PCF: Provably Resilient Flexible Routing

Chuan Jiang
Purdue University
jiang486@purdue.edu

Sanjay Rao
Purdue University
sanjay@ecn.purdue.edu

Mohit Tawarmalani
Purdue University
mtawarma@purdue.edu

ABSTRACT

Recently, traffic engineering mechanisms have been developed that guarantee that a network (cloud provider WAN, or ISP) does not experience congestion under failures. In this paper, we show that existing congestion-free mechanisms, notably FFC, achieve performance far short of the network's intrinsic capability. We propose PCF, a set of novel congestion-free mechanisms to bridge this gap. PCF achieves these goals by better modeling network structure, and by carefully enhancing the flexibility of network response while ensuring that the performance under failures can be tractably modeled. All of PCF's schemes involve relatively light-weight operations on failures, and many of them can be realized using a local proportional routing scheme similar to FFC. We show PCF's effectiveness through formal theoretical results, and empirical experiments over 21 Internet topologies. PCF's schemes provably out-perform FFC, and in practice, can sustain higher throughput than FFC by a factor of 1.11X to 1.5X on average across the topologies, while providing a benefit of 2.6X in some cases.

CCS CONCEPTS

• **Networks → Network performance evaluation**; **Data path algorithms**;

KEYWORDS

network optimization, network resilience

ACM Reference Format:
Chuan Jiang, Sanjay Rao, and Mohit Tawarmalani. 2020. PCF: Provably Resilient Flexible Routing. In *Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication (SIGCOMM '20), August 10–14, 2020, Virtual Event, NY, USA.* ACM, New York, NY, USA, 15 pages. https://doi.org/10.1145/3387514.3405858

1 INTRODUCTION

Failures are the norm in both ISP networks [27, 35], and cloud provider WANs [12, 14, 30]. Yet networks must ensure that the increasingly stringent performance requirements of business critical applications are met [18]. Many recent works [17, 20, 23] have developed flexible ways of routing traffic motivated by the goal of efficiently utilizing network capacity. However, with these schemes, the network could experience congestion on failure [26].

Motivated by these challenges, the research community has recently designed traffic engineering mechanisms that proactively ensure that the network is congestion-free (i.e., ensure that no link carries more traffic than its capacity) under typical failure scenarios [26, 32, 37]. For instance, FFC [26], a representative and state-of-the-art approach, allocates bandwidth to flows so that no congestion occurs when f or fewer links fail. To do so, FFC splits traffic from each ingress to egress along a set of pre-specified tunnels.

In this paper, we explore the performance of such congestion-free mechanisms relative to the performance that a network could achieve by responding optimally to each failure. We refer to the performance achieved when the network responds optimally as the intrinsic network capability. We make two **contributions**.

First, we show that congestion-free schemes perform much worse than optimal, and present deeper insights into the underlying reasons. In particular, we show that (i) FFC is not only conservative, but also its performance can *degrade* with an increase in the number of tunnels; and (ii) the performance of FFC can be *arbitrarily worse than optimal*, even when *exponentially many tunnels* are used. We show that these results arise because (i) FFC models network structure in a coarse fashion; and (ii) reservations are tightly coupled to paths, and the failure of a link leads to unutilized capacity on other links in the tunnel that contain the failed link.

Second, we propose PCF (Provably Congestion-free and resilient Flexible routing), a set of novel mechanisms that ensure the network is *provably congestion-free* under failures, while performing closer to the network's intrinsic capability. PCF achieves these goals by better modeling network structure, and through more flexible response strategies. The key challenge that PCF addresses is how to enhance the flexibility of network response while ensuring that the performance under failures can be tractably modeled.

We develop multiple mechanisms as part of PCF that allow the architect to trade-off the achievable performance guarantee with deployment complexity. First, we present an alternate approach for bandwidth allocation with the FFC response mechanism which (i) results in a better performance guarantee; and (ii) ensures the allocation does not degrade with additional tunnels. Second, we explore more flexible network response based on an abstraction that we term **logical sequence (LS)**. A LS from a source to a destination traverses a series of logical segments (formally defined in §3.3). The reservation on any LS for a targeted failure set is guaranteed by the logical segments constituting the sequence. Each segment may recursively route traffic over other LSs or physical tunnels servicing that segment. This allows for significant flexibility in how traffic is routed over various segments, and which nodes respond to a given failure. LSs are loosely inspired by ideas such as segment routing [16, 34] though with significant differences (§6). We show that when LSs are used, the performance can be arbitrarily better than FFC. We develop several mechanisms based on LSs, including

SIGCOMM '20, August 10–14, 2020, Virtual Event, NY, USA

ACM ISBN 978-1-4503-7955-7/20/08…$15.00
https://doi.org/10.1145/3387514.3405858

those that provably out-perform R3 [37], another congestion-free mechanism.

We show how PCF's mechanisms can be implemented in practice. For example, we show that when LSs are chosen with some restrictions, they can be realized by a simple generalization of the local proportional routing scheme used by FFC. When LSs are arbitrarily chosen (which allows for even better performance guarantees), our approach discovers a viable routing using techniques that are lighter weight than the the optimal network response strategy.

Empirical evaluations of PCF over 21 topologies from the Internet Topology Zoo show that PCF significantly out-performs FFC. PCF's schemes can sustain higher throughput than FFC by a factor of 1.11X to 1.5X on average across the topologies, while providing a benefit of 2.6X in some cases.

2 MOTIVATION

A critical task for network architects is to ensure that their network designs can sustain desired traffic over a target set of failures [26, 32, 37]. This in turn depends on the mechanisms that the network uses to respond to failures.

To illustrate these issues, consider tunnel-based forwarding [23, 26, 33], where traffic from each ingress to egress is carried over a set of pre-selected tunnels. When a tunnel is no longer available (e.g., due to the failure of an underlying link), then, traffic is redistributed across the surviving tunnels. Redistributing traffic can potentially overload some links. A congestion-free routing mechanism guarantees that the network has been proactively designed so no link would be over-loaded over a desired set of failures [26, 32, 37].

FFC [26] is a recent and representative approach set in the context of tunnel-based forwarding. Consider a network where each pair of nodes (s, t) is associated with a traffic demand d_{st}, and a set of tunnels $T(s, t)$ to route the traffic. FFC seeks to assign a bandwidth bw_{st} to each node pair such that this bandwidth can be guaranteed under all possible f simultaneous link failures. To achieve this, FFC reserves bandwidth on each tunnel, and ensures that the total reservation on all tunnels in $T(s, t)$ exceeds bw_{st} under every failure scenario of interest. We present examples to illustrate why FFC is conservative.

Coarse modeling of network structure. Consider Fig. 1 where the goal is to carry the maximum amount of traffic possible from s to t, while tolerating any possible single link failure. If the network could respond optimally for each failure scenario (by running an optimal multi-commodity flow for that scenario), it is easy to verify that the network is intrinsically capable of carrying 2 units of flow from s to t under all possible single link failures. When FFC is used, the results depend on the set of tunnels considered. We consider two schemes: (i) FFC-4 (all 4 tunnels $l1$ to $l4$ are used); and (ii) FFC-3 (only 3 tunnels $l1$ to $l3$ are used). Fig. 2 shows that both schemes perform worse than optimal, and surprisingly, FFC-4 performs worse.

We now explain why FFC is *conservative*, and why its performance may *degrade* with *more tunnels*. FFC uses a parameter p_{st} which denotes the maximum number of tunnels between s and t that share a common link. When designing to tolerate f link failures, FFC conservatively assumes that upto fp_{st} tunnels may fail, and plans a reservation that can tolerate all possible failures of fp_{st} tunnels. In Fig. 1, when FFC uses all 4 tunnels, p_{st} is 2. Hence, when designing for single link failures, FFC-4 plans for all possible combinations of two tunnel failures. This is conservative because tunnels $l1$ and $l2$ do not fail together under single link failures. With FFC-3, all tunnels are disjoint, and $p_{st} = 1$. Hence, FFC-3 only needs to be consider single tunnel failures. However, FFC-3 still cannot match the optimal since it cannot tap into the capacity of links $s - 4$ and $4 - 3$.

Fig. 2 also shows that if all two link failures must be tolerated, the throughput with the optimal, FFC-3, and FFC-4 are 1, 0.5, and 0 respectively. The reasons are similar – FFC-4 can only service traffic that can survive $p_{st}f = 2 \times 2 = 4$ tunnel failures, and hence cannot carry any traffic, while FFC-3 only needs to consider all 2 tunnel failure scenarios.

Limitations of tunnel reservations. A second issue with FFC is that it is inherently limited by the fact that reservations are made at the granularity of entire tunnels. To illustrate this, consider Fig. 3. It is easy to verify that if the network responds optimally, it can carry 2/3 units of traffic from s to t under any single link failure. Unfortunately, FFC can only achieve an optimal of 1/2. In §3.3, we will further generalize this example to show that FFC can see arbitrarily poor performance relative to optimal.

Tunnel-based allocation does not perform as well as optimal because reservations are made on all links of a tunnel, and when a link fails, the reservations on other links of that tunnel go unutilized. For example, consider a tunnel l that traverses links e_1 and e_4. When e_4 (and hence the tunnel l) fails, FFC only uses the reservations on the remaining tunnels, and the reservation on e_1 for the failed tunnel l goes unutilized. In contrast, the optimal approach is able to use all capacity on all the non-failed links.

In Fig. 3, let T_4 and T_5 respectively denote the set of tunnels from s to t that use e_4 and e_5. Let r_4 and r_5 denote FFC's reservations on each of these sets of tunnels. FFC can carry at most r_5 units of traffic when e_4 fails, and at most r_4 units when e_5 fails. Thus, FFC can guarantee at most $\min(r_4, r_5)$ traffic from s to t over all single link failures. However, $\min\{r_4, r_5\} \times 2 \leq r_4 + r_5 \leq 1$, where the second inequality is because tunnels in $T4$ and $T5$ must reserve capacity in one of the links $e1$, $e2$, or $e3$, whose combined capacity is 1 unit. Hence, FFC can carry at most 0.5 units of traffic from s to t.

3 PCF OVERVIEW

PCF's primary goal is to bridge the gap between existing congestion-free routing mechanisms, and intrinsic network capability. PCF tackles the issues raised in §2 by better modeling, and adopting more flexible response strategies.

Unfortunately, not all routing strategies are amenable to formal guarantees on worst-case performance under failures. For instance, when the network responds with an optimal multi-commodity flow (the most flexible response), the problem of determining the worst-case performance under failures is intractable [10]. Thus, a central challenge that PCF tackles is one of carefully crafting response strategies that are (i) amenable to formal worst-case guarantees; and yet (ii) perform closer to the network's intrinsic capability.

PCF achieves the above by (i) developing tractable optimization formulations that are inspired by practical response mechanisms

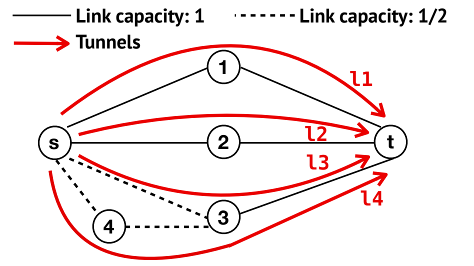

Figure 1: Example to illustrate FFC's coarse modeling of network structure.

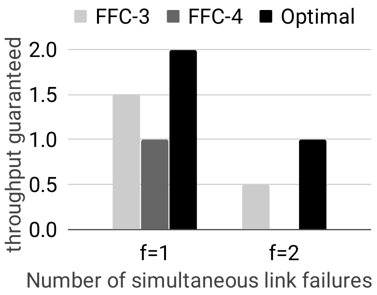

Figure 2: Throughput guarantee with FFC for different tunnel choices compared to the optimal.

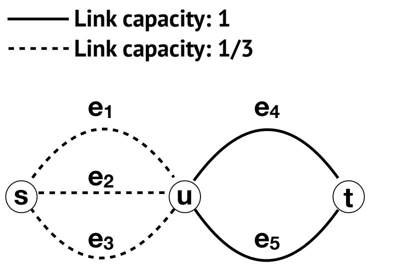

Figure 3: Example to illustrate how tunnel-based reservations can be inefficient

in networking and conservatively estimate network capability; and (ii) providing explicit response mechanisms that achieve the estimated capability. Moreover, PCF allows the network architect to incrementally dial-in additional flexibility in response as desired.

Roadmap. We introduce notation (§3.1), and present PCF-TF (§3.2), which uses FFC's response mechanism but better models network structure. We show that PCF-TF out-performs FFC, and achieves better performance with more tunnels. Despite these benefits, we show that PCF-TF (like FFC) can perform arbitrarily worse than optimal. We introduce a more flexible approach based on the logical sequence abstraction, and formally show the performance benefits over PCF-TF (§3.3). We present further generalizations in §3.4, and show how to practically realize the schemes (§4).

3.1 Notation and preliminaries

Consider a network topology, represented as a graph $G = \langle V, E \rangle$. Each link $e \in E$ is associated with a link capacity c_e. For each node pair (s, t) on the graph, we are given a traffic demand d_{st}, and a set of tunnels $T(s, t)$ to route the traffic. Each tunnel l consists of a set of links $\tau_l \subseteq E$. Below, we present a formulation for bandwidth allocation with tunnels,

$$\begin{aligned} (P1)\ \max_{z,a}\quad & \Theta(z) \\ \text{s.t.}\quad \sum_{l \in T(s,t)} & a_l(1-y_l) \ge z_{st} d_{st} \quad \forall s,t \in V, \forall y \in Y && (1) \\ & a_l \ge 0 \quad \forall s,t \in V, l \in T(s,t) && (2) \\ \sum_{\forall s,t \in V, l \in T(s,t)} & a_l \delta(e \in \tau_l) \le c_e \quad \forall e \in E. && (3) \end{aligned}$$

Here, $\delta(e \in \tau_l) = 1$ if $e \in \tau_l$ and 0 otherwise. The formulation determines a_l and z_{st}, where a_l represents the amount of reservation on tunnel l, and variable z_{st} represents the fraction of traffic from s to t that can be satisfied. Note that we use slightly different notation than FFC (see Table 2 in appendix). Y stands for the set of tunnel failure scenarios of interest, and y_l indicates whether tunnel l fails or not in a failure scenario ($y_l = 1$ indicates tunnel l fails and $y_l = 0$ otherwise.) We later discuss how Y is modeled). $\Theta(z)$ is the metric function we want to optimize. For tractability, we assume $\Theta(z)$ is a concave function, and note that this model covers common metrics such as overall throughput and maximum link utilization. For example, $\Theta(z) = \sum_{s,t} \min\{1, z_{st}\} d_{st}$ models overall throughput. Alternately, when $\Theta(z) = \min_{s,t}\{z_{st}\}$, and the optimal value is Θ^*, the model guarantees that Θ^* fraction of each flow can be sent in every failure scenario. This also means that using $1/\Theta^*$ of each link's capacity is sufficient to send all the flows. Hence, the inverse of this Θ^* is the utilization of the most congested link, also known as the Maximum Link Utilization (MLU). Thus, $\Theta(z) = \min_{s,t}\{z_{st}\}$ minimizes the MLU.

3.2 Modeling network structure

We now discuss how to model the set of failure scenarios Y. If at most p_{st} tunnels between s and t share a common link, FFC assumes that upto fp_{st} tunnels can fail under f link failures, and plans for *all possible combinations* of fp_{st} tunnel failures. As discussed in §2, this is conservative – e.g., for the network shown in Fig. 1, FFC considers the simultaneous failure of $l1$ and $l2$ even though this is impossible under single link failure. To address this, PCF more accurately models Y by better relating link and tunnel failures. Let x_e indicate if link e fails ($x_e = 1$ indicates link e fails and $x_e = 0$ otherwise). Then, PCF models Y as:

$$\begin{aligned} & \sum_{e \in E} x_e \le f \\ & x_e - y_l \le 0 \quad \forall l, e \in \tau_l \\ & y_l - \sum_{e \in \tau_l} x_e \le 0 \quad \forall l && (4) \\ & 0 \le x_e \le 1 \quad \forall e \in E \\ & 0 \le y_l \le 1 \quad \forall l. \end{aligned}$$

The first constraint bounds the maximum number of simultaneous link failures. The second ensures that the failure of an underlying link will cause the tunnel to fail. The third ensures that a tunnel only fails when at least one underlying link fails. We denote (P1) with Y modeled by (4) as PCF-TF. Observe that we do not explicitly impose that $x_e \in \{0, 1\}$ because, just as for FFC, the failure set Y may contain too many scenarios to enumerate. Instead, we conservatively relax this requirement to $x_e \in [0, 1]$. Then, the model PCF-TF (and all other models presented in this paper) can be solved using dualization to ensure the number of constraints is polynomial in the size of the network, a technique that has been widely used in prior networking papers [9, 26, 37]. Details are presented in the Appendix. Yet, we prove that (i) PCF-TF performs at least as well as FFC; and (ii) unlike FFC, the performance of PCF-TF does not degrade as more tunnels are added.

Proposition 1. *The feasible region (the set of all possible values of the variables that satisfy the constraints) of FFC is contained in the feasible region of PCF-TF, so PCF-TF performs at least as well as FFC (i.e., achieves the same objective or higher) for any metric.*

Proof. FFC models Y as

$$\begin{aligned} &\sum_{l \in T(s,t)} y_l \leq f p_{st} \quad \forall s,t \in V \\ &0 \leq y_l \leq 1 \quad \forall l. \end{aligned} \tag{5}$$

Let Y_0 be the set of tunnel failure scenarios considered by FFC (constrained by (5)) and let Y_1 be the set of tunnel failure scenarios considered by PCF-TF (constrained by (4)). We show that $\text{proj}_y Y_1 \subseteq Y_0$, where proj_y denotes projection of the set to y variables. For any $s,t \in V$, we sum the third constraint in (4) over all $l \in T(s,t)$ to get $\sum_{l \in T(s,t)} (y_l - \sum_{e \in \tau_l} x_e) \leq 0$. Then,

$$\begin{aligned} \sum_{l \in T(s,t)} y_l &\leq \sum_{l \in T(s,t)} \sum_{e \in \tau_l} x_e = \sum_{l \in T(s,t)} \sum_{e \in E} x_e \delta(e \in \tau_l) \\ &= \sum_{e \in E} x_e \sum_{l \in T(s,t)} \delta(e \in \tau_l), \end{aligned}$$

$\sum_{e \in E} x_e$ is the total number of link failures, which is no more than f. And $\sum_{l \in T(s,t)} \delta(e \in \tau_l)$ is the number of tunnels from s to t traversing link e, which is no more than p_{st}. Hence, we have $\sum_{l \in T(s,t)} y_l \leq f p_{st}$, which shows that any scenario in Y_1 also satisfies (5). Since FFC imposes (1) for each $y \in Y_0$ while PCF-TF imposes (1) for each $(x,y) \in Y_1$, PCF-TF is less constrained than FFC. □

The above proof does not depend on the objective function in the optimization problem, which means that the proposition holds for any metric. We next show that unlike FFC, PCF-TF's performance does not degrade with more tunnels. The intuition behind this is that when more tunnels are added to PCF-TF, the set of constraints that need to be satisfied does not increase. Hence, any solution feasible when fewer tunnels are employed remains feasible when tunnels are added (though new and better solutions may be possible). Thus the performance cannot get worse.

Proposition 2. *As we provide more tunnels, PCF-TF's performance cannot decrease.*

Proof. Let $\{T_0(s,t) \mid \forall s,t \in V\}$ and $\{T_1(s,t) \mid \forall s,t \in V\}$ be two sets of tunnels, and $T_0(s,t) \subseteq T_1(s,t)$ for all $s,t \in V$. Then, we show that the optimal value for (P1) with $T = T_1$ will not be worse than the optimal solution to (P1) with $T = T_0$. Let (a^*, z^*) be the optimal solution to (P1) with $T = T_0$. We construct (a', z') in the following way,

$$\begin{aligned} &a'_l = a^*_l \quad \forall s,t \in V, l \in T_0(s,t) \\ &a'_l = 0 \quad \forall s,t \in V, l \in T_1(s,t) - T_0(s,t) \\ &z'_{st} = z^*_{st} \quad \forall s,t \in V. \end{aligned} \tag{6}$$

Let Y_0 denote (4) with $T = T_0$ and Y_1 denote (4) with $T = T_1$. It is easy to see that projection of Y_1 onto the space of variables $\{x_e\}_{e \in E}$ and $\{y_l\}_{l \in T_0}$ is contained in Y_0, since all the constraints in Y_0 are present in Y_1. Now for each $y \in Y_1$,

$$\sum_{l \in T_1(s,t)} a'_l (1 - y_l) = \sum_{l \in T_0(s,t)} a^*_l (1 - y_l) \leq z^*_{st} d_{st} = z_{st} d_{st},$$

where the first equality is because $a'_l = 0$ for $l \notin T_0(s,t)$, the first inequality is because (a^*, z^*) is feasible for $T = T_0$ and the projection of Y_1 is contained in Y_0 and the last equality is by construction. Since z is not altered, the objective value remains the same. □

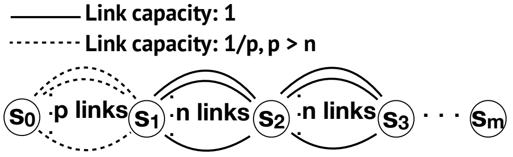

Figure 4: A topology with m nodes generalized from the previous example.

We later show in §5 that PCF-TF performs much better than FFC for real networks.

3.3 Modeling more flexible response

While PCF-TF is guaranteed to out-perform FFC, we begin by presenting a theoretical result that shows the performance of PCF-TF can still be arbitrarily worse than optimal because of the inflexibility of tunnel-based reservations. We then discuss PCF's more flexible approach.

Proposition 3. *The throughput guaranteed by PCF-TF (and hence that guaranteed by FFC) can be arbitrarily worse than the optimal even with exponentially many tunnels.*

Proof. Consider the topology in Fig. 4 (the example in Fig. 3 is a special case where $p = 3$, $n = 2$ and $m = 2$). Under any failure involving $n - 1$ links, the network can carry $1 - \frac{n-1}{p}$ units of traffic if it responded optimally. This is because under any such failure, the network can carry (i) at least 1 unit of traffic between s_i and s_{i+1}, $i > 0$; and (ii) at least $1 - \frac{n-1}{p}$ units of traffic between s_0 and s_1. Moreover, if $n - 1$ of the links between between s_0 and s_1 fail simultaneously, the traffic is no more than $1 - \frac{n-1}{p}$.

Next, consider PCF-TF, and assume that all possible tunnels between s and t are used. There are pn^{m-1} possible tunnels. We will show that PCF-TF can only guarantee traffic of $1/n$ units from s_0 to s_m under $n - 1$ simultaneous link failures. To see this, observe that the reservation across all tunnels between s and t is at most 1 (constrained by the capacity of all links between s_0 and s_1). Let r_i denote the reservation on all tunnels that use the i^{th} link between s_1 and s_2. Then, $\sum_{i=1}^{n} r_i \leq 1$, and there must exist at least one link j between s_1 and s_2 such that $r_j \leq 1/n$. Consider a failure scenario where all links between s_1 and s_2 except j fail. Under this scenario, PCF-TF can guarantee at most $1/n$ units of traffic from s_0 to s_m.

Note that $1 - \frac{n-1}{p} - \frac{1}{n} = \frac{(p-n)(n-1)}{pn} > 0$ whenever $p > n > 1$. Consider the case where $p = n^2$. Then, as n gets larger, the amount of traffic carried in the optimal solution converges to 1, while PCF-TF converges to 0. □

As discussed in §2, these issues with FFC and PCF-TF stem from the fact that reservations are made over entire tunnels, are tightly coupled to a particular network path, and are pre-allocated independent of any specific failure scenario. When a link in the tunnel fails, the corresponding capacity is unavailable in other links along the tunnel.

Logical sequences. PCF is motivated by the fact that more flexible methods of responding to failures can potentially address the limitations of FFC and PCF-TF highlighted by Proposition 3. However, even with more flexible response, PCF must proactively decide prior to any failure scenario how much traffic to admit so the network does not experience congestion over a given set of failure scenarios. Not all ways of making routing more flexible are amenable to provable congestion-free guarantees.

Instead, PCF considers a more carefully crafted flexible network response strategy, which we show is amenable to provable guarantees. Specifically, PCF introduces the notion of a *logical sequence (LS)*. A LS q from s to t consists of a series of routers $s, v_1, ..., v_m, t$ that we refer to as logical hops. Consecutive logical hops in a LS need not have a direct link between them, and in fact any pair of routers in the network could be consecutive logical hops. Traffic from s to t is required to traverse the logical hops $v_1, v_2, \ldots, v_m, t$, with significant flexibility in terms of how traffic is carried between two consecutive logical hops. In particular, traffic may be carried over physical tunnels (like FFC), or other LSs. We refer to each of $sv_1, v_1v_2, \ldots, v_mt$ as a logical segment of q. Each LS q is associated with a reservation b_q, which indicates that every segment of q is guaranteed to carry b_q traffic under all failure scenarios that PCF is designed for.

We next illustrate the potential benefits of LSs using Fig. 4. Consider the LS q which traverses the logical hops $s_0, s_1, ..., s_m$. Let each link be a tunnel. Traffic between consecutive logical hops is carried by the tunnels (links) connecting those hops. For example, traffic between s_1 and s_2 is carried on the n tunnels (links) connecting the nodes. When any link fails, only the reservation in the relevant segment of q is impacted – e.g., if a link between s_1 and s_2 fails, there is no impact on the reservation on the segment between s_0 and s_1. This is unlike FFC and PCF-TF where such a failure would cause part of the capacity on other links to be unavailable. The corollary to Proposition 3 below captures the resulting benefits.

Corollary 3.1. *For the topology in Fig. 4, PCF's performance with a single LS and polynomially many tunnels can be arbitrarily better than PCF-TF and FFC with exponentially many tunnels.*

Proof. We have already shown that FFC and PCF-TF can be arbitrarily worse than optimal. Consider PCF where LS q corresponding to $s_0, s_1, ..., s_m$ is used, with each link being a tunnel. There are $p + n(m-1)$ tunnels in total. Under any scenario involving $n-1$ simultaneous link failures, the first segment (s_0s_1) has a capacity of at least $1 - \frac{n-1}{p}$ available. All other segments have at least capacity 1 available on any $n-1$ failure scenario. Thus, q can carry at least $1 - \frac{n-1}{p}$ traffic, which meets the optimal throughput. □

We note that using the LS has at least two sources of flexibility beyond classic tunneling. First, in classic tunneling, traffic on each tunnel only carries traffic corresponding to the end points of the tunnel. Second, when there is a failure, only the source node of a tunnel may respond. In contrast, with a LS, each segment may carry traffic corresponding to different sources and destinations - for instance, in Fig. 4, the segment (and hence tunnel) between s_1 and s_2 may carry traffic between s_0 and s_m. Further, if the link between s_1 and s_2 fails, s_1 may redistribute the traffic that arrives at s_1 onto the tunnels between s_1 and s_2.

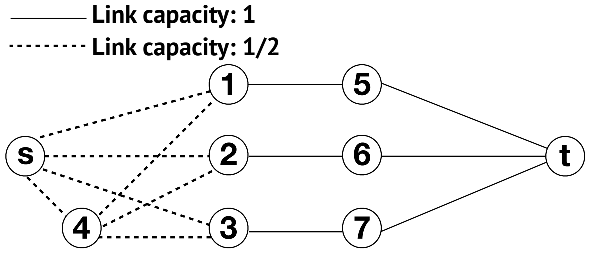

Figure 5: Illustrating conditional logical sequences

Optimal	FFC	PCF-TF	PCF-LS	PCF-CLS	R3
1	0	2/3	4/5	1	0

Table 1: Throughput of different schemes for the topology in Fig 5 under 2 simultaneous link failures.

Bandwidth allocation with LSs. We next show that bandwidth allocation with LSs can be tractably formulated. For each pair with source s and destination d, let $L(s,t)$ denote the set of LSs from s to t (with $T(s,t)$ denoting the set of tunnels as before). Note that each source destination pair is associated with zero or more LSs, and zero or more tunnels. Then, the model seeks to reserve b_q on each LS, and reserve a_l on each tunnel l as discussed below:

$$
\begin{aligned}
(P2)\ \max_{z,a,b}\quad & \Theta(z) \\
\text{s.t.}\quad & \sum_{l \in T(s,t)} a_l(1-y_l) + \sum_{q \in L(s,t)} b_q \\
& \geq \sum_{q' \in Q(s,t)} b_{q'} + z_{st}d_{st} \quad \forall s,t \in V, \forall y \in Y \qquad (7) \\
& b_q \geq 0 \quad \forall s,t \in V, q \in L(s,t) \\
& \text{Constraints (2), (3).}
\end{aligned}
$$

The most significant change relative to (4) pertains to the capacity constraint (first constraint). The LHS of this constraint captures that traffic from s to t could use both the reservations (a_l) on the physical tunnels between s and t, and the reservations (b_q) on the LSs between s and t. While the capacity on tunnel l is only available when all links on the tunnel are alive ($y_l = 0$), the reservation on the LS q is always available (though we relax this requirement in §3.4). The RHS of this constraint corresponds to the total traffic that must be carried from s to t. With FFC, this corresponds entirely to the bandwidth allocated to traffic that originates at s, and terminates at t. However, in PCF, it is possible that st is a segment of a LS q' (between a source s' and destination t'). Let $Q(s,t)$ denote the set of all such LSs. Then, the RHS also accounts for reservations on all such $q' \in Q(s,t)$. We refer to (P2) as the **PCF-LS** model. Note that the reservation on a LS is supported by the reservations on physical tunnels and other LSs. The reservations on physical tunnels themselves are supported by the capacity of underlying physical links.

3.4 Conditional Logical Sequences

As described in §3.3, each segment in a LS must guarantee the reservation associated with the LS over the entire set of failures. We next consider a generalization, that we call conditional LSs which only guarantee the reservation over a subset of failure scenarios. A conditional LS q is associated with a condition h_q, and a reservation b_q. The reservation b_q must be guaranteed over each segment of q for all scenarios where the condition h_q is met. An example condition is a given set of links being alive or dead.

Illustrating benefits of conditional LSs. We illustrate by considering Fig. 5. Table 1 shows the traffic guaranteed by different schemes for traffic from source s to destination t under single and two link failures. The table shows both FFC and PCF-TF (both schemes use all 6 tunnels from s to t) are sub-optimal (for the same reasons as (§3.2, §3.3)).

Consider now that a LS $(s, 4, t)$ is added with logical segments $s4$ (with the tunnel $s-4$), and $4t$ (with multiple tunnels from 4 to t including $4-1-5-t$, $4-2-6-t$, and $4-3-7-t$). Further, the LS is associated with a condition that the reservation is only needed when the link $s-4$ is alive. Table 1 shows the optimal is achieved with this conditional LS (PCF-CLS). Consider two link failure case. When $s-4$ is dead, at most one of the tunnels $s-1-5-t$, $s-2-6-t$, $s-3-7-t$ are dead and the remaining can carry 1 unit of flow. When $s-4$ is alive, at most 2 of these tunnels are dead. Therefore, they can cary 0.5 units of flow. Finally, LS $(s, 4, t)$ can carry 0.5 units of flow since $s4$ is alive and at most 2 of the tunnels $4-1-5-t$, $4-2-6-t$ and $4-3-7-t$ are dead.

Note that when the same LS is added but without the attached condition, the objective is not optimal. This is because, the logical segment $s4$ cannot guarantee any reservation over single link failures when only the tunnel $s-4$ is used. It is possible to add more tunnels between s and 4 (e.g., $s-1-4$, $s-2-4$, $s-3-4$), which allows the LS $(s, 4, t)$ to be more resilient to failures (PCF-LS). However, this is at the cost of reservations on the tunnels from s to t, and consequently the objective is increased but still does not achieve the optimal.

Modeling conditional LSs. We next discuss how conditional LSs are modeled. Under any given failure scenario, let h_q indicate whether LS q is active or not. Like before, let y_l indicate whether tunnel l fails or not. Let (y, h) denote all y_l and h_q variables, and let YH denote all possible combinations of (y, h) under all scenarios involving the simultaneous failure of f links. To incorporate these conditions, we replace constraint (7) in (P2) with the constraint below, and refer to the resulting model as **PCF-CLS**.

$$\sum_{l\in T(s,t)} a_l(1-y_l) + \sum_{q\in L(s,t)} b_q h_q \geq \sum_{q'\in Q(s,t)} b_{q'} h_{q'} + z_{st} d_{st} \quad \forall s,t \in V, \forall (y,h) \in YH.$$

In §5, we show that LSs activated under a simple condition (a single link being dead) is sufficient to get good performance. To handle this, we model YH by adding constraints $h_q = x_{e_q}$ for each LS q to (4), where e_q is the link whose failure activates LS q. In the Appendix, we model a more general condition, where all links in a set η_q are alive and all links in a set ξ_q are dead, which helps generalize PCF to richer failures (e.g., node failures) (§3.5).

3.5 PCF generalizations

In this section, we discuss generalizations of PCF, and its relationship with R3 [37], another congestion-free mechanism.

Heuristics for selecting LSs. We present a heuristic for selecting LSs that works well empirically (§5). Our approach involves considering a more general model based on flows and decomposing the results of that model into LSs.

We begin by introducing *logical flows*, which are a generalization of LSs in that traffic is no longer constrained to visiting a sequence of hops. A logical flow w from s to t is captured by the flow balance constraints below:

$$b_w \geq 0 \quad \forall s,t \in V, \forall w \in W(s,t)$$

$$\sum_j p_w(ij) - \sum_j p_w(ji) = \begin{cases} b_w & \forall s,t, i=s, w\in W(s,t) \\ 0 & \forall s,t, i\neq s, i\neq t, w\in W(s,t) \\ -b_w & \forall s,t, i=t, w\in W(s,t) \end{cases} . \tag{8}$$

Here, b_w is the reservation associated with the logical flow, and $p_w(ij)$ is the amount of this reservation that must be supported on logical segment ij. Each logical flow w may itself be associated with a condition h_w, which indicates the reservation associated with w is only guaranteed when h_w is satisfied. Let $W(s,t)$ be the set of all logical flows for traffic from s to t. Then, relative to (P2), the logical flow model involves adding (8), and changing (7) to

$$\sum_{l\in T(s,t)} a_l(1-y_l) + \sum_{w\in W(s,t)} b_w h_w \geq \sum_{s',t'\in V, w'\in W(s',t')} p_{w'}(st) h_{w'} + z_{st} d_{st} \quad \forall s,t \in V, \forall (y,h) \in YH.$$

The first term on the RHS captures the reservation that must be supported on (s,t) for any logical flow w' from s' to t'.

To obtain LSs, we decompose the flow into sequences [9, 23]. For each flow $w \in W(s,t)$, this approach generates a derived graph with the same nodes as the original topology. For each node pair (i,j), if $p_w(ij) > 0$, we add an edge from i to j with the weight being $p_w(ij)$. Then, we search for the widest path from s to t on this graph, and use the sequence of hops in this widest path as a LS with condition h_w.

Relationship to link bypass. While we have focused on tunnel based mechanisms so far [26], we next discuss the relationship of our work to R3 [37], another congestion-free routing scheme. Instead of tunnels, R3 [37] focuses on a link bypass mechanism, where traffic on a link $e = \langle i,j \rangle$ is re-routed upon its failure, along a pre-computed flow from i to j and this flow does not use e.

We first illustrate using Fig. 5 that our models can out-perform R3. As Table 1 shows, when R3 is applied to Fig. 5, no traffic can be carried from s to t if two link failures must be tolerated. To understand why, consider a scenario where links $1-5$, and $5-t$ fail. Since a link bypass for $1-5$ must start at 1 and end at 5, and a link bypass for $5-t$ must start at 5 and end at t, no viable bypass paths exist for either link. Instead, an obvious feasible strategy is to route the traffic along the path $s-2-6-t$, an option that is not considered by R3 because s is not an end point of either $1-5$ or $5-t$.

We now state a more formal result:

Proposition 4. *A special case of PCF's logical flow model where conditions are restricted to the no failure or single link failure scenarios, and links are tunnels, dominates (performs as well as or better than) R3.*

Proof. To see this, consider the logical flow model under the conditions above. More specifically, for each node pair (s, t), we have a flow w with the condition being no failure and we constrain the flow to exactly serve the demand, i.e., $b_w = z_{st} d_{st}$. For node pair (i, j) which has an edge, we have a flow w with the condition being the link (i, j) being dead. This model is exactly the Generalized-R3 model presented in [10] which has been shown to dominate R3. □

Shared risk links groups (SRLGs) and node failures While we have focused on link failures, a few modifications allow for the treatment of shared risk link groups (SRLGs), and node failures. An SRLG captures that a group of links may fail together (e.g., owing to failure of an underlying optical element) [27]. Each SRLG is modeled by a condition h_q which indicates all links in that SRLG fail. Observe that the first constraint in (4) is imposed on x variables that capture link failures. Instead, the constraint can be imposed on conditions dependent on the x variables. For example, a requirement that at most f SRLGs fail is modeled by requiring that $\sum_{q \in Q} h_q \leq f$, where Q is the set of SRLGs. Similarly, the failure of each node is modeled by a condition that all links incident on that router fail. Our discussion and results in §3.2 holds for node failures as well - i.e., relative to FFC, PCF-TF performs better, and PCF-TF's performance does not degrade with tunnels. Further, our models do not suffer from the weaknesses of link bypass mechanisms including R3 [37], that cannot deal with node failures (since no viable bypass paths for link $\langle i, j \rangle$ from i to j exist when node j itself fails).

4 REALIZING PCF'S MECHANISMS

In this section, we discuss how to realize PCF's network response mechanisms associated with the models in §3.

First, PCF-TF employs the same response mechanism as FFC, which we describe in the rest of this paragraph. Under any failure scenario, traffic across tunnels between a source s and destination t is carried on all live tunnels, and in proportion to the reservations on the tunnels. Consider three tunnels from s to t with reservations of $(2, 3, 5)$. When all the tunnels are alive, the (s, t) traffic is split across the tunnels in the ratio $(0.2, 0.3, 0.5)$. If the first tunnels fails, the traffic is sent across the tunnels in the ratio $(0, \frac{0.3}{0.8}, \frac{0.5}{0.8})$.

We next discuss the response mechanisms associated with our models based on LSs. First, we discuss a mechanism that works for arbitrary LSs (§4.1). We then show that when LSs are topologically sorted (more formally explained later), a response mechanism similar to FFC may be used (§4.2).

4.1 Realizing general logical sequences

Consider Fig. 6(a) which shows the physical tunnels and the LSs used with our offline PCF-LS, and PCF-CLS models for an example setting (e.g., $l1$ is a physical tunnel between A and C, while $q1$ is a LS between A and D) where traffic is carried from A to B. These models determine the reservations associated with each tunnel, and each LS (e.g., a_{l1} and b_{q1} are respectively the reservation on $l1$ and $q1$).

We discuss an approach to realize this abstract model only using tunnels (in §4.2, we discuss an alternate implementation). While in FFC, a tunnel l from i to j may carry traffic only from i to j, PCF permits some flexibility – e.g., l may carry traffic from s to t if in the abstract model, (i, j) is a segment in a LS from s to t.

Like FFC, our models are run at the granularity of several minutes to periodically recompute reservations (e.g., to handle significant shifts in traffic demands). Once computed for a given traffic matrix, we show that the traffic carried on tunnel l to destination t for any failure scenario may be computed online by solving a system of linear equations, which is much faster than solving linear programs (LPs) such as the multi-commodity flow problem (e.g., a popular approach to solving LPs involves solving many linear systems).

In describing our approach, it is helpful to consider a matrix M that summarizes the reservations. For instance, for the topology in Fig. 6, the reservation matrix M is summarized in Fig. 7. Each row and column corresponds to a node pair. The diagonal entries indicate the total reservation across all live tunnels and active logical sequences associated with that node pair. A non-diagonal entry in column i and row j indicates that the node pair j must carry traffic corresponding to column i. For instance, in the third row corresponding to the node pair (A, D), the diagonal entry $a_{l3}+b_{q1}$ is the total reservation associated with that node pair (over tunnel $l3$ and LS $q1$). Further, the entry $-b_{q2}$ reflects that (A, D) is a segment of the LS $q2$ from A to B and must be able to carry the reservation b_{q2} associated with $q2$.

A node pair (s, t) is considered to be of interest if it carries positive demand, or if it carries traffic for another node pair of interest. Let P be the set of node pairs of interest (more formally defined in the Appendix). Constraint (7) in our LS model can be equivalently expressed in matrix notation as $M \times \vec{1} \geq_v \vec{D}$. Here, $\vec{1}$ and $\vec{D}$ are $P \times 1$ column vectors. All entries of $\vec{1}$ are 1, while the p^{th} row of $\vec{D}$ has an entry $z_p d_p$ indicating the total traffic associated with pair p that can be carried. Let $\vec{U}$ be a $P \times 1$ column vector. Then, we have:

Proposition 5. *M is an invertible M-matrix*[1]*, and there is a unique solution $\vec{U}^*$ to the linear system $M \times \vec{U} = \vec{D}$, where $\forall (i, j) \in P$, $\vec{U}^*(i, j) \in [0, 1]$.*

We defer a proof to the appendix but discuss the implications here. While PCF's models determine the reservations, realizing them in practice requires determining the fraction of the reservation that is actually used in any given failure scenario. The above result indicates that such a fraction exists and may be obtained as a solution to a linear system of equations. While linear systems are already much faster to solve than LPs, the result also indicates that the matrix M is of a type for which simple and memory-efficient iterative algorithms for solving linear systems can be used [4].

For $t \in V$, let $\vec{D_t}$ be a $P \times 1$ column vector where the p^{th} row of $\vec{D_t}$ has an entry $z_p d_p$ if t is an end point of p, and 0 otherwise. Using the same argument as for Proposition 5, there is a unique solution $\vec{U_t^*}$ to the linear system $M \times \vec{U_t} = \vec{D_t}$. Then, the following holds:

[1] A matrix T is an invertible M-matrix if $T_{ij} \leq 0$ when $i \neq j$ and $Tx \geq 0$ implies that $x \geq 0$.

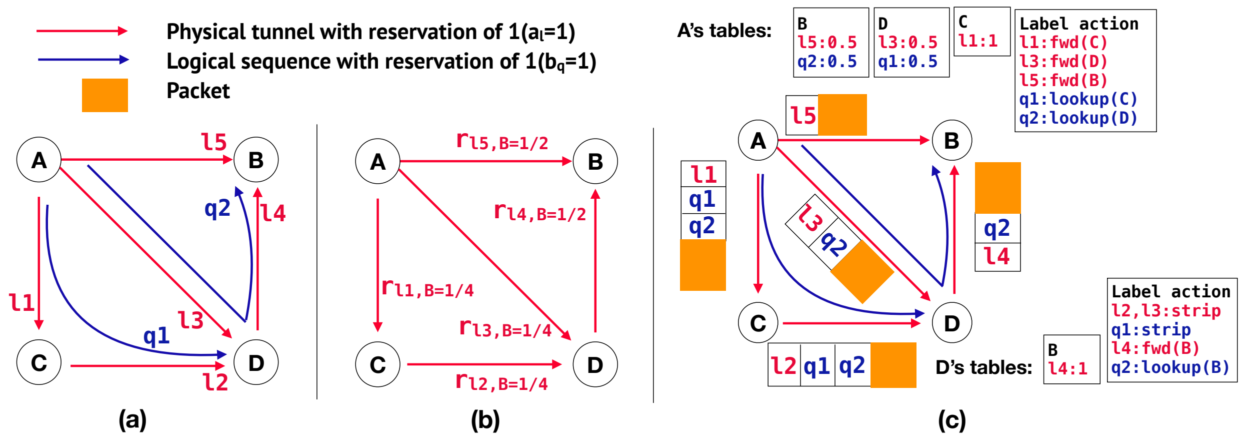

Figure 6: Realizing PCF in practice. (a) Example abstract model; (b) a practical realization using only tunnels applicable for arbitrary LSs (§4.1); (c) an alternate realization when LSs can be topologically sorted (§4.2).

$$M_1 = \begin{array}{c} \\ \end{array}\begin{array}{c} \begin{array}{ccccc} AC & CD & AD & DB & AB \end{array} \\ \begin{pmatrix} a_{l_1} & 0 & -b_{q_1} & 0 & 0 \\ 0 & a_{l_2} & -b_{q_1} & 0 & 0 \\ 0 & 0 & a_{l_3}+b_{q_1} & 0 & -b_{q_2} \\ 0 & 0 & 0 & a_{l_4} & -b_{q_2} \\ 0 & 0 & 0 & 0 & a_{l_5}+b_{q_2} \end{pmatrix} \end{array}\begin{array}{c} \\ AC \\ CD \\ AD \\ DB \\ AB \end{array}$$

Figure 7: Reservation matrix associated with Fig. 6.

Proposition 6. *For any live tunnel l from i to j and any destination t, let $r_{lt} = \vec{U_t^*}(i,j)a_l$ be the total traffic carried to destination t on tunnel l. Then r_{lt} represents a valid routing which carries all the traffic with the destination of t.*

We compute U_t^* for every node t by solving the linear system $M \times [\vec{U_{t_1}^*}, \vec{U_{t_2}^*}, ...\vec{U_{t_{|V|}}^*}] = [\vec{D_{t_1}}, \vec{D_{t_2}}, ...\vec{D_{t_{|V|}}}]$, which in turn allows r_{lt} to be computed. As computed, r_{lt} may have cycles that can be eliminated by subtracting flow associated with the cycle. Fig. 6(b) shows a concrete realization of PCF's routing on tunnels for the abstract model shown in Fig. 6(a). Each tunnel is annotated with the fraction of the traffic to destination B carried on that tunnel – e.g., $r_{l5,B} = 1/4$ indicates that $l5$ carries 1/4 of the traffic to B.

4.2 Topologically sorted logical sequences

While the approach in §4.1 works for *arbitrary* LSs, we next describe an alternate approach that works when LSs are chosen with some restrictions. Given two node pairs (i,j), and (i',j'), we say that $(i,j) > (i',j')$ if (i',j') is a segment of any active LS q in $L(i,j)$. Our approach below is applicable if all the node pairs under every failure scenario can be topologically sorted with respect to relation '>'. For example, in Fig. 6, the LSs satisfy a topological ordering with $(A,B) > (A,D)$ since $q2 \in L(A,B)$ uses the segment (A,D) (but not vice versa). Note that, essentially, we only require a strict partial order over the node pairs. The topological sort refers to any total order that extends this strict partial order and, it is well-known that such a total order exists and can be derived easily from the partial order[21].

When a topological ordering is possible, PCF implements LSs more directly (Fig. 6(c)). When A sends packets to B, traffic is split across the tunnel $l5$ and LS $q2$. Traffic to $q2$ involves pushing a label, and looking up the table entry for host D. This entry indicates traffic is split across tunnel $l3$ and LS $q1$. Traffic to $q1$ involves pushing another label and looking up the entry for host C, which indicates the traffic is to be forwarded on tunnel $l1$. When a router receives a packet, it pops labels as needed, and if it is an intermediate point of a LS takes the appropriate action. For example, when D receives a packet on tunnel $l3$ it pops the outer label $l3$, and based on the inner label $q2$, looks up the entry for B, and forwards to B along tunnel $l4$.

A key question is to decide how to split the traffic at each hop – e.g., for traffic from A to B, what fraction is sent on each of tunnels $l5$, and LS $q2$. We define *local proportional routing* as a scheme where the traffic associated with each node pair (i,j) is split across all tunnels and LSs from i to j in proportion to the reservations associated with these tunnels and LSs. This is a generalization of FFC which uses a locally proportional scheme but in a context where there are only tunnels. Then, the following holds:

Proposition 7. *The LS models can be realized by local proportional routing when the topological sort property is met.*

Proof. For a particular failure scenario x, let $T_x(s,t)$ denote the set of live tunnels from s to t, $L_x(s,t)$ denote the set of active LSs from s to t and $Q_x(s,t)$ denote the active LSs which go through segment (s,t). We show by induction along the topological sort order that locally proportional routing services the demand. Our induction hypothesis is that the pair (i,j) needs to route $\tilde{D_{ij}}$ where,

$$\tilde{D_{ij}} = \vec{D}(i,j) + \sum_{(m,n)\in P, q\in Q_x(i,j)\cap L_x(m,n)} u_{mn} b_q, \tag{9}$$

if every router distribute $\tilde{D_{ij}}$ among the tunnels ($l \in T_x(i,j)$) and LSs ($q \in L_x(i,j)$) in the proportion of their reservations, i.e., there is a constant u_{ij} such that traffic along l is $u_{ij}a_l$ and that along LS q is $u_{ij}b_q$ where

$$u_{ij} = \frac{\tilde{D_{ij}}}{\sum_{l\in T_x(i,j)} a_l + \sum_{q\in L_x(i,j)} b_q}.$$

For the base case, observe that for the topologically largest pair $p_1 = (i_1, j_1)$, the demand received is $\vec{D}(i_1, j_1)$. And the hypothesis is trivially true because $Q(i_1, j_1) = \emptyset$. For the induction step, we assume that the hypothesis is true for pairs $p_1, p_2, ..., p_n$ in topological sort order and show it holds for $p_{n+1} = (i, j)$. Observe that for any $q \in Q_x(i, j) \cap L_x(m, n)$, the traffic sent to b_q is, by the induction hypothesis, $u_{mn} b_q$ because $Q_x(i, j) \cap L_x(m, n) = \emptyset$ implies $(m, n) > (i, j)$. Then, (9) holds for (i, j), and it follows easily that if (i, j) routes $u_{ij} b_q$ along each $q \in L_x(i, j)$ and $u_{ij} a_l$ along each $l \in T_x(i, j)$ that $\sum_{l \in T_x(i,j)} u_{ij} a_l + \sum_{q \in L_x(i,j)} u_{ij} b_q = \tilde{D_{ij}}$. Since it follows easily from above that

$$u_{ij}\Big(\sum_{l \in T_x(i,j)} a_l + \sum_{q \in L_x(i,j)} b_q\Big) =$$

$$\vec{D}(i, j) + \sum_{(m,n) \in P, q \in Q_x(i,j) \cap L_x(m,n)} u_{mn} b_q,$$

it follows that $(u_{ij})_{(i,j) \in P}$ solves $M \times \vec{U} = \vec{D}$. Therefore, by proposition 5, $0 \le u_{ij} \le 1$. This implies that the routing is feasible since none of the reservations are exceeded. □

4.3 Implementation and deployment pathways

Now, we discuss how our scheme can be implemented and practically deployed. We start with the case when the logical sequences are topologically sorted. The offline computation phase determines the reservation for each tunnel and LS, similar to how FFC determines tunnel reservations. The regular forwarding operation and the failure recovery is completely distributed. Traffic associated with each node pair (i, j) is split across all physical tunnels and LSs in proportion to the reservations associated with them. When a tunnel fails or an LS is inactive, the weights are rescaled in proportion to the reservations on live tunnels and active LSs. This is similar to the existing approach of rescaling on live tunnels. Recall that LSs may have conditions attached to them and may only be active when the condition is true. Thus, for any conditional LS q from i to j, we need a mechanism to propagate the condition (e.g., link failure event) to i. For concreteness, we focus our discussion on two cases (the only cases considered in our evaluations). The first case involves LSs that do not have any conditions attached. This case is trivial to implement - such LSs are always active, and no hint propagation is needed. The second case involves LSs q between i and j which are only active when the link $i - j$ fails. This can be implemented by having i locally detect the failure of the $i - j$ link, which then results in i activating q and following the standard proportional scheme.

More generally, when logical sequences cannot be sorted in topological order, one simple implementation approach is to use a centralized controller. On each failure, the centralized controller solves a linear system which determines the new routing as discussed in §4.1. Solving a linear system is much easier than solving a linear program, as discussed earlier. While we do not explore further, we believe that it is possible to perform the operations on failure in a completely distributed fashion because the linear system we solve is of a special type (see Proposition 5) for which iterative algorithms exist. We defer further investigation to future work.

5 EVALUATIONS

We compare the performance guarantees provided by PCF's congestion-free mechanisms with FFC, the state-of-the-art congestion-free mechanism. When possible, we compare PCF with the performance achieved by the **optimal** network response which involves computing the optimal multi-commodity flow for each failure scenario. We implement all our optimization models in Python, and use Gurobi 8.0 [19] to solve them. We consider the following PCF schemes:

- *PCF-TF*. This uses FFC's mechanism to respond to failures, but models network structure more explicitly (§3.2).
- *PCF-LS*. Here, LSs are used but not associated with any condition (§3.3). For each node pair (s, t), we provide a single LS that includes the set of nodes in the shortest path from s to t. This guarantees that the topological sort assumption is met, which ensures the scheme can be implemented as a locally proportional routing scheme similar to FFC (§4.2).
- *PCF-CLS*. Here, the failure of each link $\langle i, j \rangle$ results in the activation of a LS from i to j. Further, each node pair is associated with a LS that is always active. We get these LSs by decomposing a restricted form of the logical flow model, where the only conditions are no link failures, or single link failures, with failure of link $\langle i, j \rangle$ resulting in the activation of a flow from i to j (§3.4). The LSs may not be topologically sorted. The scheme can be realized using relatively light-weight operations on each failure compared to the optimal network response (§4.1). In §5.2 we evaluate a heuristic that derives topologically sorted LSs from the above LSs, which allows for a proportional routing scheme similar to FFC.

Topologies. We evaluate our models on 21 topologies obtained from [22] and [23] (see Table 3 in the Appendix). Our two largest networks were Deltacom and Ion that contained 151 and 135 edges respectively and over a hundred nodes each. We remove one-degree nodes in the topologies recursively so that the networks are not disconnected with any single link failure. We use the gravity model [40] to generate traffic matrices with the utilization of the most congested link (MLU) in the range [0.6, 0.63] across the topologies.

5.1 Results

We start by reporting the demand scale (z) achieved by each scheme, which is the factor by which the traffic demand of all pairs can be scaled and yet supported by a given scheme. For example, $z = 0.5$ indicates that for all source destination pairs, half the demand can be served, while $z = 2$ indicates twice the demand can be handled. The MLU, or the utilization of the most congested link is the inverse of z. Later in this section, we report results with the throughput metric.

Benefits of modeling network structure. Fig. 8 shows the demand scale guaranteed by FFC when used to design for all single link failures for Deltacom (the topology with the most edges) for twelve different demands. Each curve corresponds to the number of tunnels used per node pair. We select physical tunnels so that they are as disjoint as possible, preferring shorter ones when there are multiple choices. With all our topologies, any node pair has at least two disjoint physical tunnels. When three or four tunnels are selected, it is not possible to guarantee that they are disjoint. Our strategy ensures that the failure of any link causes at most two tunnels to fail for all node pairs in the three tunnel case, and for most node pairs in the four tunnels case. The optimal is obtained by exhaustively enumerating all failure scenarios, and can take over 2 days in some settings. FFC performs significantly worse

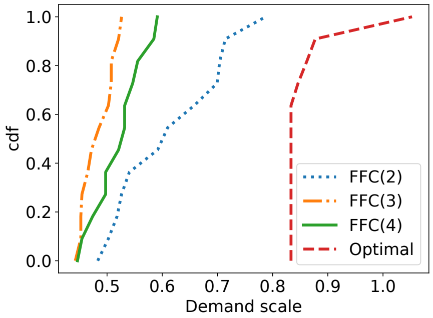

Figure 8: Impact of adding tunnels to FFC's performance

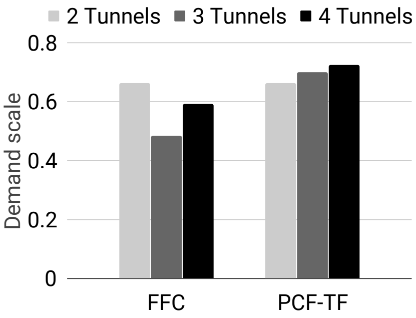

Figure 9: Performance of PCF-TF and FFC when more tunnels are added

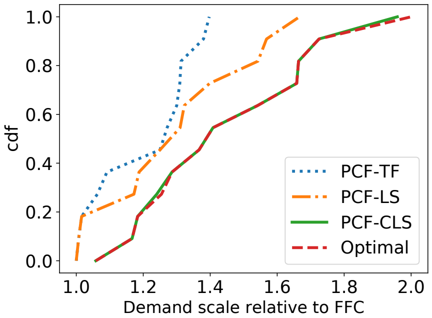

Figure 10: Benefits of PCF across multiple demands for Deltacom

than optimal, and consistently better with two tunnels (additional tunnels hurt).

Fig. 9 shows the demand scale guaranteed by PCF-TF when designing for single link failures for Deltacom, and an example traffic matrix. Results for FFC are included for comparison. PCF-TF matches FFC's performance when 2 tunnels are used, and performs better as tunnels are added given that it better models network structure. We observed similar trends with all topologies, and across demands. Henceforth, in our experiments, all our schemes use three tunnels (this is conservative as adding more tunnels improves performance), while FFC uses two tunnels (this represents the best setting for FFC and choosing more tunnels leads to poorer performance).

Benefits of more flexible response. We next evaluate the performance of our various PCF schemes relative to FFC, and report the ratio of the demand scale for a given scheme to the demand scale with FFC. We generate 12 different demands for Deltacom to model a traffic matrix every 2 hours. Fig. 10 shows a CDF of the ratios across these demands. In the median case, PCF-TF and PCF-LS achieve an improvement of $1.25X$ over FFC, while PCF-CLS achieves a $1.37X$ improvement. Further, for 25% of the traffic matrices, PCF-TF, PCF-LS and PCF-CLS achieve improvements of more than $1.3X$, $1.4X$ and $1.66X$ over FFC respectively. Finally, PCF-CLS matches the optimal for most cases. While PCF-TF's improvements arise due to better modeling of network structure, the further benefits achieved by PCF-LS and PCF-CLS are due to additional flexibility provided by logical sequences.

Analysis across topologies. Fig. 11 presents a CDF of the ratios of the demand scale for each scheme relative to FFC across topologies when designing for single link failures. All our schemes provide significant benefits, with PCF-CLS matching the optimal for most topologies. On average, PCF-TF, PCF-LS and PCF-CLS achieve improvements of more than $1.11X$, $1.22X$ and $1.44X$ over FFC respectively. For GEANT (rightmost point), PCF-LS and PCF-CLS perform $2.6X$ better.

Multiple simultaneous failures. We next consider simultaneous link failures. To avoid disconnecting the topologies, we split the capacity of each link evenly across two sub-links that fail independently. We report the performance of all schemes when designing for all possible scenarios involving the simultaneous failure of three sub-links. For all PCF schemes we pick 6 tunnels, choosing them to be as disjoint as possible. For similar reasons as above, we found FFC achieved significantly better performance with 4 tunnels (FFC resulted in a demand scale factor of 0 with 6 tunnels).[2] Fig. 12 shows a CDF of the demand scale ratios for each scheme relative to FFC. On average, PCF-TF, PCF-LS and PCF-CLS achieve improvements of more than $1.11X$, $1.25X$ and $1.50X$ over FFC respectively. Note that while the trends are similar to single failures, the absolute values of demand scales are lower for all schemes – e.g., the optimal under 3 failures is 0.42 for Deltacom, while 0.85 under single failures).

Throughput metric. Instead of demand scale, we next consider performance when the schemes optimize the throughput metric (sum of bandwidth allocated to each pair). Given a demand d_{st} for source s and destination t, and an allocated bandwidth bw_{st} ($bw_{st} \leq d_{st}$), we compute the throughput overhead $1 - \frac{\sum bw_{st}}{\sum d_{st}}$. Fig.13 shows the % reduction in throughput overhead of each scheme relative to FFC when designing for three failures. PCF provides significant benefits. In the median case, PCF-TF and PCF-LS reduce the throughput overhead of FFC by more than 16%, and the reduction with PCF-CLS is 46%. For 25% of the topologies, PCF-TF, PCF-LS and PCF-CLS reduce the throughput overhead by 27%, 41% and 55% respectively. We do not report the optimal for this metric since it requires a prohibitively large optimization formulation that simultaneously models combinatorially many routing problems, one for each failure state.

5.2 Feasibility of local yet optimal routing

As discussed earlier, PCF-TF uses a routing mechanism identical to FFC, while PCF-LS uses topologically sorted logical sequences and can be realized using a locally proportional routing (§4.2) similar to FFC. However, the LSs chosen in PCF-CLS are not guaranteed to be topologically sorted.

Interestingly, under single link failures, the LSs generated by PCF-CLS are already topologically sorted by default for 16 of our 21 topologies. For the remaining ones, we consider a new scheme, which we refer to as PCF-CLS-TopSort, that starts with the LSs initially generated by PCF-CLS, and picks a subset which are topologically sorted. To achieve this, we use a greedy algorithm that adds LSs one by one from the original set, omitting any LS that violates the topological sort property. In all cases, less than 0.59% of the LSs were pruned. Further, for 4 of the 5 topologies, PCF-CLS-TopSort performs identically with PCF-CLS for the demand scale metric. For Ion alone there was some performance degradation,

[2] It was only feasible to select 6 tunnels, with 2 sharing a common link. Under three failures, FFC must provision for the case all tunnels failed.

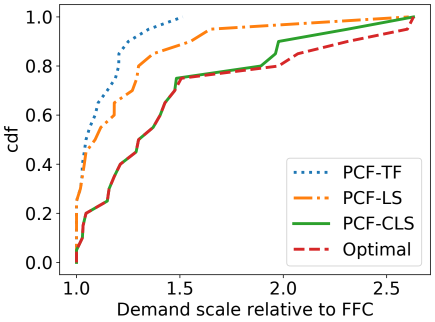

Figure 11: PCF vs FFC across topologies.

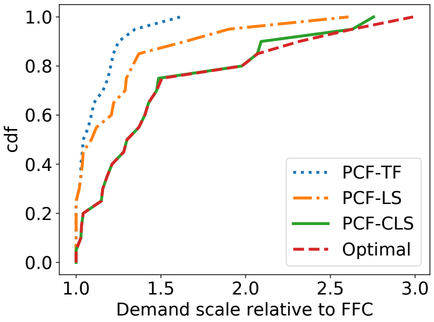

Figure 12: Performance under three simultaneous failures.

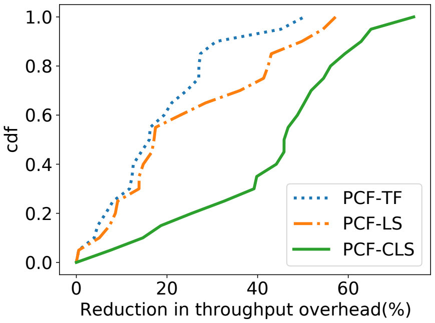

Figure 13: Reduction in throughput overhead compared to FFC

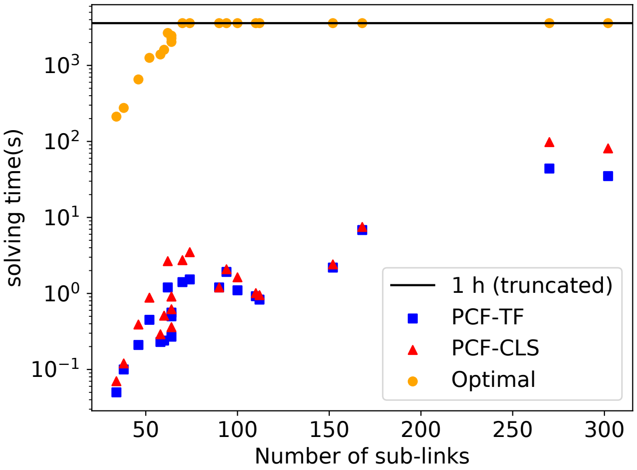

Figure 14: Solving time for PCF's schemes and optimal.

from a demand scale of 1.11 with PCF-CLS to 0.82 with PCF-CLS-TopSort, but still much better than FFC which achieved a demand scale of 0.48. Overall, the results indicate that for single failure scenarios, a local proportional routing mechanism is sufficient to ensure near optimal performance.

For multiple simultaneous failures, PCF-CLS-TopSort does not match the performance of PCF-CLS. We note however that (i) PCF-CLS-TopSort and PCF-LS still significantly out-perform FFC and are realizable as local algorithms; and (ii) while PCF-CLS nearly matches optimal, it only requires a linear system of equations to be solved on each failure as opposed to a more expensive optimization problem.

5.3 Tractability of formulations

Fig. 14 presents the solving time (Y-Axis) against topology size (X-Axis) when PCF-TF, and PCF-CLS (the most complex scheme) are used to design for all simultaneous three link failure scenarios. Each point corresponds to a topology. PCF-LS takes less time than PCF-CLS and is not shown. For most topologies, the solving times is under 10 seconds. For the two largest networks (Deltacom and Ion) with 302 and 270 sub-links (each original link comprises 2 sub-links), the solving time for PCF-TF is under 50 seconds and for PCF-CLS under 100 seconds. This is reasonable because PCF's models only need to be run at the granularity of several minutes (on failure, lighter-weight online operations are used (§4.3)).

The figure also shows the solving time for the optimal scheme (truncating the Y-Axis at 1 hour). The solving time is much larger even for the smaller topologies and did not complete within an hour for most topologies. For one of the larger topologies, it did not complete even after two days.

6 RELATED WORK

Reactive vs. congestion-free routing schemes. Many recent traffic engineering (TE) schemes [17, 20] have developed flexible ways of routing traffic motivated by the goal of efficiently utilizing network capacity. Typically, these schemes involve deciding how to optimally route traffic at a centralized controller leveraging network-wide views [17, 20, 23]. Failures are handled *reactively* by recomputing routes at the centralized controller, and updating rules at switches, a process that can take a long time, and that could congestion links in the interim [26]. A more recent work [23] derives tunnels from an oblivious routing strategy, and determines how to split traffic across tunnels so link utilizations are minimized. The scheme does not guarantee that the network would remain congestion-free on failure.

A second class of schemes [26, 32, 37] *proactively* guarantee the network remains congestion-free over a large set of failure scenarios (e.g., all scenarios with f simultaneous link failures), while only allowing for the network to respond to failures using fast and light-weight response mechanisms. For instance, FFC [26] conservatively admits traffic so the network does not experience congestion when local proportional routing is used. With such schemes, an optimization problem is only solved *offline* (i.e., prior to any failure scenario). The optimization models guarantee the congestion-free property, and are tractable in that they do not require an explicit enumeration of the large space of failure scenarios.

PCF addresses both objectives at the same time, by developing provably congestion-free light-weight mechanisms that achieve close to the optimal performance sought by reactive TE mechanisms. PCF not only out-performs existing congestion-free mechanisms, but performs close to optimal (the best possible performance that can be achieved by a reactive centralized TE scheme). Further, like other congestion-free schemes, PCF does not solve an LP on failure but only involves light-weight operations. Finally, with topologically sorted LSs, PCF uses local proportional routing, similar to FFC. Finally, while we do not explore in this paper, the tractable failure models associated with congestion-free schemes in general and PCF in particular can aid in network design tasks such as provisioning networks with sufficient capacity to protect against failures.

Other congestion-free routing schemes. Among congestion-free schemes, we have extensively discussed FFC [26]. R3, another congestion-free mechanisms based on link bypass [37], is based on flows, and cannot handle node failures (§3.5). PCF uses tunnels which are easier to deploy [23], and can tackle node failures. When flows are allowed, PCF provably out-performs R3 even for link failures (§3.5). Another work [32] addresses link failures by adding edges to the network. The original excess capacity of the network is not used, and the number of edges added may be substantial.

Rather than tackle all f failures, recent works Teavar [6], and Lancet [10] design for scenarios that occur with sufficient probability so a desired availability target is met. The techniques in Teavar [6] and Lancet [10] are respectively demonstrated with FFC and a generalization of R3. Techniques for probabilistic design are orthogonal to the design of congestion-free routing schemes. In particular, the ideas in both Teavar and Lancet are complementary to PCF, and may be potentially combined with PCF in the future to achieve better performance bounds when designing for scenarios with given probability.

A framework for analyzing the worst-case performance of centralized TE approaches was presented in [9]. The framework provides conservative performance bounds when network response can be modeled as an optimization problem. The conservative bounds may be viewed loosely equivalent to the performance of a more restricted network routing scheme that does not re-optimize on each failure. However, the bounds are obtained using optimization-theoretic relaxation methods, and it is an open question whether these abstract relaxations relate to practically realizable network response mechanisms. In contrast, all of PCF's models are associated with realizable network response mechanisms as we have discussed. Interestingly, while we do not explore in this paper, PCF's models may provide alternate and better ways to bounds the performance of centralized TE schemes – e.g., the performance of PCF-CLS under failures matched the optimal for most topologies (§5). These benefits arise because using LSs can improve the bounds for the relaxations proposed in [9]. Finally, PCF's formulations can be naturally used to augment capacities so as to meet a desired performance metric by simply making capacities variable.

Segment and pathlet routing. Logical sequences are similar to segment routing [16, 34] in that traffic is steered through a given series of hops. DEFO considers ISP carrier network settings where the traffic in each segment is carried using a (possibly legacy) mechanism such as shortest-path forwarding, and the segments may be chosen so as to optimize a traffic engineering goal [16]. In contrast, LSs are an abstraction to increase the flexibility of provably congestion-free resilient routing mechanisms. Each LS is associated with a reservation, and may only be active when some conditions are met. Our actual implementation (§4.1) may be entirely tunneling based, or use both LSs and tunnels with a local proportional routing scheme (§4.2).

In pathlet routing [13], sources concatenate fragments of paths (pathlets) into end-to-end routes in a bottom-up fashion. In contrast, with PCF, logical sequences and physical tunnels are predefined in a top-down manner. Moreover, pathlet routing is motivated by the challenges of multipath routing, while it does not provide any performance guarantee upon failures.

Other related work. While several works explore quick re-routing of traffic to restore connectivity on failures [24, 25, 28, 31, 38], PCF guarantees the network is congestion-free (not merely restore connectivity). Oblivious routing provides bounds on network performance over multiple demands, and when networks do not adapt [2, 3, 36, 39]. PCF carefully adds flexibility to network response to allow for tractable analysis of performance under failures. Robust network design under single link or node failures has received attention [3, 5, 11, 15, 29, 33, 41]. PCF scales to the large number of failure states arising from concurrent failures, and shows how networks with carefully chosen response can achieve near optimal performance.

7 CONCLUSIONS

In this paper, we have made two contributions. First, we have shown that existing mechanisms which ensure the network is congestion-free on failures achieve performance far short of the network's intrinsic capability, and shed light on the underlying reasons. Second, we have proposed PCF, a set of novel congestion-free mechanisms that bridges this gap by better modeling network structure, and by carefully enhancing the flexibility of network response to ensure that the performance under failures can be tractably modeled. Through formal theoretical results, we show PCF's schemes provably out-perform FFC. Empirical experiments over 21 Internet topologies show that PCF's schemes can sustain higher throughput than FFC by a factor of 1.11X to 1.5X on average across the topologies, providing a benefit as high as 2.6X in some cases. PCF's schemes are practically realizable, and some of them can yet achieve near optimal performance. **This work does not raise any ethical issues.**

8 ACKNOWLEDGEMENTS

We thank our shepherd, Barath Raghavan, and the anonymous reviewers for their feedback which greatly helped improve the paper. This work was funded in part by the National Science Foundation (NSF) Award 1910234.

REFERENCES

[1] Ravindra K. Ahuja, Thomas L. Magnanti, and James B. Orlin. *Network Flows.* Prentice Hall, Englewood Cliffs, New Jersey, 1993.

[2] David Applegate and Edith Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In *Proceedings of ACM SIGCOMM*, pages 313–324, 2003.

[3] David Applegate, Lee Breslau, and Edith Cohen. Coping with network failures: Routing strategies for optimal demand oblivious restoration. In *Proceedings of the Joint International Conference on Measurement and Modeling of Computer Systems*, SIGMETRICS '04/Performance '04, pages 270–281, 2004.

[4] Abraham Berman and Robert J. Plemmons. *Nonnegative Matrices in the Mathematical Sciences.* Classics in Applied Mathematics. Society for Industrial and Applied Mathematics, 1994.

[5] Randeep S. Bhatia, Murali Kodialam, T. V. Lakshman, and Sudipta Sengupta. Bandwidth guaranteed routing with fast restoration against link and node failures. *IEEE/ACM Transactions on Networking*, 16(6):1321–1330, December 2008.

[6] Jeremy Bogle, Nikhil Bhatia, Manya Ghobadi, Ishai Menache, Nikolaj Bjorner, Asaf Valadarsky, and Michael Schapira. Teavar: Striking the right utilization-availability balance in wan traffic engineering. In *Proceedings of ACM SIGCOMM*, 2019. (to appear).

[7] Jonathan M. Borwein and Adrian S. Lewis. *Convex analysis and nonlinear optimization.* Springer, New York, 2000.

[8] James H. Bramble and Bert E. Hubbard. On a finite difference analogue of an elliptic boundary problem which is neither diagonally dominant nor of non-negative type. *Journal of Mathematics and Physics*, 43(1-4):117–132, 1964.

[9] Yiyang Chang, Sanjay Rao, and Mohit Tawarmalani. Robust validation of network designs under uncertain demands and failures. In *14^{th} USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, pages 347–362, 2017.

[10] 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: 1–26, 12 2019. doi: 10.1145/3366697.

[11] Bernard Fortz and Mikkel Thorup. Robust optimization of OSPF/IS-IS weights. In *Proceedings of International Network Optimization Conference*, pages 225–230, 2003.

[12] Phillipa Gill, Navendu Jain, and Nachiappan Nagappan. Understanding network failures in data centers: Measurement, analysis, and implications. In *Proceedings of ACM SIGCOMM*, pages 350–361, 2011.

[13] P. Brighten Godfrey, Igor Ganichev, Scott Shenker, and Ion Stoica. Pathlet routing. In *Proceedings of the ACM SIGCOMM 2009 Conference on Data Communication*, page 111–122, 2009.

[14] Ramesh Govindan, Ina Minei, Mahesh Kallahalla, Bikash Koley, and Amin Vahdat. Evolve or die: High-availability design principles drawn from googles network infrastructure. In *Proceedings of ACM SIGCOMM*, pages 58–72, 2016.

[15] Fang Hao, Murali Kodialam, and T. V. Lakshman. Optimizing restoration with segment routing. In *Proceedings of IEEE INFOCOM*, pages 1–9, April 2016.

[16] Renaud Hartert, Stefano Vissicchio, Pierre Schaus, Olivier Bonaventure, Clarence Filsfils, Thomas Telkamp, and Pierre Francois. A declarative and expressive approach to control forwarding paths in carrier-grade networks. In *Proceedings of ACM SIGCOMM*, pages 15–28, 2015.

[17] 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 ACM SIGCOMM*, pages 15–26, 2013.

[18] 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 2018 Conference of the ACM Special Interest Group on Data Communication*, pages 74–87, 2018.

[19] Gurobi Optimization Inc. Gurobi optimizer reference manual, 2016. http://www.gurobi.com.

[20] 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. In *Proceedings of ACM SIGCOMM*, pages 3–14, 2013.

[21] Thomas J. Jech. *The axiom of choice.* Courier Corporation, 2008.

[22] Simon Knight, Hung Nguyen, Nickolas Falkner, Rhys Bowden, and Matthew Roughan. The internet topology zoo. *IEEE Journal on Selected Areas in Communications*, 29:1765 – 1775, October 2011.

[23] 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 18)*, pages 157–170, 2018.

[24] Kin-Wah Kwong, Lixin Gao, Roch Guérin, and Zhi-Li Zhang. On the feasibility and efficacy of protection routing in ip networks. *IEEE/ACM Transactions on Networking*, 19(5):1543–1556, October 2011.

[25] Karthik Lakshminarayanan, Matthew Caesar, Murali Rangan, Tom Anderson, Scott Shenker, and Ion Stoica. Achieving convergence-free routing using failure-carrying packets. In *Proceedings of ACM SIGCOMM*, pages 241–252, 2007.

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

[27] Athina Markopoulou, Gianluca Iannaccone, Supratik Bhattacharyya, Chen-Nee Chuah, Yashar Ganjali, and Christophe Diot. Characterization of failures in an operational ip backbone network. *IEEE/ACM Trans. Netw.*, 16(4):749–762, 2008.

[28] P. Pan, G. Swallow, and A. Atlas. Fast Reroute Extensions to RSVP-TE for LSP Tunnels. RFC 4090, May 2005.

[29] Michal Pióro and Deepankar Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks.* Morgan Kaufmann Publishers Inc., San Francisco, CA, USA, 2004. ISBN 0125571895.

[30] Rahul Potharaju and Navendu Jain. When the network crumbles: An empirical study of cloud network failures and their impact on services. In *Proceedings of the 4th Annual Symposium on Cloud Computing*, SOCC '13, pages 15:1–15:17, 2013.

[31] M. Shand and S. Bryant. IP Fast Reroute Framework. RFC 5714, January 2010.

[32] Rakesh Sinha, Funda Ergun, Kostas N. Oikonomou, and K. K. Ramakrishnan. Network design for tolerating multiple link failures using Fast Re-route (FRR). In *2014 10th International Conference on the Design of Reliable Communication Networks (DRCN)*, pages 1–8, April 2014.

[33] Martin Suchara, Dahai Xu, Robert Doverspike, David Johnson, and Jennifer Rexford. Network architecture for joint failure recovery and traffic engineering. *SIGMETRICS Perform. Eval. Rev.*, 39(1):97–108, 2011.

[34] Jeff Tantsura, Igor Milojevic, Edward Crabbe, Stefano Previdi, Martin Horneffer, Ahmed Bashandy, Rob Shakir, Clarence Filsfils, Bruno Decraene, Saku Ytti, et al. Segment routing architecture. *IETF Internet draft*, 2013.

[35] Daniel Turner, Kirill Levchenko, Alex C. Snoeren, and Stefan Savage. California fault lines: Understanding the causes and impact of network failures. In *Proceedings of the ACM SIGCOMM 2010 Conference*, pages 315–326, 2010.

[36] Hao Wang, Haiyong Xie, Lili Qiu, Yang Richard Yang, Yin Zhang, and Albert Greenberg. COPE: Traffic engineering in dynamic networks. In *Proceedings of ACM SIGCOMM*, pages 99–110, 2006.

[37] Ye Wang, Hao Wang, Ajay Mahimkar, Richard Alimi, Yin Zhang, Lili Qiu, and Yang Richard Yang. R3: Resilient routing reconfiguration. In *Proceedings of ACM SIGCOMM*, pages 291–302, 2010.

[38] Baohua Yang, Junda Liu, Scott Shenker, Jun Li, and Kai Zheng. Keep forwarding: Towards k-link failure resilient routing. In *Proceedings of IEEE INFOCOM*, pages 1617–1625, April 2014.

[39] Chun Zhang, Zihui Ge, Jim Kurose, Yong Liu, and Don Towsley. Optimal routing with multiple traffic matrices tradeoff between average and worst case performance. In *Network Protocols, 2005. ICNP 2005. 13th IEEE International Conference on*, 2005.

[40] Yin Zhang, Zihui Ge, Albert Greenberg, and Matthew Roughan. Network anomography. In *Proceedings of the 5th ACM SIGCOMM Conference on Internet Measurement*, pages 30–30, 2005.

[41] Jiaqi Zheng, Hong Xu, Xiaojun Zhu, Guihai Chen, and Yanhui Geng. We've got you covered: Failure recovery with backup tunnels in traffic engineering. In *2016 IEEE 24th International Conference on Network Protocols (ICNP)*, pages 1–10, 2016.

A APPENDIX

Appendices are supporting material that has not been peer-reviewed.

Notation in PCF and FFC. In this paper, we use slightly different notation than FFC. For example, FFC uses f to denote a flow aggregated by a pair of ingress-egress switches, while we use (s, t) to directly denote a flow by its ingress and egress switches. In Table 2, we list key notation in PCF and the corresponding notation in FFC.

Efficiently solving PCF's models leveraging LP duality. As presented, converting PCF's models into LPs would create one constraint per failure scenario which is not scalable since the failure scenarios in Y may be large. Instead, we leverage a dualization technique, which has been extensively used in prior work [26, 37]. We illustrate with PCF-TF, but note that the same approach may be used for all of PCF's schemes. We rewrite (1) as

$$\min_{y \in Y} \sum_{l \in T(s,t)} a_l(1 - y_l) \geq z_{st} d_{st} \quad \forall s, t \in V. \tag{10}$$

PCF	FFC	meaning
(s,t)	f	A flow aggregated by ingress-egress switches
d_{st}	d_f	The bandwidth demand of a flow
$T(s,t)$	T_f	The set of tunnels that are set up for a flow
f	k_e	The number of link failures that PCF (FFC) protects the network against
$e \in \tau_l$	$L[t,e]$	True (1) if tunnel l (t) uses link e and False (0) otherwise
$z_{st}d_{st}$	b_f	The bandwidth granted to a flow
a_l	$a_{f,t}$	The bandwidth allocated to a tunnel

Table 2: Different notation used in PCF and FFC

These constraints are reformulated by relaxing the integrality of y variables, and expressing the LHS as a maximization problem leveraging LP duality shown below:

$$
\begin{aligned}
(D1)\ \max_{\pi,\lambda,\sigma,\phi}\ & -(f\lambda_{st} + \sum_{e\in E}\sigma_{est} + \sum_{l\in T(s,t)}\phi_l) \\
\text{s.t.}\quad & \pi_l + \phi_l \ge a_l \quad \forall l \in T(s,t) \\
& -\sum_{l:e\in\tau_l}\pi_l + \lambda_{st} + \sigma_{est} \ge 0 \quad \forall e \\
& \pi_l \ge 0 \quad \forall l \in T(s,t) \\
& \lambda_{st} \ge 0 \\
& \sigma_{est} \ge 0 \quad \forall e \in E \\
& \phi_l \ge 0 \quad \forall l \in T(x,y).
\end{aligned}
$$

Now, we put $(D1)$ into (10) and combine it with the rest of constraints in $(P1)$ to obtain the final model below.

$$
\begin{aligned}
(D2)\quad \max_{\pi,\lambda,\sigma,\phi,z,a}\quad & \Theta(z) \\
\text{s.t.}\quad & \sum_{l\in T(s,t)} a_l - (f\lambda_{st} + \sum_{e\in E}\sigma_{est} + \sum_{l\in T(s,t)}\phi_l) \ge z_{st}d_{st} \\
& \qquad\qquad\qquad\qquad\qquad\qquad \forall s,t \in V \\
& a_l \ge 0 \quad \forall s,t \in V, l \in T(s,t) \\
& \sum_{l:\forall s,t\in V, l\in T(s,t)} a_l\delta(e \in \tau_l) \le c_e \quad \forall e \in E \\
& \pi_l + \phi_l \ge a_l \quad \forall s,t \in V, \forall l \in T(s,t) \\
& -\sum_{l:e\in\tau_l}\pi_l + \lambda_{st} + \sigma_{est} \ge 0 \quad \forall e, \forall s,t \in V \\
& \pi_l \ge 0 \quad \forall s,t \in V, \forall l \in T(s,t) \\
& \lambda_{st} \ge 0 \quad \forall s,t \in V \\
& \sigma_{est} \ge 0 \quad \forall s,t \in V, \forall e \in E \\
& \phi_l \ge 0 \quad \forall s,t \in V, \forall l \in T(s,t).
\end{aligned}
$$

More general conditions (§3.4). Let h_q be a condition that requires all links in η_q to be alive and all links in ξ_q to be dead. Then we model h_q by linearizing the constraint:

$$h_q = \prod_{e\in\xi_q} x_e \prod_{\eta_q}(1 - x_e)$$

as follows:

$$
\begin{aligned}
& (h_q - 1) + x_e \le 0 \quad \forall e \in \eta_q \\
& h_q - x_e \le 0 \quad \forall e \in \xi_q \\
& (1 - h_q) - \sum_{e\in\eta_q} x_e - \sum_{e\in\xi_q}(1 - x_e) \le 0 \\
& 0 \le h_q \le 1.
\end{aligned}
$$

We model YH by adding the above constraints for each LS q to (4).

Proof of Proposition 5. We will show $M \in \mathbb{R}^{P\times P}$ is a weakly-chained diagonally dominant matrix, where $M_{ij,i_1j_1} \le 0$ for $(i,j) \ne (i_1,j_1)$. Then, it follows from Theorems 2.1 and 2.2 in [8] that M is an invertible M-Matrix.

For a particular failure scenario x, let $T_x(s,t)$ denote the set of alive tunnels from s to t, $L_x(s,t)$ denote the set of active LSs from s to t and $Q_x(s,t)$ denote the active LSs which go through segment (s,t). We first give the formal definition of P, the set of node pairs of interest. A node pair $(i_1,j_1) \in P$ if and only if there is a sequence of node pairs (i_1,j_1), (i_2,j_2),…,(i_k,j_k), such that $z_{i_kj_k}d_{i_kj_k} > 0$ and $\forall m : 1 \le m \le k-1, \exists q \in L_x(i_{m+1},j_{m+1}) \cap Q_x(i_m,j_m)$ such that $b_q > 0$. There is a chain of LSs, such that a preceding LS serves a segment in the subsequent LS, where the last LS serves a pair with non-zero allocation and the first LS contains (i,j). For the node pairs which are not included in P, we set $U(i,j) = 0$.

Next, we formally define each entry in M. The diagonal of M is the sum of available reservations on the pair, i.e. $\forall(i,j) \in P, M_{ij,ij} = \sum_{l\in T_x(i,j)} a_l + \sum_{q\in L_x(i,j)} b_q$. Other entries of M denote how much a node pair needs to carry for other node pairs, i.e. for $(i,j) \ne (m,n)$ we set $M_{ij,mn} = -\sum_{q\in Q_x(i,j)\cap L_x(m,n)} b_q$.

It is easy to see that M is weakly diagonally dominated because $M \times \vec{1} \ge \vec{D} \ge 0$, where the first inequality is the capacity constraint and second because $z_pd_p \ge 0$ for all p.

From our definition of P, we know that $\forall(i_1,j_1) \in P$, there is a sequence (i_1,j_1), (i_2,j_2),…,(i_k,j_k), such that $z_{i_kj_k}d_{i_kj_k} > 0$ and $\forall m : 1 \le m \le k-1, \exists q \in L_x(i_{m+1},j_{m+1}) \cap Q_x(i_m,j_m) : b_q > 0$. Thus, for each row $(i,j) \in P$, there is a sequence (i_1,j_1), (i_2,j_2),…,(i_k,j_k), such that

$$
\begin{aligned}
& \sum_{(m,n)\in P} M_{i_kj_k,mn} \\
& = \sum_{l\in T_x(i_k,j_k)} a_l + \sum_{q\in L_x(i_k,j_k)} b_q - \sum_{q\in Q_x(i_k,j_k)} b_q \\
& \ge z_{i_kj_k}d_{i_kj_k} > 0,
\end{aligned}
$$

and $\forall m : 1 \le m \le k-1, M_{i_kj_k,i_{k+1}j_{k+1}} \ne 0$. Therefore, M is a weakly-chained diagonally dominant matrix. Since, for $(i,j) \ne (i_1,j_1)$, $M_{ij,i_1j_1} = -\sum_{q\in Q_x(i,j)\cap L_x(i_1,j_1)} b_q \le 0$, it follows that M is an invertible M-matrix and there is a unique solution $\vec{U}^*$ to the linear system $M \times \vec{U} = \vec{D}$.

Next, we use Brouwer fixed-point theorem [7] to prove that all entries of the solution are in $[0,1]$. Let $f(\vec{U})$ be a function mapping from $[0,1]^P$ to $\mathbb{R}^P$. We define $f(\vec{U})$ as

$$f(\vec{U})_{i,j} = \frac{\vec{D}(i,j) + \sum_{(m,n)\in P, q\in Q_x(i,j)\cap L_x(m,n)} \vec{U}(m,n)b_q}{\sum_{l\in T_x(i,j)} a_l + \sum_{q\in L_x(i,j)} b_q}. \quad (11)$$

Observe that the denominator is larger than zero. If not, it follows from weak diagonal dominance that $M_{ij,i'j'} = 0$ for all $(i',j') \ne$

(i, j), which contradicts $(i, j) \in P$. It is easy to see that $\vec{U_0}$ is a solution to $M \times \vec{U} = \vec{D}$ if $f(\vec{U_0}) = \vec{U_0}$. With $\vec{U} \in [0, 1]^P$, we have

$$f(\vec{U})_{i,j} \geq \frac{\vec{D}(i,j)}{\sum_{l\in T_x(i,j)} a_l + \sum_{q\in L_x(i,j)} b_q} \geq 0. \tag{12}$$

Moreover,

$$f(\vec{U})_{i,j} \leq \frac{\vec{D}(i,j) + \sum_{(m,n)\in P, q\in Q_x(i,j)\cap L_x(m,n)} b_q}{\sum_{l\in T_x(i,j)} a_l + \sum_{q\in L_x(i,j)} b_q} \leq 1, \tag{13}$$

where the first inequality is because $\vec{U}(m, n) \leq 1$, $b_q \geq 0$, and the denominator is positive. The second inequality is from the capacity constraint. Since f is a continuous function mapping from $[0, 1]^P$ to $[0, 1]^P$, and $[0, 1]^P$ is a compact convex set, it follows from Brouwer fixed-point theorem [7] that there is at least one point $U_0 \in [0, 1]^P$ so that $f(U_0) = U_0$, which we have already argued is the unique solution to $M \times \vec{U} = \vec{D}$. □

Proof of Proposition 6. We consider (s, t) column of M^{-1}, which exists by Proposition 5, and denote it as λ.

By definition, $M = A + B$ where A is a diagonal matrix with $A_{st,st} = \sum_{l\in T_x(s,t)} a_l$, $B_{st,st} = \sum_{q\in L_x(s,t)} b_q$, and for $(s, t) \neq (m, n)$, $B_{st,mn} = -\sum_{(m,n)\in P, q\in Q_x(s,t)\cap L_x(m,n)} b_q$. Then it follows that

$$\sum_{(m,n)\in P} \lambda_{mn} M_{mn,\cdot} = e_{st}, \tag{14}$$

where e_{st} is $(s, t)^{th}$ unit vector in $\mathbb{R}^P$ and $M_{mn,\cdot}$ denotes the column of M corresponding to the pair (m, n). It follows that

$$\sum_{(m,n)\in P} \lambda_{mn} A_{mn,mn} e_{mn} = e_{st} - \sum_{(m,n)\in P} \lambda_{mn} B_{mn,\cdot} \tag{15}$$

Now, e_{st} can be interpreted as a directed path carrying a unit flow from s to t. Moreover, we show that $B_{mn,\cdot}$ is a circulation since it can be written as an addition of cycles, one for each logical sequence servicing (m, n). To show this, we only need to show that for any $q \in L_x(i, j)$ with $b_q > 0$, if (k, l) is a logical segment in q, that is if $q \in Q_x(k, l)$, then $(k, l) \in P$. Since $(i, j) \in P$, there is a weak chain from (i, j) to a strictly dominated pair. The existence of q shows that (k, l) is connected to (i, j) since $M_{kl,ij} \leq b_q < 0$. Therefore, $(k, l) \in P$. Thus, the RHS of (15) represents a directed path flow e_{st} and some circulations $\sum_{(m,n)\in P} \lambda_{mn} B_{mn,\cdot}$. By the flow decomposition theorem (Theorem 3.5 in [1]), this flow yields a unique arc flow on the tunnel network shipping the same traffic as the directed path e_{st}. The resulting flow is then the LHS of (15). In other words, each a_l tunnel connecting (m, n) can use λ_{mn} fraction of its capacity to transmit a unit flow from s to t. Observe that the resulting flow may have loops that could be extracted in post-analysis.

Finally, since we have shown that $M^{-1}\vec{D} = \vec{U}^* \in [0, 1]^P$, it follows that

$$0 \leq A_{mn,mn} \sum_{(s,t)\in P} M^{-1}_{mn,st} z_{st} d_{st} \leq A_{mn,mn}, \tag{16}$$

which shows that the accumulated traffic on the tunnels between (m, n) will not exceed the reservation. Thus the traffic is feasible. Observe that all (s, t) pairs with $z_{st} d_{st} > 0$ are included in P. Therefore, $\vec{D}$ contains all the serviced demands and the proof is complete.

Topology	# nodes	# edges	Topology	# nodes	# edges
B4	12	19	Janet Backbone	29	45
IBM	17	23	Highwinds	16	29
ATT	25	56	BTNorthAmerica	36	76
Quest	19	30	CRLNetwork	32	37
Tinet	48	84	Darkstrand	28	31
Sprint	10	17	Integra	23	32
GEANT	32	50	Xspedius	33	47
Xeex	22	32	InternetMCI	18	32
CWIX	21	26	Deltacom	103	151
Digex	31	35	ION	114	135
IIJ	27	55			

Table 3: Topologies used in evaluation

Topologies summary (§5). Our evaluation is done using 21 topologies obtained from [22] and [23]. The number of nodes and the number of edges of each topology is shown in Table 3.