HEDGE: Traffic Engineering with Probabilistic Link Capacities

Arjun Devraj[†], Bill Owens[*], Umesh Krishnaswamy[§], Ying Zhang[¶], Rachee Singh[†]
[†]*Cornell University* [*]*NYSERNet* [§]*Microsoft* [¶]*Meta*

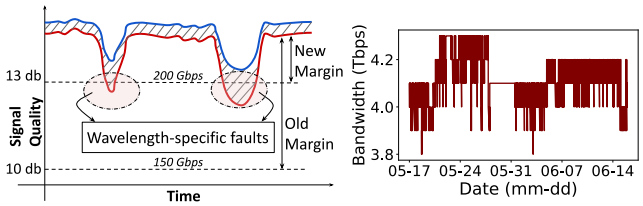
Abstract

Cloud providers have adopted higher modulation formats to achieve higher data-rate wavelengths in their optical wide-area networks. However, higher modulation formats reduce signal quality margins, making wavelengths more susceptible to wavelength-specific faults (WSFs)—temporary faults that selectively affect certain wavelengths while others remain unaffected, even though they all share the same optical fiber and equipment. WSFs cause the capacity of inter-datacenter links to fluctuate, frequently disrupting traffic engineering systems. We propose HEDGE, a system that mitigates the effects of WSFs by implementing link-local resilience and global network-wide resilience against WSFs. For local resilience, HEDGE provisions inter-datacenter links with a guaranteed minimum capacity and availability target, in spite of WSFs, while using the fewest possible constituent wavelengths. For global resilience, HEDGE optimally balances throughput and availability while allocating flows on a stochastic wide-area network with fluctuating link capacities. HEDGE sustains equivalent throughput with state-of-the-art traffic engineering systems, while dropping 12.2× less network flow in worst-case scenarios and reducing disruptions to tunnel allocations by 622× in spite of a rapidly changing topology. [1]

1 Introduction

Optical fiber underpins all long-haul connectivity in wide-area networks (WANs). Large cloud providers, Internet service providers (ISPs) and content distribution networks (CDNs) spend billions of dollars to provision and operate their long-haul fiber deployments [11, 20, 22, 42]. Therefore, to get the most out of their scarce and expensive fiber infrastructure, network operators have been pushing the limits of data transmission rates over long-haul fiber-optic wavelengths [39, 45]. These efforts have led to the deployment of wavelengths with higher modulation formats (*e.g.,* 16-QAM, 64-QAM) to achieve higher per-wavelength data rates while using the same amount of optical spectrum in cloud WANs [42].

High modulation formats reduce SNR margins. While higher modulation formats improve the spectral efficiency of fiber by packing more bits in each wavelength, their adoption has come at a cost to cloud operators. Higher modulation formats demand superior signal quality, measured in signal-to-noise ratio (SNR). This reduces the margin between a wavelength's operational SNR and the minimum SNR needed to sustain error-free communication (Fig. 1a).



(a) WSFs and low SNR margins. (b) Fluctuation in LAG capacity.

Figure 1: 1a shows new SNR margins resulting from the deployment of higher modulation formats have made some wavelengths (red wave) more susceptible to failures since they suffer higher losses compared to other wavelengths (blue wave). 1b shows the effect of such wavelength-specific faults on the capacity of LAGs, which aggregate many wavelengths to form an inter-datacenter link. Failure of a single high data-rate wavelength can manifest as smaller capacity changes of multiple LAGs due to multiplexing or muxponding of router ports.

Low SNR margins exacerbate wavelength-specific faults. Small SNR margins render certain channels on fiber vulnerable to *wavelength-specific faults* (WSFs), scenarios where ordinary operational disturbances (repairs, fiber bends, *etc.*) cause some wavelengths to fail while others remain functional, even though they share the same optical infrastructure (Fig. 1a). While it has been long understood that different wavelengths suffer different propagation losses in optical fiber [12, 47], such losses have recently become a noticeable cause of wavelength faults in cloud networks due to reduced SNR margins with higher modulation formats. Using a realistic hardware prototype of a long-haul optical link, we experimentally confirm that routine fiber impairments can selectively fail some wavelengths while others on the same fiber remain operational (§3).

WSFs cause link capacity fluctuations. Since cloud providers aggregate multiple wavelengths to form logical links between datacenters, WSFs cause fluctuations in the capacities of inter-datacenter links and in extreme cases, can fail an entire link if all constituent wavelengths fail simultaneously. Fig. 1b shows the effect of WSFs on the capacity of an inter-datacenter link in a large commercial cloud WAN. Our empirical analysis shows that such fluctuations are frequent: on average, there are more than *four* inter-datacenter link capacity fluctuations of over 100 Gbps every 5 minutes in this cloud WAN (§3).

Link capacity fluctuations are disruptive. Frequent fluctuations in the capacities of inter-datacenter links disrupt network routing mechanisms that rely on a stable network graph. For

[1] Code and data available at: https://github.com/hedge-wan/hedge

example, cloud WANs use software-defined traffic engineering (TE) systems to allocate flows along network paths [21, 27, 28]. Changes in inter-datacenter link capacities due to wavelength faults force the TE controller to frequently recompute flow allocations and reprogram network routers [20, 22]. The frequency of these fluctuations—occurring multiple times within a few minutes (§3)— causes the TE system to respond far more often than it would in a stable network, leading to high control-plane overheads. Moreover, link capacity fluctuations can cause allocated flows to be dropped, resulting in network congestion. These operational concerns in the network of a large commercial cloud provider have motivated our work.

Mitigating temporary WSFs is challenging. One way to reduce wavelength faults is to increase SNR margins by using lower modulation formats. However, naively increasing SNR margins of all wavelengths will reduce their data rates in steady state, under-utilizing the fiber. A more refined approach involves computing precise per-wavelength SNR margins to fit the characteristics of each wavelength. However, even this is challenging since it depends on both a wavelength's characteristics and the operational behavior of the optical line system, such as amplifier gain response [12]. Due to these factors, previous work has recommended adapting wavelength modulation formats in response to changes in signal quality [42, 43]. However, (1) WSFs are often too sudden (§A.1) and frequent (§3) to rapidly detect and respond to, and (2) state-of-the-art optical transponders incur wavelength downtime of several minutes while adapting modulation formats [42] (see §8 for more details), making this approach unappealing.

A fundamental impasse. Thus, operating fiber efficiently using high modulation formats can cause frequent, temporary wavelength faults that are not only disruptive to network TE, but can also cause entire inter-datacenter links to fail [42]. On the other hand, conservative techniques that operate wavelengths well below their true signal quality to avoid wavelength faults result in inefficient use of expensive fiber. This has led to a fundamental impasse where the two viable ways of operating optical fiber either incur frequent wavelength faults or severely under-utilize the optical infrastructure.

Our solution. We build HEDGE to address this impasse (§2). HEDGE builds resilience against temporary wavelength faults both *locally* (for each inter-datacenter link) and *globally* (for the entire WAN), without sacrificing the operational efficiency of optical fiber. We describe the two components of HEDGE:

- **Local resilience using wavelength aggregation (HEDGE-AGG).** HEDGE-AGG is a link-layer wavelength aggregation technique that provisions inter-datacenter links with the desired reliability guarantee (*e.g.,* 5 Tbps at least 99.9% of the time) by aggregating wavelengths of varying risk levels and mapping binary physical-link failures to capacity fluctuations in logical links. Thus, HEDGE-AGG *hedges* the impact of wavelength faults by ensuring that enough wavelengths remain available to meet the link-level reliability target. This scheme has two benefits. First, it provisions inter-datacenter links with a minimum capacity guarantee despite potential WSFs. Second, it de-risks operating wavelengths with low SNR margins since temporary wavelength faults no longer risk failing an entire link (§4).
- **Global resilience using stochastic TE (HEDGE-TE).** While HEDGE-AGG makes inter-datacenter links more resilient to WSFs, these links still have *stochastic capacity*, as with any wavelength-aggregated link (including inter-datacenter links in state-of-the-art cloud WANs). HEDGE-TE mitigates disruptions caused by links with stochastic capacity by developing a novel traffic engineering algorithm that substantially reduces the need to recompute flow allocations when link capacities fluctuate, without sacrificing throughput. HEDGE-TE computes these resilient flow allocations by incorporating the probability distribution of each link's capacity into the TE decision process (§5).

Hardware testbed and evaluation. We build a hardware prototype of an optical WAN and confirm that ordinary disturbances to the fiber can selectively and temporarily fail some wavelengths while others remain up. We use this testbed to evaluate HEDGE-AGG, and show that adapting the data rate of a subset of wavelengths does not hamper connectivity. Finally, we evaluate HEDGE-TE using per-link capacity distributions and the network topology of a large commercial cloud provider.

Results. HEDGE-AGG enables operators to provision an inter-datacenter link with a guaranteed minimum capacity and availability, in spite of WSFs, while using the fewest possible constituent wavelengths. Meanwhile, HEDGE-TE achieves 53% higher throughput compared to a conservative WAN that operates links at their minimum capacity while sustaining equivalent throughput and dropping 12.2× less network flow in worst-case scenarios than state-of-the-art WANs that operate links at their provisioned capacity. Finally, HEDGE-TE reduces disruptions to tunnel allocations by 622× with only a 0.15% reduction in throughput compared to an "oracle" WAN that can recompute flow allocations to deterministically maximize throughput in all scenarios.

2 HEDGE Design

We first discuss how wavelength-specific faults are a challenging and emergent problem in cloud WANs. We then summarize our system, HEDGE, which builds resilience against WSFs and their effects, *i.e.,* stochastic link capacity.

2.1 Need for a novel solution to an emergent problem

Recent work has proposed ways to mitigate the effects of *persistent wavelength failures*—faults that last for longer durations (*e.g.,* several hours to days). Persistent failures are caused by degraded optical equipment (*e.g.,* failing amplifier, dust on transponder) or fiber cuts [17, 50]. Mechanisms to recover from persistent wavelength failures include repairing faulty equipment [50] or migrating wavelengths from failed fiber spans [51]. In contrast, WSFs are *temporary* wavelength failures that result from ordinary operational fiber disturbances such as bending of the fiber. The wavelength automatically recovers once the

disturbance resolves. Fig. 1a shows this behavior using the red wavelength. (§A.1 shows that WSFs typically last several seconds to minutes.) Moreover, WSF-induced capacity fluctuations in cloud WANs can occur several times in every 5-minute interval (§3). Due to these characteristics, existing recovery mechanisms designed for persistent failures or longer-lasting signal quality changes [42] are not well suited for WSFs.

Need for fast recovery mechanisms. Mitigation techniques for persistent wavelength failures involve replacing or reprogramming optical switches (*e.g.,* ROADMs), which can take anywhere from minutes to hours [51]. These methods are effective for long-lasting failures that justify the recovery time, but are unsuitable for frequent, temporary WSFs, which may only last a few minutes. Further, WSFs lead to link capacity fluctuations, forcing frequent TE recomputations that are disruptive and wasteful since WSFs may only last for short time durations.

Adapting modulation is too slow. Other work suggests adjusting the modulation formats of wavelengths based on changes in signal quality [42, 43]. However, modern optical transponders take several minutes to reconfigure the modulation format due to their complex digital signal processing circuits (see [42] and §8). This renders the wavelength inactive during the adjustment period and is thus ineffective for mitigating short-lived and frequent WSFs.

2.2 HEDGE proactively mitigates WSFs

WSFs occur too frequently and are too short-lived for reactive fault mitigation to be effective. Thus, HEDGE takes a *proactive* approach to handle the challenges of WSFs. HEDGE consists of two distinct components: (1) HEDGE-AGG and (2) HEDGE-TE. Fig. 2 summarizes these components and contrasts them with the capabilities of present-day WANs.

HEDGE-AGG (§4) builds resilience to WSFs in inter-datacenter links *by design*: by bundling wavelengths of varying levels of reliability into an inter-datacenter link that has guaranteed capacity and availability regardless of WSFs. Unlike reactive approaches to wavelength failures [42, 51] that necessitate a fast response from slow optical hardware, HEDGE-AGG proactively bakes in resilience at the time of provisioning. While HEDGE-AGG builds resilience to WSFs in links, the capacity of these links may still fluctuate due to inevitable wavelength faults.

HEDGE-TE (§5) is a scalable and optimal stochastic traffic engineering algorithm that reduces control-plane overheads (*e.g.,* recomputing flows, reprogramming routers) from link capacity fluctuations while still maximizing throughput. Our key insight is that inter-datacenter link capacity is inherently stochastic, with WSFs being just one contributing factor. Instead of representing each link with a fixed capacity, HEDGE-TE models each link as having multiple "capacity states" with associated probabilities.

Key technical challenges. To proactively recover from WSFs, HEDGE solves two key technical challenges:

- Reliability and spectral efficiency are critical objectives for network operators, but they are also trade-offs. Higher modulation formats use optical spectrum efficiently at the cost of reliability. While lower modulation formats are more reliable, relying solely on them is an inefficient use of optical spectrum; their low data rates require many additional wavelengths to achieve the target bandwidth of inter-datacenter links. HEDGE-AGG strikes a balance by optimizing the selection of wavelengths to provision links that are highly reliable using the fewest wavelengths.
- Link capacity fluctuations imply that a link's capacity is actually a discrete random variable. If every link can be in one of k capacity states, there are k^n different network states, where n is the total number of links. Finding the optimal traffic allocation across this exponential search space through brute force is computationally infeasible, requiring a more efficient approach. HEDGE-TE reduces this complexity to $O(kn)$. Further, unlike prior work [6,32], HEDGE enforces availability at network design-time (via HEDGE-AGG) instead of exponential search at TE runtime.

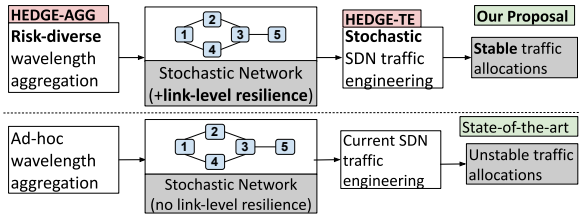


Figure 2: HEDGE Design.

3 Causes and Implications of WSFs

We investigate how some wavelengths can selectively fail while others sharing the same fiber infrastructure remain up. We have observed that such wavelength-specific faults have become disruptive to WAN traffic engineering systems in the cloud that rely on a stable network topology. We experimentally evaluate hypotheses about which factors contribute to WSFs in cloud networks (§3.1) and then empirically study the effect of WSFs on inter-datacenter link capacities (§3.2).

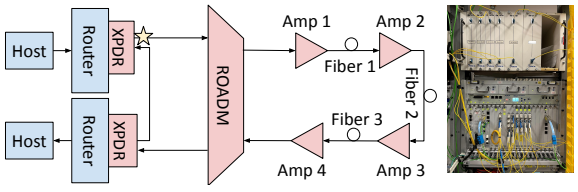


Figure 3: Our hardware prototype. Pink components are optical devices, and blue are electrical. We approximate the setup using different ports on the same electrical router (XPDR = transponder). The star is a fixed attenuator attached to one transponder to add 16 dB of loss.

Hardware testbed. We build a hardware prototype of an inter-datacenter link using equipment from a popular vendor for long-haul fiber deployments. The prototype implements a bidirectional link aggregation group (LAG) between electrical routers. The LAG combines 3 wavelengths over the optical line system (OLS), which includes fiber, a ROADM, and amplifiers. Cloud providers typically connect tens of wavelengths to form

a single LAG. The OLS (Fig. 3) in our setup uses fiber spanning over 100 km, connected through 4 cascaded amplifiers.

3.1 Experimental demonstration of WSFs

Based on our experience with an operational cloud WAN, we have two main hypotheses about factors that contribute to WSFs. Our first hypothesis is that different wavelengths experience different losses in fiber, and this loss is significant enough at high modulation formats to cause some wavelengths to fail from ordinary fiber disturbances. Our second hypothesis is that wavelengths in higher modulation formats are more susceptible to WSFs during ordinary fiber disturbances.

To test these hypotheses, we load the OLS in our prototype with three wavelengths, with the same launch power, across the C-band of the electromagnetic spectrum [30]. We select one wavelength near the start, one in the center, and one near the end of the C-band, to evaluate how a wavelength's position in the C-band affects its loss. We also use a fixed 16 dB attenuator to add more realistic loss. Initially, we use the 16-QAM modulation format for all wavelengths. Thus, each wavelength can carry 200 Gbps, and the LAG has a collective data rate of 600 Gbps. We collect network-layer metrics from the bidirectional link using the end hosts shown in Fig. 3. Given this link operates at 600 Gbps, using a line-rate traffic generator is financially impractical for an academic lab like ours. Instead, we run *iperf* measurements from the hosts, through the LAG, to monitor link uptime. Additionally, we gather physical-layer metrics from the ROADM, including power and bit error rate (BER) for all wavelengths, as we introduce attenuation via fiber disturbances.

Ordinary fiber disturbances. One operational disturbance common in fiber deployments occurs when the fiber is bent, either from routine maintenance or natural phenomena. We replicate this disturbance in our experiments by introducing a gentle bend, called a macrobend [23] of $\frac{3}{4}$ inch diameter, in the fiber. Note that all wavelengths in the link are subject to this bend.

Finding 1: Position in the electromagnetic spectrum impacts reliability (*i.e.,* frequency-dependent loss). As we bend the fiber in our testbed, we find that different wavelengths on the same fiber respond to the bend differently (Fig. 4a). As different wavelengths have different propagation losses in fiber, these wavelengths have slightly different BERs when we start to bend the fiber. Even though the blue and red wavelengths fail before the green wavelength, the LAG remains functional until the green wavelength fails, indicating the benefits of packing risk-diverse wavelengths in LAGs (Fig. 4a). Once we start the bending, the BERs of the blue and red wavelengths rise steeply until they fail, while the green wavelength's BER rises slowly and can sustain the full bend before finally briefly failing (Fig. 4b). All attributes of these wavelengths are the same—they use the same modulation format (16-QAM) on the same optical equipment—and yet some (red and blue waves) fail more quickly than others (green wave) as we continue the bend. We release the fiber bend and all three wavelengths recover.

Finding 2: Modulation format impacts reliability. To test our second hypothesis, we change the modulation formats of the three wavelengths in our testbed to PM-QPSK, 8-QAM and 16-QAM. We repeat the fiber bending and measure the BERs of all wavelengths (Fig. 4c). We find that the wavelength in the highest modulation format, 16-QAM, is the first to show an increase in BER. At this stage, the standard 25% forward error correction (FEC) on the optical transponder can compensate for the increased errors. However, as we continue to complete the bend, the BER increases past FEC's ability to correct errors, and the 16-QAM wavelength fails. Note that while the BERs of 8-QAM and PM-QPSK wavelengths increase slightly during the bend, the increase is too small to cause them to fail. The LAG continues to remain up for the entire duration of the bend since two out of three wavelengths are still up despite the bend. Again, as we release the bend, the 16-QAM wave recovers. All our experiments reinforce the binary link-state model, with a single uncorrectable error failing the link.

Summary. While our experiments test hypotheses on how WSFs occur, they do not comprehensively investigate all potential causes. Rather than identifying every cause of WSFs, we now shift our attention to exploring their implications.

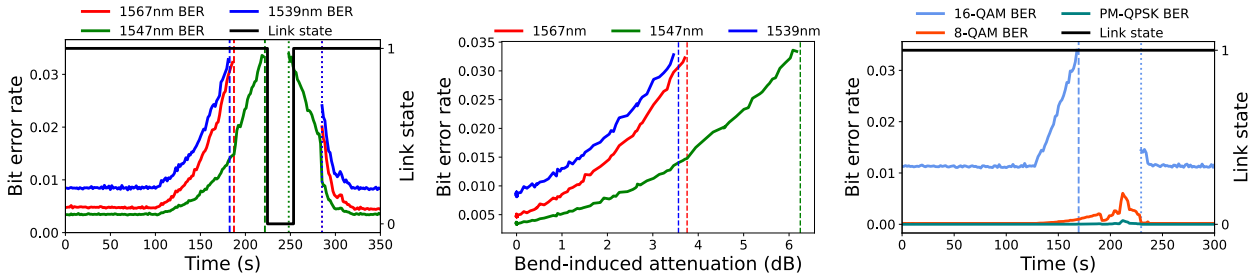
3.2 Implications of WSFs

Our hardware experiments show that ordinary disturbances to optical fiber can cause some wavelengths to fail, depending on their characteristics. Failures of wavelengths that constitute inter-datacenter LAGs cause the LAG's capacity to fluctuate. We measure the capacity of LAGs in the network of a large cloud provider for one month in 2024. We find that constituent wavelengths of inter-datacenter LAGs fail frequently, causing the capacity of over 4 inter-datacenter LAGs to fluctuate every 5 minutes (Fig. 5a). Fig. 5b shows that 35% of measured timesteps involve a substantial reduction in LAG capacity.

Therefore, regardless of the causes of WSFs, capacities of a large portion of inter-datacenter LAGs fluctuate frequently and by a significant fraction of their maximum capacity. These fluctuations force the TE system to recompute traffic allocations and reprogram routers several times within a five-minute window. This behavior is highly disruptive since each TE recomputation incurs a large control-plane overhead, a challenge that HEDGE-TE tackles.

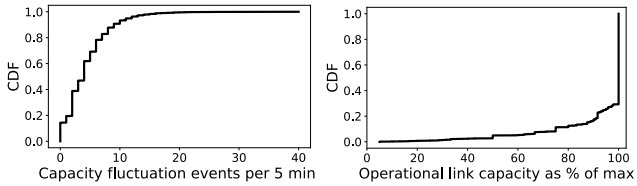
4 Building Local Resilience with HEDGE-AGG

Now that we have shown how some wavelengths in a link aggregation group (LAG) can fail selectively, we present HEDGE-AGG, which enables operators to provision inter-datacenter links with a specified capacity and availability target. HEDGE-AGG navigates the trade-off between reliability and spectral efficiency by providing reliability guarantees while using the fewest possible wavelengths. Although there may be multiple causes of link capacity fluctuations, all of which can be handled by HEDGE-TE (§5), HEDGE-AGG addresses one of the most frequent and egregious: temporary disturbances to the fiber. We consider links composed of one fiber span and later adapt the algorithm to handle optically bypassed links consisting of multiple spans (§A.4).



(a) Fiber bend's impact on different wavelengths. (b) Frequency-dependent loss profiles. (c) Fiber bend's impact on modulation formats.

Figure 4: Hardware experiments highlighting WSFs with fiber bends. In Experiment 1 (4a, 4b), a fiber containing 3 multiplexed wavelengths—distributed across the C-band (all at 16-QAM)—was bent around a cylinder with diameter $\frac{3}{4}$ in. In Experiment 2 (4c), a fiber containing multiplexed waves at different modulation formats (PM-QPSK, 8-QAM, 16-QAM) was bent around the same cylinder. The dashed vertical lines indicate the first uncorrectable FEC for each wave (when the wave fails) while the dotted ones indicate when FEC re-stabilizes (and the wave is back up).



(a) Frequency of fluctuations. (b) Magnitude of fluctuations.

Figure 5: The frequencies and magnitudes of fluctuations in LAG capacities in a large cloud WAN, which disrupt network-wide TE.

Packing diverse wavelengths. HEDGE-AGG uses LAGs at the link layer to bundle channels into one logical connection between a pair of routers. For wavelength aggregation to provide resilience, we must construct the LAG using wavelengths that do not fail simultaneously, and that provide some minimum capacity and availability guarantee. However, SNRs of wavelengths on the same fiber are likely to experience correlated effects [17, 42] because they experience similar disturbances (*i.e.,* bending of the fiber, *etc.*) and incur similar impairment losses from using similar equipment.

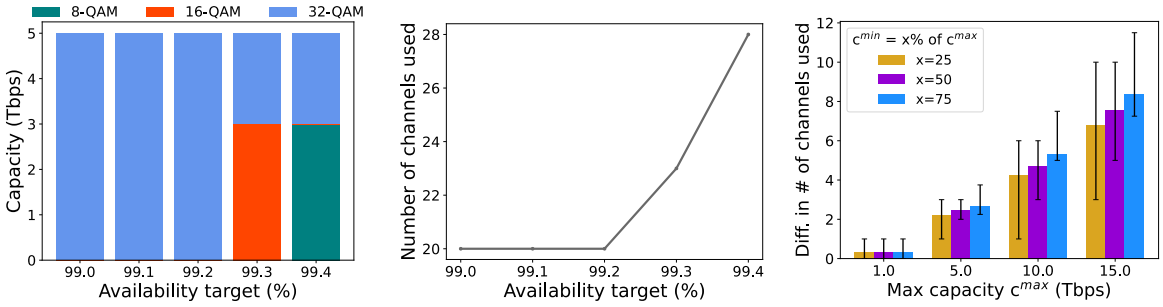
Our solution. We show that even with correlated signal quality, as long as there is sufficient diversity in the modulation formats and/or spectral widths used for wavelengths in a LAG, the LAG can provide high capacity *and* high availability. Wavelengths operated at higher modulation formats/spectral widths supply high capacity to the LAG but may fail earlier than lower-capacity, but more reliable, wavelengths operated at lower modulation formats/spectral widths. While different wavelengths of the same modulation format may also experience different SNR fluctuations due to frequency-dependent losses, these differences are far smaller than the SNR thresholds between different modulation formats, as shown in Fig. 4. To ensure reliability guarantees, we use the "worst-case" wavelength in the spectrum—the wavelength most likely to experience the greatest loss from a fiber disturbance (often the smallest wavelength [12])—but conduct an evaluation of HEDGE-AGG incorporating frequency-dependent losses in §4.3. We first show how HEDGE-AGG aggregates wavelengths to achieve desired stochastic bandwidth distributions (§4.1). Then, we evaluate the algorithm using data from an ISP WAN (§4.2), including an optimality analysis after incorporating frequency-dependent losses (§4.3). Finally, we develop a hardware prototype of a wavelength-aggregated link (§A.5).

4.1 Optimally provisioning a LAG

We formalize the task of designing links using wavelength aggregation. Our goal is to minimize the number of wavelengths required to provision a LAG between WAN routers of capacity $c^{\max}$ when all member wavelengths are up. We require that the LAG sustains at least a minimum capacity of $c^{\min}$ with likelihood β. While we use fixed-grid settings as examples, our algorithm can be extended to flex-grid ROADMs by fixing subproblems to sets of channels with equivalent spectral width.

Inputs. The operator provides the available modulation formats $i \in N$ and their hierarchy (*i.e.,* 16-QAM > 8-QAM) based on the hardware. Higher modulation formats can sustain higher data rates by packing more bits/symbol, but are more susceptible to failure from SNR variations. If all wavelengths on a fiber experience a similar SNR pattern, then SNR drops will cause modulation formats to fail in a stepwise manner; the highest modulation format will fail first, followed by the next highest, *etc.* For each modulation format $i \geq 1$, the operator also provides p_i, the *conditional* probability that modulation format i fails given that modulation format $i-1$ is up. (The *conditional* probability allows us to consider modulation-format failures in a step-wise manner.) We determine these probabilities empirically (§4.2) by using the worst-case wavelength's SNR distribution. For the lowest modulation format ($i = 0$), p_0 is simply the probability that this modulation format will fail. b_i is the data rate sustained by modulation format i for a fixed baud rate. We adopt a binary link-state model because unlike wireless networks where modulation and coding schemes can adapt continuously to signal quality, modern optical WAN links either operate at their configured data rate or fail after even a single uncorrected bit error [42] (see §8 for further details).

Algorithm. Due to SNR correlations, wavelengths using the *same* modulation format on a link will typically fail simultaneously [42]. We now define a capacity distribution for the link. Let s_i be the probability that the link is operating at its ith



(a) Constituent modulation formats. (b) Spectral channels used. (c) Suboptimality analysis.

Figure 6: 6a and 6b show our evaluation of HEDGE-AGG for different availability targets with max and min capacities of 5 and 3 Tbps, respectively. 6a shows the makeup of different modulation formats as availability targets increase. 6b shows the trade-off between number of wavelengths and the availability target. 6c shows the difference in number of wavelengths allocated by HEDGE-AGG for $\beta = 99.9\%$ using the best- vs. worst-case wavelength's SNR after applying the algorithm to all 32 wavelengths in the ISP WAN. Error bars indicate the 25th and 75th percentiles.

capacity level, in decreasing order. Thus, $i = 0$ corresponds to the link being down, and $i = N+1$ is the link operating at its maximum capacity, with $i \in \{1,..,N\}$ being intermediate capacity states, in which some, but not all, modulation formats fail. Then, s_{N+1} is the probability that no modulation format fails:

$$s_{N+1} = \prod_i (1 - p_i). \quad (1)$$

For $i < N+1$, s_i is the probability that modulation format i fails, but all lower-order modulation formats are resilient:

$$s_i = p_i \prod_{j<i} (1 - p_j), \quad \forall i. \quad (2)$$

(We prove that these probabilities are valid in §A.2.) The final output of HEDGE-AGG is the set of w_i, the number of wavelengths to operate at modulation format i, so each probability s_i corresponds to the capacity

$$c_i = \sum_{i<f} w_i \cdot b_i, \quad (3)$$

where f is the lowest modulation format that has failed in capacity state i. Note that if $f = 0$, even the lowest modulation format cannot satisfy the availability target (*i.e.,* $\beta > 1 - p_0$).

The operator provides an availability target β: the probability of sustaining at least minimum capacity $c^{\min}$. Since the capacity distribution is discrete, we can easily compute the availability constraint. First, we sort the capacity distribution in descending order of capacity, and for each capacity state i, compute

$$\sum_{j \geq i} s_j \geq \beta. \quad (4)$$

For the first (*i.e.,* highest) capacity i that satisfies this condition, we set $t = i$. Essentially, this algorithm identifies the highest capacity state t such that the link is operating at t or higher with at least probability β. Recall that each capacity level is obtained by assuming that all wavelengths of a particular modulation format (and higher orders) have failed. Thus, the optimal algorithm to minimize the number of channels used while meeting the availability target is to pack the LAG with the minimum number of wavelengths to provide capacity $c^{\min}$ at the highest modulation format h that is alive in capacity state t:

$$w_h = \lceil c^{\min} / b_h \rceil. \quad (5)$$

This ensures that the minimum number of channels is used to provision a capacity of at least $c^{\min}$ with probability β. To meet $c^{\max}$, we simply provision the remaining capacity by using the highest modulation format, as that occupies the fewest spectral channels:

$$w_N = \lceil (c^{\max} - (w_h \cdot b_h)) / b_N \rceil. \quad (6)$$

Thus, an operator only needs to operate all wavelengths of a LAG at a maximum of two different modulation formats, h and N, to provision enough capacity to meet $c^{\max}$ while ensuring $c^{\min}$ is sustained at least β proportion of the time. If $w_h + w_N$ exceeds the number of channels available, then the operator's requirements cannot be met. We discuss how noisy measurements can affect HEDGE-AGG's availability guarantee in §8.

Now, we show how HEDGE-AGG is optimal. First, consider operating any wavelength at a modulation format besides h for provisioning the LAG up to $c^{\min}$. If a higher modulation format is used, the availability target is not met since h is the highest modulation format that can satisfy β availability. If a lower modulation format is used, then the operator must use more channels to provision the same amount of minimum capacity, even though their availability target was already met with h. For the remaining capacity, using any modulation format besides N is suboptimal because it requires using more channels to provision the same amount of capacity when the availability target has already been met. We illustrate this more formally using a linear program in §A.3, but the two approaches are equivalent.

4.2 Software evaluation

We evaluate HEDGE-AGG using SNR data collected over 6 months from a wavelength in an ISP WAN (and further evaluate it in §4.3 by running it on data from all 32 links in the ISP WAN). We establish SNR cutoffs for different modulation formats using 50 GHz spectral width and 32 GBaud. Since most wavelengths in the ISP have SNRs that could sustain higher data rates (as with most WANs, which are typically over-provisioned), the modulation formats for relevant SNR cutoffs are polarization-multiplexed (PM) 8-QAM, 16-QAM, and 32-QAM. We compute the capacity-probability distribution by applying Equations 1 and 2. The value of

p_i for $i \geq 1$, is the number of timestamps in which the SNR is below the cutoff for modulation format i divided by the number of timestamps in which the SNR exceeds the cutoff for modulation format $i-1$. The value of p_0 is the proportion of timestamps in which the link falls below the 8-QAM threshold.

Using the ISP WAN data, we run HEDGE-AGG for different availability targets, using $c^{\max} = 5$ Tbps and $c^{\min} = 3$ Tbps. The results in Fig. 6a show that an operator can modulate all wavelengths at the highest modulation format, 32-QAM, to meet 3 Tbps of capacity with at most 99.2% probability. However, if the operator has a higher availability target, like 99.3%, then the 3 Tbps of minimum capacity must be provided by wavelengths at 16-QAM. At even higher availability requirements (99.4%), the 3 Tbps must be provisioned with only 8-QAM wavelengths. Note that 100% availability is infeasible because there is a small probability, p_0, that even the lowest modulation format fails. However, while an operator can satisfy higher availability targets for $c^{\min}$ by adopting lower modulation formats, the trade-off is illustrated in Fig. 6b: using lower modulation formats increases the number of spectral channels used since each wavelength can transmit less data.

4.3 Impact of frequency-dependent losses

While wavelength-specific faults broadly refer to individual failures of constituent wavelengths in an aggregated link, there are also well-documented frequency-*dependent* losses that could impact a wavelength's resilience beyond its modulation [12, 47]. Across the C-band, the worst-case wavelength (*i.e.,* highest frequency-dependent loss) can be determined experimentally by the operator, but is typically the smallest wavelength due to scattering and nonlinearities [2]. We choose the SNR of the worst-case wavelength (as defined earlier) because (1) this is the *only* design point that *guarantees* meeting the availability target β, and (2) frequency-dependent losses are insignificant compared to modulation-dependent losses, as shown in §3 and confirmed by literature [12, 42]. Because incorporating conditional failure probabilities for every `(wavelength, modulation format)` pair would be excessively complex, we use the worst-case wavelength's failure distribution and conduct a study to quantify the suboptimality in the number of channels used. We run HEDGE-AGG using the conditional failure probabilities obtained from each of the 32 wavelengths in the ISP WAN, both with the exact empirical SNR and the SNR +2 dB, assuming the best-case SNR improvement from frequency-dependent loss [12]. For each wavelength, we compute the difference of HEDGE-AGG's channel allocations between the two scenarios, and average these differences across all wavelengths to quantify the worst-case suboptimality in spectral efficiency across a range of capacity targets for $\beta = 99.9\%$; Fig. 6c indicates the overallocation, on average, is only a few channels and is proportional to the required capacity provisioning, and results with a wide range of β values replicate this finding. Higher reliability requirements (via higher $c^{\min}$) come with the inherent trade-off of using more channels to pack more reliable, but lower bandwidth, wavelengths.

However, the best-case scenario used for comparison in this study is unrealistic and overestimates the optimality gap because it assumes *all* other wavelengths operate with the same best-case SNR. This reinforces our decision to use the worst-case wavelength to construct the capacity distribution, as any other heuristic wavelength cannot guarantee the availability target without knowing the exact distribution of frequency-dependent losses. Further, imposing frequency-specific constraints complicates provisioning, as co-designing multiple LAGs in a network would then require considering exactly *which* wavelengths were used on which segments.

5 Building Global Resilience with HEDGE-TE

Wavelength-specific faults cause inter-datacenter WAN links to exhibit stochastic capacity. HEDGE-AGG builds local resilience to WSFs, but still requires dealing with variable capacity links. Furthermore, there may be other causes of capacity fluctuations that HEDGE-AGG cannot address. Prior work related to traffic engineering (TE) with variable link capacities includes ML-based TE (which has limited production deployment) with exclusive focus on MLU metrics, as opposed to availability and control-plane overheads [3]. Prior work has also examined single-wavelength scenarios in which the controller dictates capacity changes to link-local hardware, and that force the link to go down during the capacity change [42]. With HEDGE, links change capacity independently of the controller, and capacity changes are hitless; the controller only receives information about the general *distribution* of each link's Layer-3 capacity. We design a TE controller, HEDGE-TE, that enables operators to maximize their *expected* network throughput in this stochastic environment while being resilient to fluctuating link capacities. We note that HEDGE-TE is useful in *any* context that uses LAGs (whether or not they are designed with HEDGE-AGG), as all optical LAGs will exhibit link capacity distributions based on failures of some constituent wavelengths. HEDGE-AGG can be used in conjunction with HEDGE-TE to build links with even more predictable and reliable capacity distributions. Specifically, HEDGE-AGG encodes explicit availability guarantees at design time to circumvent the scaling challenges of TE methods that must iterate through exponential failure scenarios [6, 32]. However, we formulate additional approaches that enforce explicit availability guarantees in §A.7 and note that HEDGE-TE encodes availability by minimizing the flow that is lost during capacity fluctuations. While we focus on optimizing network-wide flow, HEDGE-TE also performs well for concurrent flow, enabling high *per-flow* availability without compromising on concurrent flow (§A.6).

5.1 Traffic allocations with stochasticity

Similar to standard TE controllers, the HEDGE-TE controller receives the network topology, traffic demands d_i, and tunnels $r \in R_i$ for every source-destination pair i as inputs. The optimizer solves for x_r, the flow allocations along tunnels r.

Link capacities as random variables. While conventional TE has considered link capacities as constants, this stochastic

Algorithm 1 HEDGE-TE: Maximizing Expected Throughput

Inputs:

$G\langle V,E\rangle$:	network G with vertices V and links E
$z \in Z_e$:	distribution of capacity states for link e
$p_{e,z}$:	probability of link e being in state z
$d_i \in D$:	traffic demand of source-destination pair i
$R_i \in R$:	set of tunnels for flow i
$c_{e,z}$:	capacity of link e in state z
$c_e^{\max}$:	maximum capacity of link e (*i.e.,* $\max_{z \in Z_e}\{c_{e,z}\}$)

Output:

x_r:	flow allocation along tunnel r

Maximize: $\sum_i \sum_{r \in R_i} x_r - \sum_{e \in E} \sum_{z \in Z_e} p_{e,z} l_{e,z}$

subject to:

$\sum_{r \in R_i} x_r \le d_i$	$\forall d_i \in D$
$\sum_{r \ni e} x_r \le c_e^{\max}$	$\forall e \in E$
$l_{e,z} \ge (\sum_{r \ni e} x_r) - c_{e,z}$	$\forall e \in E, z \in Z_e$
$l_{e,z} \ge 0$	$\forall e \in E, z \in Z_e$

network introduces the concept of *link capacity distributions*, in which each link can operate at different discrete capacities with varying probabilities. Thus, the HEDGE-TE controller receives a capacity distribution for each link as input. For each link $e \in E$, the set of link capacities for e is denoted by Z_e. Each capacity state $z \in Z_e$ is associated with a capacity value $c_{e,z}$ (*e.g.,* 500 Gbps) and its corresponding probability $p_{e,z}$. (It follows that $\sum_{z \in Z_e} p_{e,z} = 1$.) These probability distributions can be obtained empirically as the proportion of time the link operates at capacity $c_{e,z}$ over some time period set by the operator.

Resilience of flow allocations over network scenarios. While stochastic link capacities in prior work on stochastic link failures correspond to different sets of link failures [6], our network scenarios correspond to different combinations of link capacities. The goal of HEDGE-TE is to maximize the *expected* network throughput over all network scenarios.

Scaling challenges. A key challenge in stochastic optimization is the number of scenarios over which the objective must be maximized. Indeed, scenarios involving stochastic link capacities are exponential in the number of link capacity states. A naive formulation that optimizes over all scenarios, where each scenario corresponds to a link existing in one of its capacity states, would not scale. We show that HEDGE-TE only requires scaling with the *sum* of link capacity states instead of the product (see §5.2 and proof in §A.8).

Objective. Our objective maximizes the sum of total flow allocations with penalties for exceeding a link's capacity in a given state, weighted by the capacity state's probability and the magnitude of overallocation. As capacity distributions will vary across links due to variation in physical-layer attributes such as SNRs, HEDGE-TE maximizes the total network throughput in a *risk-aware* manner. By penalizing naive allocations that place substantial traffic on links prone to high capacity fluctuations, both in likelihood and magnitude, HEDGE-TE maximizes network throughput so that it is resilient to such variability. Specifically, the penalty for link e in capacity state z is

$$l_{e,z} = \max\{(\sum_{r \ni e} x_r) - c_{e,z}, 0\}, \tag{7}$$

which is the quantity of flow that exceeded the link's capacity in that state (trivially zero if the link's capacity is not exceeded). Note that $l_{e,z}$ can be linearized by writing each component of the max operator as a separate inequality constraint (Alg. 1). Because links are most likely to operate at maximum capacity, as lower capacities correspond to LAG members failing, we only constrain allocations to not exceed the maximum capacity for each link. This constraint is sufficient because the objective ensures that flow allocations along tunnels are risk aware. We additionally impose the standard demand constraint to ensure flows are not allocated more than their requested demand.

5.2 Scaling with network scenarios

Although network scenarios correspond to different combinations of link capacities, which grow exponentially, HEDGE-TE scales *without requiring any scenario pruning*. While the objective does not explicitly instantiate any particular combination of network scenarios, they are implicitly solved for by the algorithm. Because the first term of the objective, $\sum_i \sum_{r \in R_i} x_r$, is a global allocation independent of any scenarios/link states, we focus on the second term. In the new notation, we consider $q \in Q$ network scenarios, in which p'_q is the probability of scenario q. p'_q is the product of the probabilities of each of the link capacities occurring to create that network scenario: $p'_q = \prod_{e \in E} p_{e,z}$ given a particular capacity state z for each link e. Now, we define

$$s_{e,q} = \max\{(\sum_{r \ni e} x_r) - c_{e,q}, 0\} \tag{8}$$

as the loss associated with a particular network *scenario*. Critically, for every link e, the value of $s_{e,q}$ will be identical for all q with equivalent $c_{e,q}$ since $\sum_{r \ni e} x_r$ does not depend on q. Instead of considering network scenarios $q \in Q$, we can equivalently consider a set of link states $z \in Z_e$ for every link e, in which state z occurs with probability $p_{e,z}$ as in Alg. 1. Thus, we arrive at the equivalent, but much more easily computable, HEDGE-TE formulation with losses $l_{e,z}$ instead of $s_{e,q}$. The formulations are equivalent because we formulate the problem from groups of links that fluctuate independently (see §A.8 for more detail).

Further, three additional optimizations enable reducing the number of $l_{e,z}$ terms. First, any $l_{e,z}$ for which z is the link operating at maximum capacity can be eliminated since there is inherently no loss in this state, reducing the auxiliary variable count by $O(|E|)$. Second, $l_{e,z}$ can be entirely eliminated for all links whose capacity never fluctuates. Finally, the max function in the auxiliary variable constraint can easily be linearized, as we show in Alg. 1, maintaining the LP structure.

Link capacity independence. HEDGE-TE does not require that links fluctuate independently. If multiple links are part of the same shared risk link group (SRLG), the probabilities with

which each of their capacity states occur ($p_{e,z}$) will inherently be similar. In this case, the objective could weight the sum of loss values, $l_{e,z}$, for each link by the number of SRLGs it belongs to. A more complex case involves a link that belongs to different SRLGs for different capacity states. For example, one capacity state might be due to a LAG member sharing a physical fiber with another LAG. In this case, the objective could weight *each* loss value $l_{e,z}$ by the number of SRLGs present for link e in exactly state z.

5.3 Post-processing if any links overflow

In state-of-the-art TE systems [21, 27, 28], each link capacity fluctuation causes the controller to recompute global flow allocations. This is costly because the controller must reprogram the configuration on all routers to adjust sending rates. While HEDGE-TE is risk aware, it only bounds flows by the maximum link capacity because a link is most often operating at its provisioned capacity in our empirical analysis.

If link capacities decrease and flow allocations exceed these capacities, HEDGE-TE optimally post-processes flow allocations with minimal disruption to the controller and WAN routers using a novel algorithm (Alg. 6 in §A.9). The post-processing algorithm only takes as input flow allocations on tunnels that traverse over-allocated links, instead of all tunnels. The objective is to minimize the sum of flow reductions over these tunnels, constrained to the capacities of links in the realized scenario ($c_{e,q}$). Thus, our post-processing algorithm minimizes the overhead when reductions are required, by only impacting tunnels that traverse the offending links.

6 Large-Scale Evaluation of HEDGE-TE

Implementation Details. We evaluate HEDGE-TE on four topologies: (1) CloudWAN, a cloud WAN with ∼50 nodes and ∼200 links; (2) B4, a cloud WAN with 12 nodes and 38 links; (3) ATT (in §A.10), an ISP WAN with 25 nodes and 112 links; (4) KDL (results and implementation details in §A.11), an ISP WAN with 754 nodes and 1790 links. Of these, the ATT and B4 topologies and demand matrices are publicly available [6] whereas we collect the CloudWAN data ourselves. We synthesized demand matrices for KDL using the popular gravity model [49]. Additionally, we measured fluctuations in LAG capacities in CloudWAN over a one-month period. This data provides a distribution of time spent in states below maximum capacity for several links. Using this data as a base, we evaluate HEDGE-TE with different levels of stochasticity.

To stress-test HEDGE-TE, we assume each capacity distribution was observed over a period of 12 hours; the level of stochasticity was fixed for all topologies and TE methods. (We also evaluate varying the amount of stochasticity in §6.4.) We matched links to unique capacity distributions of similar maximum capacity (ideally, equivalent if available) for CloudWAN, and used a similar approach for other topologies since capacity fluctuations for the public topologies were not available. In cases where there were not enough distributions with the exact same maximum capacity, we used distributions with a similar maximum capacity, and normalized values in the distribution such that the maximum capacity of the assigned distribution matched the link's provisioned capacity. Thus, all other capacities in the distribution were also proportionally adjusted, to the nearest multiple of 100 Gbps. To ensure that the exact matching of capacity distributions to links did not impact our results, we report results in aggregate for 10 random permutations of the initial matching for all topologies.

6.1 Comparison Baselines.

Naive Pessimistic TE. A highly risk-averse operator would maximize total network throughput using the *minimum non-zero link capacities* for capacity constraints.

Naive Optimistic TE. An optimistic operator would maximize the total network throughput using the *maximum link capacities* for capacity constraints (as is done today).

Greedy rate-adaptive (RADWAN). A rate-adaptive WAN adjusts link capacities using SNR data in every run of the TE controller [42]. In our implementation, we assume the RADWAN controller is run every 5 simulations. During every TE computation, RADWAN solves for maximum network throughput using the *current* state of the network. Links whose capacity has fluctuated between different non-zero values (*i.e.,* links that RADWAN would have selected for a capacity change) between consecutive runs are treated as being down, since the link cannot carry any traffic for some period of time when it undergoes a capacity change.

Stochastic binary tunnels (TEAVAR). TEAVAR optimally allocates traffic subject to an availability target, using a probabilistic link failure model [6]. For fair comparison, we implement TEAVAR* [1], a variant of TEAVAR that maximizes total flow, instead of concurrent flow. Our implementation of TEAVAR* considers links as down when they are below maximum capacity, and up otherwise. Thus, the probability of the "down" state for a link is the sum of its probabilities of existing at non-maximum capacities. To enable fair comparison unhindered by risk aversion, we implement two versions of TEAVAR* with availability targets $\beta = 0.9$ (TEAVAR-90) and $\beta = 0.5$ (TEAVAR-50). We assume links fluctuate independently as with HEDGE-TE, and prune scenarios by picking scenarios with the highest probability. We ensure that β exceeds the total probability covered after pruning scenarios so that VaR_β guarantees hold. We only compare against TEAVAR-50 for ATT because 0.9 exceeds the total probability covered after scenario pruning; for CloudWAN, we do not compare against TEAVAR at all because it does not scale.

For each demand scale, we use 4-shortest paths for tunnel selection, and compute an initial flow allocation for each method using Gurobi [19]. To stress-test the controller, we use the maximum demand value over the time period for each demand pair. We then run 1000 Monte Carlo simulations for each demand scale. In each simulation, we sample from each link's capacity distribution. For each method whose initial allocation has exceeded any link capacities in the current network state, we run the post-processing algorithm to optimally reduce flow allo-

cations. Thus, for each topology, our results for every demand scale are aggregated over 10 random permutations of the topology and 1000 simulations per permutation. We use the same methodology to evaluate HEDGE-TE for the maximum concurrent flow (MCF) objective on CloudWAN and B4 in §A.6.

6.2 Throughput, availability, and overflow

HEDGE-TE defines the frontier of optimality in the throughput-availability trade-off with fluctuating link capacities. Throughput is simply the controller's original allocation for the given demand scale; we specifically show the percent change in throughput over the Naive Pessimistic baseline. *Availability* is the percent of simulations in which a controller recomputation was not required (*i.e.,* the initial allocation respects realized capacities and post-processing was not needed). Throughput and availability are inherently trade-offs because pushing more traffic to meet demand eventually requires taking risks (*i.e.,* using links more prone to fluctuating capacity) that compromise on availability. We also analyze the *magnitude* of flow reductions from post-processing when initial allocations exceed the current scenario's link capacities, particularly in worst cases. For the MCF objective, we show that HEDGE-TE performs at the frontier of *per-demand* availability and concurrent flow, achieving greater fairness (§A.6).

Throughput. For all topologies and over all demand scales, HEDGE-TE provides equivalent (maximum) throughput as Naive Optimistic and TEAVAR, as shown in Figs. 7c, 8c, 17c, and 18c, and far exceeding Naive Pessimistic. HEDGE-TE sustains 39.2% higher throughput than Naive Pessimistic at $1\times$ demand and 53.2% higher throughput at $3\times$ demand. These results indicate that a conservative approach to naively bound flows by link capacities in near-worst-case scenarios will cause the network to push significantly less traffic. RADWAN, while outperforming Naive Pessimistic, consistently sustains less throughput because each capacity up/downgrade causes the link to temporarily go down, and because RADWAN does not have knowledge of capacity distributions.

Availability. As shown in Figs. 7a, 8a, 17a, and 18a, at lower demand scales, HEDGE-TE provides the highest availability over *all* baselines, including Naive Pessimistic. At higher demand scales, Naive Pessimistic outperforms HEDGE-TE in availability, but at the cost of significantly lower throughput while HEDGE-TE's advantage over all other baselines grows. In undersubscribed environments, HEDGE-TE, unlike Naive Pessimistic, can directly encode the risk of complete link failures. In oversubscribed environments, HEDGE-TE outperforms all other baselines besides Naive Pessimistic because it can allocate traffic along tunnels that exclude links more prone to intense capacity fluctuations. While TEAVAR can encode the probability of link failures, and in our variant even the probability of fluctuations to lower capacities, it cannot differentiate between links that fluctuate with similar probabilities but between capacities of different magnitudes.

Magnitude of flow reductions in worst-case scenarios. While the percent of simulations requiring recomputation captures a notion of availability, the *magnitude* of churn induced in each scenario is another critical metric, especially in worst-case scenarios. We define this churn as the sum of flow reductions after post-processing, as this directly measures the congestion caused by greedy overallocation. In Figs. 7b, 8b, 17b, and 18b, we show the 95th percentile of this churn over demand scales. HEDGE-TE induces the minimum churn among all TE methods at lower demand scales. At higher demand scales, HEDGE-TE drops the least network flow besides Naive Pessimistic, with which it drops a similar amount of flow despite sustaining much higher throughput. HEDGE-TE drops $12.2\times$ less network flow than Naive Optimistic, $6.3\times$ less than Naive Pessimistic, and $3.13\times$ less than both TEAVAR-90 and TEAVAR-50 in worst-case scenarios in B4 (Fig. 8b). In CloudWAN, HEDGE-TE drops $4.8\times$ and $3.6\times$ less network flow than Naive Optimistic *and* Pessimistic, respectively (Fig. 7b).

Navigating the trade-off. Figs. 7d, 8d, 17d, and 18d capture the ultimate throughput-availability trade-off. Compared to all baselines and across all topologies, HEDGE-TE provides a flow allocation that is the most resilient to network scenarios (*i.e.,* requiring the fewest TE recomputations) while still providing the highest throughput. We find similar performance for the MCF objective, with HEDGE-TE sustaining more per-demand flow than Naive Pessimistic while providing more per-demand availability than Naive Optimistic and other baselines (§A.6).

6.3 Comparison to an Oracle WAN

We introduce another baseline, ORACLE, named accordingly because it will always sustain the highest throughput without ever overallocating flow on any links: ORACLE recomputes flow allocations with the maximum-throughput objective in *every* scenario. Unlike other methods, it has access to the current instantaneous topology with fixed link capacities. However, this means that the ORACLE controller must be executed *every time* the topology changes. TE recomputation is a highly disruptive process, involving not only the computation time, but also updating all WAN routers to adjust flow rates and waiting for protocol reconvergence. Thus, we compare HEDGE-TE to ORACLE at $1\times$ demand on the CloudWAN topology in terms of the number of flows disrupted with each network scenario and the *effective throughput* sustained, over all simulations. The effective throughput in a scenario is the realized throughput in the current network state: the exact allocations computed by the controller for ORACLE (since ORACLE will always respect the current state's link capacities), and the adjusted allocations after post-processing for HEDGE-TE.

Fig. 9a illustrates that running HEDGE-TE reduces the number of flow disruptions by more than $600\times$ on average, and far more in worst-case scenarios. Moreover, HEDGE-TE sustains 99.85% of ORACLE's effective throughput on average.

6.4 Performance metrics

Runtime. HEDGE-TE and post-processing can be computed in less than 34 ms and 6 ms, respectively, among the smaller topologies evaluated (Fig. 9b). Fig. 9c shows that HEDGE-TE

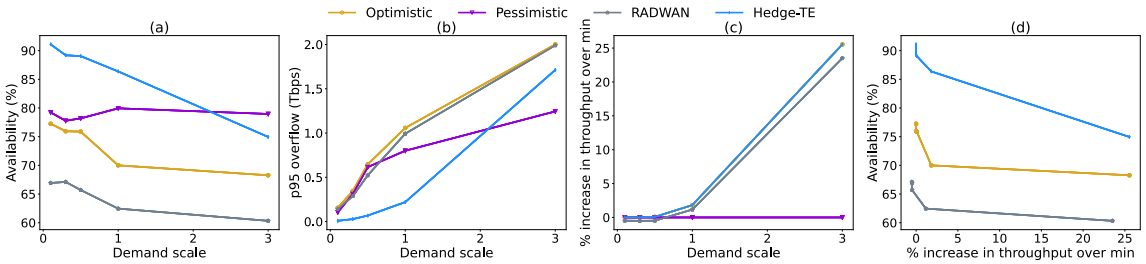


Figure 7: Evaluation on the CloudWAN topology: throughput, availability, and worst-case flow reductions across demand scales.

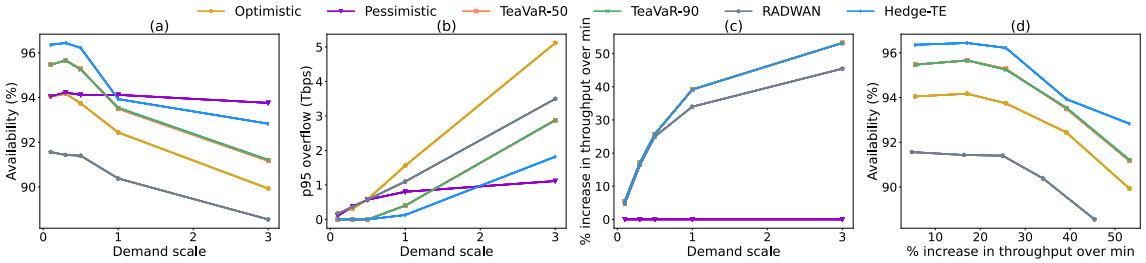
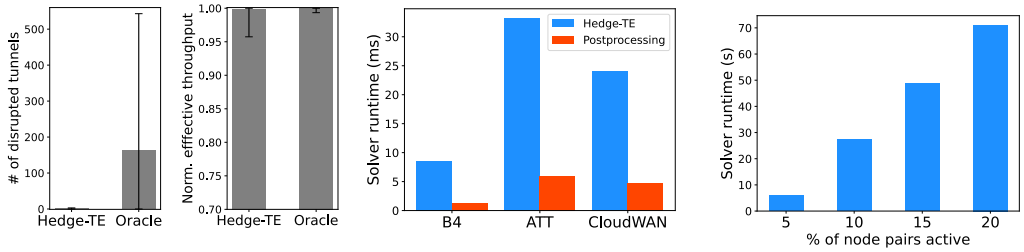


Figure 8: Evaluation on the B4 topology: throughput, availability, and worst-case flow reductions across demand scales.



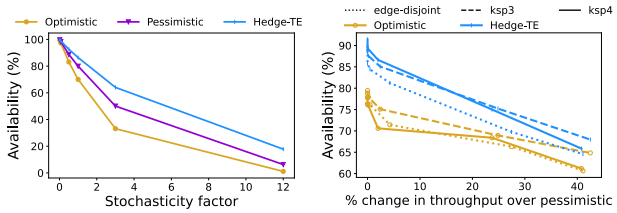
(a) Comparison to ORACLE (b) HEDGE-TE runtime. (c) HEDGE-TE runtime on KDL.

Figure 9: 9a compares HEDGE-TE's performance to ORACLE in the number of disrupted tunnels per scenario (error bars are the 1st and 99th percentiles). 9b shows the Gurobi runtime of HEDGE-TE and post-processing. 9c shows the runtime on KDL as active demand pairs increase.

scales to even the largest topology, KDL, and only requires approximately one minute to solve even with many active demand pairs (20%). Because HEDGE-TE does not prune scenarios, this runtime reflects the objective's true optimum.

Effect of stochasticity. We analyze how the level of stochasticity in link capacity distributions affects HEDGE-TE's availability. We adjusted the stochasticity of distributions by modifying our assumption of the measurement period. Fig. 10a shows that at $1\times$ demand scale, HEDGE-TE's availability exceeds that of both naive baselines for *all* stochasticity levels. As the stochasticity of capacity distributions increases (*i.e.*, increased probability at non-maximum capacities), HEDGE-TE's relative improvement in availability over both baselines decreases—though it always remains higher—while the raw availability percentage naturally increases for all methods.

Effect of tunnel selection. We compare HEDGE-TE to Naive Optimistic with three different tunnel-selection approaches: 3-shortest paths (ksp3), 4-shortest paths (ksp4), and edge-disjoint paths (edge-disjoint). Fig. 10b indicates that regardless of how tunnels are selected, HEDGE-TE outperforms Naive Optimistic in sustaining the highest availability for every level of throughput. HEDGE-TE performs slightly better with k-shortest paths as opposed to edge-disjoint paths, likely



(a) Effect of stochasticity. (b) Effect of tunnel selection.

Figure 10: 10a shows the availability of HEDGE-TE and baselines as the stochasticity level changes. A measurement period of x hours corresponds to a stochasticity factor of $x/12$. 10b shows how tunnel selection affects availability. Both use the CloudWAN topology.

because edge-disjoint paths tend to be longer on average and thus more prone to disruption from any one link fluctuating.

7 Related Work

Optical WANs. There has been extensive research on optimizing traffic in optical WANs [7,9,15,16,25,30,36,41,42,46,51]. While there have been numerous approaches for network planning in optical WANs [4,5,13,18,33,52], we introduce optimal, risk-aware provisioning for a single link through wavelength aggregation and stochastic TE to ensure global reliability. There has also been extensive work documenting

operational disturbances [37] and wavelength-dependent losses in optical fiber [12, 47]. Our methods and insights are informed by these underlying physical phenomena.

Stochastic TE. WAN TE has traditionally focused on computing optimal flow allocations in deterministic settings [20–22, 26–28, 31, 35]. Recent work has shown the effects of adversarial inputs on traditional TE [34, 35] and explored the benefits of stochastic TE to directly incorporate uncertainty into optimization problems. DOTE [38] uses stochastic optimization to handle the variation in traffic demands over time, and is complementary to our work. TeaVaR [6] introduces link failure likelihoods to TE to improve the availability of flow allocations under complete failure events. HARP [3] is an ML-based TE method that aims to mitigate the impact of changing topologies. However, HARP does not investigate the causes of changing link capacities and only focuses on ML-based TE for a specific objective. Instead, HEDGE identifies the impacts of WSFs in WANs, and their repercussions across the network stack.

Failure resilience in WANs. Recent work in TE has focused on TE that not only meets demand but is also resilient to failures [6, 8, 24, 32, 51]. While our work does not focus on resilience to traditional complete failures, it explores and mitigates a new form of failure that has been empirically observed—when a link is functional, but at a reduced capacity.

8 Discussion and Limitations

How predictive are link capacity distributions over time? HEDGE-TE uses historical link capacity distributions to compute flow allocations. Thus, HEDGE-TE benefits when past link capacity distributions closely reflect future distributions. KL divergence, a metric widely used in machine learning, offers a useful measure of a past distribution's similarity to a future distribution [29]. KL divergence (§A.12) is a non-negative metric (ranging from 0 to ∞), with smaller values indicating more similarity and 0 for identical distributions. Fig. 11 shows that the average KL divergence between capacity distributions increases as the time window used for generating distributions increases, *i.e.,* the past 30 min. of a link's capacity fluctuations are more predictive of the next 30 min. than the past day is of the next day. While most KL divergence values are sufficiently low and suggest enough similarity between past and future distributions, operators can boost performance by using shorter time windows (*i.e.,* $\leq$ 30 min.) to generate capacity distributions. However, we note that there may be outlier cases in which a past capacity distribution does not resemble future behavior. We stress-test HEDGE-TE with these adversarial inputs in §A.12 and find that it still performs on par with (or outperforms) state-of-the-art baselines. The similarity of capacity distributions over time also indicates that the failure probabilities used in HEDGE-AGG are reflective of future environments.

How does HEDGE-AGG relate to SRLGs? Cloud operators leverage a similar idea as wavelength aggregation called shared risk link groups (SRLGs). SRLGs achieve risk-diverse paths or optical circuits to connect routers. Optical circuits belonging to the same SRLG *fail simultaneously*, so routing

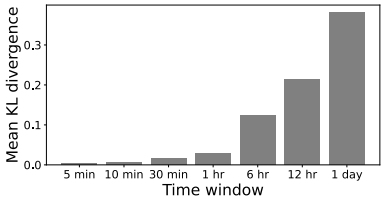


Figure 11: Average KL divergence of link capacity distributions between consecutive time-steps for different time windows.

protocols look for risk-disjoint paths to meet availability targets at a higher level of abstraction. In contrast, wavelength aggregation builds in resilience at the time of constructing links between routers and is complementary to SRLGs.

Can constellation shaping resolve WSFs? Despite advancements in digital signal processing, state-of-the-art optical transponders still cannot adjust the modulation format of wavelengths without experiencing one minute of downtime [10, 42]. While technologies like probabilistic constellation shaping (PCS) were expected to support hitless adaptation, we find that commercially available PCS-supported transponders [14] enter a low-power state when the bits/symbol setting is adjusted, taking several minutes to return to full operation.

Dependence on measured probabilities. HEDGE relies on measured probabilities to construct link-level availability guarantees and implement stochastic TE. These measurements can be reasonably integrated into TE, as shown in prior work [42], and by the HEDGE pipeline that measures SNR every 15 minutes in a live production WAN. However, empirical measurements of physical-layer phenomena can be noisy. Our experiments show that HEDGE-TE is robust even to adversarial noise in prior/predicted capacity distributions (§A.12). For HEDGE-AGG, noise is less of a concern because LAG provisioning happens infrequently and relies on extensive historical data. Any probabilistic scheme will suffer when historical and real-time conditions differ, so we suggest periodic updates to input distributions like prior work [42].

Scalability of linear programming. Recent work in TE has shown that LPs face challenges in scaling to massive networks [48]. Although HEDGE-TE scales to KDL in our evaluation, it could face challenges under all-to-all-like communication patterns in massive networks. Modern production TE systems still use LP solvers [20, 28], so scaling concerns extend beyond HEDGE-TE, which maintains a linear objective and constraints. Future directions could explore extending HEDGE-TE to emerging research approaches, like NCFlow [1] and deep learning methods.

Acknowledgements

We thank the anonymous reviewers and our shepherd, Muhammad Shahbaz, for their thoughtful feedback. AD is supported by the NSF Graduate Research Fellowship. This work was supported in part by ACE, one of the seven centers in JUMP 2.0, a Semiconductor Research Corporation (SRC) program sponsored by DARPA. The authors of this work are also supported by NSF Awards No. 2444537 and No. 2435852.

References

[1] Firas Abuzaid, Srikanth Kandula, Behnaz Arzani, Ishai Menache, Matei Zaharia, and Peter Bailis. Contracting Wide-area Network Topologies to Solve Flow Problems Quickly. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI 21)*, pages 175–200, 2021.

[2] Govind P Agrawal. *Fiber-Optic Communication Systems*. John Wiley & Sons, 2012.

[3] Abd AlRhman AlQiam, Yuanjun Yao, Zhaodong Wang, Satyajeet Singh Ahuja, Ying Zhang, Sanjay G. Rao, Bruno Ribeiro, and Mohit Tawarmalani. Transferable Neural WAN TE for Changing Topologies. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 86–102, 2024.

[4] Srivatsan Balasubramanian, Satyajeet Ahuja, Gaya Nagarajan, Andrea Celletti, and Frantisek Foston. Multilayer planning for Facebook scale worldwide network. In *International Conference on Optical Network Design and Modeling (ONDM)*, pages 1–6. IEEE, 2017.

[5] Ajay Kumar Bangla, Alireza Ghaffarkhah, Ben Preskill, Bikash Koley, Christoph Albrecht, Emilie Danna, Joe Jiang, and Xiaoxue Zhao. Capacity planning for the Google backbone network. In *International Symposium on Mathematical Programming (ISMP)*, 2015.

[6] Jeremy Bogle, Nikhil Bhatia, Manya Ghobadi, Ishai Menache, Nikolaj Bjørner, Asaf Valadarsky, and Michael Schapira. TEAVAR: striking the right utilization-availability balance in WAN traffic engineering. In *Proceedings of the ACM SIGCOMM 2019 Conference*, page 29 – 43, 2019.

[7] A. Brzezinski and E. Modiano. Dynamic reconfiguration and routing algorithms for IP-over-WDM networks with stochastic traffic. *Journal of Lightwave Technology*, 23(10):3188–3205, 2005.

[8] Yiyang Chang, Chuan Jiang, Ashish Chandra, Sanjay Rao, and Mohit Tawarmalani. Lancet: Better Network Resilience by Designing for Pruned Failure Sets. *Proceedings of the ACM on Measurement and Analysis of Computing Systems*, 3(3):1–26, 2019.

[9] Marek Denis, Yuanjun Yao, Ashley Hatch, Qin Zhang, Chiun Lin Lim, Shuqiang Zhang, Kyle Sugrue, Henry Kwok, Mikel Jimenez Fernandez, Petr Lapukhov, Sandeep Hebbani, Gaya Nagarajan, Omar Baldonado, Lixin Gao, and Ying Zhang. EBB: Reliable and Evolvable Express Backbone Network in Meta. In *Proceedings of the ACM SIGCOMM 2023 Conference*, page 346–359, 2023.

[10] Arjun Devraj, Bill Owens, and Rachee Singh. The Reality of Chasing Shannon's Limit in Optical Wide-Area Networks. In *Proceedings of the 25th ACM Internet Measurement Conference (IMC)*, 2025.

[11] Ramakrishnan Durairajan, Paul Barford, Joel Sommers, and Walter Willinger. InterTubes: A Study of the US Long-haul Fiber-optic Infrastructure. In *Proceedings of the ACM SIGCOMM 2015 Conference*, 2015.

[12] Andrea D'Amico, Stefano Straullu, Giacomo Borraccini, Elliot London, Stefano Bottacchi, Stefano Piciaccia, Alberto Tanzi, Antonino Nespola, Gabriele Galimberti, Scott Swail, and Vittorio Curri. Enhancing Lightpath QoT Computation With Machine Learning in Partially Disaggregated Optical Networks. *IEEE Open Journal of the Communications Society*, 2:564–574, 2021.

[13] John P Eason, Xueqi He, Richard Cziva, Max Noormohammadpour, Srivatsan Balasubramanian, Satyajeet Singh Ahuja, and Biao Lu. Hose-based cross-layer backbone network design with Benders decomposition. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 333 – 345, 2023.

[14] Eugene Park. Optimize Your Multi-Haul Network Capacity with the AC1200. https://acacia-inc.com/blog/optimize-your-multi-haul-network-capacity-with-the-ac1200/, (Accessed on 2022-01-20).

[15] Mark Filer, Jamie Gaudette, Monia Ghobadi, Ratul Mahajan, Tom Issenhuth, Buddy Klinkers, and Jeff Cox. Elastic optical networking in the Microsoft cloud (invited). *J. Opt. Commun. Netw.*, 8(7):A45–A54, 2016.

[16] Mark Filer, Jamie Gaudette, Yawei Yin, Denizcan Billor, Zahra Bakhtiari, and Jeffrey L. Cox. Low-margin optical networking at cloud scale (invited). *J. Opt. Commun. Netw.*, 11(10):C94–C108, 2019.

[17] Monia Ghobadi and Ratul Mahajan. Optical Layer Failures in a Large Backbone. In *Proceedings of the 2016 Internet Measurement Conference (IMC)*, page 461 – 467, 2016.

[18] Jennifer Gossels, Gagan Choudhury, and Jennifer Rexford. Robust network design for IP/optical backbones. *IEEE/OSA Journal of Optical Communications and Networking*, 11(8):478–490, 2019.

[19] Gurobi. Gurobi Optimization. https://www.gurobi.com/, (Accessed on 2024-02-02).

[20] Chi-Yao Hong, Srikanth Kandula, Ratul Mahajan, Ming Zhang, Vijay Gill, Mohan Nanduri, and Roger Wattenhofer. Achieving high utilization with software-driven WAN. In *Proceedings of the ACM SIGCOMM 2013 Conference on SIGCOMM*, pages 15–26, 2013.

[21] Chi-Yao Hong, Subhasree Mandal, Mohammad Al-Fares, Min Zhu, Richard Alimi, Kondapa Naidu B., Chandan Bhagat, Sourabh Jain, Jay Kaimal, Shiyu Liang, Kirill Mendelev, Steve Padgett, Faro Rabe, Saikat Ray, Malveeka Tewari, Matt Tierney, Monika Zahn, Jonathan Zolla, Joon Ong, and Amin Vahdat. B4 and After: Managing Hierarchy, Partitioning, and Asymmetry for Availability and Scale in Google's Software-Defined WAN. In *Proceedings of the ACM SIGCOMM 2018 Conference*, pages 74–87, 2018.

[22] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jon Zolla, Urs Hölzle, Stephen Stuart, and Amin Vahdat. B4: Experience with a Globally-deployed Software Defined Wan. 2013.

[23] John A. Jay. An overview of macrobending and microbending of optical fibers. White Paper WP1212, Corning, Inc., 2010. Accessed: 2024-09-19.

[24] Chuan Jiang, Zixuan Li, Sanjay Rao, and Mohit Tawarmalani. Flexile: Meeting bandwidth objectives almost always. In *Proceedings of the 18th International Conference on emerging Networking EXperiments and Technologies (CoNEXT)*, pages 110–125, 2022.

[25] Xin Jin, Yiran Li, Da Wei, Siming Li, Jie Gao, Lei Xu, Guangzhi Li, Wei Xu, and Jennifer Rexford. Optimizing Bulk Transfers with Software-Defined Optical WAN. In *Proceedings of the 2016 ACM SIGCOMM Conference*, pages 87–100, 2016.

[26] Alexander Krentsel, Nitika Saran, Bikash Koley, Subhasree Mandal, Ashok Narayanan, Sylvia Ratnasamy, Ali Al-Shabibi, Anees Shaikh, Rob Shakir, Ankit Singla, et al. A Decentralized SDN Architecture for the WAN. In *Proceedings of the ACM SIGCOMM 2024 Conference*, pages 938–953, 2024.

[27] Umesh Krishnaswamy, Rachee Singh, Nikolaj Bjørner, and Himanshu Raj. Decentralized cloud wide-area network traffic engineering with BLASTSHIELD. In *19th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 325–338, 2022.

[28] Umesh Krishnaswamy, Rachee Singh, Paul Mattes, Paul-Andre C Bissonnette, Nikolaj Bjørner, Zahira Nasrin, Sonal Kothari, Prabhakar Reddy, John Abeln, Srikanth Kandula, Himanshu Raj, Luis Irun-Briz, Jamie Gaudette, and Erica Lan. OneWAN is better than two: Unifying a split WAN architecture. In *20th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 515–529, 2023.

[29] Solomon Kullback and Richard A Leibler. On information and sufficiency. *The Annals of Mathematical Statistics*, 22(1):79–86, 1951.

[30] Abhishek Vijaya Kumar, Bill Owens, Nikolaj Bjørner, Binbin Guan, Yawei Yin, Paramvir Bahl, and Rachee Singh. CHISEL: An optical slice of the wide-area network. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 859–875, 2024.

[31] Praveen Kumar, Yang Yuan, Chris Yu, Nate Foster, Robert Kleinberg, Petr Lapukhov, Chiun Lin Lim, and Robert Soulé. Semi-Oblivious Traffic Engineering: The Road Not Taken. In *15th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2018.

[32] Hongqiang Harry Liu, Srikanth Kandula, Ratul Mahajan, Ming Zhang, and David Gelernter. Traffic engineering with forward fault correction. In *ACM SIGCOMM 2014 Conference*, pages 527–538, 2014.

[33] Congcong Miao, Zhizhen Zhong, Ying Zhang, Kunling He, Fangchao Li, Minggang Chen, Yiren Zhao, Xiang Li, Zekun He, Xianneng Zou, and Jilong Wang. FlexWAN: Software Hardware Co-design for Cost-Effective and Resilient Optical Backbones. In *Proceedings of the ACM SIGCOMM 2023 Conference*, page 319–332, 2023.

[34] Pooria Namyar, Behnaz Arzani, Ryan Beckett, Santiago Segarra, Himanshu Raj, and Srikanth Kandula. Minding the gap between fast heuristics and their optimal counterparts. In *Proceedings of the 21st ACM Workshop on Hot Topics in Networks (HotNets)*, pages 138–144, 2022.

[35] Pooria Namyar, Behnaz Arzani, Srikanth Kandula, Santiago Segarra, Daniel Crankshaw, Umesh Krishnaswamy, Ramesh Govindan, and Himanshu Raj. Solving Max-Min Fair Resource Allocations Quickly on Large Graphs. In *21st USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 1937–1958, 2024.

[36] Matthew Nance-Hall, Paul Barford, Klaus-Tycho Foerster, and Ramakrishnan Durairajan. Improving scalability in traffic engineering via optical topology programming. *IEEE Transactions on Network and Service Management*, 2023.

[37] Shree Patnaik, Paul Barford, Dante Fratta, Bill Jensen, Neal Lord, Matt Malloy, and Herb Wang. Internet Photonic Sensing: Using the Internet Optical Transport Signals for Vibration and Deformation Sensing. In *Proceedings of the ACM SIGCOMM 2021 Workshop on Optical Systems*, OptSys '21, page 12–17, 2021.

[38] Yarin Perry, Felipe Vieira Frujeri, Chaim Hoch, Srikanth Kandula, Ishai Menache, Michael Schapira, and Aviv Tamar. DOTE: Rethinking (predictive) WAN traffic engineering. In *20th USENIX Symposium on Networked*

Systems Design and Implementation, pages 1557–1581, 2023.

[39] Thomas Richter, Steven Searcy, Philippe Jennevé, Dimitrios Giannakopoulos, Bill Owens, Miquel A. Mestre, Ahmed Awadalla, and Sorin Tibuleac. 1Tb/s and 800 Gb/s real-time transmission at 138 GBd over a deployed ROADM network with live traffic. In *Optical Fiber Communication Conference (OFC) 2023*, 2023.

[40] Jane M. Simmons. *Optical Network Design and Planning*. Springer, 2014.

[41] Rachee Singh, Nikolaj Bjørner, Sharon Shoham, Yawei Yin, John Arnold, and Jamie Gaudette. Cost-effective capacity provisioning in wide area networks with Shoofly. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, page 534–546, 2021.

[42] Rachee Singh, Manya Ghobadi, Klaus-Tycho Foerster, Mark Filer, and Phillipa Gill. RADWAN: Rate Adaptive Wide Area Network. In *Proceedings of the ACM SIGCOMM 2018 Conference*, August 2018.

[43] Rachee Singh, Monia Ghobadi, Klaus-Tycho Foerster, Mark Filer, and Phillipa Gill. Run, walk, crawl: Towards dynamic link capacities. In *Proceedings of the 16th ACM Workshop on Hot Topics in Networks (HotNets)*, page 143–149, 2017.

[44] The R Project for Statistical Computing. Kullback-Leibler Divergence. https://search.r-project.org/CRAN/refmans/philentropy/html/KL.html.

[45] Tobias Mann. Arista, Microsoft Validate 400G ZR Optical Pluggables, (Accessed on 2024-02-01). https://www.sdxcentral.com/articles/news/arista-microsoft-validate-400g-zr-optical-pluggables/2021/02/.

[46] Shih-Hao Tseng, Saksham Agarwal, Rachit Agarwal, Hitesh Ballani, and Ao Tang. CodedBulk: Inter-Datacenter Bulk Transfers using Network Coding. In *18th USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 15–28, 2021.

[47] Stephen S. Walker. Rapid modeling and estimation of total spectral loss in optical fibers. *Journal of Lightwave Technology*, 4(8):1125–1131, 1986.

[48] Zhiying Xu, Francis Y Yan, Rachee Singh, Justin T Chiu, Alexander M Rush, and Minlan Yu. Teal: Learning-Accelerated Optimization of WAN Traffic Engineering. In *Proceedings of the ACM SIGCOMM 2023 Conference*, pages 378–393, 2023.

[49] Yin Zhang, Matthew Roughan, Nick Duffield, and Albert Greenberg. Fast accurate computation of large-scale IP traffic matrices from link loads. *ACM SIGMETRICS Performance Evaluation Review*, 31(1):206–217, 2003.

[50] Ying Zhang, Nathan Hu, Carl Verge, and Scott O'Brien. Cross-layer diagnosis of optical backbone failures. In *Proceedings of the 22nd ACM Internet Measurement Conference (IMC)*, page 419–432, 2022.

[51] Zhizhen Zhong, Manya Ghobadi, Alaa Khaddaj, Jonathan Leach, Yiting Xia, and Ying Zhang. ARROW: Restoration-Aware Traffic Engineering. In *Proceedings of the 2021 ACM SIGCOMM 2021 Conference*, pages 560—-579, 2021.

[52] Hang Zhu, Varun Gupta, Satyajeet Singh Ahuja, Yuandong Tian, Ying Zhang, and Xin Jin. Network planning with deep reinforcement learning. In *Proceedings of the ACM SIGCOMM 2021 Conference*, page 258–271, 2021.

A Appendix

A.1 Duration of WSFs

We measure the durations of all LAG capacity fluctuations in CloudWAN over a one-month period. The CDF of the duration of these capacity fluctuation events is shown in Figure 12, which indicates that most WSFs are very short-lived, lasting just a few seconds to minutes, but still substantially longer than RTT timescale. Thus, WSFs are caused by temporary disturbances to the fiber and require risk-aware algorithmic solutions as opposed to hardware upgrades, since the hardware is clearly functional after the optical recovery without manual intervention.

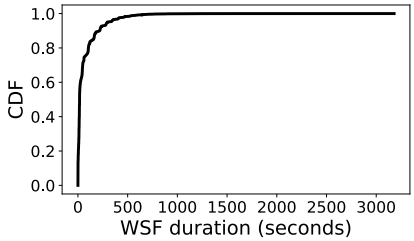


Figure 12: Duration of LAG capacity fluctuation events in CloudWAN.

A.2 Proof that LAG Capacity Probabilities Yield a Valid Distribution

First, it is trivial that $0 \leq s_i \leq 1$ for $0 \leq i \leq N+1$ as long as $0 \leq p_i \leq 1$ (following Eq. 1 and 2).

Next, we would like to show that the probabilities, s_i, over all capacity states $i = 0,\ldots,N+1$ sum to 1. That is,

$$\begin{aligned} 1 &\stackrel{?}{=} \sum_{i=0}^{N+1} s_i \\ &= s_{N+1} + \sum_{i=0}^{N} s_i \\ &= \prod_{i=0}^{N} (1-p_i) + \sum_{i=0}^{N} p_i \prod_{j<i} (1-p_j) \end{aligned}$$

First, we expand and factor the terms such that

$$\begin{aligned}
&\prod_{i=0}^{N}(1-p_i)+\sum_{i=0}^{N}p_i\prod_{j<i}(1-p_j)\\
&=(1-p_N)(1-p_{N-1})\cdots(1-p_0)+p_N(1-p_{N-1})\cdots\\
&\quad(1-p_0)+\cdots+p_1(1-p_0)+p_0\\
&=(1-p_0)[(1-p_N)(1-p_{N-1})\cdots(1-p_1)+\\
&\quad p_N(1-p_{N-1})\cdots(1-p_1)+\cdots+p_1]+p_0\\
&=(1-p_0)\sigma_1+p_0,
\end{aligned}$$

where $\sigma_1=(1-p_N)(1-p_{N-1})\cdots(1-p_1)+p_N(1-p_{N-1})\cdots(1-p_1)+\cdots+p_1$. More generally, we define

$$\sigma_k=\prod_{i=k}^{N}(1-p_i)+\Big(\sum_{i=k+1}^{N}p_i\prod_{j=k}^{i-1}(1-p_j)\Big)+p_k. \tag{9}$$

Furthermore, our expansion has shown that

$$\sum_{i=0}^{N+1}s_i=(1-p_0)\sigma_1+p_0. \tag{10}$$

We now prove by induction that $\sigma_i=1$ for non-negative integers $i\leq N$.

Proof. **Base case:**

$$\begin{aligned}
\sigma_N&=\prod_{i=N}^{N}(1-p_i)+\Big(\sum_{i=N+1}^{N}p_i\prod_{j=N}^{i-1}(1-p_j)\Big)+p_N\\
&=(1-p_N)+0+p_N\\
&=1
\end{aligned}$$

Inductive step: Assume $\sigma_n=1$ for some non-negative integer $n\leq N$. If this holds, we show that

$$\begin{aligned}
\sigma_{n-1}&=\prod_{i=n-1}^{N}(1-p_i)+\Big(\sum_{i=n}^{N}p_i\prod_{j=n-1}^{i-1}(1-p_j)\Big)+p_{n-1}\\
&=\prod_{i=n-1}^{N}(1-p_i)+\Big(p_n(1-p_{n-1})+\sum_{i=n+1}^{N}p_i\prod_{j=n-1}^{i-1}(1-p_j)\Big)\\
&\quad+p_{n-1}\\
&=(1-p_{n-1})\prod_{i=n}^{N}(1-p_i)+\Big(p_n(1-p_{n-1})+\\
&\quad\sum_{i=n+1}^{N}p_i(1-p_{n-1})\prod_{j=n}^{i-1}(1-p_j)\Big)+p_{n-1}\\
&=(1-p_{n-1})\prod_{i=n}^{N}(1-p_i)+(1-p_{n-1})\Big(p_n+\\
&\quad\sum_{i=n+1}^{N}p_i\prod_{j=n}^{i-1}(1-p_j)\Big)+p_{n-1}\\
&=(1-p_{n-1})\Big[\prod_{i=n}^{N}(1-p_i)+\Big(\sum_{i=n+1}^{N}p_i\prod_{j=n}^{i-1}(1-p_j)\Big)\\
&\quad+p_n\Big]+p_{n-1}\\
&=(1-p_{n-1})\sigma_n+p_{n-1}\\
&=(1-p_{n-1})(1)+p_{n-1}\\
&=1-p_{n-1}+p_{n-1}\\
&=1
\end{aligned}$$

□

Thus, we have shown that $\sigma_i=1$ for non-negative integers $i\leq N$. It follows that $\sum_{i=0}^{N+1}s_i=(1-p_0)\sigma_1+p_0=1$. Therefore, the probabilities, s_i—derived from conditional failure probabilities of individual modulation formats—of the LAG capacity distribution sum to 1.

A.3 Wavelength Aggregation as a Formal ILP

As opposed to the intuitive algorithm described in §4.1 for HEDGE-AGG, we can also solve an integer linear program to obtain the optimal number of wavelengths to operate at each modulation format in order to achieve $c^{\min}$ with at least probability β while still provisioning enough capacity for $c^{\max}$ (all while minimizing the number of spectral channels used). Recall that we initially sort the capacity distribution in descending order of capacity, and identify t as the highest capacity state that can meet the operator's availability target of β; this value of t is used as input, alongside the other parameters, to Algorithm 2. This ILP outputs the same results as the approach described in §4.1, and will also fail to provide a result when the physical hardware and/or underlying SNR cannot support it (*i.e.,* the value of β cannot even be met by the lowest modulation format, or the capacity requirements, alongside the availability target, exceed the number of spectral channels available).

Algorithm 2 HEDGE-AGG ILP

Inputs:

$c^{\max}$:	standard capacity of link
$c^{\min}$:	minimum capacity of link
k:	number of spectral channels available
b_i:	data rate supported by modulation format i
t:	output of availability algorithm

Output:

w_i:	number of wavelengths using modulation format i

Minimize: $\sum_i w_i$
subject to:

$$\sum_i b_i w_i = c^{\max}$$
$$\sum_i w_i \leq k$$
$$\sum_{i\leq t} b_i w_i \geq c^{\min}$$

A.4 HEDGE-AGG's Implications for Optical Bypass

In §4, we used HEDGE-AGG to design resilient LAGs for point-to-point WANs. To design a LAG in a WAN that

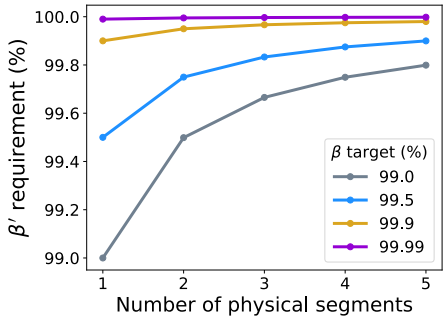


Figure 13: The per-segment availability requirement (β') scales with the number of physical segments per optically bypassed path, with a pattern dependent on the logical link's availability target (β).

is equipped with optical bypass, we must use the same assignment of wavelengths and modulation formats across *all* physical segments that comprise a bypassed logical link [40]. However, we must now translate the β availability target to meet $c^{\min}$ capacity for the logical link into a β' for each physical fiber that makes up the logical link. To simplify the problem, we assume that disturbances that cause temporary SNR fluctuations such as fiber bends are independent across physical fibers in a WAN. While this may not be true for some cases, such as widespread climate events, it likely holds for the many frequent disturbances that occur to a fiber (the focus of our work). Consider a bypass link with capacity l composed of k physical segments with capacities $l_1,\ldots,l_k$, respectively. Thus,

$$P(l \geq c^{\min}) = P(l_1 \geq c^{\min},\ldots,l_k \geq c^{\min}) = \prod_{i=1}^{k} P(l_i \geq c^{\min}) \tag{11}$$

because l will fall below $c^{\min}$ as long as *any* of its physical segments falls below $c^{\min}$. Since $P(l \geq c^{\min}) \geq \beta$ and $P(l_i \geq c^{\min}) \geq \beta'$ for $i \in \{1,\ldots,k\}$, it follows that $\beta = (\beta')^k$ and thus

$$\beta' = \beta^{\frac{1}{k}}. \tag{12}$$

Because $0 < \beta < 1$, it follows that $\beta' = \beta^{\frac{1}{k}} > \beta$, and the larger k is (*i.e.,* the longer the bypass path) the higher β' will be. Therefore, optical bypass requires engineering each physical span of the logical link with a *higher* individual availability target, $\beta^{\frac{1}{k}}$, to meet the overall β availability for the logical link. This implies diminishing returns for longer bypass paths, which have typically been lauded as a cost-saving mechanism [41]. Fig. 13 shows this effect, as longer bypass paths increase the per-segment availability requirement.

Across all wavelengths in the ISP WAN, the average value of $1 - p_0$ (the probability that the lowest modulation format does not fail) was ~ 0.9989. Using $\beta' = 99.89\%$ as an upper bound, we compute the maximum possible length of a bypass path to meet β availability as $\lfloor \log_{0.9989} \beta \rfloor$, which follows from $(\beta')^k = \beta$. For $\beta = 99\%$, this limits paths to 9 physical segments, and for $\beta = 99.5\%$, bypass paths are limited to a maximum of 4 segments. Fig. 13 shows this effect, as longer bypass paths increase the per-segment availability requirement.

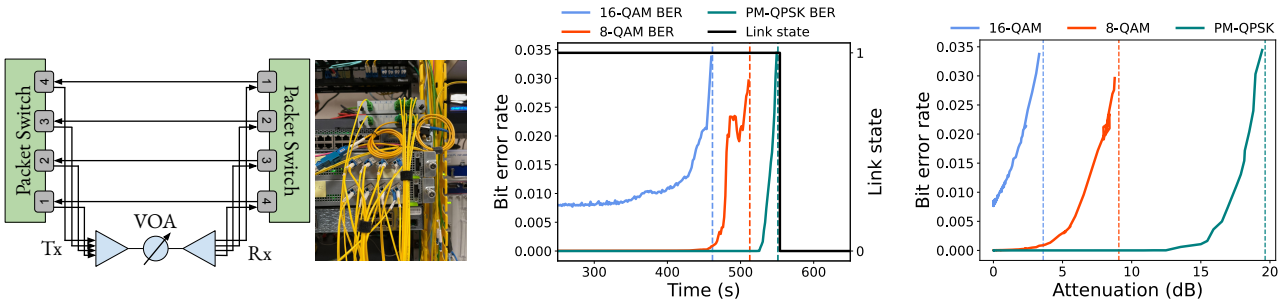
A.5 Hardware evaluation of HEDGE-AGG

Our hardware testbed (Fig. 14a) contains two packet switches, with four network interfaces (1-4) each. We built bidirectional fiber connections between the 4 interfaces. Each connection uses a bandwidth-variable transponder that can modulate signals in 3 formats (PM-QPSK, 8-QAM, and 16-QAM). We aggregate 4 wavelengths, one at PM-QPSK, two at 8-QAM, and one at 16-QAM, between the packet switches. Each packet switch is connected to a host server that generates traffic. The host machines generating traffic in our setup have modest CPU configurations, allowing them to generate up to 5 Gbps of traffic. While not line rate, flow of this traffic establishes the availability of the connection despite signal degradation.

Our goal in the hardware evaluation is to examine the impact of SNR changes in a "diverse" LAG consisting of multiple wavelengths operated at different modulation formats. We use a variable optical attenuator (VOA) to degrade the transmit (Tx) signals from one packet switch to the other. We establish four *iperf* sessions consisting of 12 parallel UDP streams each between the two end hosts. We then gradually increase the attenuation of the VOA, thereby reducing the SNR over time, and observe the impact to both Layer 1 and Layer 3 metrics. As shown in Fig. 14b, the higher the modulation format, the earlier the bit-error-rate begins to spike, and the earlier the wavelength fails due to *uncorrectable errors*; that is, lower modulation formats like PM-QPSK can handle significantly more loss (*i.e.,* lower SNR) than higher modulation formats. Critically, while the 16-QAM and 8-QAM wavelengths can no longer sustain transmission $\sim$450 and $\sim$500 seconds into the experiment, respectively, all UDP traffic continues to be transmitted until the PM-QPSK wavelength fails, at $\sim$ 550 seconds. Our experiments show that aggregating wavelengths of different modulation formats together in a LAG ensures that data can continue to be transmitted, albeit at a lower rate, if traffic exceeds the capacity afforded by the remaining modulation formats, even with significant reductions in SNR.

A.6 HEDGE-TE for Maximizing Concurrent Flow

We formulate a version of HEDGE-TE suitable for the maximum concurrent flow (MCF) objective, shown in Algorithm 3. MCF ensures that all flows are able to send $\gamma \in [0,1]$ proportion of their requested demand, and HEDGE-TE formulation enables optimizing such that these flows in aggregate do not suffer as much loss when link capacities fluctuate. We evaluate HEDGE-TE for the MCF objective on the CloudWAN and B4 topologies, using the exact same experimental scheme as described in §6 (*i.e.,* for every demand scale, 10 random permutations per topology and 1000 simulations per permutation, yielding 10,000 total data points per demand scale). However, we modify the definitions of *availability* and *throughput* to better capture fairness across commodities. Now, instead of measuring the total aggregate throughput, we specifically measure the concurrent flow (since every flow can only send the same percentage of demand, this provides a proxy metric for total throughput too). Importantly, while availability



(a) Hardware setup. (b) Bit-error rates & Layer-3 link state. (c) Bit-error rates vs. loss.

Figure 14: 14a shows a schematic and photograph of our hardware testbed. 14b shows that traffic can still be carried as long as the PM-QPSK wavelength is up. 14c shows how bit-error rates vary for different modulation formats as attenuation is increased. The dashed vertical lines in 14b and 14c indicate the first uncorrectable FEC for each modulation format, when the wavelength fails.

Algorithm 3 HEDGE-TE: Maximum Concurrent Flow

Inputs:

$G\langle V,E\rangle$:	network G with vertices V and links E
$z \in Z_e$:	distribution of link capacities for link e
$p_{e,z}$:	probability of link e being in state z
$d_i \in D$:	traffic demand of source-destination pair i
$R_i \in R$:	set of tunnels for flow i
$c_{e,z}$:	capacity of link e in state z
$c_e^{\max}$:	maximum capacity of link e (*i.e.,* $\max_{z \in Z_e}\{c_{e,z}\}$)

Output:

x_r:	flow allocation along tunnel r
γ:	concurrent flow (in [0,1])

Maximize: $\gamma \sum_i d_i - \sum_{e \in E} \sum_{z \in Z_e} p_{e,z} l_{e,z}$

subject to:

$$\sum_{r \in R_i} x_r = \gamma d_i \qquad \forall d_i \in D$$
$$\sum_{r \ni e} x_r \le c_e^{\max} \qquad \forall e \in E$$
$$l_{e,z} \ge \left(\sum_{r \ni e} x_r\right) - c_{e,z} \qquad \forall e \in E, z \in Z_e$$
$$l_{e,z} \ge 0 \qquad \forall e \in E, z \in Z_e$$

in prior experiments corresponded to the percentage of simulations not requiring any post-processing to be run (*i.e.,* the initial allocation's resilience to fluctuating link capacities across all simulations), availability is now calculated on a *per-demand (per-commodity) basis*. Specifically, we calculate the availability of each individual demand pair over all simulations and plot the lowest value, hence capturing the demand pair that experiences the most flow reductions due to fluctuating link capacities, in order to satisfy MCF's fairness criterion and individually reason about guarantees for commodities.

The results are shown in Figures 15 and 16, and indicate that HEDGE-TE's performance gains can also extend to the MCF objective without losing its linear programming structure. We note that while HEDGE-TE's MCF formulation does not explicitly encode per-demand availability, any approach that would do so would sacrifice on scalability and quickly become quadratic. This is because reasoning about an individual flow's contribution to link capacity over-allocation under fluctuation events quickly becomes computationally challenging. Instead, the HEDGE-TE MCF objective, which incorporates both per-demand concurrent flow and global availability (via link overflows), is successful in practice to both maximize the concurrent flow on par with the Naive Optimistic MCF formulation while ensuring strong per-demand availability. Cases in which HEDGE-TE underperforms baselines for per-demand availability are because these baselines sustain substantially less concurrent flow, similar to the findings in §6; however, HEDGE-TE continues to best trade off concurrent flow for per-demand availability, just as with aggregate throughput and total availability. We note that the Naive Pessimistic baseline significantly under-allocates concurrent flow due to some demands being particularly constrained by the minimum link capacities. Finally, we emphasize Figures 15d and 16d, which show that for every level of concurrent flow sustained, HEDGE-TE provides the greatest per-demand availability.

A.7 HEDGE-TE with Explicit Availability Guarantees

HEDGE-AGG provides explicit availability guarantees while HEDGE-TE focuses on optimizing the *expected* value of the traditional TE objective (*i.e.,* total flow/throughput, concurrent flow, *etc.*) under link capacity fluctuations but still providing substantial improvements in availability at no cost of throughput (see §6). Enforcing explicit availability guarantees with HEDGE-TE will inherently under-utilize the network because links are most often operating at their maximum (provisioned) capacity, so HEDGE's design intentionally offloads explicit reasoning about availability requirements to the time of capacity provisioning, similar to a network operator's approach. Further, providing availability guarantees at design time circumvents the challenge of enumerating the exponential number of failure scenarios at TE runtime that inhibits prior work from scaling [6, 32]. However, for completeness, we provide two approaches that could be used to augment HEDGE-TE to enforce explicit availability guarantees for traffic engineering with stochastic link capacity.

CVaR minimization. We use the approach proposed by

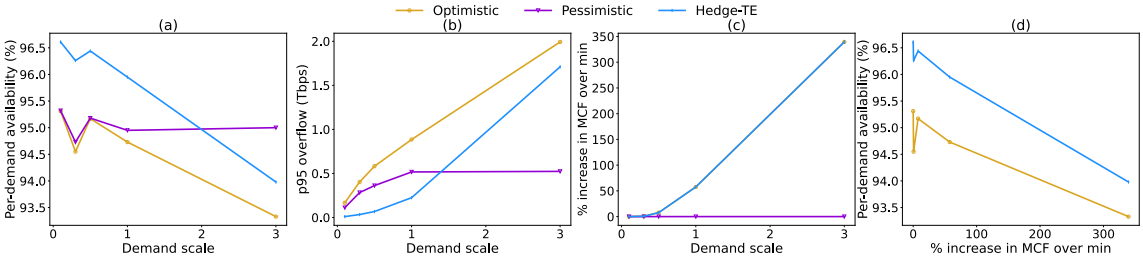


Figure 15: Evaluation on the CloudWAN topology using the maximum concurrent flow (MCF) objective.

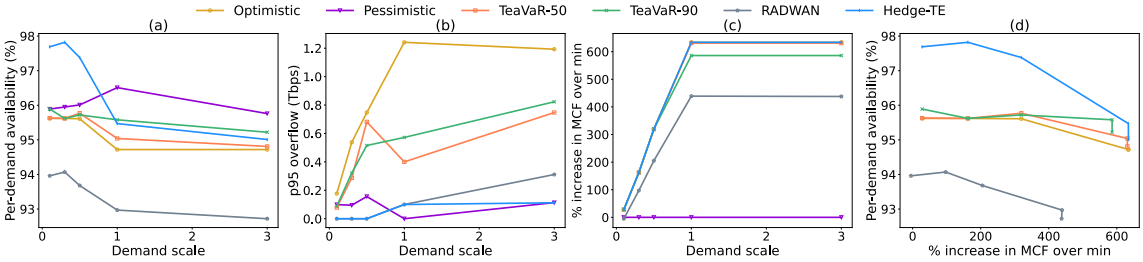


Figure 16: Evaluation on the B4 topology using the maximum concurrent flow (MCF) objective.

prior work [6] to provide explicit availability guarantees in TE using the CVaR minimization technique. This approach aims to bound the risk of "severe" reductions in availability using an operator-specified availability target $\beta \in [0,1]$. In [6], β corresponds to the percentage of the time that computed TE flow allocations can be realized, in spite of binary link failures. In our formulation of the CVaR minimization approach (Algorithm 4), β specifies the percent of the time that computed TE flow allocations can be realized, in spite of link capacity fluctuations. (In Algorithm 4, we show HEDGE-TE's CVaR formulation for maximizing total flow, but the expression for t_q can be changed back to the original TEAVAR expression for $t_{i,q}$ to maximize concurrent flow instead.) Similar to TEAVAR, this approach scales exponentially in the number of links $|E|$, but the runtime complexity is $O(k^{|E|})$ for k capacity states per link instead of $O(2^{|E|})$; to reduce the time complexity, a similar scenario pruning strategy as TEAVAR's can be used. However, the approach *guarantees* meeting the operator's availability target β (see the proof in [6]), but potentially at the cost of network flow and substantial runtime scaling challenges. We note that the scenario loss function $y_r(q)$ for our CVaR minimization approach in Alg. 4 now defines the "displaced flow" for the tunnel by the tunnel's bottleneck link in that scenario, ensuring that all flows proportionally reduce their flow by *at least* the percent capacity reduction for their bottleneck links. This is adapted from the original $y_r(q)$ used in [6], but is even more combinatorially expensive to compute.

Capacity constraint reformulation. Another approach to approximate a guaranteed β availability target for the operator is to enumerate all possible network scenarios (via independently fluctuating link capacities) and then rank scenarios from highest combined link capacity to lowest combined link capacity (in cases of ties, the scenarios can be ordered by increasing probability). Then, iterating from the highest-capacity scenario to the lowest-capacity scenario, we sum the probability to generate a running-sum cumulative probability measure, A. As soon as $A \geq \beta$, we select the "crossing scenario", *i.e.,* the last scenario before $A \geq \beta$, and set the capacity constraint for each link equal to the link's capacity in this scenario. Aside from this modification to the capacity constraint, everything about the HEDGE-TE formulation (Algorithm 1) remains the same, though additional speedups can be gained by eliminating $p_{e,z}$ and $z \in Z_e$ for which $c_{e,z}$ is less than the value of c_e for the crossing index. In general, this approach ensures that the TE allocation achieves approximately β availability for the operator, while also intelligently allocating traffic (via the HEDGE-TE objective) for scenarios that fall outside the scope of β. However, this approach would be highly risk-averse and underutilize the network, since it bounds flow allocations by the link capacities at the crossing index, even though links are most often operating at their maximum (provisioned) capacity. Further, the ranking and iterating process still approximates the availability target, so there may be minor discrepancies.

A.8 Original HEDGE-TE Formulation and Proof of Scale Reduction

Our original approach to develop HEDGE-TE was to consider *network scenarios*, as TEAVAR does, but with each scenario corresponding to a unique combination of link capacities. Network scenarios with stochastic link capacities increase scale even more rapidly than a binary model of link failures (as in TEAVAR) because there can often be more than one state per link, and a link can develop new states (and lose old states) over time. We share the original formulation in Algorithm 5.

Next, we highlight the complete proof of scale reduction, which shows that Algorithm 5 is equivalent to the scale-reduced HEDGE-TE solution, shown in Algorithm 1. The objective in Algorithm 5 is to maximize

$$\sum_{q \in Q} p'_q \Big(\sum_i \sum_{r \in R_i} x_r - \sum_{e \in E} s_{e,q} \Big) \tag{13}$$

Algorithm 4 HEDGE-TE: CVaR Minimization

Inputs:

$G\langle V,E\rangle$:	network G with vertices V and links E
β:	target availability level
$q \in Q$:	network scenario with different link capacities
p'_q:	probability of network state q
$d_i \in D$:	traffic demand of source-destination pair i
$R_i \in R$:	set of tunnels for flow i
$c_{e,q}$:	capacity of link e in scenario q
$c_e^{\max}$:	maximum capacity of link e $(\max_{z \in Z_e} c_{e,z})$
$y_{r,q}$:	relative capacity of tunnel r in scenario q $\left(\min_{e \in r} \frac{c_{e,q}}{c_e^{\max}}\right)$

Auxiliary Variables:

s_q:	loss in scenario q

Outputs:

x_r:	flow allocation along tunnel r
α:	the value at risk

Minimize: $\alpha + \frac{1}{1-\beta}\sum_{q \in Q} p'_q s_q$

subject to:

$$\sum_{r \in R_i} x_r \leq d_i \qquad \forall d_i \in D$$
$$\sum_i \sum_{r \ni e} x_r \leq c_e^{\max} \qquad \forall e \in E$$
$$s_q \geq t_q - \alpha \qquad \forall q \in Q$$
$$s_q \geq 0 \qquad \forall q \in Q$$
$$t_q = \sum_i (d_i - \sum_{r \in R_i} x_r y_{r,q}) \qquad \forall q \in Q$$

This objective function is equivalent to

$$\sum_{q \in Q} p'_q \sum_i \sum_{r \in R_i} x_r - \sum_{q \in Q} p'_q \sum_{e \in E} s_{e,q} \tag{14}$$

Because $\sum_{q \in Q} p'_q = 1$ and x_r has no dependency on q, this can be rewritten as

$$\sum_i \sum_{r \in R_i} x_r - \sum_{q \in Q} p'_q \sum_{e \in E} s_{e,q} \tag{15}$$

Now that we have dealt with the first term, we rewrite the second term to avoid scaling in $|Q|$. Recall that p'_q represents the probability of network state q, and p'_q is the product of the probabilities of each of the link capacities occurring to create that network state. Thus, each link e has a discrete probability distribution of capacities, in which e can have capacity z with probability $p_{e,z}$, and because each link fluctuates independently, we have $p'_q = \prod_{e \in E} p_{e,z}$ (fixing a capacity value z for each link e in order to create scenario q).

Note that in Algorithm 5, we have defined

$$s_{e,q} = \max\{(\sum_{r \ni e} x_r) - c_{e,q}, 0\}. \tag{16}$$

For every link e, the value of $s_{e,q}$ will be equivalent $\forall q$ in which $c_{e,q}$ is the same value since the $\sum_{r \ni e} x_r$ term does not depend on q. Thus, instead of considering network scenarios

Algorithm 5 Original HEDGE-TE Optimization

Inputs:

$G\langle V,E\rangle$:	network G with vertices V and links E
$q \in Q$:	network scenario with different link capacities
p'_q:	probability of network state q
$d_i \in D$:	traffic demand of source-destination pair i
$R_i \in R$:	set of tunnels for flow i
$c_{e,q}$:	capacity of link e in network state q
$c_e^{\max}$:	maximum capacity of link e (i.e., $\max_{q \in Q}\{c_{e,q}\}$)

Output:

x_r:	flow allocation along tunnel r

Maximize: $\sum_{q \in Q} p'_q (\sum_i \sum_{r \in R_i} x_r - \sum_{e \in E} s_{e,q})$

subject to:

$$\sum_{r \in R_i} x_r \leq d_i \qquad \forall d_i \in D$$
$$\sum_{r \ni e} x_r \leq c_e^{\max} \qquad \forall e \in E$$
$$s_{e,q} = \max\{(\sum_{r \ni e} x_r) - c_{e,q}, 0\} \qquad \forall e \in E, q \in Q$$

$q \in Q$, we can instead consider a set of link states $z \in Z_e$ for every link e, where each state z occurs with probability $p_{e,z}$ $(\forall e : \sum_{z \in Z_e} p_{e,z} = 1)$. We now define

$$l_{e,z} = \max\{(\sum_{r \ni e} x_r) - c_{e,z}, 0\} \tag{17}$$

where $c_{e,z}$ is the capacity of edge e in state z. Note that for every $z \in Z_e$, $c_{e,z}$ will inherently have a different value, unlike for $c_{e,q}$ with respect to $q \in Q$. Therefore,

$$\sum_{q \in Q} p'_q \sum_{e \in E} s_{e,q} = \sum_{e \in E} \sum_{z \in Z_e} p_{e,z} l_{e,z} \tag{18}$$

since $\sum_{q \in Q} p'_q \mathbb{1}_{\{c_{e,q}=k\}} = p_{e,z}$ for a given link e with capacity k in state z. Thus, our objective can now be written as maximizing

$$\sum_i \sum_{r \in R_i} x_r - \sum_{e \in E} \sum_{z \in Z_e} p_{e,z} l_{e,z} \tag{19}$$

which means that our solution no longer scales exponentially in the number of link states, but instead linearly.

A.9 Post-Processing HEDGE-TE's Allocations

In worst-case scenarios where HEDGE-TE *must* take calculated risks that could result in some link capacities being exceeded in reduced states (due to the demand being so high), the controller can run Alg. 6 to reduce flow allocations to comply with link capacities in the current state. HEDGE-TE's high availability over all demand scales means that post-processing is not run often, but when it is run, as shown in Fig. 9, it impacts very few tunnels and can be computed in less than 4 milliseconds. We note that post- processing minimizes the amount of flow that must be reduced compared to naively reducing allocations by the magnitude of edge overallocation because it accounts for the fact that multiple links along the same tunnel can fluctuate simultaneously. In such cases, it may be possible to reduce the allocation by less than the sum total of link overal-

Algorithm 6 Post-processing Algorithm

Inputs:

$G\langle V,E\rangle$:	network G with vertices V and edges E
q:	current network state (fixed link capacities)
$R_i \in R$:	set of tunnels for flow i
$c_{e,q}$:	capacity of edge e in the current network state q
x_r:	flow allocation on tunnel r
F_q:	subset of edges with exceeded capacity

Output:

ε_r:	quantity of flow to reduce along tunnel r

Minimize: $\sum_{e\in F_q}\sum_{r\ni e}\varepsilon_r$
subject to:

$$\begin{array}{ll} \sum_{r\ni e}(x_r-\varepsilon_r)\le c_{e,q} & \forall e\in F_q \\ \sum_{r\ni e}(x_r-\varepsilon_r)\ge 0 & \forall e\in F_q \\ \varepsilon_r\ge 0 & \forall r \end{array}$$

locations along the tunnel, since reducing flow along the tunnel by some amount can count "doubly" towards the reduction.

A.10 Evaluation on the ATT Topology

We also evaluated HEDGE on the ATT topology. The results in Fig. 17 highlight a similar pattern, with HEDGE presenting the best performance in the throughput-availability trade-off.

A.11 HEDGE-TE Performance on KDL

In addition to the runtime analysis, we also analyze the performance of HEDGE-TE on KDL using the maximum throughput objective. Due to the lack of any publicly available demand data, we synthesize a demand matrix using the popular gravity model [49]. Based on prior work [28] and our empirical analysis of demands in CloudWAN, we keep the top 20% of demand pairs as active (in reality, 20% is a high proportion and functions as a stress-test of the controller). We also lack any link capacity distributions for KDL, and unlike the other topologies, which were smaller than CloudWAN, KDL contains substantially more links, making the process of assigning capacity distributions to links especially challenging. Thus, in our evaluation, we create the stochastic topology by (1) assuming that 50% of links will fluctuate, and (2) for each link with stochastic capacity, randomly selecting a link capacity distribution observed in CloudWAN and then normalizing this capacity distribution by the KDL link's provisioned capacity. Similar to the approach for other topologies, we repeat this random matching process 10 times, to generate 10 different stochastic topologies (in terms of which links have which capacity distributions) and thereby eliminate any confounders with how exact links are matched to capacity distributions. Then, for each topology variant, we run 100 simulations of sampling independently from link capacity distributions to simulate a network state, and measure throughput, availability, and overflow identically as is described in §6. Thus, our results from KDL, shown in Figure 18 are drawn from 1000 total simulations. We note that TEAVAR did not scale, and we exclude RADWAN from the figures because it significantly underperformed the other baselines due to the frequent capacity upgrades and downgrades that resulted in link downtime.

Figure 18 shows that HEDGE-TE continues to outperform both the Naive Optimistic and Pessimistic baselines and best trades off throughput and availability across all demand scales. We note that the low raw availability levels seen on the y-axis are likely due to the definition of availability as the percentage of simulations in which an initial allocation had to be reduced due to a link capacity fluctuation; because KDL has many more links, and we stress-tested the controller by assuming 50% of links have stochastic capacity, this is just more likely to happen as the topology size increases (with each additional stochastic link, the baseline availability inherently reduces).

A.12 HEDGE-TE with Adversarial Input Distributions

As mentioned in §8, we use the KL divergence [29] between consecutive distributions to measure how well historical link capacity distributions capture the current state of the network. KL divergence is calculated as

$$D_{KL}(P||Q) = \sum_{x\in X} P(x)\log\left(\frac{P(x)}{Q(x)}\right), \tag{20}$$

where P is the current/future link capacity distribution and Q is the historical distribution that would be used as input to HEDGE-TE. (Note that D_{KL} is asymmetric, meaning $D_{KL}(P||Q) \neq D_{KL}(Q||P)$, so we particularly choose Q as the previous distribution and P as the next since this captures Q's ability to predict P.) Since $\log\left(\frac{P(x)}{Q(x)}\right)$ is undefined when either $P(x)$ or $Q(x)$ is 0 (*i.e.,* some state is observed in one distribution but never in the other within that time window), we use the default value of 1×10^{-5} as the probability (value of $P(x)$ or $Q(x)$) in such scenarios, as is common practice [44]. In our analysis, we pick a time window to compute link capacity distributions, and for each link, calculate the KL divergence between consecutive distributions for each timestep (where each timestep is of the time window's length) using the empirical data from link capacity distributions in CloudWAN. Results are reported by averaging the KL divergence values across all consecutive timesteps and all links for each time window we analyzed, ranging from 5 minutes to 1 day. The results are shown in Fig. 11. While the average KL divergence values tend to be close to zero regardless of the time window used for calculating link capacity distributions—though D_{KL} on average tends to increase as the time window gets larger—there can be outliers with high KL divergence (*i.e.,* the past link capacity distribution is not very predictive of the future).

We examined these outlier situations with high KL divergence and found that most often they resemble two scenarios:

- The probabilities of existing link capacity states change substantially from one time period to the next.

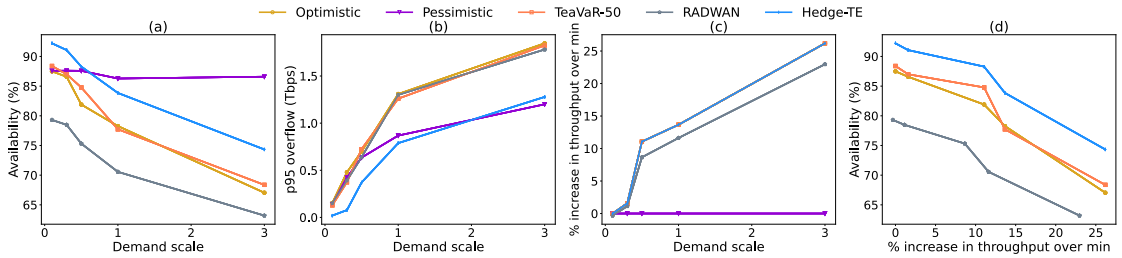


Figure 17: Evaluation on the ATT topology: throughput, availability, and worst-case flow reductions across demand scales.

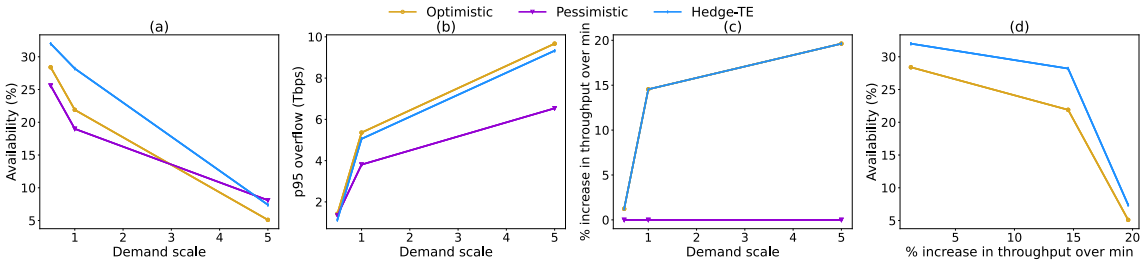


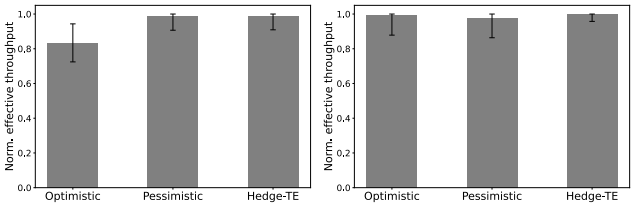
Figure 18: Evaluation on the KDL topology using the maximum throughput objective.

- One (or more) new link capacity states are present in the next time period that were not observed previously.

Not only do these edge cases result in a high KL divergence, but they also would inherently present the greatest challenge to HEDGE-TE. We evaluate HEDGE-TE in these edge cases (on the CloudWAN topology) with two experiments, one for each type of scenario.

Experiment 1: Flipping the probability of states between the historical and observed distributions. We take the two most probable non-zero capacities for each link and assign them a probability, where the most probable capacity (c_1) retains its probability p and the next most probable (c_2) is given probability $1-p$. (Note that the most probable capacity in our data from CloudWAN is always the maximum capacity of the link.) Thus, the link capacity distribution would be $Q = \{c_1 : p,\ c_2 : 1-p\}$. We use this distribution to compute HEDGE-TE's flow allocations as well as the Naive Optimistic and Pessimistic baselines. Then, we conduct 1000 Monte Carlo simulations for each of the 10 random topology permutations but in each simulation, links are sampled independently from an adversarial distribution where the probabilities of capacities are flipped, *i.e.,* $P = \{c_1 : 1-p, c_2 : p\}$.

Experiment 2: Adding new states that were not observed in the historical distribution when computing flow allocations. To construct the historical distribution, Q, we remove the second- and third-most probable non-zero capacities if more than two states exist, or just the second-most probable non-zero capacity otherwise, and normalize probabilities to sum to 1. (We do not remove the most probable capacity because this is the link operating at its provisioned capacity—a state that would always be seen or at least directly encoded into the input—or zero capacity because it is just the probability of traditional link failure.) We use this distribution to compute HEDGE-TE's flow allocations as well as the Naive Optimistic and Pessimistic baselines. Then, we conduct 1000 Monte Carlo simulations for each of the 10 random topology permutations, but in each simulation, links are sampled from their *original* distribution P, which contains at least one or two capacity states that were removed from Q. We note that the higher probability capacities were removed from Q intentionally to stress-test the controller.



(a) Flipping state probabilities. (b) Adding previously unseen states.

Figure 19: HEDGE-TE's performance on adversarial inputs. The height of the bars is the average over all 10,000 simulations, and error bars denote the 1st and 99th percentiles.

For both experiments, we report the normalized effective throughput—raw throughput minus flow reductions due to over-allocations in simulations where links have reduced capacity, divided by the maximum value across all methods and simulations—for all three methods at $1\times$ demand scale in the CloudWAN network. As shown in Fig. 19, HEDGE-TE has superior performance to the state-of-the-art Optimistic baseline (maximize throughput using the provisioned/maximum link capacities) in both experiments. HEDGE-TE outperforms the Pessimistic baseline in Experiment 2, when previously unseen capacities are sampled, and performs on par in Experiment 1, when state probabilities are flipped. The reason why Pessimistic outperforms Optimistic in Experiment 1 is because the simulations in this experiment most often contain links at their reduced/minimum state, since there are only 2 states and the reduced capacity is less likely in the historical distribution but more likely in the simulated distribution.