Improving Scalability in Traffic Engineering via Optical Topology Programming

Matthew Nance-Hall*, Paul Barford†, Klaus-Tycho Foerster‡, Ramakrishnan Durairajan*
*University of Oregon. †University of Wisconsin—Madison. ‡TU Dortmund.

***Abstract*—We present a novel framework, GreyLambda, to improve the scalability of traffic engineering (TE) systems. TE systems continuously monitor traffic and allocate network resources based on observed demands. The temporal requirement for TE is to have a time-to-solution in five minutes or less. Additionally, traffic allocations have a spatial requirement, which is to enable all traffic to traverse the network without encountering an over-subscribed link. However, the multi-commodity flow-based TE 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 due to changing traffic demands, resulting in oversubscribed links or infeasible solutions. To satisfy both these requirements, we utilize optical topology programming (OTP) to rapidly reconfigure optical wavelengths in critical network paths and provide localized bandwidth scaling and new paths for traffic forwarding. GreyLambda integrates OTP into TE 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 TE systems, SMORE and NCFlow in real-world topologies with challenging traffic and link failure scenarios.**

***Index Terms*—Optical Networks, Wide Area Networks, Control and Data Plane Programmability, Computer Simulation Experiments, Monitoring and Measurements.**

I. 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 [1], [2] to switch stacks [3]–[5] to network interface cards [6]. This has led to the replacement of ad hoc traffic engineering (TE) in wide-area networks (WANs) with software-defined systems [7]–[15], 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 [8], [11], [15]. This time constraint is referred to as the "temporal requirement." Furthermore, flow allocations onto links must not oversubscribe those links within a geographical scope of the network; this is referred to as the "spatial requirement" of the TE system.

In modern TE applications, the approach to satisfying both temporal and spacial requirements is to solve a multi-commodity flow (MCF) problem. At its core, MCF is a polynomial time algorithm that solves for the routing paths and flow allocations among paths in an application-defined optimal way [16]. Although the algorithm is solvable in polynomial time, it 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 [9] and NCFlow [10] have relaxed MCF constraints in order to meet the temporal requirement. For example, SMORE opts for a selection of semi-oblivious paths [17]. These paths, although longer than the shortest routes, deliberately avoid traversing the core directly. By doing so, SMORE mitigates the likelihood of congestion occurring in the most central-links of the network. In NCFlow, the network is divided into multiple connected components. Routing within each component is independently resolved in parallel. Following this step, the remaining demand between clusters is addressed by utilizing the remaining available bandwidth. Our evaluation shows that these relaxed constraints can lead to cases of link oversubscription, where traffic demand on a link exceeds the link's capacity, (and, consequently, throughput drops) 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

Manuscript received February 24, 2023. This paper is an expanded paper from the ACM SIGCOMM 2021 Workshop on Optical Systems (OptSys 2021) held virtually (online) on August 23, 2021. This work was funded in part by the National Science Foundation under Grants CNS-2212590, CNS-2145813, CNS-1703592, CNS-2039146 and SaTC-2132651, and in part by the University of Oregon Doctoral Research Fellowship. This work benefited from access to the University of Oregon high performance computing cluster, Talapas.

Corresponding author: M. Nance-Hall (e-mail: mhall@cs.uoregon.edu).

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 scales 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. The scaling of capacity on two links is possible at nodes that have two transponders that are unused and a neighbor across each link with at least one available transponder. 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 [9] and NCFlow [18], by integrating the two components of GreyLambda and evaluating the results in real-world topologies with challenging traffic and link failure scenarios.

This paper makes the following novel contributions.

- A first-of-its-kind framework, GreyLambda, for scaling TE systems using optical topology programming (OTP).
- Spatial scoping algorithms that is theoretically grounded and that satisfies spatial requirement of TE by identifying specific critical links in a network.
- A fast topology programming mechanism to reduce the wavelength reconfiguration latencies of links identified by the spatial scoping algorithms, achieving the temporal requirements of TE.
- A simulator to demonstrate the efficacy of spatial scoping algorithms and topology programming mechanism, and how they can be paired with different TE systems [1].
- Extensive evaluation using the simulator that demonstrates how OTP can scale two state-of-the-art TE systems and outperform two topology programming techniques even in the face of unsatisfiable demands.

[1]Source code at https://github.com/mattall/topology-programming

II. Background and Motivation

A. 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 [8] 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.) [8], [9], [11], [12], [14], [19]–[22]. 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 [15], [23]. 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 [8], [9], [11], [12], [14], [19], [20]. These efforts have been prompted by the shortfalls of greedy, shortest path routing, for managing inter-datacenter traffic at scale [8] and advances in programmable network monitoring and control software [5], [24], [25]. TE optimization solvers are expected to compute as well as provision traffic paths and flow allocations on those paths approximately once every five minutes [8], [11], [15]. 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 [8]. In light of this, a host of TE systems have been proposed to address the temporal challenge while maintaining high throughput throughout the network [9], [10], [14]. In short, any solution to scale the performance of TE systems should satisfy the temporal requirement.

B. Motivation: 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, the TE optimization runs globally, addressing the spatial requirement in a roundabout way, i.e., by provisioning flow tunnels along edge-disjoint paths [9] or by reserving *headroom* on all network links in case of an unforeseen event [12]. Prior efforts have pointed

out that events typically have a local spatial scope [19], [26], 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 [27] 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 § VI 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 with MCF (SMORE [9]), unrestricted MCF (simply noted denoted at MCF), and NCFlow [10] in Figures 1, 2, and 3.

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 1a, 2a, and 3a, 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 1b, 2b, and 3b, 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 1c, 2c, and 3c 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.

NCFlow trades off a higher throughput guarantee compared to SMORE and MCF in favor of speed. It divides the network into components and solves flow allocations within each component before routing flows across the network in an iterative fashion. As a result, Figures 1d, 2d, and 3d show that the number of congestion loss events on a per link basis are higher than MCF and SMORE, which both take a global view of the network when optimizing routes for traffic. The links that are congested most frequently in NCFlow are the border links between network partitions.

NCFlow and SMORE are the latest of these four TE strategies and therefore we exclude ECMP and MCF from the rest of the paper.

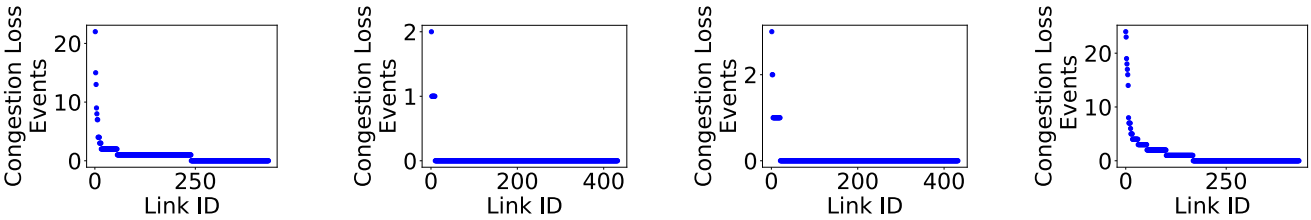
Takeaway: There is a small set of critical network paths that are affected by a diverse set of congestion-causing events (including flash crowds and multi-link failures). Unfortunately, many of the TE solvers run globally i.e. without considering the right scope of the critical paths as part of the spatial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

Limitation 2: Not Considering the Temporal and Spatial Requirements of TE Simulatenously. NCFlow [10] 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 [13] 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 [9] 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 1c, 2c, and 3c 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 [12] and TeaVar [11]. 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.

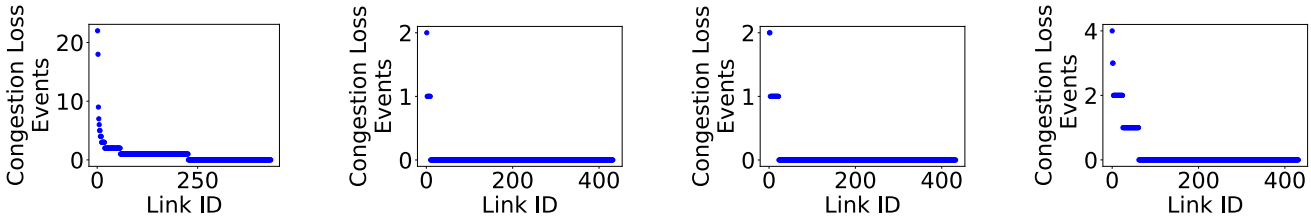
Recently, there has been a promising line of work highlighting packet-optical network co-optimization and topology reconfiguration in response to events such as link failures. For example, Arrow [15] enables partial restoration of lost link capacity by using transponders at the ends of a failed link 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 [15], 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



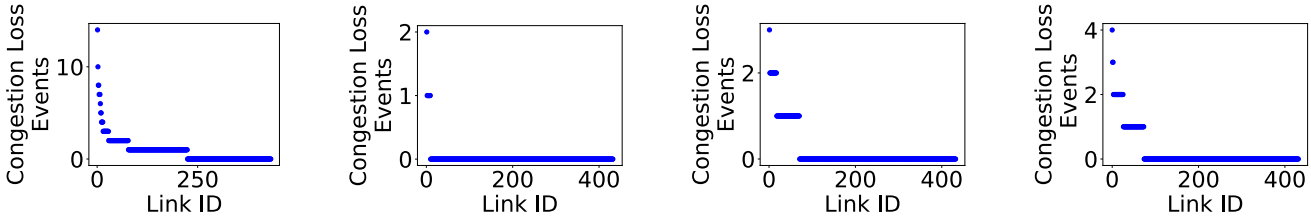
(a) ECMP (b) MCF (c) SMORE (d) NCFlow

Figure 1: Total Congestion Loss events per link in Azure with flash crowds with various TE schemes.



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

Figure 2: 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 (d) NCFlow

Figure 3: Total Congestion Loss events per link in Azure with flash crowds and *two link failures* with various TE schemes.

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.

III. Optical Topology Programming: Opportunity, Challenges, and Design Approach

A. 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. These opportunities increase the flexibility and control that today's SDN control technologies offer when faced with adverse demand and link failure scenarios.

To illustrate these opportunities, Figure 4a shows a simple graph/network with two nodes v, w connected via edges/fiber, with the number of wavelengths per edge indicated. The presence of multiple fiber links between two nodes, as shown in this figure, may come about in a variety of cases; for example, an operator from a local research and enterprise network confirmed to us that many of their IP links have redundant fiber lines, e.g., going along disjoint south-bound and north-bound conduits along freeways. It is also possible to route an optical circuit passively across intermediate nodes between v and w. Figure 4b shows an *optimally resilient* static allocation of three wavelengths in the sense that for any two fiber cuts, as in 4c, 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 4d.

Figure 5a 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 5a, whereas TE+OTP can adapt to the situation and obtain a throughput of 4 as shown in 5b.

B. Challenges

Leveraging OTP to scale TE systems entails three unique challenges:

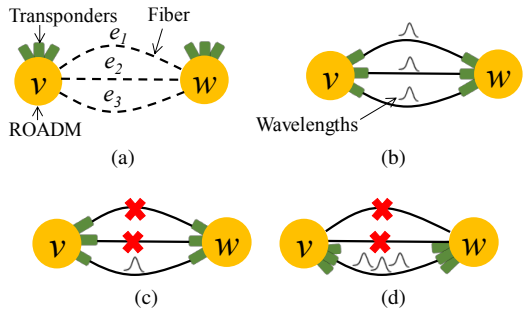
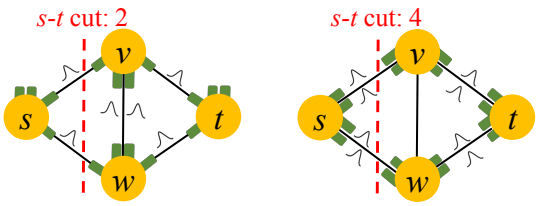


Figure 4: A physical graph with three transponders at every node in (a). The most resilient way to *statically* allocate wavelengths is shown in (b), as two fiber cuts are survivable, as in (c). With OTP, however, we can recover from these two fiber cuts and retain three wavelengths between v, w as in (d).



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

Figure 5: 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.

(**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 § II 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 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 [28]. 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 § V-B). In light of these limitations, packet and optical network innovations have generally occurred independently of each other [29], 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.

C. Design Approach, Novelty, 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 employs a formal model with theoretical guarantees to reduce the scope of TE optimization and identifies certain critical optical layer links (§ IV).

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

(**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 (§ VI).

IV. Reducing the Scope of TE Optimization for Spatial Requirement

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. Here, critical links refer to those where Traffic Engineering (TE) alone cannot ensure a 100% throughput in scenarios involving multiple traffic and link failures. (**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, we prove (§IV-A) that the throughput and utilization gain factor of enabling optical topology programming is between one and $O(\Delta)$, for wavelengths between neighboring nodes, where $\Delta = max_{v \in \mathcal{V}} \delta(v)$ is the maximum node degree $\delta(v)$ 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 § IV-B) 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.

A. Formal Model and Theoretical Guarantees of OTP

As we formally prove tight bounds on the gains of optical topology programming, we first give a precise model setting for the case of regeneration and wavelength conversion at each node. A list of notations used is given in Table I.

Variable	Description
$G = (\mathcal{V}, \mathcal{L}, \sigma)$	Graph representing a physical network
$\mathcal{V}$	ROADM nodes
$v, w \in \mathcal{V}$	Exemplary distinct elements of $\mathcal{V}$
$\mathcal{L}$	Set of all fiber links (multi-edges)
$e = (v, w) \in \mathcal{L}$	A single fiber link (edge).
$\delta(v) \in \mathbb{N}$	Degree of node v, #edges incident to v
$\lvert N_1(v) \rvert \in \mathbb{N}$	# of neighboring nodes adjacent to v
$\Delta \in \mathbb{N}$	Maximum degree (of a node) in a graph
$\sigma : \mathcal{V} \to \mathbb{N}$	Number of transponders that can be allocated to v's edges
$\mu(e) \in \mathbb{N}$	Maximum possible wavelengths on the edge e
$\Lambda : \mathcal{V} \times \mathcal{L} \to \mathbb{N}$	A wavelength allocation, the number of wavelengths from a node onto an edge.
$c(e) \in \mathbb{N}$	Total number of wavelengths on edge e
$u(e) \in \mathbb{R}^{[0,1]}$	Utilization of edge e
G_Λ	Wavelength allocation Λ applied to G
$Y : G \times \mathcal{D} \times \mathcal{F} \to \mathbb{R}^{[1, O(\Delta]}$	Gain factor for moving from a static to dynamic capacity WAN.
$T : G \times \mathcal{D} \times \mathcal{F} \to \mathbb{R}^{[0,1]}$	Maximum link utilization in topology G, for demand matrices $D \in \mathcal{D}$ and for failure scenarios $F \in \mathcal{F}$.
$D \in \mathcal{D}$	Demand matrix, $\{\mathcal{V}^2 \to \mathbb{R}\}$
$F \in \mathcal{F}$	Failure scenario, $F \subseteq \mathcal{L}$, set of edges for which $c(e) = 0$

Table I: Notations used in the formal model.

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) $\delta(v)$ 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.[2]

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 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.[3] Given this, we define a wavelength allocation Λ of a graph G by $\Lambda : \mathcal{V} \times \mathcal{L} \to \mathbb{N}$ 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 the number of wavelengths on an edge, $c(e)$, is not greater than the number of transponders allocating wavelengths onto e from either end, v or w; $c(e) = \min(\Lambda(v, e), \Lambda(w, e), \mu(e))$ where $(v, w) = e$. We denote G_Λ when wavelength allocation Λ is applied to G.

Static WAN: We briefly describe the baseline for our model's comparison, a 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 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

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

[3]Recent work shows this assumption is not optimal; there are gains in building a unidirectional WAN using unidirectional transponders [30], but we leave this discussion for future work.

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. In order to prove the theorem, we will first (1) show that there is always a feasible static wavelength allocation Λ_A, (2) whose gain is at most a factor of $O(\Delta)$ away from some other static wavelength configuration Λ_B, where Λ_B possibly infeasibly allocates more wavelengths than available at the nodes / permitted on the links. Both Λ_A and Λ_B are chosen independently of the traffic or demands present. We will then show (3) that Λ_B can match any throughput or utilization that any dynamically adjusted feasible wavelength allocation Λ_C could serve.

(1) To this end, we create Λ_A as follows: For Λ_A, each node v distributes its $\sigma(v)$ transponders evenly over all its neighbors, possibly wasting up to $|N_1(v)| - 1 \leq \Delta - 1$ transponders to obtain identical numbers for all neighbors, where $|N_1(v)|$ is the number of neighboring nodes adjacent to v. 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 optimal static performance.

(2) 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. 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.

(3) Lastly, 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, as in Λ_B, every node v assigns $\sigma(v)$ transponders to *each* neighbor adjacent to v, whereas in Λ_C, each node v has to distribute its $\sigma(v)$ transponders over *all* its incident links in total. □

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.

B. Model-based Bandwidth Scaling Algorithm

Result from Theorem 1 has implications for both the (offline) provisioning of network resources and the (online) orchestration of those resources according to dynamic traffic and network event scenarios. Regarding the provisioning of resources, we see 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 with degree two or higher, knowing that their benefit will be the greatest at nodes with high degree. Nodes with degree one only need one fallow transponder, as they have only one out-bound fiber on which two activate an optical link. By placing two fallow transponders at each node with degree two or greater, and one at every one with degree one, we can activate and end-to-end path with increased bandwidth between and two nodes in the network. 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 § VI we give a detailed analysis of the cost and benefit of static topologies compared dynamic topologies. Finally, strategically provisioning fallow transponders dramatically aids in reducing computational complexity for optimally choosing where and when to activate reconfigurable links.

By provisioning the network with these fallow transponders, we can call on them opportunistically to adapt and scale bandwidth on network links during adverse scenarios such as flash crowds and link failures. We provide two intuitive heuristic algorithms (henceforth known as *spatial scoping algorithms*) that describe how to orchestrate the fallow transponders to quickly adapt the topology in response to an event. The first assumes a slow TE solver which takes several minutes or more to compute flow allocations when network conditions change (e.g., traffic shifts or link failures). In such a case, we look locally at the links surrounding the event of interest and quickly attempt to scale capacity on those links. In light of the temporal requirement, this algorithm provides a mechanism for increasing capacity along a congested path quickly, on the order of seconds, without re-invoking a potentially slow TE optimization. The second heuristic algorithm assumes a TE optimization that satisfies the temporal requirement and can

compute flow allocations in less than one second for the entire network. It uses the TE optimization to predict which links will suffer congestion loss and increases bandwidth directly on them.

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 algorithm first narrows the scope for a bandwidth scaling solution down to a set of critical links, i.e., high degree nodes where fallow transponders are available, leveraging the model described in § IV-A. This model-based scoping is the key novelty of our approach, as it enables us to quickly find links whose performance is faltering and interject by scaling bandwidth on them. 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 along an alternate path 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 naive bandwidth scaling algorithm (i.e., Algorithm 1), is bound by the shortest path algorithm, and is therefore $O(\mathcal{V}\mathcal{L})$.

The ACTIVATELINK 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.

In cases where the TE algorithm can compute flow allocations and predict utilization quickly, e.g., in sub-second timescales, we can leverage the speediness of the TE optimization to inform our bandwidth scaling decision. Algorithm 2 describes this mechanism concretely. This algorithm runs in a centralized process on the network's SDN controller. When a link failure occurs, the process simulates flow throughput and link utilization in the network assuming every remaining link with any fallow transponders in the network has double the capacity. Now, any link whose utilization is above 50% would have been congested due to the failure in the current topological configuration. Therefore, we identify those links from the simulation and then scale bandwidth up on those links only.

The complexity of the TE-informed bandwidth scaling algorithm (i.e., Algorithm 2) is similar to the complexity of the underlying TE algorithm and has an added linear term with respect to the number of edges in the network because it iterates over each edge in the network twice.

The novelty of spatial scoping algorithms 1 and 2 is that they narrow the scope of the TE optimization problem directly to the localized area in which traffic or network events disrupt throughput for network-wide traffic. Another system

Algorithm 1 Naïve Bandwidth Expansion Algorithm

Require: $G = (\mathcal{V}, \mathcal{L}, \sigma)$ ▷ Network, G, of nodes, $\mathcal{V}$, and links, $\mathcal{L}$ and transponders σ

Require: $f : G \rightarrow L$ ▷ Scope links based on model, taking G as input and returning a set of critical links L

procedure ACTIVATELINK(v, w)
  $e \leftarrow (v, w)$
  $\Lambda(v, e) \leftarrow \Lambda(v, e) + 1$
  $\Lambda(w, e) \leftarrow \Lambda(w, e) + 1$
end procedure

for $(v, w) \in L$ **do**
  if $\sigma(v) > 0$ **AND** $\sigma(w) > 0$ **then**
    ACTIVATELINK(v, w)
  else
    $P \leftarrow$ `shortest_path`(v, w) s.t. $(v, w) \notin P$
    `can_activate` $\leftarrow$ `True`
    for $(\hat{v}, \hat{w}) \in P$ **do**
      if $\sigma(\hat{v}) = 0$ **OR** $\sigma(\hat{w}) = 0$ **then**
        `can_activate` $\leftarrow$ `False`
      end if
    end for
    if `can_activate` **then**
      for $(\hat{v}, \hat{w}) \in P$ **do**
        ACTIVATELINK$(\hat{v}, \hat{w})$
      end for
    end if
  end if
end for

that operate this way is NCFlow [10]. NCFlow partitions the network into a set of connected components, and routes traffic locally in each component before routing inter-component demands. We compare our performance with NCFlow in section VI.

V. REDUCING RECONFIGURATION DELAYS TO ACHIEVE THE TEMPORAL REQUIREMENT

Toward the goal of satisfying temporal requirement with OTP, we seek to establish a performance baseline for wavelength reconfiguration. The baseline will serve as a starting point for developing methods for identifying opportunities to reduce reconfiguration latencies from minutes to sub-seconds in the absence of ASE noise generators. To establish this baseline, we conduct a series of experiments on commodity optical networking equipment using best practices for adding/removing high-capacity circuits in WANs. All of our experiments are conducted on an emulated long-haul optical path with seven amplifiers and 200 km of single-mode optical fiber.

A. Objectives and Testbed

To assess the feasibility and thresholds for dynamic reconfiguration of wavelengths on long-haul paths, we conduct a series of laboratory-based experiments on standard optical networking equipment that can be found in today's enterprise or service provider infrastructures. The specific goal of our tests is to measure the time taken by an optical path to stabilize to the point where it can be used to transport data after adding/removing wavelengths. To this end, we create a testbed

Algorithm 2 TE Informed Bandwidth Scaling

Require: $G = (\mathcal{V}, \mathcal{L}, \sigma)$ ▷ Network, G, of nodes, $\mathcal{V}$, and links, $\mathcal{L}$ and transponders σ
Require: $F \subset \mathcal{L}$ s.t. $c(e) = 0 \ \forall \ e \in F$ ▷ Set of failed links
Require: $D = \{\mathcal{V}^2 \rightarrow \mathbb{R}\}$ ▷ Traffic demands between nodes
Require: $t : (G, D) \rightarrow \{(e, u_e)\}$ ▷ TE function where u_e is the utilization of edge e
Require: $f : G \rightarrow L$ ▷ Scope links based on model, taking G as input and returning a set of critical links L

```
procedure SimulateBandwidthScaling(G, F, D, t, f)
    S = ∅                                  ▷ Set of links to increase bandwidth on
    if F ≠ ∅ then
        H = G − F                          ▷ The network with failed links removed
        L = f(H)
        for e ∈ L do
            c(e) = 2 × c(e)                ▷ Double capacity of remaining edges
        end for
        U = t(H, D)                        ▷ Run the TE algorithm, t on H with D
        for e ∈ U do                       ▷ Find links with utilization above 50%
            if u(e) > 50% then
                S = S + e                  ▷ Add them to the set of scaling links
            end if
        end for
    end if
    return S
end procedure

S = SimulateBandwidthScaling(G, F, D, t, f)
for v, w ∈ S do
    ActivateLink(v, w)
end for
```

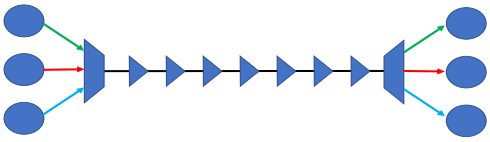


Figure 6: Configuration of our lab-based experiments: six optical TXPs (circles). Three of these transmit 100 Gbps of data each into a DWDM (trapezoid), over seven amplifiers (triangles), and out of a DWDM at the other end before the data is received and decoded by the three TXPs at the other end.

setup that includes equipment found in points of presence (optical switches) and on long-haul paths (amplifiers).

Testbed: Our experimental testbed is shown in Figure 6. The testbed is symmetric with three transponders (TXPs) transceiving data along two simple fiber paths; all of the experiments reported below utilize a single path from West to East. We employ two types of TXPs in our testbed: TXP 1 sends 100 Gbps of data via dual polarization-quadrature amplitude modulation (DP-QPSK), and TXPs 2 & 3 deliver 100 Gbps of data each with on-off keying (OOK). Throughout our experiments, all TXPs send and receive streams at the max data rate of 100 Gbps constantly unless otherwise stated. The data carried are optical data units (ODUs), and can contain SONET, Ethernet, or IP packets in a live deployment. TXPs 1–3 employ forward error correction (FEC) to decode the ODU data and performance monitoring systems built into the TXPs continuously monitor the quality of the optical signals via correctable/uncorrectable FEC bits errors, signal-to-noise ratio (SNR), and Q factor. Note that these monitored values are always available, even while receiving empty ODUs.

Together, the TXPs provide the capacity for 300 Gbps data transmission in our testbed. They are connected to dense wavelength division multiplexers (DWDMs), which can optically multiplex up to 400 Gbps. The testbed is also equipped with seven erbium-doped fiber amplifiers (EDFAs) with an operating range of 20 to 27.5 dB. Short (1 m) jumper fibers connect the amplifiers across the path and are attenuated at 20 dB, emulating signal loss from a span of 80 km between amplifiers. We have one unattenuated span following the third amplifier on the path; the third and fourth amplifiers are spanned by 100 km of single-mode fiber.

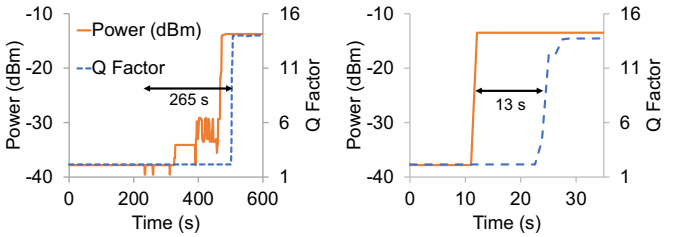
The optical equipment we use in our experiments is representative of equipment that would have been deployed in operational networks over the past ten years. IP routers with suitable TXP interfaces can connect directly to our DWDMs. The EDFAs are high power, capable of transmitting up to 100 km, and operate in the C-band (1550 nm frequency). Amplifiers similar to those used in our setup are often arranged in series to enable the transmission of signals over hundreds to thousands of kilometers.

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 EDFA and Q factor at the receiving TXP. We measure *add-time* for a circuit as the time that it takes for power and Q factor to stabilize after a wavelength is added. We take measurements using an optical signal analyzer (OSA) to measure power levels directly on the fiber, as well as SNMP management information base (MIB) values available from the administrator interface. Unfortunately, the sampling period from the OSA is 5 seconds, so it is only useful for confirmation of signal strength and as visual confirmation of the addition or removal of wavelengths. After repeated tests, we found that the update frequency of SNMP MIB values across devices is ∼1 s. Given the slightly higher precision from SNMP over the OSA and the ability to probe all system devices in parallel, we use the SNMP MIB values in our results.

B. 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 completely stable and predictable state after certain events, including adding/dropping wavelengths. These tests, which take place by default, can be time-consuming and thus pose a challenge in our experiments, which specifically seek to assess a *minimal time* required for power levels on a path to stabilize after adding/dropping wavelengths. To address this issue, we disabled all default testing and safeguards on the ingress and egress amplifiers in most of our experiments (what we refer to as *manual mode*). More details about experiments in automatic vs. manual mode are described in § V-C.

Using our testbed, we evaluate the add/drop time that can be reasonably expected by hardware operating in "automatic" mode. In automatic mode, amplifiers and TXPs use a handshake protocol to negotiate the appropriate sending power level between the TXP and the first amplifier hop. From there, amplifiers adjust their gain at each hop. Eventually, the amplifiers on the path determine the appropriate gain value and relay the wavelength to the receiving TXP. This automatic procedure assures a stable path without the need for operator involvement.



(a) In *automatic mode*, the add time for the wavelength is 4 min & 25 s.

(b) In *manual mode*, the add time for the wavelength is 13 s—over 19× faster than automatic mode.

Figure 7: Wavelength add-time comparison for automatic and manual modes as measured by comparing the time between power level increase for the wavelength at the *first amp-hop* with the time for Q factor increase beyond 11 at the receiver.

Figure 7a shows the ingress power to the first amplifier hop plotted with the Q factor of a corresponding wave 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 the Q factor above 11 at the receiver. In this instance, the add-time for this wave is 4 min & 25 s.

Main finding and implication: The add-time for a long-haul optical circuit, in practice, is on the order of minutes. This delay is primarily due to two factors: *(i)* TXPs incrementally and conservatively increase 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.[4]

If the appropriate power level is known a priori for a TXP on an optical path, then the 4 min spent ramping up power can be saved by automatically applying that power as we show next. We address factor *(ii)* in § V-E.

C. Reconfiguration Delay From Minutes to Seconds

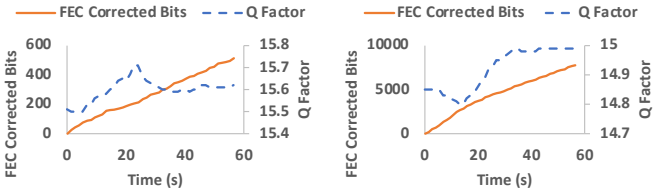
Next, we investigate a method for reducing add-time via intervention in the protocol between the TXPs and their ingress amplifier. Typically, in "automatic" mode, the launch power for a wave is determined by a protocol between the TXP and the ingress multiplexing amplifier. However, there is a configuration parameter on the amplifier and TXP, which enables us to side-step this negotiation process and set the launch power explicitly. Thus, we take the TXP and amplifier 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 amplifier receives it within a hardware-specific target range. In our case, the amplifier excepts to receive signals in the range -14 to -12.5 dBm from any single wavelength. Thus, we set the TXP's sending power such that it hits the target. This value only needs to be determined once for any TXP/ingress amplifier pair.

Figure 7b shows the additional time for a circuit across 7 amplifiers with TXPs operating in manual mode. We set the launch power to 0.5 dBm, and use a variable optical attenuator (VOA) to add/drop the signal instantaneously (i.e., by setting attenuation effectively to infinity or zero). At ∼8 s, when we add attenuation, we see the wave drop, and the power into the amplifier drops to -35 dBm. When attenuation is set to zero again, power at the ingress mux jumps to -13.5 in one time step (1 s). 13 s later, the Q factor for the received signal increase beyond 11, then settles to 13.73.

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, the warm-up phase for optical TXPs unnecessarily stretches the add-time for long-haul circuits. This result suggests a way forward toward achieving our objectives for OTP, however, the stability of the paths must be assured.

D. Impact of Reconfiguration on Witness Waves

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) TXP 2

(b) TXP 3

Figure 8: QoT measurements for witness waves while adding/dropping the wavelength from TXP 1. During the add/drop process the Q factors of the witness waves are relatively constant—varying by +/- 0.1. We saw errors accumulated at a linear rate as is expected in a live transport network. We saw 100% of error bits corrected in hardware while running traffic over TXP 2 & 3.

Figure 8 shows the Q factor and FEC-corrected bits for wavelengths generated by TXP 2 and TXP 3 while adding TXP 1's wavelength; these measurements are taken simultaneously with those shown in Figure 7b. From figure 8, we see that, although we add 50% more power to the circuit from a third TXP, the quality of transmission (QoT) measures of the witness waves

[4]We focus all our attention on add-time, and not drop-time, because dropping optical circuits is trivial, and our evaluations on the effect of such actions on other waves were negligible.

from TXP 2 and TXP 3 are not impaired. More concretely, the Q factor for the two waves varies *only* by +/- 0.1 and there are no FEC-uncorrected bit errors. To further assess the impact of adding/dropping waves, we install a client device that sends 10 Gbps of IP performance traffic via TXP 2. Analysis of the performance traffic over the verifies that no packets are dropped for the witness wave while adding/dropping the wavelength from TXP 1.

Main finding and implication: From these results, we find that quickly 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.

E. Toward Millisecond Reconfiguration Delays

In § V-B and § V-C, we measure the latency in add-times for automatic and manual mode respectively. In this section, we propose a new mechanism that uses a lookup table to choose gain values at each amplifier, further reducing reconfiguration delays. 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 modes with the ones obtained using our proposed lookup mechanism.

An important reason for the long reconfiguration delay in manual and automatic modes, as shown earlier, is that today's optical-layer infrastructure is not designed to support rapid reconfiguration. For example, the amplifiers operate with no knowledge of their past configurations. To solve this, and bring reconfiguration delay down further, we propose creating amplifier tables for TXPs on a light path. The table keeps track of optical configurations on the path and gain settings for amplifiers of the path.

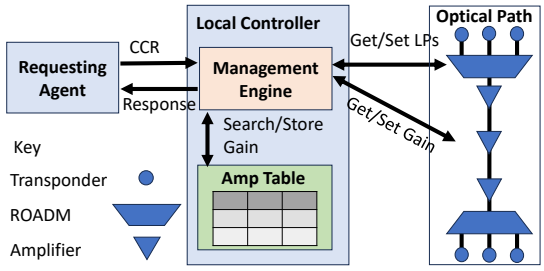


Figure 9: The LC receives a CCR and applies it to an optical path. It stores the gain at each amplifier in the *Amplifier (Amp) Table*. If a CCR is cached in the Table, then the *Management Engine* applies those gains to the amplifiers.

Amplifier table: In our broader vision of OTP, the local controller (LC) is the key point of coordination for various optical components. An LC, as shown in Figure 9, resides on a virtual machine (VM) near TXPs for an optical path (OP) and maintains a table that relates an optical configuration (OC) (i.e., set of active wavelengths) to gain at each amplifier and QoT.[5] OCs in the table are aggregated by power level, thereby keeping the size of the table manageable by a single 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 (TXPs, ROADMs, Amplifiers). The amplifier table 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., to activate TXP pair (s, t) on OP x), it checks the amplifier 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: TL1 and SNMP. These are the two APIs available for querying the EDFAs in our testbed pragmatically. We use both interfaces 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 3.43 s, over 50% faster than the time to activate a light path in manual mode. We also find that with SNMP, we can reduce this latency to ∼0.5 s. 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 over 276× speedup over the expected configuration time for light paths in automatic mode (see Figure 10).

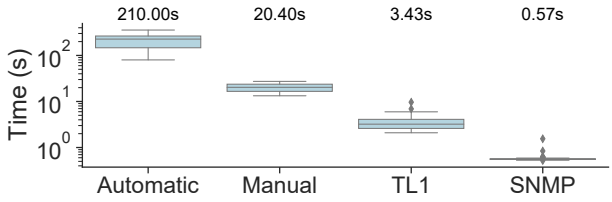


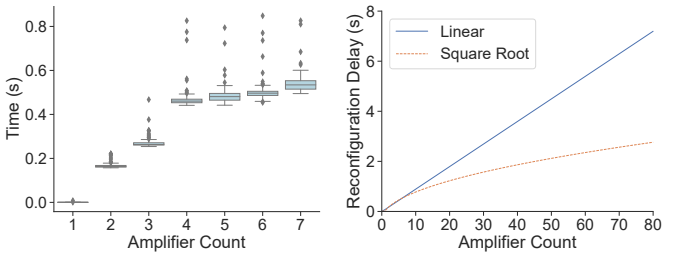
Figure 10: Comparison of reconfiguration delay for various modes. The numbers above the figure show the sample median for the distribution below.

Performance: As shown in Figure 10, the expected time for adding a wavelength in manual mode, with no gain information available, is about 20 s. Therefore any new configuration added to the path will be installed in 20 s. After the configuration metrics are stored in the amplifier table, any future request for that configuration can be added in 0.57 s.

Validation: We collect Q factor and latency data on a 100 Gbps circuit. We find 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.

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

We use a layer-3 traffic generator to produce packets of various sizes (95, 1500, and 9216 bytes) and find that round-trip time varies by no more than +/- 0.1 μs. The average jitter did not deviate from 0.0 μs. 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.



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

(b) Linear (red) and Non-linear (blue) regression ($\sqrt{x}$) model projected for 80 amplifiers.

Figure 11: Gain retrieval time for a path of seven amplifiers (left), and projected reconfiguration time for longer paths (right).

Longer paths: 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 collect data by polling different subsets of amplifiers with parallel SNMP queries (the same method used to benchmark our SNMP approach in figure 10). For each set of amplifiers tested, we repeat our measurement for the *gain retrieval time* 100 times. Figure 11 shows the data we collected (11a) and the model (11b). According to the model, an optical path with 25 amplifiers can be reconfigured in 1.5 to 2.3 s.

VI. 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 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.

A. Simulator

Evaluating OTP requires access to a WAN backbone which we do not have. To address this challenge, we build a Python-based discrete event simulator, the *GreyLambda simulator*. Figure 12 illustrates the architecture of the GreyLambda simulator. While TE simulators in recent work [10], [31] have taken topology as a fixed input to show how routing decisions affect performance as a function of the traffic, the GreyLambda

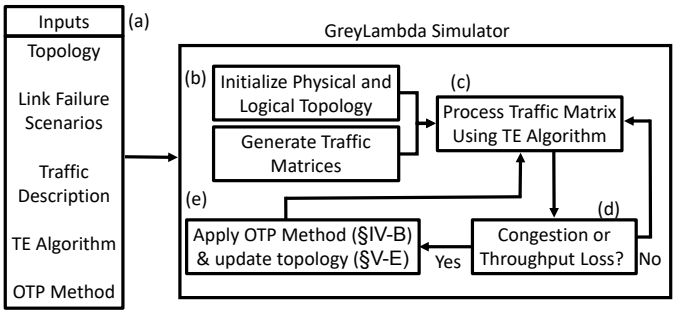


Figure 12: Architecture of the GreyLambda simulator.

simulator aims to show how topology *and* routing decisions affect performance as a function of traffic.

The GreyLambda 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 TE control loop. The third is to enable the prototyping of different OTP methods in conjunction with different TE algorithms. The simulator is written with ~6.7 k lines of Python 3 code[6].

The inputs to the GreyLambda simulator are shown in figure 12 (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 12 (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., [32]) 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 12 (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

[6]https://github.com/mattall/topology-programming

threshold (e.g., 100%), as shown in Figure 12 (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 12 (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 § IV-B. 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 V-E serve as a baseline for this step as the GreyLambda simulator estimates link reconfiguration times using experimental data and the model given in Figure 11b. After this process is complete, the GreyLambda simulator returns to step Figure 12 (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 V-E 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 [33] and manual transcription of publicly available network infrastructure maps [27], [34]. These topologies were chosen because they are large real world ISP and cloud service networks. A summary of the topology information is given in table II.

Network		Nodes	Edges
B4	[34]	54	118
Zayo	[33]	96	110
Verizon	[33]	116	151
Azure	[27]	113	216
Comcast	[33]	149	195

Table II: Network topologies used in this study.

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 constraint is satisfied by leaving spectrum available for a single optical wavelength on each fiber. We are confident that these are reasonable assumptions based on our recent discussions with a regional network operator; optical fiber is abundant in backbone networks to the extent that it is often leased to third parties as an additional revenue stream. 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 [35]. While fallow transponders are *not* currently deployed in networks, we place them in the network for our simulation to assess their benefit for achieving scalability in TE with OTP.

IP path selection and flow allocation: We compare the performance of two recent state-of-the-art TE algorithms, namely SMORE [9] and NCFlow [18]. 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. Given that these are recent state-of-the-art TE solvers whose performance has previously been measured against other TE optimization solvers, we are confident that they are relevant choices for the TE parameter in our simulation.

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 3 shows our flash crowd generation method explicitly.

We chose *aggregate_strength* to be $2\times$ the capacity of the link chosen for each traffic matrix while constructing one matrix for every link in the network. We consider these as reasonable parameters because a $2\times$ demand for capacity on a link will need to be routed intelligently through a network, as single-shortest path routing is guaranteed to result in a total throughput not greater than 50% of demand, and by repeating this for every link in the network we get a full picture of the TE's performance and a clear picture of whether or not the fixed topology is a limiting factor for any of these demands.

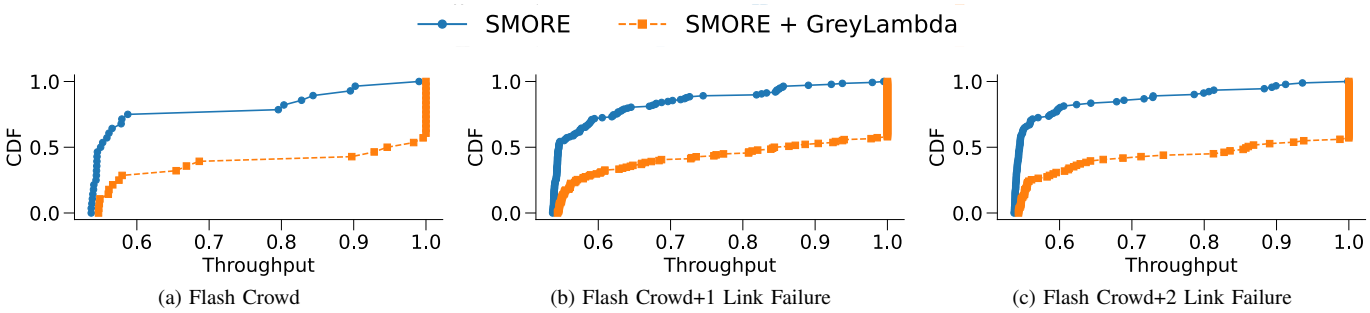
Algorithm 3 Flash Crowd Traffic Matrix Generation

Require: $G = (V, E)$ ▷ Network topology G of vertices V and edges E
Require: $f : (u, v) \rightarrow list : paths$ ▷ Map from each link in the 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

B. SMORE Comparison

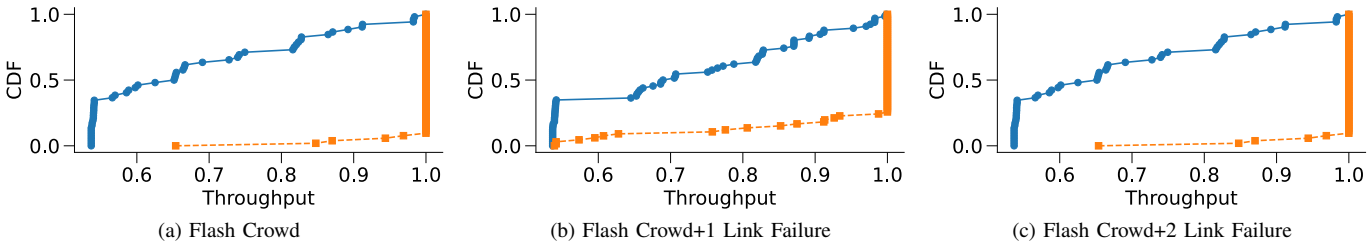
We emulate flash crowd events, each with an aggregate strength of 2x link capacity against every link in the five large CDN and ISP topologies while removing up to two links. We then compare the performance of SMORE vs. SMORE+GreyLambda in these scenarios. Figures 13– 17 show the results for aggregate network throughput. Overall, we find that GreyLambda can increase the throughput of SMORE for all traffic and link-failure scenarios.

Spatial requirement: In all figures 13– 17, we see a gap in the CDF for throughput between SMORE and



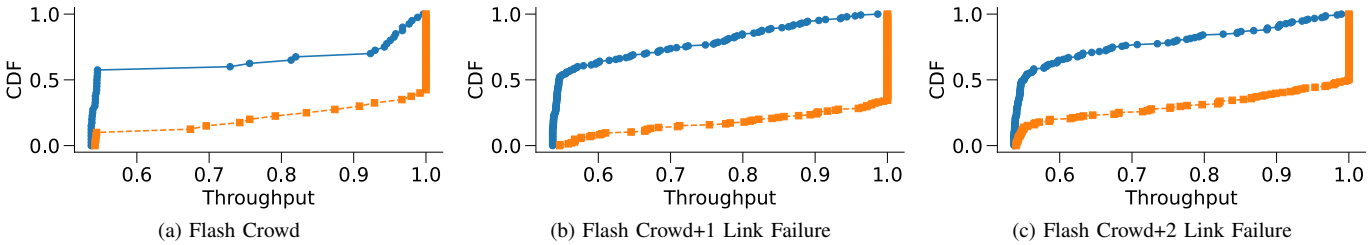
(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failure

Figure 13: Throughput in Zayo under flash crowds combined with one and two link failures.



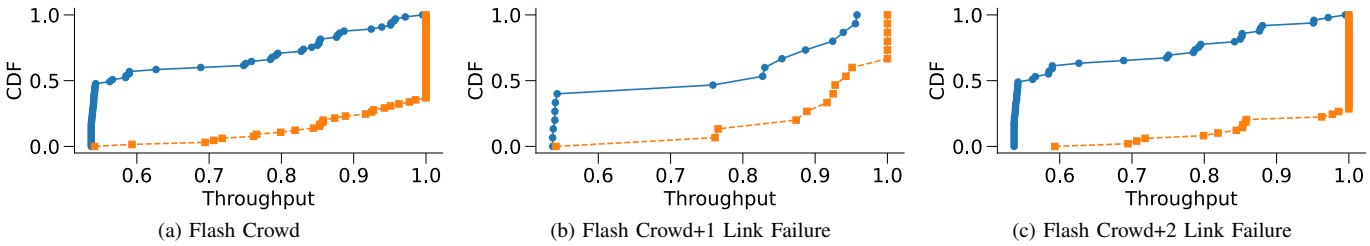
(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failure

Figure 14: Throughput in B4 under flash crowds combined with one and two link failures.



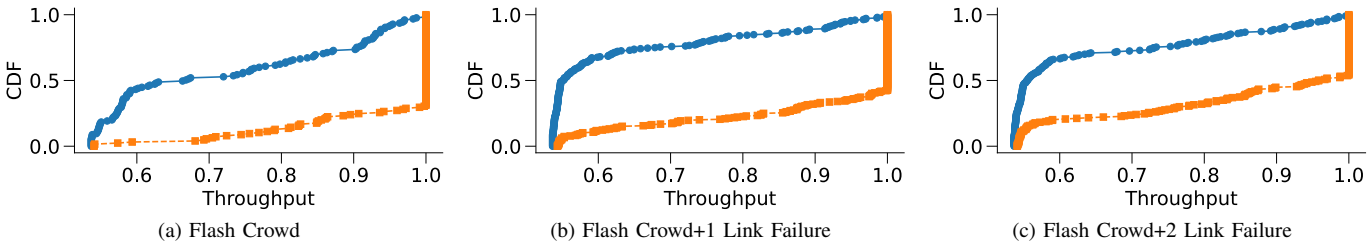
(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failure

Figure 15: 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 Failure

Figure 16: 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 Failure

Figure 17: Throughput in Comcast under flash crowds combined with one and two link failures.

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 § V 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.

C. NCFlow Comparison

We compare the performance of NCFlow [10] by itself versus NCFlow+GreyLambda. This analysis uses the latest available version of the NCFlow simulator [18]. 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. In this experiment, we used Algorithm 2 because NCFlow's optimization step was solvable quickly (in fewer than 4 seconds for every network and traffic matrix considered). 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.

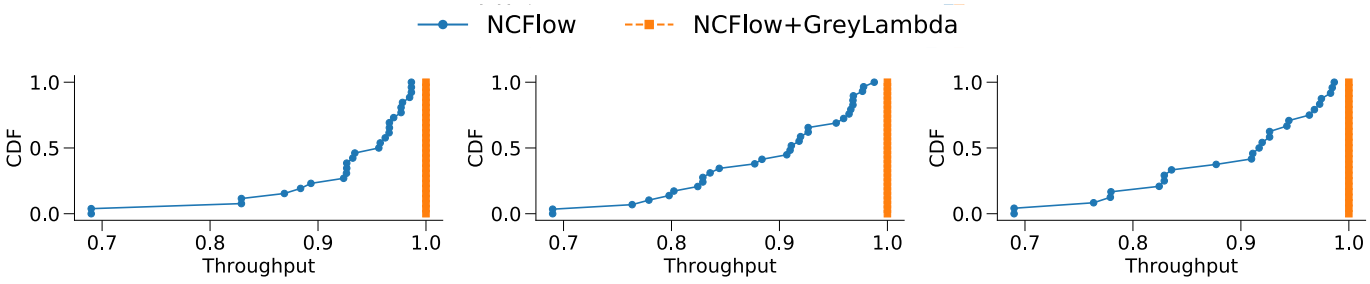
Extended discussion on NCFlow: It may be surprising that NCFlow+GreyLambda results are flawless concerning throughput. The reason we can guarantee such performance comes down to the speediness with which NCFlow solves its optimization function; when a link failure or traffic surge occurs, we can simulate the network throughput assuming every link has 2× 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. This *TE-informed* OTP method (Algorithm 2) dramatically improves the scalability of NCFlow concerning diverse traffic matrices and link failure scenarios.

D. GreyLambda vs. Other Topology Programming Techniques

In the following two examples we illustrate the performance of GreyLambda against two other topology-programming techniques. We present Comcast for both of these illustrations because it is the largest and most complex network in our data set. The results we observe for Comcast were qualitatively similar for the other networks presented earlier, but those results have been excluded from this paper due to space constraints. We have performed the following tests using two underlying traffic engineering methods, namely optimal MCF and ECMP. In our analysis, MCF offers roughly 2× more throughput than ECMP when all other parameters are the same (including topology, traffic, and topology programming method or absence thereof).

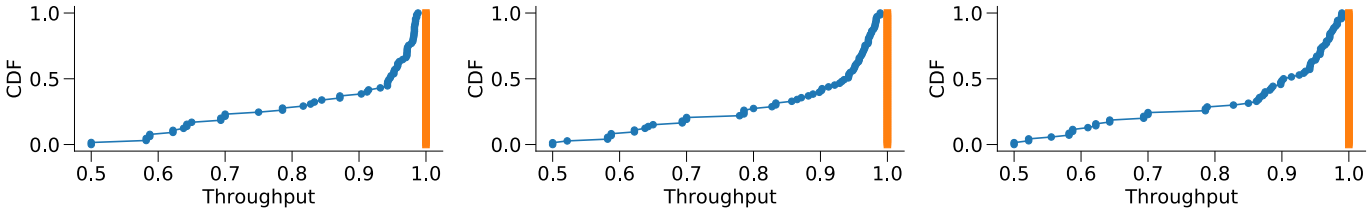
GreyLambda vs. Temporary Bandwidth Expansion. First, we compare GreyLambda against an alternative SDN-based technique that approximates topology programming. In this comparison, we consider *temporary bandwidth expansion* (TBE), as proposed in Spiffy [36] and optimal traffic engineering as derived by solving the multi-commodity flow (MCF) problem.

In TBE, the bandwidth of a link is artificially constricted. This approximates bandwidth on demand by allowing the restriction to be lifted in times of need. In Spiffy, Kang et al. specifically implemented it along with a rate-change test on traffic and ultimately determined whether flows can be classified as malicious or not. However, in the general case of flash crowds, all of the traffic could be legitimate and therefore all flows may pass this rate-change test. In such a case, TBE simply reduces the optimal performance of TE.



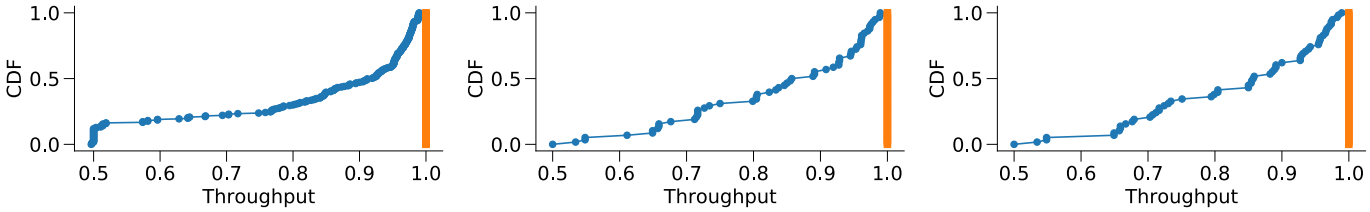
(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failure

Figure 18: Throughput in Zayo with NCFlow and NCFlow+GreyLambda.



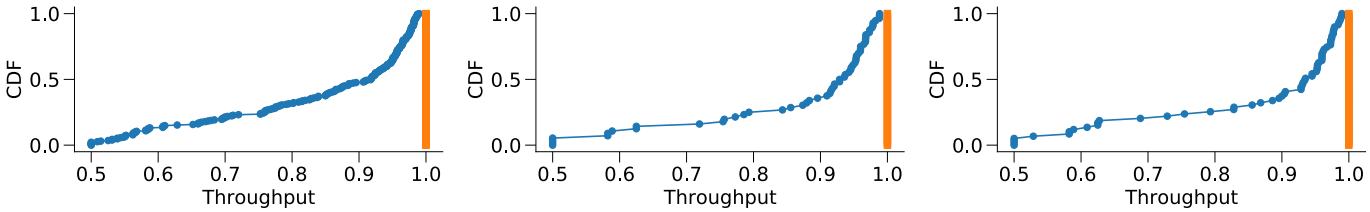
(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failure

Figure 19: Throughput in B4 with NCFlow and NCFlow+GreyLambda.



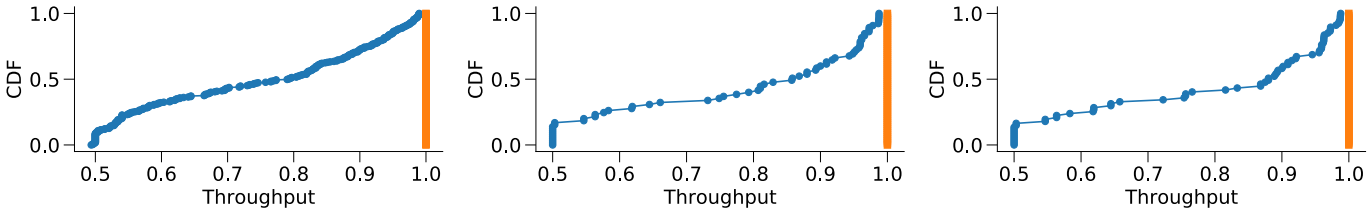
(a) Flash Crowd (b) Flash Crowd+1 Link Failure (c) Flash Crowd+2 Link Failure

Figure 20: Throughput in Verizon with NCFlow and NCFlow+GreyLambda.



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

Figure 21: Throughput in Azure with NCFlow and NCFlow+GreyLambda.



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

Figure 22: Throughput in Comcast with NCFlow and NCFlow+GreyLambda.

Figures 23 and 24 show the total network throughput through Comcast as a function of demand scale during a flash-crowd event. In this case, traffic is added to a gravity model matrix such that the flash-crowd traffic is concentrated on a small set of core links within the network. A demand scale of 1× represents the highest scale at which all systems enable 100% throughput and scale value 2× represents a case wherein the same matrix scaled up twice.

Figure 23 shows that GreyLambda is able to satisfy all demand without loss for a traffic matrix 1.4× the magnitude of the optimal TE allocation. In Figure 24 we see the max link congestion across all links in the network and observe that as demand scale increase beyond 1.4×, the max link congestion stalls at ∼1.38 (meaning the most congested link in the network has 1.38× more demand than bandwidth) while the demand scale of the entire traffic matrix is between 1.8× and 2.0×. This is because GreyLambda attempts to light new paths on links that are congested and the set of congested links has grown to include links that weren't congested at the 1.6× scale. Thus, GreyLambda activates new optical paths along these links and thereby stalls congestion loss until the demand scale grows past 2.0×.

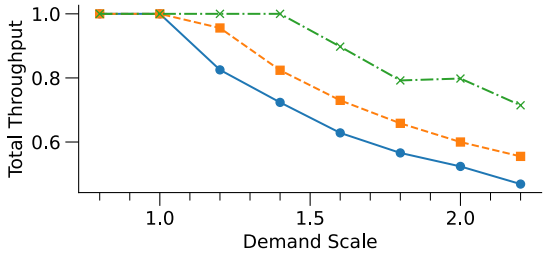


Figure 23: Total Throughput as demand increases for a scaled gravity model traffic matrix on Comcast's topology

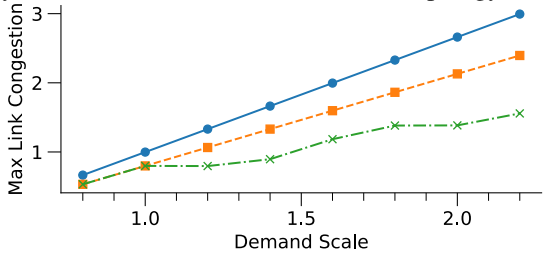


Figure 24: Max link congestion as demand increases for a scaled gravity model traffic matrix on Comcast's topology

GreyLambda vs. Bandwidth Variable Transceivers. Second, we compare GreyLambda against a complimentary topology programming technique wherein bandwidth variable transceivers (BVTs) can adapt their modulation and reduce their sending rate at times when the signal-to-noise ratio (SNR) of the link falls (due to transient events of physical disturbances). Modern transceivers in wide-area networks are augmented with coherent digital signal processing (coherent DSP) elements. The coherent DSPs allow fixed-modulation transceivers to tolerate and correct physical layer bit-errors up to a certain rate by employing forward error correction (FEC) in hardware. However, when SNR falls below the FEC maximum allowable BER, these errors are not corrected fast enough and the link is effectively out of service for the network. BVTs, in contrast to fixed rate transponders, can adapt their modulation and reduce their bandwidth when SNR falls and thereby remain online (albeit with less bandwidth) rather than shut off completely. The application of BVTs to traffic engineering was proposed in [37] and deployed in [19].

In this experiment, we simulate a transient link failure induced by poor SNR of a WAN link. We show three scenarios for the failure. First, with no topology programming, the link is simply removed from the network. Second, with BVT, the link remains online with its bandwidth reduced by 25%. We note that this capacity drop is consistent with the stepping down from a 16-QAM modulation to 8-QAM [38] and that signals modulated with 16-QAM have been transmissible over continental and trans-Atlantic distances for more than a decade [39]; therefore these parameters are feasible and justified for all links in the Comcast topology which we study in this experiment. The third scenario shows BVTs in conjunction with GreyLambda. For all of these experiments, the underlying traffic forwarding policy is derived by solving for the multi-commodity flow allocation.

Figures 25 and 26 show the total network throughput through Comcast as a function of demand scale during an SNR-induced link failure. A demand scale of 1 indicates the maximum scale at which the optimal multi-commodity flow allocation can achieve 100% throughput. We observe that with BVT, the max demand scale is ∼1.8×. Beyond this scale, the link's capacity is over-subscribed thus inducing a drop in throughput. With BVT and GreyLambda, however, we can continue to scale the demand as high as 2.2× with 100% throughput. At the demand scale 2.6× we see a modest throughput drop with a loss of 1.08% of aggregate traffic. Figure 26 shows that BVT reduces the rate of change for the max link congestion with respect to demand scale. BVT mirrors this benefit until that point at which congestion loss occurs for BVT alone; it is at this point which GreyLambda activates a new optical path, keeping the max link congestion below 1 and improving on the permissible demand scale for the network.

VII. Related Work

SDN based traffic engineering: Optimizing WAN network performance via TE has been of interest to both industrial and academic communities [7], [8], [10], [40], [41]. Approaches include B4 [7], SWAN [8], Owan [40], SMORE [9], Shoofly [42] and others [13], [19], [43]–[55], each of which aims at improving the utilization of inter-datacenter WANs. A survey of related efforts is available here [56]–[58]. We posit that deployment of the techniques described in these studies along with OTP has the potential to improve performance results.

Optics and SDN for link failures: Efforts complementary to ours include [15], [45]–[48]. These share our goal of introducing programmability to optical layer. However, with the exception of [15], these efforts do not address the performance penalties incurred by reconfiguring optical components in WANs. While [15] 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.

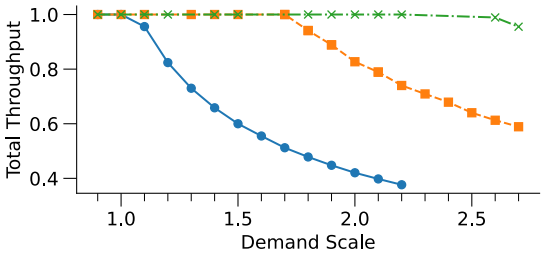


Figure 25: Total Throughput as demand increases for a scaled gravity model traffic matrix on Comcast's topology

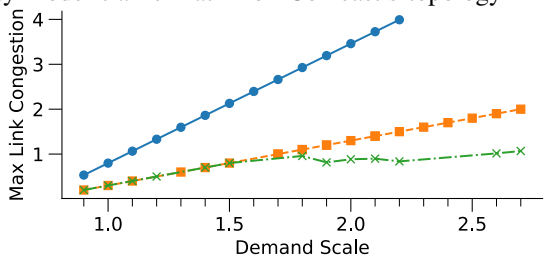


Figure 26: Max link congestion as demand increases for a scaled gravity model traffic matrix on Comcast's topology

Capacity planning: Provisioning infrastructure to enhance the robustness of networked systems has been a focus of many prior studies including backup routing [59], preventive routing via risk analysis [60], shortcuts [42], management system for provisioning [61], and backup paths via IGP link weight optimization [62] or RSVP-TE's *fast reroute* mechanism [63]. 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., [64]–[66]). Our work can be used in conjunction with these efforts since it augments the network capacity by dynamically allocating wavelengths to the recovery paths.

Technological trends towards programmable networks: OpenConfig [67] provides the optical networking community with an "open" system [68] 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 paper. 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.

Dynamic inter-AS optical networking: 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 [69]–[74] and references therein. These broker-based approaches realize cross-domain light path provisioning with Nash bargaining-type cooperative games [72], [73]. Whereas the experiments described in this paper focus on demonstrating and quantifying the performance of OTP. In contrast, the work on a distributed multi-continental infrastructure reported in [74] 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 [75]. 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.

Bandwidth variable transceivers: A method for optimizing bandwidth globally using bandwidth variable transceivers (BVTs) is considered by Ives et al. in [54]. A followup effort [76] 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.

Optical amplifier tuning: Similar to our effort, stabilizing optical paths via predetermined amplifier gains are explored in [77]–[79]. In particular, Oliveira et al. [78] show that they can use a cognitive approach to select amplifier gain. Building on top of [78], Moura et al. [77] 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.

VIII. 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.

References

[1] P. Patel, D. Bansal, L. Yuan, A. Murthy, A. Greenberg, D. A. Maltz, R. Kern, H. Kumar, M. Zikos, H. Wu, C. Kim, N. Karri, Ananta: Cloud scale load balancing, in: ACM SIGCOMM, 2013, pp. 207–218.

[2] D. E. Eisenbud, C. Yi, C. Contavalli, C. Smith, R. Kononov, E. Mann-Hielscher, A. Cilingiroglu, B. Cheyney, W. Shang, J. D. Hosein, Maglev: A fast and reliable software network load balancer, in: USENIX NSDI, 2016, pp. 523–535.

[3] D. Loher, SONiC: Software for open networking in the cloud (accessed Feb. 2023), https://sonic-net.github.io/SONiC/ (2023).

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

[5] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, J. Turner, Openflow: enabling innovation in campus networks, ACM SIGCOMM Computer Communication Review (2008).

[6] D. Firestone, SmartNIC: Accelerating Azure's network with FPGAs on OCS servers (accessed Feb. 2023), https://ocpussummit2016.sched.com/event/68u4/ (2016).

[7] S. Jain, A. Kumar, S. Mandal, J. Ong, L. Poutievski, A. Singh, S. Venkata, J. Wanderer, J. Zhou, M. Zhu, et al., B4: Experience with a globally-deployed software defined WAN, ACM SIGCOMM (2013).

[8] C.-Y. Hong, S. Kandula, R. Mahajan, M. Zhang, V. Gill, M. Nanduri, R. Wattenhofer, Achieving high utilization with software-driven WAN, in: ACM SIGCOMM, 2013, pp. 15–26.

[9] P. Kumar, Y. Yuan, C. Yu, N. Foster, R. Kleinberg, P. Lapukhov, C. L. Lim, R. Soulé, Semi-oblivious traffic engineering: The road not taken, in: USENIX NSDI, USENIX Association, Renton, WA, 2018, pp. 157–170.

[10] F. Abuzaid, S. Kandula, B. Arzani, I. Menache, M. Zaharia, P. Bailis, Contracting wide-area network topologies to solve flow problems quickly, in: USENIX NSDI, 2021, pp. 175–200.

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

[12] H. H. Liu, S. Kandula, R. Mahajan, M. Zhang, D. Gelernter, Traffic engineering with forward fault correction, in: ACM SIGCOMM, ACM, 2014, pp. 527–538.

[13] U. Krishnaswamy, R. Singh, N. Bjørner, H. Raj, Decentralized cloud wide-area network traffic engineering with {BLASTSHIELD}, in: 19th USENIX Symposium on Networked Systems Design and Implementation (NSDI 22), 2022, pp. 325–338.

[14] S. Mandal, 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).

[15] Z. Zhong, M. Ghobadi, A. Khaddaj, J. Leach, Y. Xia, Y. Zhang, Arrow: restoration-aware traffic engineering, in: ACM SIGCOMM, 2021, pp. 560–579.

[16] S. Even, A. Itai, A. Shamir, On the complexity of time table and multi-commodity flow problems, in: 16th Annual Symposium on Foundations of Computer Science (sfcs 1975), 1975, pp. 184–193. doi:10.1109/SFCS.1975.21.

[17] H. Räcke, Optimal hierarchical decompositions for congestion minimization in networks, in: C. Dwork (Ed.), Proceedings of the 40th Annual ACM Symposium on Theory of Computing, Victoria, British Columbia, Canada, May 17-20, 2008, ACM, 2008, pp. 255–264. doi:10.1145/1374376.1374415.
URL https://doi.org/10.1145/1374376.1374415

[18] NCFlow github repository (accessed Feb. 2023), https://github.com/stanford-futuredata/pop-ncflow (2021).

[19] R. Singh, M. Ghobadi, K.-T. Foerster, M. Filer, P. Gill, RADWAN: Rate adaptive wide area network, in: ACM SIGCOMM, SIGCOMM '18, ACM, New York, NY, USA, 2018, pp. 547–560.

[20] S. Kandula, I. Menache, R. Schwartz, S. R. Babbula, Calendaring for wide area networks, in: ACM SIGCOMM, SIGCOMM '14, ACM, New York, NY, USA, 2014, p. 515–526.

[21] A. Kumar, S. Jain, U. Naik, A. Raghuraman, N. Kasinadhuni, E. C. Zermeno, C. S. Gunn, J. Ai, B. Carlin, M. Amarandei-Stavila, M. Robin, A. Siganporia, S. Stuart, A. Vahdat, BwE: Flexible, hierarchical bandwidth allocation for WAN distributed computing, in: ACM SIGCOMM, 2015, pp. 1–14.

[22] C.-Y. Hong, S. Mandal, M. Al-Fares, M. Zhu, R. Alimi, C. Bhagat, S. Jain, J. Kaimal, S. Liang, K. Mendelev, et al., B4 and after: Managing hierarchy, partitioning, and asymmetry for availability and scale in Google's software-defined wan, in: ACM SIGCOMM, 2018, pp. 74–87.

[23] Z. Yang, Y. Cui, B. Li, Y. Liu, Y. Xu, Software-defined wide area network (SD-WAN): Architecture, advances and opportunities, in: International Conference on Computer Communication and Networks, IEEE, 2019, pp. 1–9.

[24] P. Berde, M. Gerola, J. Hart, Y. Higuchi, M. Kobayashi, T. Koide, B. Lantz, B. O'Connor, P. Radoslavov, W. Snow, 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.

[25] P.-W. Tsai, C.-W. Tsai, C.-W. Hsu, C.-S. Yang, Network monitoring in software-defined networking: A review, IEEE Systems Journal 12 (4) (2018) 3958–3969.

[26] M. Crovella, E. Kolaczyk, 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), Vol. 3, IEEE, 2003, pp. 1848–1857.

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

[28] M. Nance-Hall, P. Barford, K.-T. Foerster, M. Ghobadi, W. Jensen, R. Durairajan, Are WANs ready for optical topology programming?, in: Proceedings of the ACM SIGCOMM 2021 Workshop on Optical Systems, 2021, pp. 28–33.

[29] M. Nance-Hall, R. Durairajan, Bridging the optical-packet network chasm via secure enclaves (2020).

[30] W. Zhang, B. G. Bathula, Breaking the bidirectional link paradigm, in: Optical Fiber Communication Conference, Optical Society of America, 2016, p. Th1E.3.

[31] P. Kumar, C. Yu, Y. Yuan, N. Foster, R. Kleinberg, R. Soulé, YATES: Rapid prototyping for traffic engineering systems, in: Proceedings of the Symposium on SDN Research, ACM, 2018, p. 11.

[32] TMgen (accessed Feb. 2023), https://github.com/progwriter/TMgen/blob/master/docs/quickstart.rst (2018).

[33] R. Durairajan, S. Ghosh, X. Tang, P. Barford, B. Eriksson, Internet Atlas: A geographic database of the Internet, in: Proceedings of ACM HotPlanet, 2013, pp. 15–20.

[34] 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).

[35] Tunable DWDM transceivers (accessed Feb. 2023), https://ii-vi.com/product/400g-zr-qsfp-dd-dco-high-tx-output-power-optical-transceiver/ (2023).

[36] M. S. Kang, V. D. Gligor, V. Sekar, Spiffy: Inducing cost-detectability tradeoffs for persistent link-flooding attacks., in: NDSS, 2016, pp. 53–55.

[37] R. Singh, M. Ghobadi, K.-T. Förster, M. Filer, P. Gill, Run, walk, crawl: Towards dynamic link capacities, in: Proceedings of the 16th ACM HotNets 2017, 2017, pp. 143–149.

[38] X. Zhou, Q. Zhuge, M. Qiu, M. Xiang, F. Zhang, B. Wu, K. Qiu, D. V. Plant, On the capacity improvement achieved by bandwidth-variable transceivers in meshed optical networks with cascaded roadms, Opt. Express 25 (5) (2017) 4773–4782. doi:10.1364/OE.25.004773.
URL https://opg.optica.org/oe/abstract.cfm?URI=oe-25-5-4773

[39] H. Zhang, J.-X. Cai, H. G. Batshon, C. R. Davidson, Y. Sun, M. Mazurczyk, D. G. Foursa, A. Pilipetskii, G. Mohs, N. S. Bergano, 16qam transmission with 5.2 bits/s/hz spectral efficiency over transoceanic distance, Opt. Express 20 (11) (2012) 11688–11693. doi:10.1364/OE.20.011688.
URL https://opg.optica.org/oe/abstract.cfm?URI=oe-20-11-11688

[40] X. Jin, Y. Li, D. Wei, S. Li, J. Gao, L. Xu, G. Li, W. Xu, J. Rexford, Optimizing bulk transfers with software-defined optical WAN, in: ACM SIGCOMM, 2016, pp. 87–100.

[41] N. Laoutaris, M. Sirivianos, X. Yang, P. Rodriguez, Inter-datacenter bulk transfers with NetStitcher, in: ACM SIGCOMM, 2011, pp. 74–85.

[42] R. Singh, N. Bjorner, S. Shoham, Y. Yin, J. Arnold, J. Gaudette, Cost-effective capacity provisioning in wide area networks with shoofly, in: Proceedings of the 2021 ACM SIGCOMM 2021 Conference, 2021, pp. 534–546.

[43] Z. Xu, F. Y. Yan, R. Singh, J. T. Chiu, A. M. Rush, M. Yu, Teal: Learning-accelerated optimization of wan traffic engineering, in: Proceedings of the ACM SIGCOMM 2023 Conference, 2023, pp. 378–393.

[44] R. Singh, N. Bjørner, U. Krishnaswamy, Traffic engineering: from isp to cloud wide area networks, in: Proceedings of the Symposium on SDN Research, 2022, pp. 50–58.

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

[46] A. Giorgetti, F. Paolucci, F. Cugini, P. Castoldi, Dynamic restoration with GMPLS and SDN control plane in elastic optical networks, Journal of Optical Communications and Networking (2015).

[47] D. C. Kilper, Y. Li, Optical physical layer SDN: Enabling physical layer programmability through open control systems, in: Optical Fiber Communications Conference, 2017, pp. W1H–3.
[48] Y. Xiong, Y. Li, B. Zhou, R. Wang, G. N. Rouskas, SDN enabled restoration with triggered precomputation in elastic optical inter-datacenter networks, Journal of Optical Communications and Networking (2018).
[49] 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).
[50] S. Jia, X. Jin, G. Ghasemiesfeh, J. Ding, J. Gao, Competitive analysis for online scheduling in software-defined optical WAN, in: INFOCOM, 2017, pp. 1–9.
[51] M. Filer, J. Gaudette, M. Ghobadi, R. Mahajan, T. Issenhuth, B. Klinkers, J. Cox, Elastic optical networking in the Microsoft cloud (invited), J. Opt. Commun. Netw. 8 (7) (2016) A45–A54.
[52] T. Hofmeister, V. Vusirikala, B. Koley, How can flexibility on the line side best be exploited on the client side?, in: Optical Fiber Communication Conference, Optical Society of America, 2016, p. W4G.4.
[53] P. Zhu, J. Li, D. Wu, Z. Wu, Y. Tian, Y. Chen, D. Ge, X. Chen, Z. Chen, Y. He, Demonstration of elastic optical network node with defragmentation functionality and SDN control, in: Optical Fiber Communication Conference, Optical Society of America, 2016, p. Th3I.3.
[54] D. J. Ives, P. Bayvel, S. J. Savory, Routing, modulation, spectrum and launch power assignment to maximize the traffic throughput of a nonlinear optical mesh network, Photonic Network Communications 29 (3) (2015) 244–256.
[55] S. Oda, M. Miyabe, S. Yoshida, T. Katagiri, Y. Aoki, J. C. Rasmussen, M. Birk, K. Tse, Demonstration of an autonomous software controlled living optical network that eliminates the need for pre-planning, in: Optical Fiber Communication Conference, Optical Society of America, 2016, p. W2A.44.
[56] A. S. Thyagaturu, A. Mercian, M. P. McGarry, M. Reisslein, W. Kellerer, Software defined optical networks (SDONs): A comprehensive survey, IEEE Communications Surveys & Tutorials (2016).
[57] R. Alvizu, G. Maier, N. Kukreja, A. Pattavina, R. Morro, A. Capello, C. Cavazzoni, Comprehensive survey on T-SDN: Software-defined networking for transport networks, IEEE Communications Surveys & Tutorials (2017).
[58] K.-T. Foerster, S. Schmid, Survey of reconfigurable data center networks: Enablers, algorithms, complexity, ACM SIGACT News 50 (2019) 62–79.
[59] L. Gao, T. Griffin, J. Rexford, Inherently safe backup routing with BGP, in: IEEE INFOCOM, 2001, pp. 547–556.
[60] B. Eriksson, R. Durairajan, P. Barford, RiskRoute: A framework for mitigating network outage threats, in: ACM CoNEXT, 2013, pp. 405–416.
[61] R. Boutaba, W. Golab, Y. Iraqi, Lightpaths on demand: A web-services-based management system, IEEE Communications Magazine (2004).
[62] B. Fortz, M. Thorup, Optimizing OSPF/IS-IS weights in a changing world, IEEE JSAC (2002).
[63] P. Pan, G. Swallow, A. Atlas, RFC 4090: Fast reroute extensions to RSVP-TE for LSP tunnels, http://www.ietf.org/rfc/rfc4090.txt (2005).
[64] T. Holterbach, E. C. Molero, M. Apostolaki, A. Dainotti, S. Vissicchio, L. Vanbever, Blink: Fast connectivity recovery entirely in the data plane, in: USENIX NSDI, 2019, pp. 161–176.
[65] A. K. Atlas, A. Zinin, RFC 5286: Basic specification for IP fast reroute: Loop-free alternates, https://www.ietf.org/rfc/rfc5286.txt (2008).
[66] M. Shand, S. Bryant, RFC 5714: IP fast reroute framework, https://www.ietf.org/rfc/rfc5714.txt (2010).
[67] OpenConfig, Vendor-neutral, model-driven network management designed by users (accessed Feb. 2023), http://www.openconfig.net/ (2016).
[68] J. Cox, SDN control of a coherent open line system, in: Optical Fiber Communications Conference, 2015, pp. 1–1.
[69] A. Castro, L. Velasco, L. Gifre, C. Chen, J. Yin, Z. Zhu, R. Proietti, S.-J. B. Yoo, Brokered orchestration for end-to-end service provisioning across heterogeneous multi-operator (multi-AS) optical networks, IEEE Journal of Lightwave Technology (2016).
[70] S. Yoo, Multi-domain cognitive optical software defined networks with market-driven brokers, in: European Conference on Optical Communication, IEEE, 2014, pp. 1–3.
[71] D. Marconett, S. Yoo, FlowBroker: Market-driven multi-domain SDN with heterogeneous brokers, in: Optical Fiber Communication Conference, Optical Society of America, 2015, pp. Th2A–36.
[72] X. Chen, J. Yin, C. Chen, Z. Zhu, A. Casales, S. B. Yoo, Multi-broker based market-driven service provisioning in multi-domain SD-EONs in noncooperative game scenarios, in: European Conference on Optical Communication, IEEE, 2015, pp. 1–3.
[73] L. Sun, X. Chen, Z. Zhu, Multibroker-based service provisioning in multidomain SD-EONs: Why and how should the brokers cooperate with each other?, IEEE Journal of Lightwave Technology (2017).
[74] A. Castro, L. Gifre, C. Chen, J. Yin, Z. Zhu, L. Velasco, S.-J. B. Yoo, 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).
[75] A. Mahimkar, A. Chiu, R. Doverspike, M. D. Feuer, P. Magill, E. Mavrogiorgis, J. Pastor, S. L. Woodward, J. Yates, Bandwidth on demand for inter-data center communication, in: Proceedings of the 10th ACM Workshop on Hot Topics in Networks, ACM, 2011, p. 24.
[76] D. J. Ives, A. Alvarado, S. J. Savory, Throughput gains from adaptive transceivers in nonlinear elastic optical networks, Journal of Lightwave Technology 35 (6) (2017) 1280–1289.
[77] U. Moura, M. Garrich, H. Carvalho, M. Svolenski, A. Andrade, A. C. Cesar, J. Oliveira, E. Conforti, Cognitive methodology for optical amplifier gain adjustment in dynamic DWDM networks, Journal of Lightwave Technology 34 (8) (2016) 1971–1979.
[78] J. R. Oliveira, A. Caballero, E. Magalhães, U. Moura, R. Borkowski, G. Curiel, A. Hirata, L. Hecker, E. Porto, D. Zibar, et al., Demonstration of EDFA cognitive gain control via GMPLS for mixed modulation formats in heterogeneous optical networks, in: Optical Fiber Communication Conference, Optical Society of America, 2013, pp. OW1H–2.
[79] Z. Zhong, M. Ghobadi, M. Balandat, S. Katti, A. Kazerouni, J. Leach, M. McKillop, Y. Zhang, Bow: First real-world demonstration of a firewall-based bayesian optimization system for wavelength deployment, in: Optical Fiber Communications Conference, IEEE, 2021, pp. 1–3.

Matthew Nance-Hall is a Ph.D. candidate in computer science from the University of Oregon where he was recognized with the University of Oregon Dissertation Research fellowship. His research interests are in computer networking with a focus wide-area network performance and network security.

Paul Barford (Fellow, IEEE) received the Ph.D. degree in computer science from Boston University in 2000. He is a Professor of Computer Science with the University of Wisconsin–Madison, where he is Associate Chair and the Founder and the Director of Wisconsin Advanced Internet Laboratory. His research interests are in computer networking with a focus on measurement and analysis of Internet performance, robustness, security, and topological structure. His most recent interest is in fiber-optic-based techniques for ground motion sensing. He is the Founder to three successful startup companies, and a Fellow of ACM.

Klaus-Tycho Foerster received the Ph.D. degree in computer science from ETH Zurich, Switzerland, in 2016. He is a Professor for Networked and Distributed Systems at TU Dortmund, Germany, in the Faculty of Computer Science. His research focuses on fundamental problems of networked and distributed systems, especially software-defined networks, optical networks, resilience, consistency, and fault-tolerance.

Ramakrishnan Durairajan is an Associate Professor in the Department of Computer Science at the University of Oregon. His research has been recognized with multiple NSF awards including the NSF CAREER award, a Ripple faculty fellowship, a UO faculty research award, and several best paper awards, and has been covered in several fora.