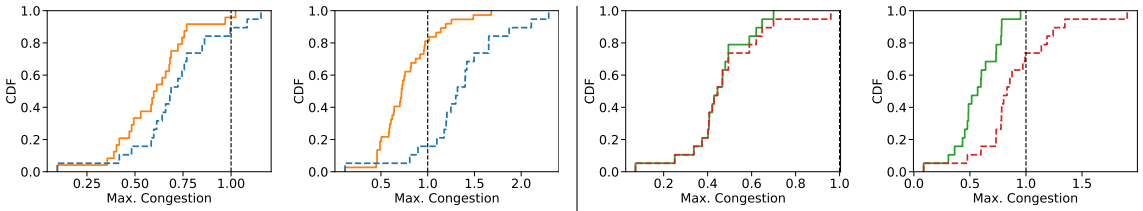
8.8 Summary

LFAs present a particularly insidious and difficult-to-defend-against form of DDoS attacks. While some early work has proposed LFA defenses, the techniques treat the network topology as a static resource and only alter the forwarding behavior for traffic. Consequently, they incur fundamental limitations in terms of tackling attacks, or worse inducing collateral damage elsewhere in the network. Our vision is to leverage optical layer advancement called topology programming to augment existing LFA defense capabilities. Our framework, ONSET, paves the way for this feat. ONSET jointly optimizes topology and routing, using fallow transponders at nodes in the network to create opportunistic links. We show via *what-if* style analysis that ONSET amplifies the benefits of existing LFA defenses for diverse terabit attack scenarios and for a diverse set of networks.



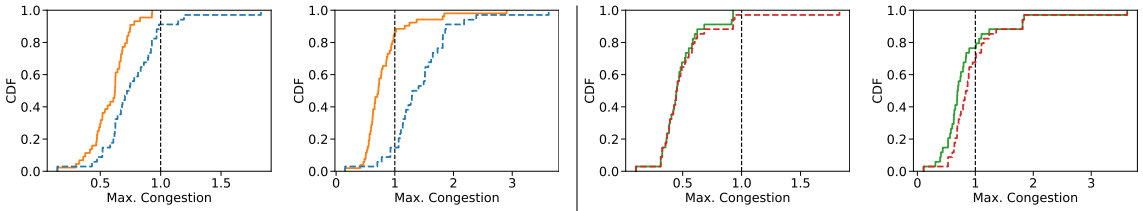
(a) 100 Gbps (b) 200 Gbps (c) 100 Gbps (d) 200 Gbps

Figure 57. All Crossfire Attacks on Sprint.



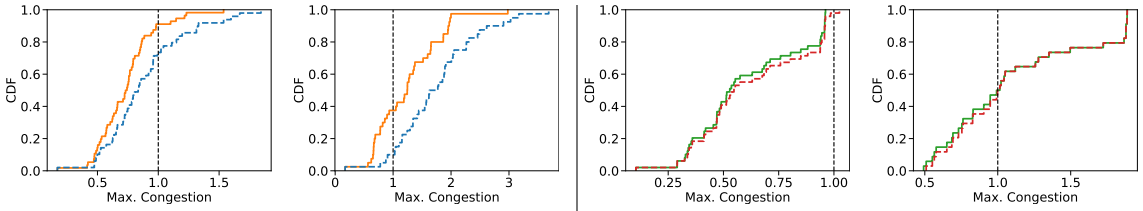
(a) 100 Gbps (b) 200 Gbps (c) 100 Gbps (d) 200 Gbps

Figure 58. All Crossfire Attacks on ANS.



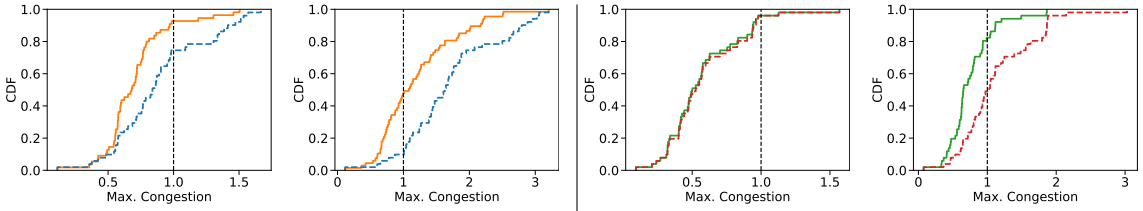
(a) 100 Gbps (b) 200 Gbps (c) 100 Gbps (d) 200 Gbps

Figure 59. All Crossfire Attacks on CRL.



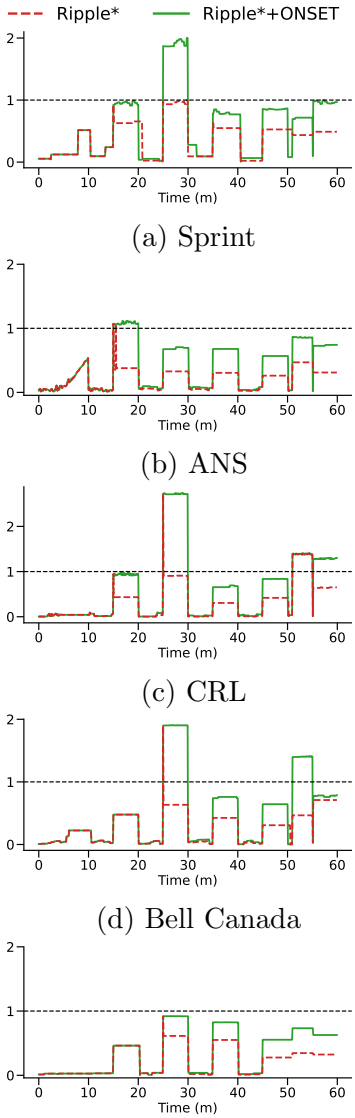
(a) 100 Gbps (b) 200 Gbps (c) 100 Gbps (d) 200 Gbps

Figure 60. All Crossfire Attacks on Bell Canada.



(a) 100 Gbps (b) 200 Gbps (c) 100 Gbps (d) 200 Gbps

Figure 61. All Crossfire Attacks on Surf Net.



(a) Sprint

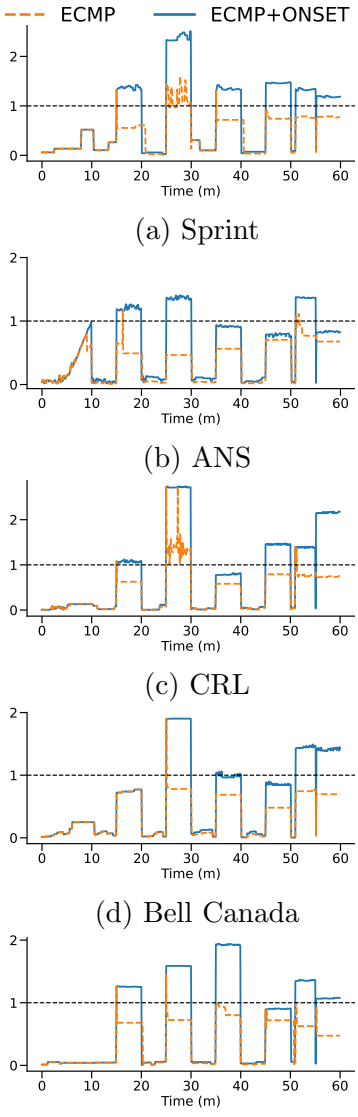
(b) ANS

(c) CRL

(d) Bell Canada

(e) SurfNet

Figure 62. Max. Link Congestion During Rolling Attacks on different networks, Ripple* vs. Ripple*+ONSET



(a) Sprint

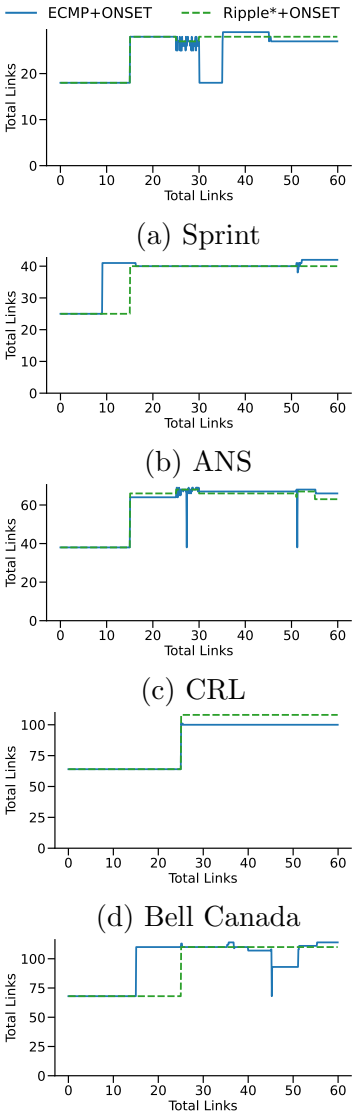
(b) ANS

(c) CRL

(d) Bell Canada

(e) SurfNet

Figure 63. Max. Link congestion During Rolling Attacks on different networks, ECMP vs. ECMP+ONSET



(a) Sprint

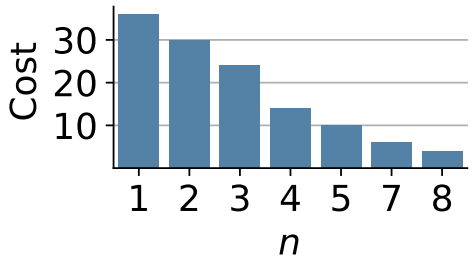
(b) ANS

(c) CRL

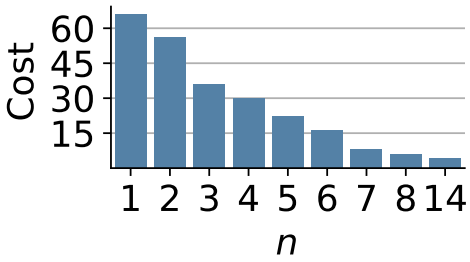
(d) Bell Canada

(e) SurfNet

Figure 64. Total Network Links Active During Rolling Attacks on different networks, ECMP+ONSET vs. Ripple*+ONSET.

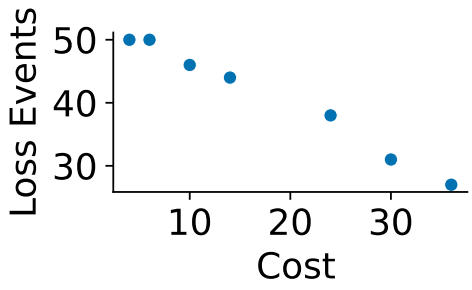


(a) ANS

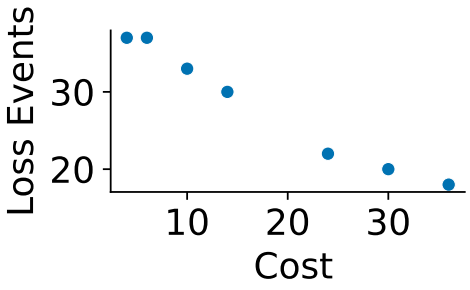


(b) CRL

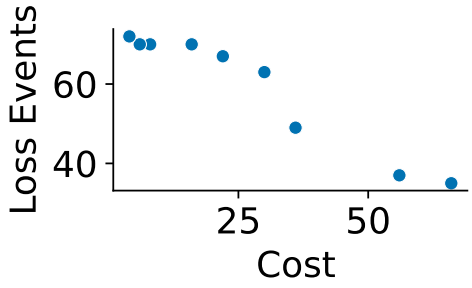
Figure 65. Cost vs. n where cost is the number of fallow transponders allocated to the network for different values of n. n is defined as the minimum rank a node must have to be allocated fallow transponders. When n is equal to one all nodes receive fallow transponders.



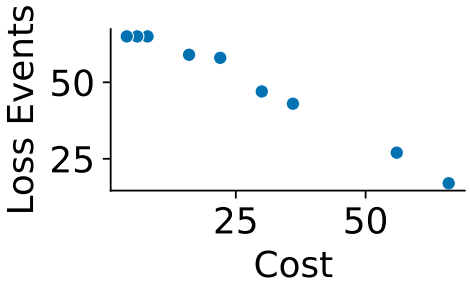
(a) ANS - ECMP



(b) ANS - Ripple*



(c) CRL - ECMP



(d) CRL - Ripple*

Figure 66. Cost vs. Loss Events for various networks under ECMP or Ripple*. As cost increases and fallow transponders are deployed more liberally, the number of Loss Events for the set of attacks falls. An operator may use charts similar to these, with their own network and historical attack data sets, to determine which level of defense they would like to achieve based on their budget.

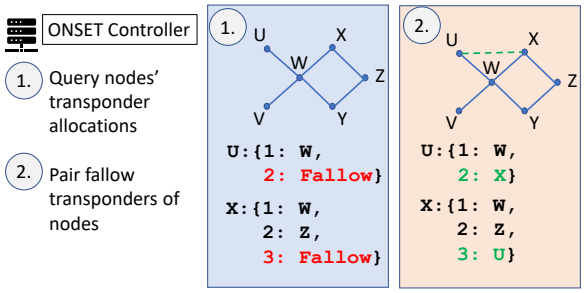


Figure 67. The ONSET controller leverages its optical layer API to query the set of transponders at the two nodes, U and X. It finds that the pair of nodes each have a fallow transponder. It maps the fallow transponder at U to X and the fallow transponder at X to U. After the transponders are mutually paired the link is active and able to forward traffic.

CHAPTER IX

FUTURE WORK

These studies show a significant benefit can be unlocked by considering topology as a non-static entity in the network, but more work and experimentation is needed to deploy these solutions in a live network.

First, we need to test these systems on a live optical network. We have done some early work to characterize the type of lab equipment needed to implement the frameworks for GreyLambda, ONSET, and Doppler at scale. A network with three ROADMs could allow four distinct topologies. We could use this to look at the IP performance for stateful connections before, during, and after topology reconfiguration.

Second, there is work to be done on designing a system to orchestrate and manage the state of the network, including power sent/received, amplifier gain settings, and the set of all optical circuits in the WAN. The system will need to abstract all of this data for an operator so that they can oversee and troubleshoot any problems that occur. The system should be designed to minimize the chances for any such problem. In the event that a problem occurs, the operator should be given sufficient information to find and resolve it. Diagnosing failures in an optical network is an ongoing area of research, and we can leverage the recent developments in the field, e.g. [107, 158, 304], applying and extending them to the reconfigurable topology setting.

Third, simulating network traffic and topology is an exciting domain to expand into more deeply all on its own. There are many ongoing efforts in network simulation that are being pushed by the industry to enhance data center and wide area network performance, e.g., [91, 221, 289]. The Topology Programming simulator we develop is open to extension [209]. At present, it does not support NS3 but building in

that support would allow us to study some of the properties of reconfigurable optical networks mentioned above, e.g., the performance and optimization of stateful TCP connections and UDP flows on a dynamic topology.

Work is also ongoing to apply optical topology programming to data center networks built for training large language models [82,244]. Our framework outlined in the studies from this dissertation can be extended and apply here as well. For example, we could look at the different characteristic traffic matrices that are generated from the various ML training phases and adapt the topology programming loop to optimize latency between GPU elements during these phases.

REFERENCES CITED

[1] Abuzaid, F. NCFlow github repository (accessed Feb. 2024). `https://github.com/stanford-futuredata/pop-ncflow`, 2021.

[2] Abuzaid, F., Kandula, S., Arzani, B., Menache, I., Zaharia, M., and Bailis, P. Contracting wide-area network topologies to solve flow problems quickly. In *USENIX NSDI* (2021), pp. 175–200.

[3] Akamai. Akamai security solutions. `https://www.akamai.com/us/en/products/security/`, 2019.

[4] Aladaileh, M. A., Anbar, M., Hasbullah, I. H., Chong, Y.-W., and Sanjalawe, Y. K. Detection techniques of distributed denial of service attacks on software-defined networking controller–A review. *IEEE Access 8* (2020), 143985–143995.

[5] Alcoz, A. G., Strohmeier, M., Lenders, V., and Vanbever, L. Aggregate-based congestion control for pulse-wave DDoS defense. In *Proceedings of the ACM SIGCOMM 2022 Conference* (2022), pp. 693–706.

[6] Alistarh, D., Ballani, H., Costa, P., Funnell, A., Benjamin, J., Watts, P. M., and Thomsen, B. A high-radix, low-latency optical switch for data centers. *Computer Communication Review 45*, 5 (2015), 367–368.

[7] Alvizu, R., Maier, G., Kukreja, N., Pattavina, A., Morro, R., Capello, A., and Cavazzoni, C. Comprehensive survey on T-SDN: Software-defined networking for transport networks. *IEEE Communications Surveys & Tutorials* (2017).

[8] Anand, S., Garg, N., and Kumar, A. Resource augmentation for weighted flow-time explained by dual fitting. In *Proceedings of the twenty-third annual ACM-SIAM symposium on Discrete Algorithms* (2012), SIAM, pp. 1228–1241.

[9] Atlas, A. K., and Zinin, A. RFC 5286: Basic specification for IP fast reroute: Loop-free alternates. `https://www.ietf.org/rfc/rfc5286.txt`, 2008.

[10] Avin, C., Haeupler, B., Lotker, Z., Scheideler, C., and Schmid, S. Locally self-adjusting tree networks. In *Proceedings of the 2013 IEEE 27th International Symposium on Parallel and Distributed Processing* (Washington, DC, USA, 2013), IPDPS '13, IEEE Computer Society, pp. 395–406.

[11] Avin, C., Hercules, A., Loukas, A., and Schmid, S. rdan: Toward robust demand-aware network designs. In *Information Processing Letters (IPL)* (2018).

[12] Avin, C., Mondal, K., and Schmid, S. Demand-aware network designs of bounded degree. In *31st International Symposium on Distributed Computing (DISC 2017)* (Dagstuhl, Germany, 2017), A. Richa, Ed., vol. 91 of *Leibniz International Proceedings in Informatics (LIPIcs)*, Schloss Dagstuhl – Leibniz-Zentrum für Informatik, pp. 5:1–5:16.

[13] Avin, C., Mondal, K., and Schmid, S. Demand-aware network design with minimal congestion and route lengths. In *Proc. IEEE INFOCOM* (2019).

[14] Avin, C., Mondal, K., and Schmid, S. Dynamically optimal self-adjusting single-source tree networks. In *Proc. Latin American Theoretical Informatics Symposium (LATIN)* (2020).

[15] Avin, C., Salem, I., and Schmid, S. Working set theorems for routing in self-adjusting skip list networks. In *Proc. IEEE INFOCOM* (2020).

[16] Avin, C., and Schmid, S. Toward demand-aware networking: A theory for self-adjusting networks. In *ACM SIGCOMM Computer Communication Review (CCR)* (2018).

[17] Avin, C., and Schmid, S. ReNets: Statically-optimal demand-aware networks. In *Proc. SIAM Symposium on Algorithmic Principles of Computer Systems (APOCS)* (2021).

[18] AWS. AWS Shield: Managed DDoS protection. `https://aws.amazon.com/shield/`, 2019.

[19] Azimi, N. H., Qazi, Z. A., Gupta, H., Sekar, V., Das, S. R., Longtin, J. P., Shah, H., and Tanwer, A. Firefly: A reconfigurable wireless data center fabric using free-space optics. In *ACM SIGCOMM* (2014), ACM, pp. 319–330.

[20] Microsoft global network (accessed Feb. 2023). `https://docs.microsoft.com/en-us/azure/networking/microsoft-global-network`, 2023.

[21] Google cloud networking in depth: Cloud CDN (accessed Feb. 2023). `https://cloud.google.com/blog/products/networking/google-cloud-networking-in-depth-cloud-cdn`, 2023.

[22] Baker, F., and Savola, P. Ingress filtering for multihomed networks. RFC 3704, RFC Editor, March 2004.

[23] Ballani, H., Costa, P., Behrendt, R., Cletheroe, D., Haller, I., Jozwik, K., Karinou, F., Lange, S., Shi, K., Thomsen, B., and Williams, H. Sirius: A flat datacenter network with nanosecond optical switching. In *SIGCOMM* (2020), ACM, pp. 782–797.

[24] Bannister, J. A., Fratta, L., and Gerla, M. Topological design of the wavelength-division optical network. In *IEEE INFOCOM'90* (1990), IEEE Computer Society, pp. 1005–1006.

[25] Bao, J., Dong, D., Zhao, B., Luo, Z., Wu, C., and Gong, Z. FlyCast: Free-space optics accelerating multicast communications in physical layer. *Computer Communication Review 45*, 5 (2015), 97–98.

[26] Baxter, G., Frisken, S., Abakoumov, D., Zhou, H., Clarke, I., Bartos, A., and Poole, S. Highly programmable wavelength selective switch based on liquid crystal on silicon switching elements. In *2006 Optical Fiber Communication Conference and the National Fiber Optic Engineers Conference* (2006), IEEE, pp. 3–pp.

[27] Benzaoui, N., Gonzalez, M. S., Estarán, J. M., Mardoyan, H., Lautenschlaeger, W., Gebhard, U., Dembeck, L., Bigo, S., and Pointurier, Y. Deterministic dynamic networks (DDN). *Journal of Lightwave Technology 37*, 14 (2019), 3465–3474.

[28] Berde, P., Gerola, M., Hart, J., Higuchi, Y., Kobayashi, M., Koide, T., Lantz, B., O'Connor, B., Radoslavov, P., Snow, W., et al. ONOS: Towards an open, distributed SDN OS. In *Proceedings of the third workshop on Hot topics in software defined networking* (2014), pp. 1–6.

[29] Bernier, E., Vukovic, M., Goodwill, D., Daspit, P., and Wang, G. Omninet: A metropolitan 10 Gb/s DWDM photonic switched network trial. In *Optical Fiber Communication Conference* (2004), Optical Society of America, p. WH4.

[30] Beverly, R., Durairajan, R., Plonka, D., and Rohrer, J. P. In the IP of the beholder: Strategies for active IPv6 topology discovery. In *Proceedings of the Internet Measurement Conference 2018* (2018), pp. 308–321.

[31] Bienkowski, M., Fuchssteiner, D., Marcinkowski, J., and Schmid, S. Online dynamic b-matching with applications to reconfigurable datacenter networks. In *Proc. 38th International Symposium on Computer Performance, Modeling, Measurements and Evaluation (PERFORMANCE)* (2020).

[32] Billah, M. R., Blaicher, M., Hoose, T., Dietrich, P.-I., Marin-Palomo, P., Lindenmann, N., Nesic, A., Hofmann, A., Troppenz, U., Moehrle, M., Randel, S., Freude, W., and Koos, C. Hybrid integration of silicon photonics circuits and InP lasers by photonic wire bonding. *Optica 5*, 7 (Jul 2018), 876–883.

[33] Birk, M., Gerard, P., Curto, R., Nelson, L., Zhou, X., Magill, P., Schmidt, T., Malouin, C., Zhang, B., Ibragimov, E., Khatana, S., Glavanovic, M., Lofland, R., Marcoccia, R., Saunders, R., Nicholl, G., Nowell, M., and Forghieri, F. Coherent 100 Gb/s PM-QPSK field trial. *Communications Magazine, IEEE 48* (08 2010), 52–60.

[34] Bloom, S., Korevaar, E., Schuster, J., and Willebrand, H. Understanding the performance of free-space optics. *Journal of optical Networking 2*, 6 (2003), 178–200.

[35] Bogle, J., Bhatia, N., Ghobadi, M., Menache, I., Bjørner, N., Valadarsky, A., and Schapira, M. TeaVaR: Striking the right utilization-availability balance in WAN traffic engineering. In *ACM SIGCOMM* (New York, NY, USA, 2019), SIGCOMM '19, Association for Computing Machinery, p. 29–43.

[36] Bosshart, P., Daly, D., Gibb, G., Izzard, M., McKeown, N., Rexford, J., Schlesinger, C., Talayco, D., Vahdat, A., Varghese, G., and Walker, D. P4: Programming protocol-independent packet processors. *ACM SIGCOMM Computer Communications Review 44*, 3 (July 2014), 87–95.

[37] Boutaba, R., Golab, W., and Iraqi, Y. Lightpaths on demand: A web-services-based management system. *IEEE Communications Magazine* (2004).

[38] Braess, D., Nagurney, A., and Wakolbinger, T. On a paradox of traffic planning. *Transportation science 39*, 4 (2005), 446–450.

[39] Brzezinski, A., and Modiano, E. Dynamic reconfiguration and routing algorithms for IP-over-WDM networks with stochastic traffic. *Journal of Lightwave Technology 23*, 10 (2005), 3188.

[40] Bursztein, E. Inside Mirai the infamous IoT botnet: A retrospective analysis (accessed Mar. 2024). `https://elie.net/blog/security/inside-mirai-the-infamous-iot-botnet-a-retrospective-analysis/`, Dec 2018.

[41] Castro, A., Gifre, L., Chen, C., Yin, J., Zhu, Z., Velasco, L., and Yoo, S.-J. B. Experimental demonstration of brokered orchestration for end-to-end service provisioning and interoperability across heterogeneous multi-operator (multi-AS) optical networks. *European Conference on Optical Communication* (2015).

[42] Castro, A., Velasco, L., Gifre, L., Chen, C., Yin, J., Zhu, Z., Proietti, R., and Yoo, S.-J. B. Brokered orchestration for end-to-end service provisioning across heterogeneous multi-operator (multi-AS) optical networks. *IEEE Journal of Lightwave Technology* (2016).

[43] CenturyLink. CenturyLink DDoS mitigation (accessed Mar. 2024). `https://web.archive.org/web/20181123020241/https://www.centurylink.com/asset/business/enterprise/brochure/ddos-mitigation.pdf`, 2018.

[44] Chamania, M., Caria, M., and Jukan, A. Achieving IP routing stability with optical bypass. *Optical Switching and Networking 7*, 4 (2010), 173–184.

[45] Chang, C., Lee, D., and Jou, Y. Load balanced Birkhoff-von Neumann switches, part I: one-stage buffering. *Comput. Commun. 25*, 6 (2002), 611–622.

[46] Channegowda, M., Nejabati, R., and Simeonidou, D. Software-defined optical networks technology and infrastructure: Enabling software-defined optical network operations. *Journal of Optical Communications and Networking* (2013).

[47] Chatterjee, B. C., Ba, S., and Oki, E. Fragmentation problems and management approaches in elastic optical networks: A survey. *IEEE Communications Surveys & Tutorials 20*, 1 (2017), 183–210.

[48] Chen, H., Fontaine, N. K., Ryf, R., and Neilson, D. T. LCoS-based photonic crossconnect. In *Optical Fiber Communication Conference (OFC) 2019* (2019), Optical Society of America, p. Th1E.6.

[49] Chen, J., Wosinska, L., Machuca, C. M., and Jaeger, M. Cost vs. reliability performance study of fiber access network architectures. *IEEE Communications Magazine 48*, 2 (2010), 56–65.

[50] Chen, K., Singla, A., Singh, A., Ramachandran, K., Xu, L., Zhang, Y., Wen, X., and Chen, Y. OSA: An optical switching architecture for data center networks with unprecedented flexibility. *IEEE/ACM Trans. Netw. 22*, 2 (2014), 498–511.

[51] Chen, K., Wen, X., Ma, X., Chen, Y., Xia, Y., Hu, C., Dong, Q., and Liu, Y. Toward a scalable, fault-tolerant, high-performance optical data center architecture. *IEEE/ACM Transactions on Networking 25*, 4 (2017), 2281–2294.

[52] Chen, L., Chen, K., Zhu, Z., Yu, M., Porter, G., Qiao, C., and Zhong, S. Enabling wide-spread communications on optical fabric with MegaSwitch. In *USENIX NSDI* (Boston, MA, 2017), USENIX Association, pp. 577–593.

[53] Chen, X., Yin, J., Chen, C., Zhu, Z., Casales, A., and Yoo, S. B. Multi-broker based market-driven service provisioning in multi-domain SD-EONs in noncooperative game scenarios. In *European Conference on Optical Communication* (2015), IEEE, pp. 1–3.

[54] Chen, X., Zhu, Z., Proietti, R., and Yoo, S. J. B. On incentive-driven VNF service chaining in inter-datacenter elastic optical networks: A hierarchical game-theoretic mechanism. *IEEE Transactions on Network and Service Management 16*, 1 (2019), 1–12.

[55] Cheung, S., Su, T., Okamoto, K., and Yoo, S. J. B. Ultra-compact silicon photonic 512×512 25 GHz arrayed waveguide grating router. *IEEE Journal of Selected Topics in Quantum Electronics 20*, 4 (2014), 310–316.

[56] Chiu, A. L., Choudhury, G., Clapp, G., Doverspike, R., Feuer, M., Gannett, J. W., Jackel, J., Kim, G. T., Klincewicz, J. G., Kwon, T. J., Li, G., Magill, P., Simmons, J. M., Skoog, R. A., Strand, J., Lehmen, A. V., Wilson, B. J., Woodward, S. L., and Xu, D. Architectures and protocols for capacity efficient, highly dynamic and highly resilient core networks (invited). *IEEE/OSA Journal of Optical Communications and Networking 4*, 1 (January 2012), 1–14.

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

[58] Christodoulopoulos, K., Kokkinos, P., and Varvarigos, E. M. Indirect and direct multicost algorithms for online impairment-aware RWA. *IEEE/ACM Transactions on Networking 19*, 6 (2011), 1759–1772.

[59] Ciena. The history of Optical and Ethernet (accessed Mar. 2024). `https://www.ciena.com/insights/infographics/Packet-Optical-Convergence-Infographic-prx.html`.

[60] Cisco. Cisco ONS 15216 EDFA3 operations guide, release 1.0, chapter 7, TL1 turn up (accessed Mar. 2024). `https://www.cisco.com/c/en/us/td/docs/optical/spares/15216/edfa3/operations/EDFA3/EDFA3PT1.html`, Aug. 2013.

[61] Cisco. Cisco crosswork hierarchical controller (accessed Feb. 2023). `https://www.cisco.com/c/en/us/products/cloud-systems-management/crosswork-hierarchical-controller/index.html`, 2023.

[62] Clark, K. A., Cletheroe, D., Gerard, T., Haller, I., Jozwik, K., Shi, K., Thomsen, B., Williams, H., Zervas, G., Ballani, H., et al. Synchronous subnanosecond clock and data recovery for optically switched data centres using clock phase caching. *Nature Electronics 3*, 7 (2020), 426–433.

[63] Cloudflare. Advanced DDoS attack protection (accessed Mar. 2024). https://www.cloudflare.com/ddos/, 2019.

[64] Cox, J. SDN control of a coherent open line system. In *Optical Fiber Communications Conference* (March 2015), pp. 1–1.

[65] Crovella, M., and Kolaczyk, E. Graph wavelets for spatial traffic analysis. In *IEEE INFOCOM 2003. Twenty-second Annual Joint Conference of the IEEE Computer and Communications Societies (IEEE Cat. No. 03CH37428)* (2003), vol. 3, IEEE, pp. 1848–1857.

[66] Cui, Y., Xiao, S., Wang, X., Yang, Z., Yan, S., Zhu, C., Li, X., and Ge, N. Diamond: Nesting the data center network with wireless rings in 3-d space. *IEEE/ACM Trans. Netw. 26*, 1 (2018), 145–160.

[67] DDoS Attacks Up By 84% in Q1 (accessed Mar. 2024). https://www.cybersecurityintelligence.com/blog/ddos-attacks-up-by-84-in-q1-4346.html.

[68] Dai, W., Foerster, K.-T., Fuchssteiner, D., and Schmid, S. Load-optimization in reconfigurable networks: Algorithms and complexity of flow routing. In *Proc. 38th International Symposium on Computer Performance, Modeling, Measurements and Evaluation (PERFORMANCE)* (2020).

[69] De Felipe, D., Kleinert, M., Zawadzki, C., Polatynski, A., Irmscher, G., Brinker, W., Moehrle, M., Bach, H.-G., Keil, N., and Schell, M. Recent developments in polymer-based photonic components for disruptive capacity upgrade in data centers. *Journal of Lightwave Technology 35*, 4 (2016), 683–689.

[70] Dihuni. Every day big data statistics – 2.5 quintillion bytes of data created daily. https://www.dihuni.com/2020/04/10/every-day-big-data-statistics-2-5-quintillion-bytes-of-data-created-daily/ (Accessed Feb. 2023), April 2020.

[71] Dinitz, M., and Moseley, B. Scheduling for weighted flow and completion times in reconfigurable networks. In *INFOCOM* (2020).

[72] Dukic, V., Khanna, G., Gkantsidis, C., Karagiannis, T., Parmigiani, F., Singla, A., Filer, M., Cox, J. L., Ptasznik, A., Harland, N., Saunders, W., and Belady, C. Beyond the mega-data center: networking multi-data center regions. In *SIGCOMM* (2020), ACM, pp. 765–781.

[73] Durairajan, R., Barford, P., Sommers, J., and Walter, W. GreyFiber: A system for providing flexible access to wide-area connectivity. *1807.05242 abs/1807.05242* (2018).

[74] Durairajan, R., Barford, P., Sommers, J., and Willinger, W. InterTubes: A study of the US long-haul fiber-optic infrastructure. In *ACM SIGCOMM* (2015).

[75] Durairajan, R., Ghosh, S., Tang, X., Barford, P., and Eriksson, B. Internet Atlas: A geographic database of the Internet. In *Proceedings of ACM HotPlanet* (2013), pp. 15–20.

[76] Edmonds, J. Paths, trees and flowers. *Canad. J. Math 17* (1965), 449–467.

[77] Eisenbud, D. E., Yi, C., Contavalli, C., Smith, C., Kononov, R., Mann-Hielscher, E., Cilingiroglu, A., Cheyney, B., Shang, W., and Hosein, J. D. Maglev: A fast and reliable software network load balancer. In *USENIX NSDI* (2016), pp. 523–535.

[78] Ellis, A. D., Zhao, J., and Cotter, D. Approaching the non-linear shannon limit. *Journal of Lightwave Technology 28*, 4 (2009), 423–433.

[79] Enns, R., Björklund, M., Bierman, A., and Schönwälder, J. Network Configuration Protocol (NETCONF). RFC 6241, June 2011.

[80] Eriksson, B., Durairajan, R., and Barford, P. RiskRoute: A framework for mitigating network outage threats. In *ACM CoNEXT* (2013), pp. 405–416.

[81] Even, S., Itai, A., and Shamir, A. On the complexity of time table and multi-commodity flow problems. In *16th annual symposium on foundations of computer science (sfcs 1975)* (1975), IEEE, pp. 184–193.

[82] Fariborz, M., Samani, M., Fotouhi, P., Proietti, R., Yi, I.-M., Akella, V., Lowe-Power, J., Palermo, S., and Yoo, S. B. LLM: Realizing low-latency memory by exploiting embedded silicon photonics for irregular workloads. In *International Conference on High Performance Computing* (2022), Springer, pp. 44–64.

[83] Farrington, N., Forencich, A., Porter, G., Sun, P. ., Ford, J. E., Fainman, Y., Papen, G. C., and Vahdat, A. A multiport microsecond optical circuit switch for data center networking. *IEEE Photonics Technology Letters 25*, 16 (Aug 2013), 1589–1592.

[84] Farrington, N., Porter, G., Fainman, Y., Papen, G., and Vahdat, A. Hunting mice with microsecond circuit switches. In *HotNets* (2012), ACM, pp. 115–120.

[85] Farrington, N., Porter, G., Radhakrishnan, S., Bazzaz, H. H., Subramanya, V., Fainman, Y., Papen, G., and Vahdat, A. Helios: A hybrid electrical/optical switch architecture for modular data centers. In *ACM SIGCOMM* (2010), pp. 339–350.

[86] Fayaz, S. K., Tobioka, Y., Sekar, V., and Bailey, M. Flexible and elastic DDoS defense using Bohatei. In *Proc. USENIX Security* (Washington, D.C., Aug. 2015), USENIX Association, pp. 817–832.

[87] Fedor, M., Schoffstall, M. L., Davin, J. R., and Case, D. J. D. Simple Network Management Protocol (SNMP). RFC 1157, May 1990.

[88] Fenz, T., Foerster, K.-T., Schmid, S., and Villedieu, A. Efficient non-segregated routing for reconfigurable demand-aware networks. In *18th IFIP Networking Conference (IFIP Networking)* (May 2019).

[89] Ferguson, A. D., Gribble, S. D., Hong, C.-Y., Killian, C. E., Mohsin, W., Muehe, H., Ong, J., Poutievski, L., Singh, A., Vicisano, L., et al. Orion: Google's software-defined networking control plane. In *NSDI* (2021), pp. 83–98.

[90] Ferguson, P., and Senie, D. Network ingress filtering: Defeating denial of service attacks which employ IP source address spoofing. RFC 2827, RFC Editor, May 2000.

[91] Ferrari, A., Filer, M., Balasubramanian, K., Yin, Y., Rouzic, E. L., Kundrát, J., Grammel, G., Galimberti, G., and Curri, V. GNPy: An open source application for physical layer aware open optical networks. *Journal of Optical Communications and Networking 12*, 6 (March 2020), C31–C40.

[92] Figueira, S., Naiksatam, S., Cohen, H., Cutrell, D., Daspit, P., Gutierrez, D., Hoang, D. B., Lavian, T., Mambretti, J., Merrill, S., et al. DWDM-RAM: Enabling grid services with dynamic optical networks. In *IEEE International Symposium on Cluster Computing and the Grid, 2004. CCGrid 2004.* (2004), IEEE, pp. 707–714.

[93] Filer, M., Gaudette, J., Ghobadi, M., Mahajan, R., Issenhuth, T., Klinkers, B., and Cox, J. Elastic optical networking in the Microsoft cloud (invited). *J. Opt. Commun. Netw. 8*, 7 (Jul 2016), A45–A54.

[94] Filer, M., Gaudette, J., Yin, Y., Billor, D., Bakhtiari, Z., and Cox, J. L. Low-margin optical networking at cloud scale [invited]. *J. Opt. Commun. Netw. 11*, 10 (Oct 2019), C94–C108.

[95] Firestone, D. SmartNIC: Accelerating Azure's network with FPGAs on OCS servers (accessed Feb. 2023). `https://ocpussummit2016.sched.com/event/68u4/`, 2016.

[96] Foerster, K., Schmid, S., and Vissicchio, S. Survey of consistent software-defined network updates. *IEEE Communications Surveys and Tutorials 21*, 2 (2019), 1435–1461.

[97] Foerster, K.-T., Ghobadi, M., and Schmid, S. Characterizing the algorithmic complexity of reconfigurable data center architectures. In *ANCS* (2018), IEEE/ACM.

[98] Foerster, K.-T., Luo, L., and Ghobadi, M. Optflow: A flow-based abstraction for programmable topologies. In *Proceedings of the Symposium on SDN Research* (2020), pp. 96–102.

[99] Foerster, K.-T., Pacut, M., and Schmid, S. On the complexity of non-segregated routing in reconfigurable data center architectures. *ACM SIGCOMM Computer Communication Review (CCR)* (2019).

[100] Foerster, K.-T., and Schmid, S. Survey of reconfigurable data center networks: Enablers, algorithms, complexity. *SIGACT News 50*, 2 (July 2019), 62–79.

[101] Foremski, P., Plonka, D., and Berger, A. W. Entropy/IP: Uncovering structure in IPv6 addresses. In *Proceedings of the 2016 ACM Internet Measurement Conference, IMC'16, Santa Monica, CA, USA, November 14-16, 2016* (New York, New York, USA, 2016), P. Gill, J. S. Heidemann, J. W. Byers, and R. Govindan, Eds., ACM, pp. 167–181.

[102] Forencich, A., and Papen, G. C. O-Net: An "open" optical networking framework. In *Proceedings of the ACM SIGCOMM 2019 Workshop on Optical Systems Design* (New York, NY, USA, 2019), OptSys '19, Association for Computing Machinery.

[103] Fortz, B., and Thorup, M. Optimizing OSPF/IS-IS weights in a changing world. *IEEE JSAC* (2002).

[104] Gaiser, M. How much monetary damage was done during the Oct 21, 2016 DDoS of DynDNS? (accessed Mar. 2024). `https://www.quora.com/How-much-monetary-damage-was-done-during-the-Oct-21-2016-DDOS-of-DynDNS`, Oct 2016.

[105] Gao, L., Griffin, T., and Rexford, J. Inherently safe backup routing with BGP. In *IEEE INFOCOM* (2001), pp. 547–556.

[106] Gerstel, O., Filsfils, C., Telkamp, T., Gunkel, M., Horneffer, M., Lopez, V., and Mayoral, A. Multi-layer capacity planning for IP-optical networks. *IEEE Communications Magazine 52*, 1 (2014), 44–51.

[107] Ghobadi, M., and Mahajan, R. Optical layer failures in a large backbone. In *Proceedings of the 2016 Internet Measurement Conference* (2016), ACM, pp. 461–467.

[108] Ghobadi, M., Mahajan, R., Phanishayee, A., Blanche, P.-A., Rastegarfar, H., Glick, M., and Kilper, D. Design of mirror assembly for an agile reconfigurable data center interconnect. Tech. Rep. MSR-TR-2016-1139, Microsoft, June 2016.

[109] Ghobadi, M., Mahajan, R., Phanishayee, A., Devanur, N., Kulkarni, J., Ranade, G., Blanche, P.-A., Rastegarfar, H., Glick, M., and Kilper, D. ProjecToR: Agile reconfigurable data center interconnect. In *Proceedings of the 2016 ACM SIGCOMM Conference* (2016), ACM, pp. 216–229.

[110] Giorgetti, A., Paolucci, F., Cugini, F., and Castoldi, P. Dynamic restoration with GMPLS and SDN control plane in elastic optical networks. *Journal of Optical Communications and Networking* (2015).

[111] Global Research NOC Systems Engineering. GlobalNOC routerproxy (accessed Feb. 2024). `https://routerproxy.grnoc.iu.edu/`.

[112] Goel, A., Kapralov, M., and Khanna, S. Perfect matchings in O(nlog n) time in regular bipartite graphs. *SIAM J. Comput. 42*, 3 (2013), 1392–1404.

[113] Gossels, J., Choudhury, G., and Rexford, J. Robust network design for IP/optical backbones. *IEEE/OSA Journal of Optical Communications and Networking 11*, 8 (2019), 478–490.

[114] Govil, J., and Govil, J. On the investigation of transactional and interoperability issues between IPv4 and IPv6. In *2007 IEEE International Conference on Electro/Information Technology* (2007), IEEE, pp. 604–609.

[115] Gringeri, S., Bitar, N., and Xia, T. J. Extending software defined network principles to include optical transport. *IEEE Communications Magazine 51*, 3 (March 2013), 32–40.

[116] Gunes, M. H., and Saraç, K. Inferring subnets in router-level topology collection studies. In *Proceedings of the 7th ACM SIGCOMM Internet Measurement Conference, IMC 2007, San Diego, California, USA, October 24-26, 2007* (New York, New York, USA, 2007), C. Dovrolis and M. Roughan, Eds., ACM, pp. 203–208.

[117] Gunkel, M., Autenrieth, A., Neugirg, M., and Elbers, J.-P. Advanced multilayer resilience scheme with optical restoration for IP-over-DWDM core networks. In *2012 IV International Congress on Ultra Modern Telecommunications and Control Systems* (2012), IEEE, pp. 657–662.

[118] Guo, J., and Zhu, Z. When deep learning meets inter-datacenter optical network management: Advantages and vulnerabilities. *J. Lightwave Technol. 36*, 20 (Oct 2018), 4761–4773.

[119] Gupta, H., Curran, M., and Zhan, C. Near-optimal multihop scheduling in general circuit-switched networks. In *CoNEXT* (2020), ACM, pp. 31–45.

[120] Hamza, A. S., Deogun, J. S., and Alexander, D. R. Wireless communication in data centers: A survey. *IEEE Commun. Surv. Tutorials 18*, 3 (2016), 1572–1595.

[121] Hamza, A. S., Deogun, J. S., and Alexander, D. R. Classification framework for free space optical communication links and systems. *IEEE Commun. Surv. Tutorials 21*, 2 (2019), 1346–1382.

[122] Hamza, A. S., Yadav, S., Ketan, S., Deogun, J. S., and Alexander, D. R. OWCell: Optical wireless cellular data center network architecture. In *ICC* (2017), IEEE, pp. 1–6.

[123] Hart, P. E., Nilsson, N. J., and Raphael, B. A formal basis for the heuristic determination of minimum cost paths. *IEEE Transactions on Systems Science and Cybernetics 4*, 2 (1968), 100–107.

[124] Hatori, N., Shimizu, T., Okano, M., Ishizaka, M., Yamamoto, T., Urino, Y., Mori, M., Nakamura, T., and Arakawa, Y. A hybrid integrated light source on a silicon platform using a trident spot-size converter. *Journal of Lightwave Technology 32*, 7 (2014), 1329–1336.

[125] Heike Hofmann, H. W., and Kafadar, K. Letter-value plots: Boxplots for large data. *Journal of Computational and Graphical Statistics 26*, 3 (2017), 469–477.

[126] Heorhiadi, V. TMgen (accessed Feb. 2023). `https://github.com/progwriter/TMgen/blob/master/docs/quickstart.rst`, 2018.

[127] Hofmeister, T., Vusirikala, V., and Koley, B. How can flexibility on the line side best be exploited on the client side? In *Optical Fiber Communication Conference* (2016), Optical Society of America, p. W4G.4.

[128] Holterbach, T., Molero, E. C., Apostolaki, M., Dainotti, A., Vissicchio, S., and Vanbever, L. Blink: Fast connectivity recovery entirely in the data plane. In *USENIX NSDI* (2019), pp. 161–176.

[129] Hong, C.-Y., Kandula, S., Mahajan, R., Zhang, M., Gill, V., Nanduri, M., and Wattenhofer, R. Achieving high utilization with software-driven WAN. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (New York, NY, USA, 2013), SIGCOMM '13, Association for Computing Machinery, p. 15–26.

[130] Hong, C.-Y., Mandal, S., Al-Fares, M., Zhu, M., Alimi, R., Bhagat, C., Jain, S., Kaimal, J., Liang, S., Mendelev, K., et al. B4 and after: Managing hierarchy, partitioning, and asymmetry for availability and scale in Google's software-defined WAN. In *ACM SIGCOMM* (New York, NY, USA, 2018), SIGCOMM '18, Association for Computing Machinery, pp. 74–87.

[131] Hou, T., Wang, T., Lu, Z., and Liu, Y. Combating adversarial network topology inference by proactive topology obfuscation. *IEEE/ACM Transactions on Networking 29*, 6 (2021), 2779–2792.

[132] Huang, X. S., Sun, X. S., and Ng, T. S. E. Sunflow: Efficient optical circuit scheduling for coflows. In *CoNEXT* (2016), ACM, pp. 297–311.

[133] International Telecommunication Union. *Forward error correction for high bit-rate DWDM submarine systems*, 7 2013. Approved in 2013-07-12.

[134] Iovanna, P., Sabella, R., and Settembre, M. A traffic engineering system for multilayer networks based on the GMPLS paradigm. *IEEE Network 17*, 2 (2003), 28–37.

[135] Ives, D. J., Alvarado, A., and Savory, S. J. Throughput gains from adaptive transceivers in nonlinear elastic optical networks. *Journal of Lightwave Technology 35*, 6 (2017), 1280–1289.

[136] Ives, D. J., Bayvel, P., and Savory, S. J. Routing, modulation, spectrum and launch power assignment to maximize the traffic throughput of a nonlinear optical mesh network. *Photonic Network Communications 29*, 3 (Jun 2015), 244–256.

[137] Jain, S., Kumar, A., Mandal, S., Ong, J., Poutievski, L., Singh, A., Venkata, S., Wanderer, J., Zhou, J., Zhu, M., Zolla, J., Hölzle, U., Stuart, S., and Vahdat, A. B4: experience with a globally-deployed software defined WAN. *ACM SIGCOMM* (2013), 3–14.

[138] Jia, S., Jin, X., Ghasemiesfeh, G., Ding, J., and Gao, J. Competitive analysis for online scheduling in software-defined optical WAN. In *INFOCOM* (2017), IEEE, pp. 1–9.

[139] Jin, C., Wang, H., and Shin, K. G. Hop-count filtering: an effective defense against spoofed DDoS traffic. In *Proceedings of the 10th ACM conference on Computer and communications security* (2003), ACM, pp. 30–41.

[140] Jin, X., Li, Y., Wei, D., Li, S., Gao, J., Xu, L., Li, G., Xu, W., and Rexford, J. Optimizing bulk transfers with software-defined optical WAN. In *ACM SIGCOMM* (2016), ACM, pp. 87–100.

[141] Jin, X., Liu, H. H., Gandhi, R., Kandula, S., Mahajan, R., Zhang, M., Rexford, J., and Wattenhofer, R. Dynamic scheduling of network updates. In *ACM SIGCOMM* (2014), ACM, pp. 539–550.

[142] Jinno, M., Kozicki, B., Takara, H., Watanabe, A., Sone, Y., Tanaka, T., and Hirano, A. Distance-adaptive spectrum resource allocation in spectrum-sliced elastic optical path network [topics in optical communications]. *IEEE Communications Magazine 48*, 8 (August 2010), 138–145.

[143] Jørgensen, A., Kong, D., Henriksen, M., Klejs, F., Ye, Z., Helgason, O., Hansen, H., Hu, H., Yankov, M., Forchhammer, S., et al. Petabit-per-second data transmission using a chip-scale microcomb ring resonator source. *Nature Photonics* (2022), 1–5.

[144] Kachris, C., and Tomkos, I. A survey on optical interconnects for data centers. *IEEE Communications Surveys & Tutorials 14*, 4 (2012), 1021–1036.

[145] Kalmbach, P., Zerwas, J., Babarczi, P., Blenk, A., Kellerer, W., and Schmid, S. Empowering self-driving networks. In *Proc. ACM SIGCOMM 2018 Workshop on Self-Driving Networks (SDN)* (2018).

[146] Kandula, S., Menache, I., Schwartz, R., and Babbula, S. R. Calendaring for wide area networks. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (New York, NY, USA, 2014), SIGCOMM '14, Association for Computing Machinery, p. 515–526.

[147] Kandula, S., Padhye, J., and Bahl, P. Flyways to de-congest data center networks. In *HotNets* (2009), ACM ACM SIGCOMM.

[148] Kang, M. S., Gligor, V. D., and Sekar, V. SPIFFY: Inducing cost-detectability tradeoffs for persistent link-flooding attacks. In *NDSS* (2016), pp. 53–55.

[149] Kang, M. S., Lee, S. B., and Gligor, V. D. The Crossfire attack. In *2013 IEEE Symposium on Security and Privacy, SP 2013, Berkeley, CA, USA, May 19-22, 2013* (Washington, DC, USA, 2013), IEEE Computer Society, pp. 127–141.

[150] Kassing, S., Valadarsk, A., Shahaf, G., Schapira, M., and Singla, A. Beyond fat-trees without antennae, mirrors, and disco-balls. In *ACM SIGCOMM* (2017), HotNets '16, ACM, pp. 64–70.

[151] Kilper, D., and Bergman, K. TURBO: Terabits/s using reconfigurable bandwidth optics (final report), 5 2020.

[152] Kilper, D. C., and Li, Y. Optical physical layer SDN: Enabling physical layer programmability through open control systems. In *Optical Fiber Communications Conference* (2017), pp. W1H–3.

[153] Kim, J., Marin, E., Conti, M., and Shin, S. Equalnet: A secure and practical defense for long-term network topology obfuscation. In *29th Annual Network and Distributed System Security Symposium, NDSS Symposium 2022, San Diego, California, USA, April 24-28, 2022* (Reston, Virginia, USA, 2022), The Internet Society, p. 18.

[154] Kim, J., Nam, J., Lee, S., Yegneswaran, V., Porras, P., and Shin, S. Bottlenet: Hiding network bottlenecks using SDN-based topology deception. *IEEE Transactions on Information Forensics and Security 16* (2021), 3138–3153.

[155] Knight, S., Nguyen, H. X., Falkner, N., Bowden, R., and Roughan, M. The Internet topology zoo. *IEEE Journal on Selected Areas in Communications (JSAC) 29*, 9 (October 2011), 1765–1775.

[156] Kodialam, M., and Lakshman, T. V. Integrated dynamic IP and wavelength routing in IP over WDM networks. In *IEEE INFOCOM* (2001), vol. 1, pp. 358–366.

[157] Kokkinos, P., Soumplis, P., and Varvarigos, E. A. Pattern-driven resource allocation in optical networks. *IEEE Transactions on Network and Service Management 16*, 2 (2019), 489–504.

[158] Kong, J., Morgan, R., Qin, C., Guan, B., Yin, Y., Ma, H., and Gaudette, J. Secure and high-available cloud optical network data collecting and analysis system. In *2023 Optical Fiber Communications Conference and Exhibition (OFC)* (2023), pp. 1–3.

[159] Kozdrowski, S., Żotkiewicz, M., and Sujecki, S. Optimization of optical networks based on CDC-ROADM technology. *Applied Sciences 9*, 3 (2019), 399.

[160] Krishnaswamy, R. M., and Sivarajan, K. N. Design of logical topologies: A linear formulation for wavelength-routed optical networks with no wavelength changers. *IEEE/ACM Transactions On Networking 9*, 2 (2001), 186–198.

[161] Krishnaswamy, U., Singh, R., Bjørner, N., and Raj, H. Decentralized cloud wide-area network traffic engineering with BLASTSHIELD. In *USENIX NSDI* (2022), pp. 325–338.

[162] Kulkarni, J., Schmid, S., and Schmidt, P. Scheduling opportunistic links in two-tiered reconfigurable datacenters. *arXiv preprint arXiv:2010.07920* (2020).

[163] Kumar, A., Jain, S., Naik, U., Raghuraman, A., Kasinadhuni, N., Zermeno, E. C., Gunn, C. S., Ai, J., Carlin, B., Amarandei-Stavila, M., Robin, M., Siganporia, A., Stuart, S., and Vahdat, A. BwE: Flexible, hierarchical bandwidth allocation for WAN distributed computing. In *ACM SIGCOMM* (2015), pp. 1–14.

[164] Kumar, M. N., Sujatha, P., Kalva, V., Nagori, R., Katukojwala, A. K., and Kumar, M. Mitigating economic denial of sustainability (EDoS) in cloud computing using in-cloud scrubber service. In *2012 Fourth International Conference on Computational Intelligence and Communication Networks* (2012), IEEE, pp. 535–539.

[165] Kumar, P., Yu, C., Yuan, Y., Foster, N., Kleinberg, R., and Soulé, R. YATES: Rapid prototyping for traffic engineering systems. In *Proceedings of the Symposium on SDN Research* (New York, NY, USA, 2018), SOSR '18, Association for Computing Machinery.

[166] Kumar, P., Yuan, Y., Yu, C., Foster, N., Kleinberg, R., Lapukhov, P., Lim, C. L., and Soulé, R. Semi-oblivious traffic engineering: The road not taken. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI 18)* (2018), pp. 157–170.

[167] Labovitz, C. Internet traffic 2009-2019. *Presentation at NANOG 76* (2019), 9–12.

[168] Lange, S., Raja, A., Shi, K., Karpov, M., Behrendt, R., Cletheroe, D., Haller, I., Karinou, F., Fu, X., Liu, J., Lukashchuk, A., Thomsen, B., Jozwik, K., Costa, P., Kippenberg, T. J., and Ballani, H. Sub-nanosecond optical switching using chip-based soliton microcombs. In *Optical Fiber Communication Conference (OFC'20)* (March 2020), The Optical Society (OSA).

[169] Lantz, B., Díaz-Montiel, A. A., Yu, J., Rios, C., Ruffini, M., and Kilper, D. Demonstration of software-defined packet-optical network emulation with Mininet-Optical and ONOS. In *Optical Fiber Communication Conference (OFC) 2020* (2020), Optical Society of America, p. M3Z.9.

[170] Laoutaris, N., Sirivianos, M., Yang, X., and Rodriguez, P. Inter-datacenter bulk transfers with NetStitcher. In *ACM SIGCOMM* (2011), pp. 74–85.

[171] Lehmen, A. V., Doverspike, R., Clapp, G., Freimuth, D. M., Gannett, J., Kim, K., Kobrinski, H., Mavrogiorgis, E., Pastor, J., Rauch, M., Ramakrishnan, K. K., Skoog, R., Wilson, B., and Woodward, S. L. CORONET: Testbeds, cloud computing, and lessons learned. In *Optical Fiber Communication Conference* (2014), Optical Society of America, p. W4B.1.

[172] Levin, S. L., and Schmidt, S. IPv4 to IPv6: Challenges, solutions, and lessons. *Telecommunications Policy 38*, 11 (2014), 1059–1068.

[173] Lewis, B., Broadbent, M., and Race, N. P4ID: P4 enhanced intrusion detection. In *2019 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)* (2019), pp. 1–4.

[174] Li, H., Zhang, H.-y., Wang, L., Li, Y.-b., Lai, J.-s., Tang, R., Zhao, W.-y., Wu, B.-b., Wang, D., Zhao, X., et al. Field trial of network survivability based on OTN and ROADM hybrid networking. In *Asia Communications and Photonics Conference* (2017), Optical Society of America, pp. M3C–2.

[175] Li, M., Zaccarin, D., and Barnard, C. Reconfigurable optical add-drop multiplexer, Feb. 27 2007. US Patent 7,184,666.

[176] Liu, H., Lu, F., Forencich, A., Kapoor, R., Tewari, M., Voelker, G. M., Papen, G., Snoeren, A. C., and Porter, G. Circuit switching under the radar with REACToR. In *USENIX NSDI* (2014), USENIX, USENIX Association, pp. 1–15.

[177] Liu, H., Mukerjee, M. K., Li, C., Feltman, N., Papen, G., Savage, S., Seshan, S., Voelker, G. M., Andersen, D. G., Kaminsky, M., Porter, G., and Snoeren, A. C. Scheduling techniques for hybrid circuit/packet networks. In *CoNEXT* (2015), ACM, pp. 41:1–41:13.

[178] Liu, H., Urata, R., Yasumura, K., Zhou, X., Bannon, R., Berger, J., Dashti, P., Jouppi, N., Lam, C., Li, S., Mao, E., Nelson, D., Papen, G., Tariq, M., and Vahdat, A. Lightwave fabrics: At-scale optical circuit switching for datacenter and machine learning systems. In *Proceedings of the ACM SIGCOMM 2023 Conference* (New York, NY, USA, 2023), ACM SIGCOMM '23, Association for Computing Machinery, p. 499–515.

[179] Liu, H. H., Kandula, S., Mahajan, R., Zhang, M., and Gelernter, D. Traffic engineering with forward fault correction. In *Proceedings of the 2014 ACM Conference on SIGCOMM* (New York, NY, USA, 2014), SIGCOMM '14, Association for Computing Machinery, p. 527–538.

[180] Liu, Y. J., Gao, P. X., Wong, B., and Keshav, S. Quartz: A new design element for low-latency DCNs. In *ACM SIGCOMM* (2014), ACM, pp. 283–294.

[181] Liu, Z., Namkung, H., Nikolaidis, G., Lee, J., Kim, C., Jin, X., Braverman, V., Yu, M., and Sekar, V. Jaqen: A high-performance switch-native approach for detecting and mitigating volumetric DDoS attacks with programmable switches. In *30th USENIX Security Symposium (USENIX Security 21)* (2021), pp. 3829–3846.

[182] Loher, D. SONiC: Software for open networking in the cloud (accessed Feb. 2023). `https://sonic-net.github.io/SONiC/`, 2023.

[183] Lu, Y., and Gu, H. Flexible and scalable optical interconnects for data centers: Trends and challenges. *IEEE Communications Magazine 57*, 10 (October 2019), 27–33.

[184] Luckie, M. Scamper: A scalable and extensible packet prober for active measurement of the internet. In *Proceedings of the 10th ACM SIGCOMM conference on Internet measurement* (2010), pp. 239–245.

[185] Luo, L., Foerster, K., Schmid, S., and Yu, H. Deadline-aware multicast transfers in software-defined optical wide-area networks. *IEEE Journal on Selected Areas in Communications* (2020).

[186] Luo, L., Foerster, K., Schmid, S., and Yu, H. SplitCast: Optimizing multicast flows in reconfigurable datacenter networks. In *INFOCOM* (2020), IEEE.

[187] Luo, L., Yu, H., Foerster, K.-T., Noormohammadpour, M., and Schmid, S. Inter-datacenter bulk transfers: Trends and challenges. *IEEE Network to appear* (2020).

[188] Mahimkar, A., Chiu, A., Doverspike, R., Feuer, M. D., Magill, P., Mavrogiorgis, E., Pastor, J., Woodward, S. L., and Yates, J. Bandwidth on demand for inter-data center communication. In *Proceedings of the 10th ACM Workshop on Hot Topics in Networks* (2011), ACM, p. 24.

[189] Mandal, S. Lessons learned from B4, Google's SDN WAN (accessed Feb. 2023). `https://www.usenix.org/sites/default/files/conference/protected-files/atc15_slides_mandal.pdf`, 2015.

[190] Mani, S. K., Nance Hall, M., Durairajan, R., and Barford, P. Characteristics of metro fiber deployments in the US. In *Proceedings of the Network Traffic Measurement and Analysis Conference* (June 2020).

[191] Marconett, D., and Yoo, S. FlowBroker: Market-driven multi-domain SDN with heterogeneous brokers. In *Optical Fiber Communication Conference* (2015), Optical Society of America, pp. Th2A–36.

[192] Marom, D. M., Colbourne, P. D., D'errico, A., Fontaine, N. K., Ikuma, Y., Proietti, R., Zong, L., Rivas-Moscoso, J. M., and Tomkos, I. Survey of photonic switching architectures and technologies in support of spatially and spectrally flexible optical networking. *IEEE/OSA Journal of Optical Communications and Networking 9*, 1 (2017), 1–26.

[193] Mas, C., Tomkos, I., and Tonguz, O. K. Failure location algorithm for transparent optical networks. *IEEE Journal on Selected Areas in Communications 23*, 8 (2005), 1508–1519.

[194] Matsumoto, T., Kurahashi, T., Konoike, R., Tanizawa, K., Suzuki, K., Uetake, A., Takabayashi, K., Ikeda, K., Kawashima, H., Akiyama, S., and Sekiguchi, S. In-line optical amplification for silicon photonics platform by flip-chip bonded InP-SOAs. In *2018 Optical Fiber Communications Conference and Exposition (OFC)* (2018), pp. 1–3.

[195] 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 38*, 2 (2008), 69–74.

[196] Meier, R., Tsankov, P., Lenders, V., Vanbever, L., and Vechev, M. NetHide: Secure and practical network topology obfuscation. In *27th USENIX Security Symposium (USENIX Security 18)* (2018), pp. 693–709.

[197] Mellette, W. M., Das, R., Guo, Y., McGuinness, R., Snoeren, A. C., and Porter, G. Expanding across time to deliver bandwidth efficiency and low latency. *CoRR abs/1903.12307* (2019).

[198] Mellette, W. M., McGuinness, R., Roy, A., Forencich, A., Papen, G., Snoeren, A. C., and Porter, G. Rotornet: A scalable, low-complexity, optical datacenter network. In *ACM SIGCOMM* (2017), ACM, pp. 267–280.

[199] Mellette, W. M., Schuster, G. M., Porter, G., Papen, G., and Ford, J. E. A scalable, partially configurable optical switch for data center networks. *Journal of Lightwave Technology 35*, 2 (Jan 2017), 136–144.

[200] Michel, O., and Keller, E. SDN in wide-area networks: A survey. In *2017 Fourth International Conference on Software Defined Systems (SDS)* (2017), pp. 37–42.

[201] Misra, J., and Gries, D. A constructive proof of Vizing's theorem. *Inf. Process. Lett. 41*, 3 (1992), 131–133.

[202] Mitre. Network denial of service: Direct network flood (accessed Mar. 2024). `https://attack.mitre.org/versions/v10/techniques/T1498/001/`, 2022.

[203] Monsanto, C., Reich, J., Foster, N., Rexford, J., and Walker, D. Composing software defined networks. In *10th USENIX Symposium on Networked Systems Design and Implementation* (2013).

[204] Morea, A., and Paparella, A. Cost and algorithm complexity of elastic optical networks. In *Optical Fiber Communication Conference* (2016), Optical Society of America, p. M2K.4.

[205] Moura, U., Garrich, M., Carvalho, H., Svolenski, M., Andrade, A., Cesar, A. C., Oliveira, J., and Conforti, E. Cognitive methodology for optical amplifier gain adjustment in dynamic DWDM networks. *Journal of Lightwave Technology 34*, 8 (2016), 1971–1979.

[206] Moura, U., Garrich, M., Cesar, A. C., Oliveira, J., and Conforti, E. Execution time improvement for optical amplifier cognitive methodology in dynamic WDM networks. In *2017 SBMO/IEEE MTT-S International Microwave and Optoelectronics Conference (IMOC)* (2017), IEEE, pp. 1–5.

[207] Mukerjee, M. K., Canel, C., Wang, W., Kim, D., Seshan, S., and Snoeren, A. C. Adapting TCP for reconfigurable datacenter networks. In *17th USENIX Symposium on Networked Systems Design and Implementation (NSDI 20)* (2020), pp. 651–666.

[208] Müller-Hannemann, M., and Schwartz, A. Implementing weighted b-matching algorithms: Insights from a computational study. *ACM Journal of Experimental Algorithmics 5* (2000), 8.

[209] Nance-Hall, M. OTP simulator (accessed Feb. 2024). `https://github.com/mattall/topology-programming`, 2024.

[210] Nance-Hall, M., Barford, P., Foerster, K.-T., and Durairajan, R. Improving scalability in traffic engineering via optical topology programming. *IEEE Transactions on Network and Service Management* (2023).

[211] Nance-Hall, M., Barford, P., Foerster, K.-T., Ghobadi, M., Jensen, W., and Durairajan, R. Are WANs ready for optical topology programming? In *Proceedings of the ACM SIGCOMM 2021 Workshop on Optical Systems* (New York, NY, USA, 2021), OptSys '21, Association for Computing Machinery, p. 28–33.

[212] Nance-Hall, M., and Durairajan, R. Bridging the optical-packet network chasm via secure enclaves, 2020.

[213] Nance-Hall, M., Durairajan, R., and Sekar, V. Fighting fire with light: A case for defending DDoS attacks using the optical layer. *CoRR abs/2002.10009* (2020).

[214] Nance-Hall, M., Foerster, K., Schmid, S., and Durairajan, R. A survey of reconfigurable optical networks. *Opt. Switch. Netw. 41* (2021), 100621.

[215] Nance-Hall, M., Foerster, K.-T., Schmid, S., and Durairajan, R. A survey of reconfigurable optical networks. *Optical Switching and Networking 41* (2021), 100621.

[216] Nance-Hall, M., Liu, G., Durairajan, R., and Sekar, V. Fighting fire with light: Tackling extreme terabit DDoS using programmable optics. In *Proceedings of the Workshop on Secure Programmable Network Infrastructure* (New York, NY, USA, 2020), SPIN '20, Association for Computing Machinery, p. 42–48.

[217] Nance-Hall, M., Liu, Z., Sekar, V., and Durairajan, R. Analyzing the benefits of optical topology programming for mitigating link-flood DDoS attacks. *(To appear) IEEE Transactions on Dependable and Secure Computing* (2024).

[218] Nance-Hall, M., Salamatian, L., and Durairajan, R. From fibers to fortresses: Combating modern reconnaissance via optical topology programming. *in submission* (2024).

[219] Neghabi, A. A., Navimipour, N. J., Hosseinzadeh, M., and Rezaee, A. Load balancing mechanisms in the software defined networks: a systematic and comprehensive review of the literature. *IEEE access 6* (2018), 14159–14178.

[220] NETSCOUT. Issue 8: Findings from 2nd half 2021. *Netscout Systems, Inc., Tech. Rep* (2021).

[221] Nguyen, H. X., Trestian, R., To, D., and Tatipamula, M. Digital twin for 5G and beyond. *IEEE Communications Magazine 59*, 2 (2021), 10–15.

[222] Oda, S., Miyabe, M., Yoshida, S., Katagiri, T., Aoki, Y., Rasmussen, J. C., Birk, M., and Tse, K. Demonstration of an autonomous software controlled living optical network that eliminates the need for pre-planning. In *Optical Fiber Communication Conference* (2016), Optical Society of America, p. W2A.44.

[223] Oda, S., Miyabe, M., Yoshida, S., Katagiri, T., Aoki, Y., Rasmussen, J. C., Birk, M., and Tse, K. A learning living network for open ROADM networks. In *ECOC 2016; 42nd European Conference on Optical Communication* (2016), VDE, pp. 1–3.

[224] Oliveira, J. R., Caballero, A., Magalhães, E., Moura, U., Borkowski, R., Curiel, G., Hirata, A., Hecker, L., Porto, E., Zibar, D., et al. Demonstration of EDFA cognitive gain control via GMPLS for mixed modulation formats in heterogeneous optical networks. In *Optical Fiber Communication Conference* (2013), Optical Society of America, pp. OW1H–2.

[225] Ollivier, Y. Ricci curvature of markov chains on metric spaces. *Journal of Functional Analysis 256*, 3 (2009), 810–864.

[226] OpenConfig. Vendor-neutral, model-driven network management designed by users (accessed Feb. 2023). `http://www.openconfig.net/`, 2016.

[227] Ozdaglar, A. E., and Bertsekas, D. P. Routing and wavelength assignment in optical networks. *IEEE/ACM Transactions on Networking 11*, 2 (2003), 259–272.

[228] Pan, P., Swallow, G., and Atlas, A. RFC 4090: Fast reroute extensions to RSVP-TE for LSP tunnels. `http://www.ietf.org/rfc/rfc4090.txt`, 2005.

[229] Papanikolaou, P., Christodoulopoulos, K., and Varvarigos, E. Joint multilayer planning of survivable elastic optical networks. In *Optical Fiber Communication Conference* (2016), Optical Society of America, pp. M2K–3.

[230] Patel, P., Bansal, D., Yuan, L., Murthy, A., Greenberg, A., Maltz, D. A., Kern, R., Kumar, H., Zikos, M., Wu, H., Kim, C., and Karri, N. Ananta: cloud scale load balancing. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM* (New York, NY, USA, 2013), SIGCOMM '13, Association for Computing Machinery, p. 207–218.

[231] Patri, S. K., Autenrieth, A., Rafique, D., Elbers, J.-P., and Machuca, C. M. HeCSON: Heuristic for configuration selection in optical network planning. In *Optical Fiber Communication Conference (OFC) 2020* (2020), Optical Society of America, p. Th2A.32.

[232] Peres, B., de Oliveira Souza, O. A., Goussevskaia, O., Avin, C., and Schmid, S. Distributed self-adjusting tree networks. In *INFOCOM* (2019), IEEE.

[233] Peres, B., Goussevskaia, O., Schmid, S., and Avin, C. Concurrent self-adjusting distributed tree networks. In *Proc. International Symposium on Distributed Computing (DISC)* (2017).

[234] Porter, G., Strong, R. D., Farrington, N., Forencich, A., Sun, P., Rosing, T., Fainman, Y., Papen, G., and Vahdat, A. Integrating microsecond circuit switching into the data center. In *ACM SIGCOMM* (2013), D. M. Chiu, J. Wang, P. Barford, and S. Seshan, Eds., ACM, pp. 447–458.

[235] Poutievski, L., Mashayekhi, O., Ong, J., Singh, A., Tariq, M., Wang, R., Zhang, J., Beauregard, V., Conner, P., Gribble, S., Kapoor, R., Kratzer, S., Li, N., Liu, H., Nagaraj, K., Ornstein, J., Sawhney, S., Urata, R., Vicisano, L., Yasumura, K., Zhang, S., Zhou, J., and Vahdat, A. Jupiter evolving: Transforming Google's datacenter network via optical circuit switches and software-defined networking. In *Proceedings of ACM SIGCOMM 2022* (2022).

[236] Richter, P., Allman, M., Bush, R., and Paxson, V. A primer on ipv4 scarcity. *Comput. Commun. Rev. 45*, 2 (2015), 21–31.

[237] Russell, J. The world's largest DDoS attack took GitHub offline for fewer than 10 minutes (accessed Mar. 2024). `https://techcrunch.com/2018/03/02/the-worlds-largest-ddos-attack-took-github-offline-for-less-than-tens-minutes/`, 2018.

[238] Salamatian, L., Anderson, S., Matthews, J., Barford, P., Willinger, W., and Crovella, M. Curvature-based analysis of network connectivity in private backbone infrastructures. *Proceedings of the ACM Measurement and Analysis of Computing Systems 6*, 1 (2022), 5:1–5:32.

[239] Salamatian, L., Arnold, T., Cunha, Í., Zhu, J., Zhang, Y., Katz-Bassett, E., and Calder, M. Who squats IPv4 addresses? *Comput. Commun. Rev. 53*, 1 (2023), 48–72.

[240] Salman, S., Streiffer, C., Chen, H., Benson, T., and Kadav, A. DeepConf: Automating data center network topologies management with machine learning. In *Proceedings of the 2018 Workshop on Network Meets AI & ML* (New York, NY, USA, 2018), NetAI'18, ACM, pp. 8–14.

[241] Sambo, N., Castoldi, P., D'Errico, A., Riccardi, E., Pagano, A., Moreolo, M. S., Fabrega, J. M., Rafique, D., Napoli, A., Frigerio, S., et al. Next generation sliceable bandwidth variable transponders. *IEEE Communications Magazine 53*, 2 (2015), 163–171.

[242] Schmid, S., Avin, C., Scheideler, C., Borokhovich, M., Haeupler, B., and Lotker, Z. Splaynet: Towards locally self-adjusting networks. *IEEE/ACM Trans. Netw. 24*, 3 (2016), 1421–1433.

[243] Schwartz, R., Singh, M., and Yazdanbod, S. Online and offline greedy algorithms for routing with switching costs. *arXiv preprint arXiv:1905.02800* (2019).

[244] Shah, A. Google AI supercomputer shows the potential of optical interconnects (accessed Feb. 2024). `https://www.hpcwire.com/2023/04/10/google-ai-supercomputer-shows-the-potential-of-optical-interconnects/`, April 2023.

[245] Shakeri, A., Garrich, M., Bravalheri, A., Careglio, D., Solé-Pareta, J., and Fumagalli, A. Traffic allocation strategies in WSS-based dynamic optical networks. *Journal of Optical Communications and Networking 9*, 4 (2017), B112–B123.

[246] Shand, M., and Bryant, S. RFC 5714: IP fast reroute framework. `https://www.ietf.org/rfc/rfc5714.txt`, 2010.

[247] Shen, Y., Goodfellow, R., Glick, M. S., Bartlett, G., and Bergman, K. Optical mitigation of DDoS attacks using silicon photonic switches. In *Metro and Data Center Optical Networks and Short-Reach Links III* (2020), A. K. Srivastava, M. Glick, and Y. Akasaka, Eds., vol. 11308, International Society for Optics and Photonics, SPIE, pp. 111 – 117.

[248] Shieh, A., Kandula, S., Greenberg, A. G., and Kim, C. Seawall: Performance isolation for cloud datacenter networks. In *HotCloud* (2010).

[249] Shin, J., Sirer, E. G., Weatherspoon, H., and Kirovski, D. On the feasibility of completely wirelesss datacenters. *IEEE/ACM Trans. Netw. 21*, 5 (2013), 1666–1679.

[250] Singh, R., Bjorner, N., Shoham, S., Yin, Y., Arnold, J., and Gaudette, J. Cost-effective capacity provisioning in wide area networks with Shoofly. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference* (2021), pp. 534–546.

[251] Singh, R., Ghobadi, M., Foerster, K., Filer, M., and Gill, P. Run, walk, crawl: Towards dynamic link capacities. In *HotNets* (2017), ACM.

[252] Singh, R., Ghobadi, M., Foerster, K.-T., Filer, M., and Gill, P. RADWAN: Rate adaptive wide area network. In *ACM SIGCOMM* (New York, NY, USA, 2018), SIGCOMM '18, ACM, ACM, pp. 547–560.

[253] Singh, R., Ghobadi, M., Förster, K.-T., Filer, M., and Gill, P. Run, walk, crawl: Towards dynamic link capacities. In *Proceedings of the 16th ACM HotNets 2017* (New York, NY, USA, 2017), HotNets-XVI, ACM, pp. 143–149.

[254] Singla, A., Hong, C., Popa, L., and Godfrey, P. B. Jellyfish: Networking data centers randomly. In *USENIX NSDI* (2012), USENIX.

[255] Singla, A., Singh, A., Ramachandran, K., Xu, L., and Zhang, Y. Proteus: A topology malleable data center network. In *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks* (2010), ACM, ACM, p. 8.

[256] Skoog, R. A., and Neidhardt, A. L. A fast, robust signaling protocol for enabling highly dynamic optical networks. In *2009 Conference on Optical Fiber Communication-incudes post deadline papers* (2009), IEEE, pp. 1–3.

[257] Smith, J. M., and Schuchard, M. Routing around congestion: Defeating DDoS attacks and adverse network conditions via reactive BGP routing. In *2018 IEEE Symposium on Security and Privacy (SP)* (2018), IEEE, pp. 599–617.

[258] Soto, P., Maya, P., and Botero, J. F. Resource allocation over EON-based infrastructures in a network virtualization environment. *IEEE Transactions on Network and Service Management 16*, 1 (2019), 13–26.

[259] Stone, R. CenterTrack: An IP overlay network for tracking DoS floods. In *9th USENIX Security Symposium (USENIX Security 00)* (2000), vol. 21, p. 114.

[260] Studer, A., and Perrig, A. The Coremelt attack. In *Computer Security - ESORICS 2009, 14th European Symposium on Research in Computer Security, Saint-Malo, France, September 21-23, 2009. Proceedings* (New York, New York, USA, 2009), M. Backes and P. Ning, Eds., vol. 5789 of *Lecture Notes in Computer Science*, Springer, pp. 37–52.

[261] Sun, L., Chen, X., and Zhu, Z. Multibroker-based service provisioning in multidomain SD-EONs: Why and how should the brokers cooperate with each other? *IEEE Journal of Lightwave Technology* (2017).

[262] Sun, X. S., and Ng, T. S. E. When creek meets river: Exploiting high-bandwidth circuit switch in scheduling multicast data. In *ICNP* (2017), IEEE Computer Society, pp. 1–6.

[263] Sun, X. S., Xia, Y., Dzinamarira, S., Huang, X. S., Wu, D., and Ng, T. S. E. Republic: Data multicast meets hybrid rack-level interconnections in data center. In *ICNP* (2018), IEEE Computer Society, pp. 77–87.

[264] Szymkiewicz, D. Une conlribution statistique à la géographie floristique. *Acta Societatis Botanicorum Poloniae 11*, 3 (1934), 249–265.

[265] Taboada, J. M., Maki, J. J., Tang, S., Sun, L., An, D., Lu, X., and Chen, R. T. Thermo-optically tuned cascaded polymer waveguide taps. *Applied physics letters 75*, 2 (1999), 163–165.

[266] Team, A. N. S. 2022 in review: DDoS attack trends and insights. https://www.microsoft.com/en-us/security/blog/2023/02/21/2022-in-review-ddos-attack-trends-and-insights/ (Accessed Feb. 2023), February 2023.

[267] Teh, M. Y., Hung, Y.-H., Michelogiannakis, G., Yan, S., Glick, M., Shalf, J., and Bergman, K. TAGO: Rethinking routing design in high performance reconfigurable networks. In *Proc. Int. Conf. for High Perform. Comput., Netw., Storage and Anal.* (2020), SC '20, IEEE Press.

[268] Teh, M. Y., Zhao, S., and Bergman, K. METTEOR: Robust multi-traffic topology engineering for commercial data center networks. *CoRR abs/2002.00473* (2020).

[269] Teh, M. Y., Zhao, S., Cao, P., and Bergman, K. COUDER: Robust topology engineering for optical circuit switched data center networks. *CoRR abs/2010.00090* (2020).

[270] TeleGeography. Marea submarine cable. https://www.submarinecablemap.com/submarine-cable/marea, 2024.

[271] Terzi, C., and Korpeoglu, I. 60 GHz wireless data center networks: A survey. *Computer Networks 185* (2021), 107730.

[272] Thomson, D., Zilkie, A., Bowers, J. E., Komljenovic, T., Reed, G. T., Vivien, L., Marris-Morini, D., Cassan, E., Virot, L., Fédéli, J.-M., et al. Roadmap on silicon photonics. *Journal of Optics 18*, 7 (2016), 073003.

[273] Thyagaturu, A. S., Mercian, A., McGarry, M. P., Reisslein, M., and Kellerer, W. Software defined optical networks (SDONs): A comprehensive survey. *IEEE Commun. Surv. Tutorials 18*, 4 (2016), 2738–2786.

[274] Toh, A. Azure DDoS protection—2021 Q3 and Q4 DDoS attack trends. https://azure.microsoft.com/en-us/blog/azure-ddos-protection-2021-q3-and-q4-ddos-attack-trends/, 2021.

[275] Toshiyoshi, H., and Fujita, H. Electrostatic micro torsion mirrors for an optical switch matrix. *Journal of Microelectromechanical systems 5*, 4 (1996), 231–237.

[276] Trichili, A., Cox, M. A., Ooi, B. S., and Alouini, M.-S. Roadmap to free space optics. *Journal of the Optical Society of America B 37*, 11 (Nov 2020), A184–A201.

[277] Truong-Huu, T., Mohan, P. M., and Gurusamy, M. Virtual network embedding in ring optical data centers using markov chain probability model. *IEEE Transactions on Network and Service Management 16*, 4 (2019), 1724–1738.

[278] Tsai, J., and Wu, M. C. A high port-count wavelength-selective switch using a large scan-angle, high fill-factor, two-axis mems scanner array. *IEEE Photonics Technology Letters 18*, 13 (2006), 1439–1441.

[279] Tsai, P.-W., Tsai, C.-W., Hsu, C.-W., and Yang, C.-S. Network monitoring in software-defined networking: A review. *IEEE Systems Journal 12*, 4 (2018), 3958–3969.

[280] Tseng, S. Perseverance-aware traffic engineering in rate-adaptive networks with reconfiguration delay. In *ICNP* (2019), IEEE, pp. 1–10.

[281] Tsirilakis, I., Mas, C., and Tomkos, I. Cost comparison of IP/WDM vs. IP/OTN for european backbone networks. In *Proceedings of 2005 7th International Conference Transparent Optical Networks, 2005.* (2005), vol. 2, IEEE, pp. 46–49.

[282] Tunable DWDM transceivers (accessed Feb. 2023). `https://ii-vi.com/product/400g-zr-qsfp-dd-dco-high-tx-output-power-optical-transceiver/`, 2023.

[283] Turkcu, O., and Subramaniam, S. Performance of optical networks with limited reconfigurability. *IEEE/ACM Trans. Netw. 17*, 6 (Dec. 2009), 2002–2013.

[284] ur Rasool, R., Wang, H., Ashraf, U., Ahmed, K., Anwar, Z., and Rafique, W. A survey of link flooding attacks in software defined network ecosystems. *Journal of Network and Computer Applications 172* (2020), 102803.

[285] Valiant, L. G. A scheme for fast parallel communication. *SIAM J. Comput. 11*, 2 (1982), 350–361.

[286] Venkatakrishnan, S. B., Alizadeh, M., and Viswanath, P. Costly circuits, submodular schedules and approximate carathéodory theorems. In *SIGMETRICS* (2016), ACM, pp. 75–88.

[287] Wang, G., Andersen, D. G., Kaminsky, M., Kozuch, M., Ng, T. S. E., Papagiannaki, K., Glick, M., and Mummert, L. B. Your data center is a router: The case for reconfigurable optical circuit switched paths. In *HotNets* (2009), ACM SIGCOMM.

[288] Wang, G., Andersen, D. G., Kaminsky, M., Papagiannaki, K., Ng, T. E., Kozuch, M., and Ryan, M. c-Through: Part-time optics in data centers. In *SIGCOMM* (2010), ACM, pp. 327–338.

[289] Wang, H., Wu, Y., Min, G., and Miao, W. A graph neural network-based digital twin for network slicing management. *IEEE Transactions on Industrial Informatics 18*, 2 (2022), 1367–1376.

[290] Wang, H., Xia, Y., Bergman, K., Ng, T. S. E., Sahu, S., and Sripanidkulchai, K. Rethinking the physical layer of data center networks of the next decade: using optics to enable efficient *-cast connectivity. *Computer Communication Review 43*, 3 (2013), 52–58.

[291] Wang, H., Yu, X., Xu, H., Fan, J., Qiao, C., and Huang, L. Integrating coflow and circuit scheduling for optical networks. *IEEE Transactions on Parallel and Distributed Systems* (2019).

[292] Wang, M., Cui, Y., Xiao, S., Wang, X., Yang, D., Chen, K., and Zhu, J. Neural network meets DCN: traffic-driven topology adaptation with deep learning. *POMACS 2*, 2 (2018), 26:1–26:25.

[293] Wang, M., Zong, L., Mao, L., Marquez, A., Ye, Y., Zhao, H., and Vaquero Caballero, F. J. LCoS SLM study and its application in wavelength selective switch. *Photonics 4*, 2 (2017).

[294] Wang, Y., McNulty, Z., and Nguyen, H. Network virtualization in spectrum sliced elastic optical path networks. *Journal of Lightwave Technology 35*, 10 (2017), 1962–1970.

[295] Warburton, D., Ojeda, E., and Heath, M. 2022 application protection report: DDoS attack trends. `https://www.f5.com/labs/articles/threat-intelligence/2022-application-protection-report-ddos-attack-trends`, 2022. F5.

[296] Williams, K., Liu, X., Matters-Kammerer, M., Meighan, A., Spiegelberg, M., van der Tol, J., Trajkovic, M., Wale, M., Yao, W., and Zhang, X. Indium phosphide photonic circuits on silicon electronics. In *Optical Fiber Communication Conference (OFC) 2020* (2020), Optical Society of America, p. M3A.1.

[297] Woodward, S. L., Feuer, M. D., Kim, I., Palacharla, P., Wang, X., and Bihon, D. Service velocity: Rapid provisioning strategies in optical roadm networks. *Journal of Optical Communications and Networking 4*, 2 (2012), 92–98.

[298] Xia, W., Wen, Y., Foh, C. H., Niyato, D., and Xie, H. A survey on software-defined networking. *IEEE Communications Surveys & Tutorials 17*, 1 (2014), 27–51.

[299] Xia, Y., Ng, T. S. E., and Sun, X. S. Blast: Accelerating high-performance data analytics applications by optical multicast. In *INFOCOM* (2015), IEEE, pp. 1930–1938.

[300] Xia, Y., Sun, X. S., Dzinamarira, S., Wu, D., Huang, X. S., and Eugene Ng, T. S. A tale of two topologies: Exploring convertible data center network architectures with flat-tree. In *SIGCOMM* (2017), ACM.

[301] Xing, J., Wu, W., and Chen, A. Ripple: A programmable, decentralized link-flooding defense against adaptive adversaries. In *30th USENIX Security Symposium (USENIX Security 21)* (Vancouver, B.C., Aug. 2021), USENIX Association.

[302] Xiong, Y., Li, Y., Zhou, B., Wang, R., and Rouskas, G. N. SDN enabled restoration with triggered precomputation in elastic optical inter-datacenter networks. *Journal of Optical Communications and Networking* (2018).

[303] Xu, D., Li, G., Ramamurthy, B., Chiu, A. L., Wang, D., and Doverspike, R. D. On provisioning diverse circuits in heterogeneous multi-layer optical networks. *Comput. Commun. 36*, 6 (2013), 689–697.

[304] Xu, X., Chen, H., Simsarian, J. E., Ryf, R., Mazur, M., Dallachiesa, L., Fontaine, N. K., and Neilson, D. T. Optical network diagnostics using graph neural networks and natural language processing. In *Optical Fiber Communication Conference (OFC) 2023* (2023), Optica Publishing Group, p. M3G.5.

[305] Yaar, A., Perrig, A., and Song, D. StackPi: New packet marking and filtering mechanisms for DDoS and IP spoofing defense. *IEEE Journal on Selected Areas in Communications 24*, 10 (2006), 1853–1863.

[306] Yang, H., Robertson, B., Wilkinson, P., and Chu, D. Small phase pattern 2D beam steering and a single LCOS design of 40 1×12 stacked wavelength selective switches. *Opt. Express 24*, 11 (May 2016), 12240–12253.

[307] Yang, M., Rastegarfar, H., and Djordjevic, I. B. Physical-layer adaptive resource allocation in software-defined data center networks. *Journal of Optical Communications and Networking 10*, 12 (2018), 1015–1026.

[308] Yang, Z., Cui, Y., Li, B., Liu, Y., and Xu, Y. Software-defined wide area network (SD-WAN): Architecture, advances and opportunities. In *International Conference on Computer Communication and Networks* (2019), IEEE, pp. 1–9.

[309] Yeniay, A., Gao, R., Takayama, K., Gao, R., and Garito, A. F. Ultra-low-loss polymer waveguides. *Journal of lightwave technology 22*, 1 (2004).

[310] Yoachimik, O. Cloudflare DDoS threat report for 2022 Q4. https://blog.cloudflare.com/ddos-threat-report-2022-q4/, 2022.

[311] Yoachimik, O. Cloudflare mitigates 26 million request per second DDoS attack. https://blog.cloudflare.com/26m-rps-ddos/ (Accessed Feb. 2023), June 2022.

[312] Yoachimik, O. DDoS attack trends for 2022 Q2. https://blog.cloudflare.com/ddos-attack-trends-for-2022-q2/, 2022.

[313] Yoo, S. Multi-domain cognitive optical software defined networks with market-driven brokers. In *European Conference on Optical Communication* (2014), IEEE, pp. 1–3.

[314] Zagar, D., and Grgic, K. IPv6 security threats and possible solutions. In *2006 World Automation Congress* (2006), IEEE, pp. 1–7.

[315] 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 1*, 1 (2000), 47–60.

[316] Zargar, S. T., Joshi, J., and Tipper, D. A survey of defense mechanisms against distributed denial of service (DDoS) flooding attacks. *IEEE communications surveys & tutorials 15*, 4 (2013), 2046–2069.

[317] Zhang, M., Li, G., Wang, S., Liu, C., Chen, A., Hu, H., Gu, G., Li, Q., Xu, M., and Wu, J. Poseidon: Mitigating volumetric DDoS attacks with programmable switches. In *the 27th Network and Distributed System Security Symposium (NDSS 2020)* (2020).

[318] Zhang, W., and Bathula, B. G. Breaking the bidirectional link paradigm. In *Optical Fiber Communication Conference* (2016), Optical Society of America.

[319] Zhang, X. J., Kim, S., and Lumetta, S. S. Opportunity cost analysis for dynamic wavelength routed mesh networks. *IEEE/ACM Transactions on Networking 19*, 3 (2011), 747–759.

[320] Zhang, Z., You, Z., and Chu, D. Fundamentals of phase-only liquid crystal on silicon (lcos) devices. *Light: Science & Applications 3*, 10 (2014), e213.

[321] Zhong, Z., Ghobadi, M., Balandat, M., Katti, S., Kazerouni, A., Leach, J., McKillop, M., and Zhang, Y. BOW: First real-world demonstration of a firewall-based bayesian optimization system for wavelength deployment. In *Optical Fiber Communications Conference* (2021), IEEE.

[322] Zhong, Z., Ghobadi, M., Khaddaj, A., Leach, J., Xia, Y., and Zhang, Y. ARROW: Restoration-aware traffic engineering. In *ACM SIGCOMM* (2021), pp. 560–579.

[323] Zhong, Z., Hua, N., Tornatore, M., Li, J., Li, Y., Zheng, X., and Mukherjee, B. Provisioning short-term traffic fluctuations in elastic optical networks. *IEEE/ACM Trans. Netw. 27*, 4 (2019), 1460–1473.

[324] Zhou, X., Zhang, Z., Zhu, Y., Li, Y., Kumar, S., Vahdat, A., Zhao, B. Y., and Zheng, H. Mirror mirror on the ceiling: flexible wireless links for data centers. In *ACM SIGCOMM* (2012), ACM, pp. 443–454.

[325] Zhu, P., Li, J., Wu, D., Wu, Z., Tian, Y., Chen, Y., Ge, D., Chen, X., Chen, Z., and He, Y. Demonstration of elastic optical network node with defragmentation functionality and SDN control. In *Optical Fiber Communication Conference* (2016), Optical Society of America, p. Th3I.3.

APPENDIX

A.1 Lab Hardware Description

Our experimental testbed is shown in Figure A.68. The testbed is symmetric with two simple fiber paths; all of the experiments reported below utilize a single path from West to East. We employ two types of transponders in our testbed; one pair of Advanced Optical Transport Network Line Modules (AOLMs) (Infinera AOLM-500-T4-1-C6), and two pairs of Digital Line Modules (Infinera DLM-n-C2). Throughout our experiments, all transponders send/receive streams of empty Optical Data Units at 100 Gbps. Each transponder sends ODUs on ten individual wavelengths called an Optical Carrier Group (OCG). Signals in an OCG are spaced at 200 GHz. OCG properties are summarized in Table A.9.

Together, the transponders provide capacity to light up to 30 wavelengths in each direction in our testbed. The transponders are connected to a series of Bandwidth Multiplexing Modules (BMMs) (Infinera BMM2-4-CX2-MS-A cards in separate DTC-A chassis), which can optically multiplex up to 40 wavelengths (channels) onto fibers (organized in OCGs). The BMMs are also equipped with two Erbium Doped Fiber

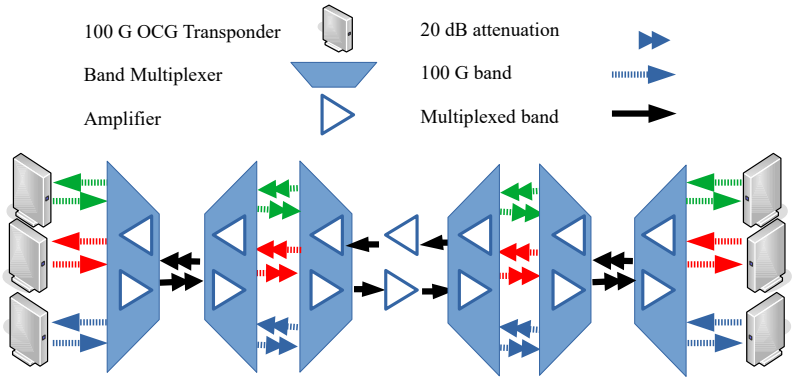


Figure A.68. Configuration used in our lab-based experiments: six optical transponders, each of which generate 100 Gbps of Optical Data Unit (ODU) traffic over seven amplifiers.

Amplifiers (EDFAs) (one in each direction) with an operating range of 20 to 27.5 dB. The BMMs are connected via one-meter fiber jumpers, attenuated at 20 dB to simulate fiber loss from a span of 80 km. The third BMM in the series is connected to a 100 km span of single-mode fiber, and then to an amplifier (Infinera OAM-CXH2-MS in an OTC-1 amplifier chassis), which is used to regenerate signals on long haul paths. The path beyond the OAM is symmetrical to the path leading to it.

OCG	Range (THz)	Range (nm)	Modulation
1	191.75 - 193.55	1563.45 - 1548.91	DP-QPSK
3	191.85 - 193.65	1562.64 - 1548.12	OOK
5	193.95 - 195.75	1545.72 - 1531.51	OOK

Table A.9. Optical Carrier Group (OCG) wavelength ranges and modulations used in our experiments.

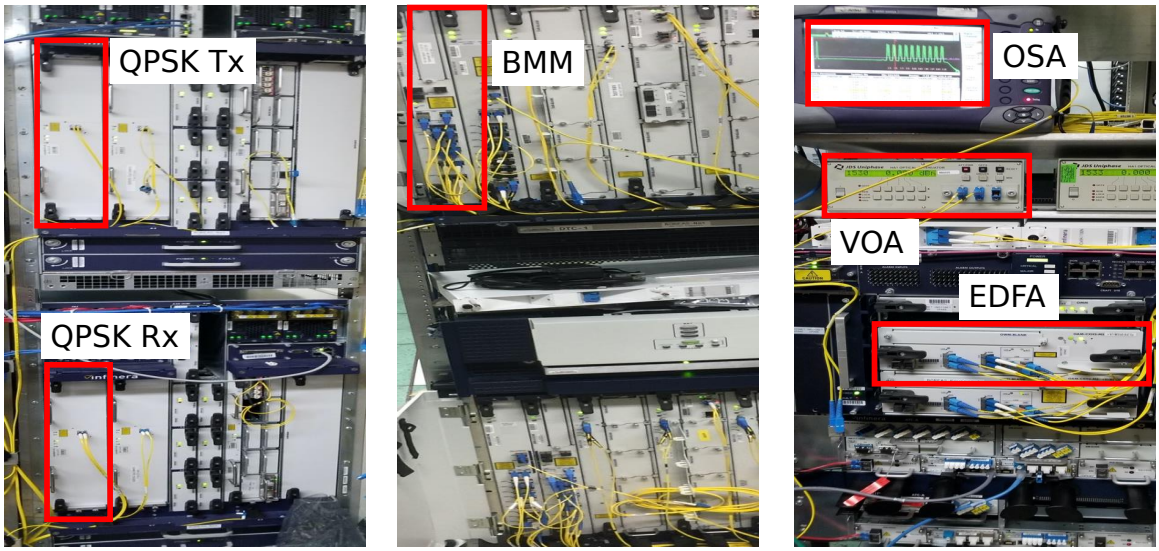


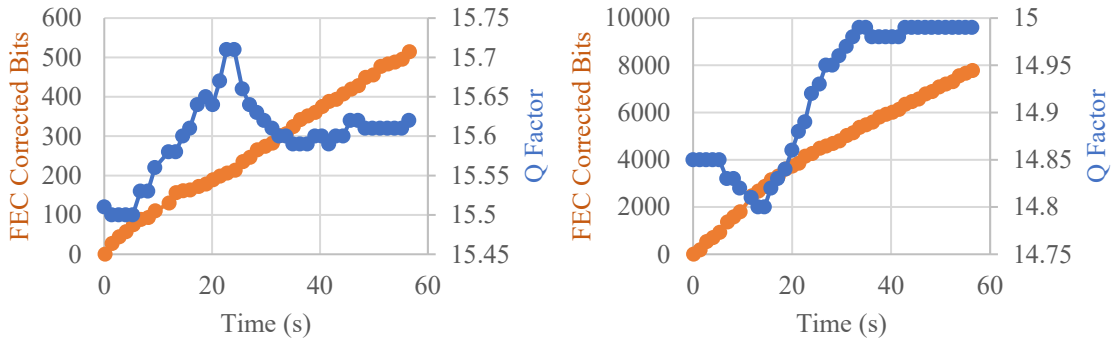
Figure A.69. 100 Gbps QPSK transponders (left), band multiplexer (center), optical spectrum analyzer, variable optical attenuator, and erbium doped fiber amplifiers (right).

The equipment used in our lab is shown in figures A.69. The optical equipment we use in our experiments is representative of equipment that is deployed in operational networks. The BMMs and amplifiers are high power (can transmit 80 to 100 km) and

operate in the C-band (1550 nm frequency). IP routers with suitable transponder interfaces can connect directly to these BMMs. Amplifiers similar to those used in our setup are often arranged in series to enable transmission of signals over hundreds of kilometers.

A.2 Quality of Transmission

Next, we turn our attention to the following fundamental question: what effect does adding or dropping a set of wavelengths have on persistent connections, i.e., those optical frequencies sharing spectrum on a fiber with a dynamic DWDM channel? We call these persistent connections "witnesses" for short because they witness the addition or subtraction of a wave (or set of waves) within the fiber they traverse.



(a) OCG 3 (b) OCG 5

Figure A.70. QoT measurements for witness waves while adding/dropping OCG1. During the add/drop, Q factor for the witness waves is relatively constant—varying by +/- 0.1. Errors accumulate at a linear rate as expected in a live transport network; 100% are corrected with FEC while running traffic over OCGs 3 and 5.

Figure A.70 shows the Q factor and corrected/uncorrected bits from forward error correction (FEC) for a wave in OCGs 3 and 5; these measurements correspond to those shown in Figure 13b. From figure A.70, we see that, although we add 50% more power to the circuit in the form of a third OCG, the Quality of Transmission (QoT) measures of the witness waves in OCGs 3 and 5 are not impaired. More concretely, the Q factor for the two waves varies *only* by +/- 0.1; FEC corrected

all physical bit errors. To further assess the impact of adding/dropping waves, we installed a Tributary Optical Module 10G (TOM-10G-SR1) (which maps electrical signals to an optical 10 Gbps wave) to run IP perf traffic over a wave in OCG 3. This tool is commonly used for diagnostics/testing of optical WAN circuits. Analysis of the perf traffic over the TOM verifies that no packets were dropped for the witness wave while adding/dropping OCG 1.

We conduct more extensive tests of the impact on QoT for witness waves while adding/dropping random OCGs. In this test, we apply every permutation of the three OCGs on the fiber. We see that adding/removing from the spectrum did not negatively impact any of the witness waves.

Main finding and implication. From these results, we find that adding 100 Gbps of capacity to an optical path does not adversely affect the witness waves on that path. Therefore, we conclude that it is safe to add/drop waves in manual mode to increase the agility of the physical layer via OTP.