OptFlow: A Flow-based Abstraction for Programmable Topologies

Klaus-Tycho Foerster
Faculty of Computer Science
University of Vienna
Austria

Long Luo
University of Electronic
Science and Technology of China
P.R. China

Manya Ghobadi
Computer Science and Artificial
Intelligence Laboratory, MIT
USA

ABSTRACT

The rapid adoption of Reconfigurable Optical Add-Drop Multiplexers (ROADMs) is setting the stage for the dynamic reconfiguration of the network topology in optical backbones. The conventional approach to enable programmability in the physical layer requires solving a cross-layer optimization formulation that captures the interplay between the IP and optical layers. However, as the network scales, the complexity and run time of cross-layer optimization formulations grow prohibitively, resulting in heuristic-based solutions that sacrifice optimality for scalability. We propose a flow-based graph abstraction, called OptFlow, that is able to find the optimal allocation faster than a cross-layer optimization formulation. The key idea in OptFlow is that topology programmability is abstracted by "network flows," enabling service providers to use multi-commodity flow formulations, such as conventional Traffic Engineering, to solve a cross-layer optimization. OptFlow augments the physical graph and uses it as input to the unmodified flow-based Traffic Engineering algorithm, capturing a variety of IP-layer optimization goals such as max throughput, min hop count, and max-min fairness. Due to its flow-based nature, OptFlow inherently provides an abstraction for consistent network updates. To benchmark our key assumptions in OptFlow, we build a small testbed prototype consisting of four ROADMs. To evaluate the optimality and run time of large networks, we simulate five WAN topologies with up to 100 nodes and 390 links. Our results show that OptFlow matches the throughput performance of an optimal cross-layer formulation but has faster computation times. The run time speed-up of OptFlow increases as the network scales, with up to 8× faster execution times in our simulations.

CCS CONCEPTS

• **Networks → Programmable networks**; **Wide area networks**;
• **Theory of computation → Network flows**.

KEYWORDS

Traffic Engineering, Wide Area Networks, Optics

ACM Reference Format:
Klaus-Tycho Foerster, Long Luo, and Manya Ghobadi. 2020. OptFlow: A Flow-based Abstraction for Programmable Topologies. In *Symposium on SDN Research (SOSR '20), March 3, 2020, San Jose, CA, USA.* ACM, New York, NY, USA, 7 pages. https://doi.org/10.1145/3373360.3380840

SOSR '20, March 3, 2020, San Jose, CA, USA

ACM ISBN 978-1-4503-7101-8/20/03…$15.00
https://doi.org/10.1145/3373360.3380840

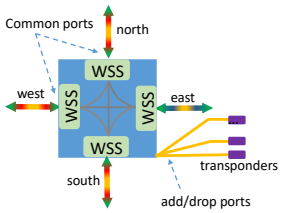
1 INTRODUCTION

Wide-area networks (WANs) are a key component of cloud providers' complex infrastructure. Today, all communication in WANs is carried over optical wavelengths. Optical backbones cost billions of dollars, and as online services are becoming an integral part of today's online services, these backbones need to be highly efficient. Inspired by reconfigurable topologies in data center networks [10, 12, 15, 21, 31, 36, 37, 39, 44, 49, 62], recent research has shown that enabling software-defined IP/optical backbones leads to greater efficiency and cost savings [22, 30, 45, 48]. As a result, service providers are looking toward enabling *Topology Programming* (TP) together with *Traffic Engineering* (TE) to boost efficiency [16, 25, 47].

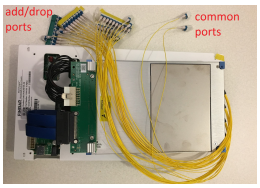
TP in a WAN is achieved using modern optical devices, such as ROADMs: Reconfigurable Optical Add-Drop Multiplexers [1, 51]. Flexible-grid capable ROADMs [50] are considered one of the most significant advances in Dense Wavelength Division Multiplexing (DWDM) systems technology over the last decade [16]. They allow incoming wavelengths to be switched from an input port to any output port, enabling the network operator to program the allocation of wavelengths across fibers during deployment and maintenance. While ROADMs are widely deployed in WANs [16] they are not being used to their full potential of dynamically adding/dropping wavelengths on-the-fly, because two challenges need to be solved simultaneously: (*i*) slow amplifiers, and (*ii*) unscalable cross-layer optimization formulations. The first challenge stems from an engineering perspective: amplifiers introduce several seconds of reconfiguration delay [22]. The good news is that recent research shows next-generation ROADMs are capable of programming wavelengths in tens to hundreds of milliseconds [13, 27, 30, 35]. However, even with fast amplifiers, the second challenge requires providers to throw away their existing TE formulations, write a new joint TP+TE formulation, and ensure it achieves the same goals as those intended by the original TE—a task that is so tedious that it hinders adoption. Service providers are understandably reluctant to embark on such an engineering-heavy endeavor without a reliable estimation of its ability to meet the required goals.

We solve this challenge by keeping the TE and TP problems decoupled while sliding TP under the TE algorithm. More specifically, we use a *flow-based* abstraction to program the topology without having to modify the TE formulation. Our abstraction, the "OptFlow graph," is an augmentation of the physical topology used as the new input to the TE algorithm. In our method, we mimic a TP+TE joint optimization by solving the unmodified TE formulation on the OptFlow graph instead of the original graph, hence achieving a more scalable solution.

The key concept in the OptFlow graph abstraction is that we represent wavelength programmability as "network flows," enabling us to use multi-commodity flow formulations to solve a cross-layer



(a) Depiction of a 4-way ROADM.



(b) The ROADM evaluation boards used in our testbed.

Figure 1: A ROADM uses Wavelength Selective Switches (WSS) to steer wavelengths between its ports. The common ports carry the light to/from other ROADMs. The add/drop ports are used to add/drop a wavelength to/from the common ports.

optimization. In doing so, we show that a large class of TE optimization objectives carry over to the OptFlow abstraction where solving the unmodified TE on OptFlow is the equivalent of solving the cross-layer IP/Optical formulation. In particular, OptFlow supports all linear throughput maximization TEs as well as max-min fairness TEs, such as SWAN [26] and B4 [28]. Furthermore, because OptFlow relies on network flows, instead of commonly used augmentation concepts such as node capacitation [23], several benefits of current TE practices, such as consistent updates [20], carry over transparently. In other words, the service provider does not have to implement a new consistent update strategy for the physical layer when adding or removing wavelengths.

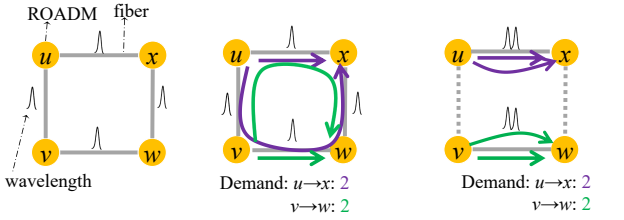
To evaluate the performance of OptFlow, we use synthetic traffic data with four different WAN topologies. Our results show that OptFlow matches the throughput of an optimal cross-layer integer linear program formulation while speeding up the run time up to a factor of 8. Finally, to showcase the feasibility of the OptFlow abstraction, we use a small testbed to benchmark key abstraction assumptions in practice and illustrate an end-to-end working scheme.

2 BACKGROUND AND OPPORTUNITIES

We begin with a brief overview of optical backbone networks with a focus on programmability and its opportunities.

Reconfigurable Optical Add/Drop Multiplexer. A ROADM is an optical device that combines multiple wavelengths into fiber cores while adding/dropping wavelengths to/from the existing multi-wavelength signal. This is achieved through the use of Wavelength Selective Switching (WSS) modules. ROADMs allow operators to dynamically reconfigure ports to carry any combination of wavelengths while adding/drop-ping any wavelength at any time [1, 6]. Fig. 1(a) illustrates the structure of a 4-way ROADM capable of steering wavelengths between its north, south, west, east common ports while adding/dropping any of the wavelengths. The wavelengths are modulated/demodulated from/to binary signals in the electrical domain. The total number of wavelengths assigned across all attached fibers is limited by the number of transponders. The frequency, modulation, and number of optical wavelengths that are multiplexed onto fiber depend on the technology [1, 3, 4, 6, 7].

Fig. 1(b) shows the evaluation board used in our testbed; it consists of 40 add/drop ports and 2 common ports [3]. ROADMs open the door to research on orchestrating IP and optical layers; recent research shows ROADMs are capable of programming wavelengths in in tens to hundreds of milliseconds [13, 30].



(a) Physical graph (b) TE throughput: 1+1 (c) (TE + TP) throughput: 2+2

Figure 2: Enabling topology programmability (TP) on top of traffic engineering (TE) can lead to throughput gains.

TP + TE on a reconfigurable graph can beat TE. Fig. 2 provides a simple example of how ROADMs can enable throughput gains by reprogramming wavelengths on adjacent fibers. Each node in Fig. 2(a) corresponds to a ROADM, and each edge represents a fiber. In this example, each ROADM has two wavelengths and by allocating one wavelength to each fiber, the operator ensures a connected static topology. Assume the traffic demand is as shown in Fig. 2(b), where nodes u and v have two units of traffic demand to x and w, respectively, and the arrows represent the TE routes. In this case, the maximum throughput (with optimal TE) is two units, thus leaving half of the demand unsatisfied. For simplicity, we assume each wavelength carries one unit of demand. Now assume TP and TE are working together as depicted in Fig. 2(c). By programming the topology and assigning two wavelengths to (u,x) and (v,w) fibers, the total throughput becomes four units (a 2 × gain over Fig. 2(b)).

3 A GRAPH ABSTRACTION FOR RECONFIGURABLE NETWORKS

Section 2 of this paper and [22, 29, 30, 40] suggest the advantages of programmable WANs, notably their ability to adapt to changes in traffic demand caused by, e.g., equipment failures [22]. Before we embrace programmable topologies, however, we need to address an important question:

What is the best approach for service providers to enable productionized TE algorithms [26, 28] with topology programmability?

We propose an abstraction that makes both layers indistinguishable from an optimization point of view. The TE can compute flow allocations on this abstract layer, and we can map the TE's solution to reconfigurations in the physical layer and flow routes in the network layer, thus achieving joint optimization.

3.1 Abstraction Concept

The intuition for our abstraction is presented in Fig. 3. Each node (ROADM) has a set of wavelengths (e.g., v in Fig. 3(a) has six wavelengths). For simplicity, in this example, we assume that the nodes can support the same number of incoming and outgoing wavelengths. Hence, v can transmit six wavelengths to neighboring nodes and also receive six wavelengths. To reflect this in our abstraction, we introduce two dummy nodes v_{in} and v_{out}, shown in Fig. 3(a), where all incoming traffic goes through v_{in} and all outgoing traffic has to pass v_{out}, and the edges have a capacity of six units. The number of dummy nodes is independent of the number of ROADM ports.

Now consider the graph in Fig. 3(b), with three nodes u, v, and w and three wavelengths on each fiber (a total of six wavelengths at v). Consider a non-symmetric traffic demand: $u \to w$: 4 and $w \to u$: 2.

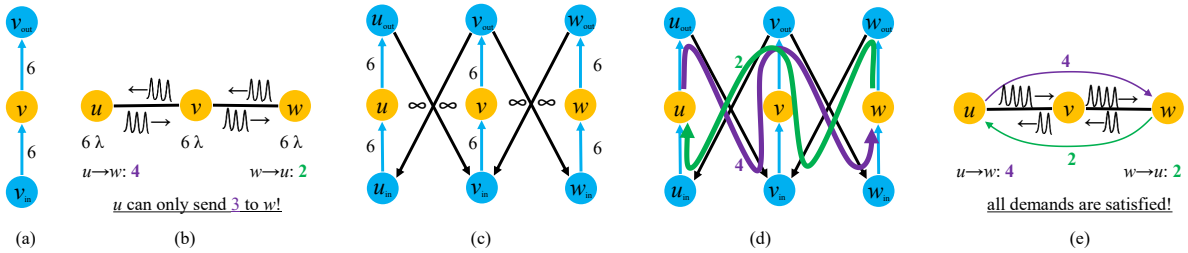


Figure 3: Illustration of the OptFlow graph abstraction idea, assuming each node can support six incoming and outgoing wavelengths. By solving the multi-commodity flow problem on (c), with the demands $u \to w : 4$, $w \to u : 2$, in (d), we obtain the wavelength allocation for v, as shown in (e): v assigns 2 incoming and 4 outgoing wavelengths to the fiber to w, and 4 incoming and 2 outgoing wavelengths to the fiber to u.

Given the reprogrammability at v, we would like to assign the wavelengths on v's adjacent fibers to support this traffic demand.

Note that the current assignment in Fig. 3(b), with three wavelengths on each fiber, cannot satisfy the demand $u \to w : 4$. To do so, we repeat the process of Fig. 3(a) by adding dummy nodes for each node in the graph and connecting them in a way that maps the connectivity in the original graph (shown in Fig. 3(c)). For simplicity, we set ∞ as the edge capacities between the in- and out-nodes. We later adapt them to the maximum number of wavelengths permitted on the fiber. The result is the OptFlow graph corresponding to Fig. 3(b). We prove that solving the multi-commodity flow problem on the OptFlow graph also indicates how wavelengths need to be reprogrammed. Fig. 3(d) shows the optimal flow allocations on the OptFlow graph ($u \to w : 4$ and $w \to u : 2$). The output of the maximized throughput on Fig. 3(d), translated back to wavelength allocations and IP routing, is shown in Fig. 3(e). This shows how solving TE on the OptFlow graph can jointly optimize both TP and TE problems.

3.2 Optimal Throughput

Our abstraction concept needs to overcome additional challenges before achieving joint layer optimization: the resulting wavelength assignments could be fractional, as optimal throughput solutions will require fractional flow sizes and integral wavelength assignments.

Dual flows. Not surprisingly, finding optimal joint solutions for integral wavelength assignments is NP-hard, even for polynomial TE schemes (we omit the proof). However, network operators solve NP-hard problems in practice all the time, in particular by routing flows along paths under capacity constraints. Hence, we believe that the natural answer to this challenge is to represent the wavelengths as flows of unit size, as TE algorithms know how to handle flows.

We use a primal-dual computation where each wavelength assignment (the dual) enforces a threshold on the throughput. The task is to find the optimal dual that allows maximum throughput. We call these wavelength flows *dual flows* and add them to the augmented input of the TE. Fig. 4(a) outlines an example where the traffic demands are $v \to u : 2.5$ and $v \to w : 3.5$. Assume v has only six wavelengths available, and the capacity of each fiber (C) is also six wavelengths. Then, the fractional solution computed in Fig. 4(b) is not feasible because it would require $3 + 4 = 7$ wavelengths. To solve this problem, we add six high-priority dual flows (unit sized) as shown in Fig. 4(c) from v to either u or w. The key idea is that the IP traffic flows combined with the dual flows may not exceed edge capacities. A possible optimal TE solution is shown in Fig. 4(d); v can still send 3.5 units of traffic to w, but only 2 units

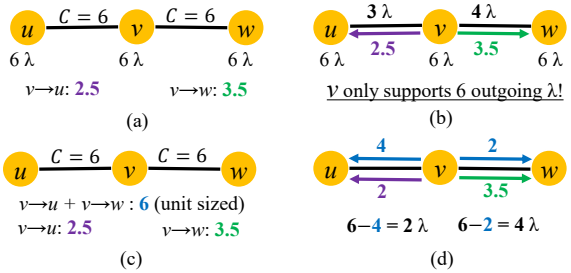


Figure 4: Obtaining optimal solutions. In (a), v has 4 outgoing wavelengths and demands of $v \to u : 2.5$, $v \to w : 3.5$. Using the abstraction idea from Fig. 3 results in the output in (b): all demands can be satisfied, but $\lceil 2.5 \rceil + \lceil 3.5 \rceil = 7$ wavelengths are required! To enforce wavelength constraints, we introduce unit sized high-priority dual flows in (c), which can be distributed among u and w. After assigning those dual flows, the remaining capacities to u, w are integral. Hence, the resulting IP flow allocation, shown in (d), is feasible and achieves optimal throughout under the network constraints.

to u. Separating the layers, we obtain 4 wavelengths from v to w and 2 wavelengths from v to u. The OptFlow graph abstraction construction is formalized in the following Theorem:

Theorem 1. *Let G be a programmable topology with demand matrix D. A OptFlow graph G_R with demand matrix D_R can be created in polynomial time s.t. the solution of multi-commodity flow throughput optimization on (G_R, D_R) translates in polynomial time to throughput-optimal routing and wavelength allocation on G for D.*

Augmenting the graph and demand matrix. The proof (sketched here) relies on a combinatorial construction that enables a standard multi-commodity flow solver to handle the concepts illustrated in Fig. 4. To this end, for each node v, we create dummy sources v'_{in}, v'_{out} and destinations v''_{in}, v''_{out} for its dual flows, one pair for the incoming and one for the outgoing wavelengths. The dual flows can then be routed along so-called dual flow paths, which enforce the wavelength integrality constraints. The idea is that a routing of the dual flows along other paths cannot increase the overall throughput. This graph augmentation is illustrated in Algorithm 1.

Input: Physical graph $G = (V, E, \sigma)$, demand matrix D
Output: OptFlow graph $G_R = (V_R, E_R)$, demand matrix D_R

(1) $V_R = V$, $E_R = E$, $D_R = D$
(2) $\forall v \in V$: $V_R = V_R \cup \{v'_{in}, v''_{in}, v'_{out}, v''_{out}\}$
(3) $\forall e = (u, v) \in E$: $E_R = E_R \cup \{\textit{dual flow paths}(v', v'')\}$
(4) $\forall v \in V$: $D_R = D_R \cup \{\textit{dual flows}(v', v'')\}$

Algorithm 1: OptFlow graph G_R and D_R construction.

Input: Physical graph $G=(V,E,\sigma)$, demand matrix D
Output: Flow $\mathcal{F}$, wavelength allocation Λ with capacity $c(e)$ $\forall e \in E$
(1) Compute OptFlow graph $G_R=(V_R,E_R)$ and D_R using Algorithm 1
(2) $\mathcal{F}_R \leftarrow$ solve TE on G_R
(3) $\mathcal{F} \leftarrow \mathcal{F}_R$ minus dual flows
(4) Compute wavelength allocation Λ for all $(v,e)=e \in E$:
  (a) Set $\lceil c(e) - \textit{dual flows} \rceil$ as wavelength allocation on e

Algorithm 2: OptFlow graph abstraction.

Running the abstraction. We solve the TE on the augmented graph G_R and traffic matrix D_R. We remove the dual flows from the resulting multi-commodity flow F_R, obtaining the routing $\mathcal{F}$ for the non-augmented graph. To obtain the transponder-to-fiber allocations for the wavelengths, we use the size of $\mathcal{F}$ on the edges, rounded up to integral values. These steps are in Algorithm 2.

3.3 Tackling Reconfiguration Delay

During optical layer reconfiguration, wavelengths become temporarily unavailable and packets can be dropped. We discuss next how our abstraction can handle such delays.

Consistent network updates. The idea of consistent network updates [19], designed to deal with transient congestion and packet drops in the IP layer, cannot yet handle a cross-layer migration of wavelengths and traffic flows. But we argue that graph abstractions enable us to use consistent update techniques *as is* by hiding wavelength allocations as flow allocations. As such, consistent network updates are performed on the OptFlow graph itself, where a migration of dual flows corresponds to shifting wavelengths by adjusting transponder-to-fiber mappings. When both old and new network states are provided as an input to consistent flow migration schemes, intermediate network states are computed [61], corresponding to consistent cross-layer network updates. Both wavelength allocations and IP traffic route changes are covered, when 1) changing the path of dual flows maps directly onto changing wavelength assignments and 2) changing the path of IP traffic in our abstraction maps onto route changes in the IP layer. In other words, a single network update can contain both optical and IP layer changes, providing an abstraction for optimal (e.g., minimum schedule) cross-layer migration. In this context, it would be interesting to investigate reconfigurations where, e.g., the number of changed wavelength assignments is minimized.

3.4 Expanding the Abstraction Coverage

We have shown how OptFlow can express throughput objectives such as multi-commodity flow (MCF). However, the quality of an abstraction is also measured by its expressiveness. To this end, we describe how OptFlow can be extended to further scenarios.

Bidirectional wavelengths. So far, our abstraction has only handled unidirectional wavelengths, but current deployments feature bidirectional wavelength technology. Instead of being decoupled, sending and receiving components are bundled in pairs by the vendors for convenience reasons. OptFlow can be adapted to bidirectional wavelengths by making the dual flows bidirectional. The enabler is a small combinatorial gadget that enforces the dual flow to pass through both directions of the augmented edge, even though the TE scheme is implicitly unaware of this restriction for the unidirectional flow. Using further combinatorial extensions, OptFlow can also incorporate wavelengths of different capacities and remove the requirement of a higher priority class for dual flows.

TE objective	Support	Key idea	Restriction
Throughput	✓	dual flows	none
Concurrent sharing [52]	✓	bottleneck edges for dual flows	none
Max-min fairness [46]	✓	splitting dual flow commodities	max dual flow demand ≤ min real flow
Flow priorities [9]	✓	dual flows highest priority	assign priorities as input
Hierarchical bandwidth [33]	✓	dual flows at top priority weight	assign hierarchies as input
Path properties [26]	✓	combinatorial graph extension of weight 0	edge lengths of 0 allowed
FFC [38] (faults)	●	dual flow paths not protected	protection scheme as input
k-shortest paths [24]	●	combinatorially unrestricted dual flow paths	$k \geq \Delta$ (max degree)
Min-max load [34]	✗	dual flows fill up the edge load	future work

Table 1: Coverage of OptFlow, with key ideas and restrictions. ✓ denotes *full* support, ● *partial* support, and ✗ *no* current support.

Fairness and further considerations. Max-min fairness [46] is captured by splitting dual flows into multiple commodities, each originating from its own source. OptFlow obtains max-min fairness when the largest dual flow has less demand than the smallest non-dual flow allocation. To capture concurrent flow objectives [52] in OptFlow, i.e., to consider the fairness of the fraction of demand allocated instead of absolutes, we route dual flows through bottleneck edges, whose capacity corresponds to transponder numbers. By setting the dual flow demands to high values, the fairness objective enforces the correct allocation of dual flows. Due to space constraints, we summarize further extension proposals in Table 1.

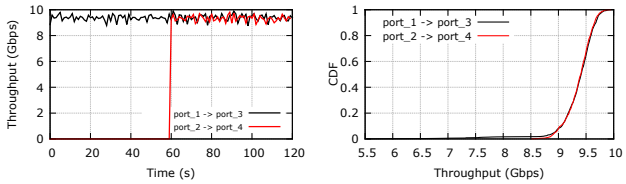
4 TESTBED EVALUATION

In this section, we first investigate further practical concerns of reconfigurable WANs, showing an end-to-end working OptFlow scheme in a small testbed. This is followed by performance simulations, to benchmark the impact [11] of our abstraction.

4.1 Evaluation

Practicality. Abstraction-based topology programming depends on two fundamental assumptions: (*i*) the time it takes to reprogram the topology is short; (*ii*) reprogramming wavelengths in the topology is not destructive to the IP traffic of other wavelengths. Together, these assumptions enable the TE engine to handle wavelength reprogramming similar to the IP layer's flow reprogramming. With respect to the first, recent research already shows ROADMs are capable of programming wavelengths in tens to hundreds of milliseconds [13, 30].

Impact on IP traffic. We use a testbed to verify the second assumption. Our testbed consists of four Finisar evaluation boards, each with 40 add/drop ports supporting data rates of 10, 40, 100, and 400 Gbps [3]. To generate traffic with different wavelengths, we use 10 Gbps tunable DWDM transceivers from Finisar [5] plugged into a 7050s Arista switch. We run an experiment where we connect two ROADMs through their common port, adding a new wavelength



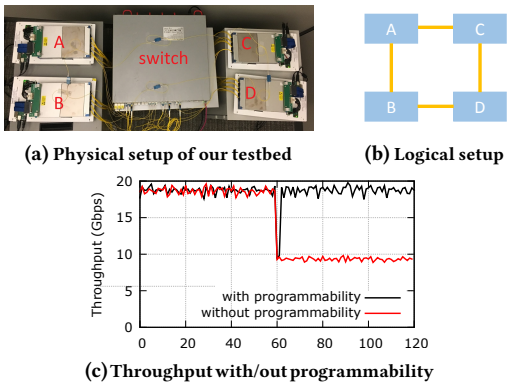
(a) A new wavelength does not disturb the existing wavelength's throughput. **(b) CDF of throughput.**

Figure 5: Adding/dropping wavelengths does not impact the throughput of other wavelengths sharing the same fiber.

while a flow is already running on an established different wavelength. Figure 5(a) shows that both flows are capable of achieving 10 Gbps and adding a new wavelength does not negatively impact the throughput of the on-going flow. We repeat this experiment 1000 times and measure the throughput of both flows (during the time they are both active). Figure 5(b) shows the CDF of throughput for both flows is overlapping, suggesting that programming wavelengths does not have a destructive effect on ongoing traffic.

Putting it all together: An end-to-end working scheme We demonstrate the power of topology programmability in practice by connecting four ROADMs to create a rectangular topology, as shown in Fig. 6(a). The logical setup is provided in Fig. 6(b). We start the experiment by generating 20 Gbps of traffic between nodes A and C using two wavelengths.

One wavelength takes the $A \rightarrow C$ direct edge, and the other takes the $A \rightarrow B \rightarrow D \rightarrow C$ path. This way, the topology can support the entire 20 Gbps demand; see Fig. 6(c). Next, we simulate a fiber cut by manually disconnecting the fiber between A, B. This will cause a loss of capacity and throughput is dropped to 10 Gbps. However, we detect the loss of light and drop the wavelength on the $A \rightarrow B$ edge and add it to the $A \rightarrow C$ edge; this allows the throughput to be restored to 20 Gbps (black curve in Fig. 6(c)). We assume that the wavelengths do not collide; i.e., they use different frequencies. Without programmability, the throughput remains at 10 Gbps (red curve in Fig. 6(c)), and slower reactions negatively impact the system's performance.



(a) Physical setup of our testbed **(b) Logical setup**

(c) Throughput with/out programmability

Figure 6: Enabling the OptFlow abstraction can mitigate the impact of fiber cuts by programming idle wavelengths based on the current traffic demand.

Topology	#Nodes	#Links
Google (G-Scale) [28]	12	38
Internet2 [30]	40	100
IDN [26]	40	390
AS 1221 (Telstra) [2]	104	306

Table 2: Network topologies used in our simulations

4.2 Performance Simulations

In this section we evaluate the performance of OptFlow in data-driven simulations. To this end, we pick 4 commonly used WAN topologies with synthetic traffic data, analogous to [60], to stress-test the performance of OptFlow. The WAN topologies vary in size and connectivity to get a good performance estimate; see Table 2.

We compare throughput and solver computation times of OptFlow to those of an optimal cross-layer formulation. To do so, we choose k-shortest path routing and maximize the total throughput. We pick a standard Integer-Linear-Program (ILP) formulation with MOSEK [8] to perform cross-layer optimization, where traffic and directed wavelength allocation are jointly optimized. OptFlow, in turn, uses the same program, but without wavelength allocation—that part is handled by the built-in abstraction. In other words, the traffic engineering formulation is applied to the OptFlow graph, oblivious to the physical layer programmability.

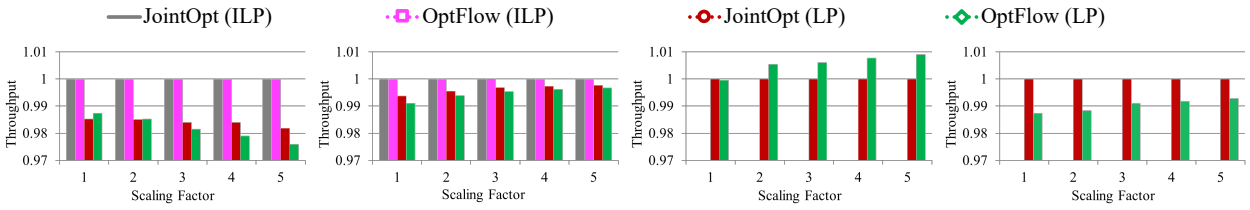
We use synthetic models to generate traffic demands and assume one demand between 10-50% of the node pairs, integrating the average demand under an exponential distribution with a mean of 200 Gbps. We use different scaling factors (from 1 to 5, 20 runs) to scale the #traffic demands. Per node, we set 16× its degree as #transponders, where the max number of 10Gbps wavelengths on each directed edge is randomly generated from [10,60].

To speed up the computation, but at the cost of reduced throughput, we relax the integral wavelength formulation, as done via rounding in [40]. We can directly apply these ideas to OptFlow, by allowing the dual flows to be fractional as well. By so doing, we lose, at most, one wavelength per node for each connected fiber, yielding a good approximation, as over 90% of the wavelengths remain deployable.

We plot the throughput and run time results of the cross-layer (JointOpt) and OptFlow simulations in Figs. 7 and 8; only the LP variants are run in both larger topologies. The throughput of the LP formulations differ by only about 1% in Figs. 7a to 7d and are, at most, 2.5% smaller than the ILP variants, whereas the ILP results match, as seen in Figs. 7b and 7a. Fig. 8 shows that the OptFlow abstraction improves the run time in all 4 topologies, with the benefit increasing with topology size. Even though the graph abstraction induces a small overhead, the smaller number of constraints and variables in OptFlow greatly outweighs this downside.

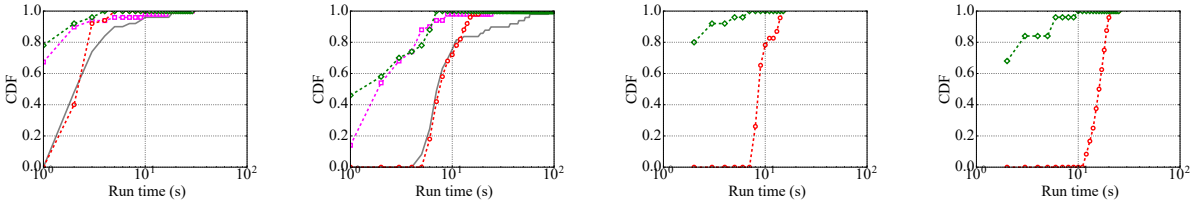
For the smallest topology with 12 nodes in Fig. 8a, the cross-layer formulation is faster than OptFlow in $\approx 10\%$ of the cases, but OptFlow is still over 4 times faster in average, for both ILP and LP variants. For Internet2 in Fig. 8b, OptFlow is faster over 95% of the time, with an average speed of over 6× faster.

In the topologies with significantly more links, OptFlow performs even better. The average speed increases over 7× for IDN in Fig. 8c and over 8× for AS 1221 in Fig. 8d. In both cases, OptFlow is always faster, especially in IDN which has the highest number of links per node: here, OptFlow is always at least 2× faster.



(a) Google (b) Internet2 (c) IDN (d) AS 1221 (Telstra)

Figure 7: Throughput comparison, normalized by JointOpt (ILP) in 7a and 7b, and by JointOpt (LP) in 7c and 7d. Note that 7c and 7d only contain the LP variants, as the run time of the ILP formulation on the large networks is significantly longer. As we only consider non-optimal LP formulations in 7c and 7d, either one can provide more throughput, e.g., OptFlow in 7c and the cross-layer formulation in 7d.



(a) Google (b) Internet2 (c) IDN (d) AS 1221 (Telstra)

Figure 8: Run time comparison taken from the simulations in Fig. 7, using the same color keys. Note the *logarithmic time* axes. 8c and 8d only contain the LP variants, as the run time of the ILP formulation on the large networks is significantly longer.

5 RELATED WORK

Our work builds on several lines of related research. The work most closely related to ours is by Jin et al. [30] on cross-layer optimization of IP and optical layers. These authors show optical reconfiguration can provide latency gains for deadline-driven bulk transfers. A follow-up study examines the makespan of scheduling algorithms in programmable topologies, where transfers must be be across single hops [29]. The authors consider both online and offline variants, analyzing competitive ratios of scheduling problems. This work is extended by Dinitz and Moseley [14]. More recently, Long et al. [40, 41] employ a partially relaxed ILP in the multicast setting of [30]. These works optimize the IP and optical layers together with a cross-layer algorithm, whereas we provide a mechanism to use current TEs without having to rewrite a cross-layer TE.

We are inspired by the large body of work on reconfigurable topologies in data center networks, as well. Prior work shows the benefits of reconfigurable topologies in data center networks when wireless/optical edges are added to the electrical topology [15, 17, 18, 31, 36, 37, 49, 62], or optical data center interconnects are created [10, 12, 21, 39, 44]. The main difference between the WAN and data centers is the geographical scale of the WAN. While the underlying assumption in a reconfigurable data center is that optical edges are either connected via a single optical switch or have line-of-sight, the sheer scale of the WAN creates physical limitations on the fiber paths and the physical topology itself making it impossible to directly adopt the data center proposals. Similarly, Expander-based [32, 43, 55, 56, 58] topologies are becoming popular, but realizing such topologies is extremely challenging in the WAN: we are limited by the placement and cost of the optical fibers – cabling suggestions which leverage the close proximity of the racks and bundle the cables together, like those suggested in [56, 58], are not achievable in the WAN. Optical multicast has also attracted more interest lately [42, 57, 59] in data center networks, and while the concepts are not directly transferable to the WAN, it would be interesting to see if OptFlow could be extended to cover multicast.

Lastly, recent work on rate adaptive links shows further reconfiguration possibilities in the WAN [53, 54]. OptFlow can be extended in this direction, by combinatorially representing rate adaptivity as different flow sizes.

6 CONCLUSION

In this paper, we take the first steps toward the practical deployment of a programmable physical layer in the WAN by presenting the OptFlow abstraction to enable current IP-layer TE algorithms to perform cross-layer optimization.

We show the expressiveness of OptFlow captures a wide spectrum of TE objectives and can also handle reconfiguration delay via consistent network updates. Our small testbed results positively benchmark key assumptions in practice, showing an end-to-end working scheme. Moreover, performance evaluations with synthetic traffic data show that OptFlow matches the throughput of optimal cross-layer formulations, at significantly better computation times. In future work, we plan to benchmark and analyze more TE objectives, explore the handling of amplifiers, and evaluate further real-world traffic patterns and scenarios.

Acknowledgements. We thank Anja Feldmann for shepherding our paper and the anonymous reviewers for providing valuable feedback. This project has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 864228, AdjustNet: Self-Adjusting Networks).

REFERENCES

[1] [n.d.]. ADVA ROADMs. https://adva.com/en/products/technology/roadm.
[2] [n.d.]. DEFO. https://sites.uclouvain.be/defo/.
[3] [n.d.]. Dual Wavelength Selective Switch (WSS). https://www.finisar.com/roadms-wavelength-management/fws0120bscfal.
[4] [n.d.]. Open ROADM Multi-Source Agreement. http://openroadm.org.
[5] [n.d.]. Tunable DWDM transceivers. https://www.finisar.com/optical-transceivers/ftlx6872mcc.
[6] 2014. Infinera introduces flexible grid 500G super-channel ROADM. http://www.gazettabyte.com/home/2014/3/14/infinera-introduces-flexible-grid-500g-super-channel-roadm.html.
[7] 2018. Next-Generation ROADM Networks. https://resource.lumentum.com/s3fs-public/technical-library-items/next-genroadm-wp-oc-ae.pdf.
[8] 2019. MOSEK. https://www.mosek.com/.
[9] D. Awduche, J. Malcolm, J. Agogbua, M. O'Dell, and J. McManus. 1999. Requirements for Traffic Engineering Over MPLS, RFC:2702.
[10] Navid Hamed Azimi, Zafar Ayyub Qazi, Himanshu Gupta, Vyas Sekar, Samir R. Das, Jon P. Longtin, Himanshu Shah, and Ashish Tanwer. 2014. FireFly: a reconfigurable wireless data center fabric using free-space optics. In *SIGCOMM*.
[11] Andreas Blenk, Arsany Basta, Wolfgang Kellerer, and Stefan Schmid. 2019. On the Impact of the Network Hypervisor on Virtual Network Performance. In *Networking*. IEEE.
[12] Li Chen, Kai Chen, Zhonghua Zhu, Minlan Yu, George Porter, Chunming Qiao, and Shan Zhong. 2017. Enabling Wide-Spread Communications on Optical Fabric with MegaSwitch. In *NSDI*. USENIX.
[13] Angela L. Chiu et al. 2012. Architectures and Protocols for Capacity Efficient, Highly Dynamic and Highly Resilient Core Networks (Invited). *IEEE/OSA Journal of Optical Communications and Networking* 4, 1 (January 2012), 1–14.
[14] Michael Dinitz and Benjamin Moseley. 2020. Scheduling for Weighted Flow and Completion Times in Reconfigurable Networks. In *INFOCOM*.
[15] Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2010. Helios: a hybrid electrical/optical switch architecture for modular data centers. In *SIGCOMM*. ACM.
[16] Mark Filer, Jamie Gaudette, Monia Ghobadi, Ratul Mahajan, Tom Issenhuth, Buddy Klinkers, and Jeff Cox. 2016. Elastic Optical Networking in the Microsoft Cloud (invited). *J. Opt. Commun. Netw.* 8, 7 (Jul 2016), A45–A54.
[17] Klaus-Tycho Foerster, Manya Ghobadi, and Stefan Schmid. 2018. Characterizing the algorithmic complexity of reconfigurable data center architectures. In *ANCS*.
[18] Klaus-Tycho Foerster, Maciej Pacut, and Stefan Schmid. 2019. On the Complexity of Non-Segregated Routing in Reconfigurable Data Center Architectures. *Computer Communication Review* 49, 2 (2019), 2–8.
[19] Klaus-Tycho Foerster, Stefan Schmid, and Stefano Vissicchio. 2019. Survey of Consistent Software-Defined Network Updates. *IEEE Communications Surveys and Tutorials* 21, 2 (2019), 1435–1461.
[20] K.-T. Foerster and S. Schmid. 2019. Survey of Reconfigurable Data Center Networks: Enablers, Algorithms, Complexity. *SIGACT News* 50, 2 (July 2019), 62–79.
[21] Monia Ghobadi, Ratul Mahajan, Amar Phanishayee, Nikhil R. Devanur, Janardhan Kulkarni, Gireeja Ranade, Pierre-Alexandre Blanche, Houman Rastegarfar, Madeleine Glick, and Daniel C. Kilper. 2016. ProjecToR: Agile Reconfigurable Data Center Interconnect. In *SIGCOMM*.
[22] Jennifer Gossels, Gagan Choudhury, and Jennifer Rexford. 2019. Robust network design for IP/optical backbones. *J. Opt. Commun. Netw.* 11, 8 (Aug 2019), 478–490.
[23] Frieda Granot and Refael Hassin. 1986. Multi-terminal maximum flows in node-capacitated networks. *Discrete Applied Mathematics* 13, 2-3 (1986), 157–163.
[24] Jiayue He and Jennifer Rexford. 2008. Toward internet-wide multipath routing. *IEEE Network* 22, 2 (2008), 16–21.
[25] Tad Hofmeister, Vijay Vusirikala, and Bikash Koley. 2016. How can Flexibility on the Line Side Best be Exploited on the Client Side?. In *Optical Fiber Communication Conference*. Optical Society of America, W4G.4.
[26] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri, and Roger Wattenhofer. 2013. Achieving high utilization with software-driven WAN. In *SIGCOMM*. ACM, 15–26.
[27] Yishen Huang, Craig L. Gutterman, Payman Samadi, Patricia B. Cho, Wiem Samoud, Cédric Ware, Mounia Lourdiane, Gil Zussman, and Keren Bergman. 2017. Dynamic mitigation of EDFA power excursions with machine learning. *Opt. Express* 25, 3 (Feb 2017), 2245–2258. https://doi.org/10.1364/OE.25.002245
[28] Sushant Jain, Alok Kumar, et al. 2013. B4: experience with a globally-deployed software defined wan. In *SIGCOMM*. ACM.
[29] Su Jia, Xin Jin, Golnaz Ghasemiesfeh, Jiaxin Ding, and Jie Gao. 2017. Competitive Analysis for Online Scheduling in Software-Defined Optical WAN. In *INFOCOM*.
[30] Xin Jin, Yiran Li, Da Wei, Siming Li, Jie Gao, Lei Xu, Guangzhi Li, Wei Xu, and Jennifer Rexford. 2016. Optimizing Bulk Transfers with Software-Defined Optical WAN. In *SIGCOMM*. ACM.
[31] Srikanth Kandula, Jitendra Padhye, and Paramvir Bahl. 2009. Flyways To De-Congest Data Center Networks. In *HotNets*. ACM SIGCOMM.
[32] Simon Kassing, Asaf Valadarsk, Gal Shahaf, Michael Schapira, and Ankit Singla. 2017. Beyond fat-trees without antennae, mirrors, and disco-balls. In *SIGCOMM*.
[33] Alok Kumar et al. 2015. BwE: Flexible, Hierarchical Bandwidth Allocation for WAN Distributed Computing. In *SIGCOMM*. ACM.
[34] Amund Kvalbein, Constantine Dovrolis, and Chidambaram Muthu. 2009. Multipath load-adaptive routing: putting the emphasis on robustness and simplicity. In *ICNP*. IEEE Computer Society.
[35] Y. Li and D. C. Kilper. 2018. Optical physical layer SDN [invited]. *IEEE/OSA Journal of Optical Communications and Networking* 10, 1 (Jan 2018), A110–A121.
[36] He Liu et al. 2014. Circuit Switching Under the Radar with REACToR. In *NSDI*.
[37] He Liu et al. 2015. Scheduling techniques for hybrid circuit/packet networks. In *CoNEXT*. ACM.
[38] Hongqiang Harry Liu, Srikanth Kandula, Ratul Mahajan, Ming Zhang, and David Gelernter. 2014. Traffic engineering with forward fault correction. In *SIGCOMM*.
[39] Yunpeng James Liu, Peter Xiang Gao, Bernard Wong, and Srinivasan Keshav. 2014. Quartz: a new design element for low-latency DCNs. In *SIGCOMM*. ACM.
[40] Long Luo, Klaus-Tycho Foerster, Stefan Schmid, and Hongfang Yu. 2019. DaRTree: deadline-aware multicast transfers in reconfigurable wide-area networks. In *IWQoS*. ACM.
[41] Long Luo, Klaus-Tycho Foerster, Stefan Schmid, and Hongfang Yu. 2020. Deadline-Aware Multicast Transfers in Software-Defined Optical Wide-Area Networks. *IEEE Journal on Selected Areas in Communications* (2020).
[42] Long Luo, Klaus-Tycho Foerster, Stefan Schmid, and Hongfang Yu. 2020. SplitCast: Optimizing Multicast Flows in Reconfigurable Datacenter Networks. In *INFOCOM*. IEEE.
[43] William M. Mellette, Rajdeep Das, Yibo Guo, Rob McGuinness, Alex C. Snoeren, and George Porter. 2020. Expanding across time to deliver bandwidth efficiency and low latency. In *NSDI*.
[44] William M. Mellette, Rob McGuinness, Arjun Roy, Alex Forencich, George Papen, Alex C. Snoeren, and George Porter. 2017. RotorNet: A Scalable, Low-complexity, Optical Datacenter Network. In *SIGCOMM*.
[45] Annalisa Morea and Andrea Paparella. 2016. Cost and Algorithm Complexity of Elastic Optical Networks, In Optical Fiber Communication Conference. *Optical Fiber Communication Conference*, M2K.4.
[46] Dritan Nace and Michal Pióro. 2008. Max-min fairness and its applications to routing and load-balancing in communication networks: A tutorial. *IEEE Communications Surveys and Tutorials* 10, 1-4 (2008), 5–17.
[47] Shoichiro Oda, Masatake Miyabe, Setsuo Yoshida, Toru Katagiri, Yasuhiko Aoki, Jens C. Rasmussen, Martin Birk, and Kathy Tse. 2016. Demonstration of an Autonomous Software Controlled Living Optical Network that Eliminates the Need for Pre-planning. In *Optical Fiber Communication Conference*. W2A.44.
[48] Panos Papanikolaou, Kostas Christodoulopoulos, and Emmanuel (Manos) Varvarigos. 2016. Joint Multilayer Planning of Survivable Elastic Optical Networks. In *Optical Fiber Communication Conference*. Optical Society of America, M2K.3.
[49] George Porter, Richard D. Strong, Nathan Farrington, Alex Forencich, Pang-Chen Sun, Tajana Rosing, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2013. Integrating microsecond circuit switching into the data center. In *SIGCOMM*.
[50] Peter Roorda and Brandon Collings. 2008. Evolution to Colorless and Directionless ROADM Architectures. In *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference*. Optical Society of America, NWE2.
[51] Akio Sahara, Yukio Tsukishima, Tetsuo Takahashi, Youhei Okubo, Kazuhisa Yamada, Kazuhiro Matsuda, and Atsushi Takada. 2009. Demonstration of Colorless and Directed/Directionless ROADMs in Router Network. In *Optical Fiber Comm. Conference and National Fiber Optic Engineers Conference*. NMD2.
[52] Farhad Shahrokhi and D. W. Matula. 1990. The Maximum Concurrent Flow Problem. *J. ACM* 37, 2 (April 1990), 318–334.
[53] Rachee Singh, Monia Ghobadi, Klaus-Tycho Foerster, et al. 2017. Run, Walk, Crawl: Towards Dynamic Link Capacities. In *HotNets*.
[54] Rachee Singh, Monia Ghobadi, Klaus-Tycho Foerster, Mark Filer, and Phillipa Gill. 2018. RADWAN: Rate Adaptive Wide Area Network. In *SIGCOMM*. ACM.
[55] Ankit Singla. 2016. Fat-FREE Topologies. In *HotNets*.
[56] Ankit Singla, Chi-Yao Hong, Lucian Popa, and P. Brighten Godfrey. 2012. Jellyfish: Networking Data Centers Randomly. In *NSDI*.
[57] Xiaoye Steven Sun and TS Eugene Ng. 2017. When creek meets river: Exploiting high-bandwidth circuit switch in scheduling multicast data. In *ICNP*.
[58] Asaf Valadarsky, Gal Shahaf, Michael Dinitz, and Michael Schapira. 2016. Xpander: Towards Optimal-Performance Datacenters. In *CoNEXT*.
[59] Yiting Xia, TS Eugene Ng, and Xiaoye Steven Sun. 2015. Blast: Accelerating high-performance data analytics applications by optical multicast. In *INFOCOM*.
[60] Hong Zhang, Kai Chen, Wei Bai, Dongsu Han, Chen Tian, Hao Wang, Haibing Guan, and Ming Zhang. 2017. Guaranteeing Deadlines for Inter-Data Center Transfers. *IEEE/ACM Trans. Netw.* 25, 1 (2017), 579–595.
[61] Jiaqi Zheng, Hong Xu, Guihai Chen, and Haipeng Dai. 2015. Minimizing Transient Congestion during Network Update in Data Centers. In *ICNP*.
[62] Xia Zhou, Zengbin Zhang, Yibo Zhu, Yubo Li, Saipriya Kumar, Amin Vahdat, Ben Y. Zhao, and Haitao Zheng. 2012. Mirror mirror on the ceiling: flexible wireless links for data centers. In *SIGCOMM*. ACM, 443–454.