# Coping with Network Failures: Routing Strategies for Optimal Demand Oblivious Restoration

David Applegate
AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
david@research.att.com

Lee Breslau
AT&T Labs–Research
75 Willow Road
Menlo Park, CA 94025, USA
breslau@research.att.com

Edith Cohen
AT&T Labs–Research
180 Park Avenue
Florham Park, NJ 07932, USA
edith@research.att.com

## ABSTRACT

Link and node failures in IP networks pose a challenge for network control algorithms. Routing restoration, which computes new routes that avoid failed links, involves fundamental tradeoffs between efficient use of network resources, complexity of the restoration strategy and disruption to network traffic. In order to achieve a balance between these goals, obtaining routings that provide good performance guarantees under failures is desirable.

In this paper, building on previous work that provided performance guarantees under uncertain (and potentially unknown) traffic demands, we develop algorithms for computing optimal restoration paths and a methodology for evaluating the performance guarantees of routing under failures. We then study the performance of route restoration on a diverse collection of ISP networks. Our evaluation uses a competitive analysis type framework, where performance of routing with restoration paths under failures is compared to the best possible performance on the failed network. We conclude that with careful selection of restoration paths one can obtain restoration strategies that retain nearly optimal performance on the failed network while minimizing disruptions to traffic flows that did not traverse the failed parts of the network.

## Categories and Subject Descriptors

C.2 [**Communication Networks**]: C.2.2 Network Protocols;C.2.3 Network Operations

## General Terms

Algorithms,Design,Management,Performance,Reliability

## Keywords

routing; restoration; demand-oblivious routing

## 1. INTRODUCTION

The algorithms and protocols that are used to compute routes in a data network have a significant impact on the performance and operation of that network. End user performance can be improved by reducing load (and therefore congestion) on network links. Furthermore, the ability to route traffic in a way that reduces network utilization can reduce operational costs by avoiding the need to increase network capacity.

Optimal routing algorithms can produce a set of routes that, as an example, minimize the maximum link utilization in the network [16]. However, developing and deploying practical algorithms that achieve a desired performance objective remains a challenge for several reasons. First and foremost, optimal routing algorithms have required as input a traffic matrix, representing the traffic demand between each origin and destination (or ingress and egress point) in the network. In practice, knowing this traffic matrix has been elusive [6, 21]. In large operational networks, knowing the traffic matrix requires a large data collection effort that is often not undertaken by network operators.

The challenges and limitations of knowing the appropriate traffic matrix have been addressed by the development of measurement and modeling tools that provide the ability to extrapolate and estimate traffic demands [8, 9, 10, 15, 20]. Nonetheless, these tools cannot address the additional problem brought about by the dynamic nature of traffic demands. Traffic demands change on varying timescales, presenting a dual challenge for traffic estimation and routing. On the one hand, the traffic estimation process must track changing demand since routes computed for one demand matrix may be inappropriate for another. On the other hand, computing new routes to mirror these changes implies both a computational cost and a performance cost if routes are changed dynamically.

In recent work [4], we addressed the problem of uncertainty in the traffic matrix. Specifically, we leveraged recent algorithmic advances to derive methods for efficiently computing routes that do not depend on any assumptions about the traffic carried by the network. These *demand oblivious* routings were found to perform nearly as well as an optimal routing algorithm based on known traffic demands. Further, these algorithms have the added benefit that they are more robust than previous algorithms to uncertainty in the traffic matrix. That is, making no assumptions about traffic demand, these algorithms often perform better than demand-based algorithms when actual traffic deviates from the input traffic matrix.

Although this recent work addresses uncertainty in the traffic matrix, an additional challenge remains. Routing algorithms assume a static network topology. However, topology can change as links fail and are restored in a network. The challenge for routing is to determine a new set of routes when such topology changes occur. It is this question of path restoration in response to link failures that we address in this paper. In particular, we consider routing


*SIGMETRICS/Performance'04,* June 12–16, 2004, New York, NY, USA.

and restoration strategies that provide useful performance guarantees both on the original network and on the network after one or more link failures.

Any restoration strategy, which computes new routes in response to a topology change, must balance conflicting goals. On the one hand, one would like the resulting performance to be as good as possible. This implies computing an entirely new routing based on the new (failed) network. On the other hand, one would like to reduce the cost of the restoration strategy. One way this cost is manifested is in terms of protocol and computational overhead of computing new routes. While our analysis is not tied to any specific routing protocol, we assume any protocol will incur overhead in distributing or computing routing information when routes change. A second cost is the disruption to existing traffic. If routes are changed dynamically, traffic using those routes may experience temporary outages or noticeable delays. Therefore, more traffic flows whose routes are modified implies a larger disruption caused by the link failure and resulting restoration.

We use these tradeoffs to guide our investigation. At one extreme, after a link fails, we reapply our algorithm for demand oblivious routing to all traffic on the modified network topology. This extreme represents the best one can do in terms of performance, while incurring the maximum cost in terms of disruption to traffic that was not directly affected by the failure. We use this as a benchmark against which we compare other strategies.

A second strategy we consider involves minimal disruption to traffic not affected by the failure. In this case, we compute bypass routes for flows that previously crossed the failed link. These bypass routes carry traffic from one endpoint of the failed link to the other by routing around the failure. The endpoints of the traffic flow need not be aware of, or participate in, the bypass routing. In this case, the effect of the failure is limited, but the resulting performance is likely to be worse than in the first strategy.

In a third strategy, we also re-route only traffic flows that crossed the affected link, but in this case we do so end-to-end rather than by bypassing the failed link locally. Unaffected traffic is left in place, and new end-to-end routes for flows crossing the failed link are computed using a similar optimization to the one by which the original routes are computed (but with modified constraints which force the routing to remain the same on unaffected routes.)

We should note that the idea of path restoration in general, and the specific strategies we propose are not new. Previous work, for example in the context of ATM networks, has considered both optimal local and end-to-end restoration [2, 18, 22]. In contrast to our work, however, this prior work assumes knowledge of the traffic matrix. By developing techniques that address this issue in the context of unknown or uncertain traffic demands, our work advances the state of the art in this area.

We use a competitive analysis framework where under failure we compare the performance of our routing strategies to the best possible routing under failure. Testing these algorithms on a diverse collection of ISP networks, we find that with careful selection of restoration paths one can obtain restoration strategies that retain nearly optimal performance on the failed network while minimizing disruptions to traffic flows that did not traverse the failed link.

The contributions of this paper are threefold. First, we derive a methodology and metrics for evaluating restoration in the event of link failures. Second, we develop algorithms for constructing optimal restoration paths for a given routing and a wide range of traffic demands. Finally, we evaluate these algorithms and compare their performance under different failure scenarios. Consequently, we have filled an important gap by applying techniques from demand oblivious routing to the problem of path restoration.

We organize our presentation as follows. In Section 2 we provide some preliminaries and in Section 3 we present in detail our restoration models. Our evaluation setup is described in Section 4 and the results are presented in Section 5. In Section 6 we state the LP models and algorithms and in Section 7 we give worst-case bounds on the performance of our restoration strategies.

# 2. PRELIMINARIES

## 2.1 Routing

A *routing* specifies how traffic between each Origin-Destination (OD) pair is routed across the network. Typically, there are multiple paths for each OD pair, and a fraction of the traffic is routed over each path.

For our purposes, the relevant characterization of each routing is what fraction of traffic, for each OD pair, is routed along each link. Thus, the routing is specified by a set of values $f_{ab}(i,j)$ that specifies the fraction of demand from $a$ to $b$ that is routed on the link $(i,j)$. Note that the values $f_{ab}(i,j)$ for a given OD pair $a \rightarrow b$, should specify a flow of value 1 from $a$ to $b$. When the routing routes a demand $d_{ab}$ for the OD pair $a \rightarrow b$, the contribution of this demand to the flow on a link $(i,j)$ is $d_{ab}f_{ab}(i,j)$.

The protocols Open Shortest Path First (OSPF) and Intermediate System-Intermediate System (IS-IS) [17] specify a routing through a set of link weights. The traffic between each pair is always routed on shortest path(s) between the origin and destination (with respect to these weights). The MPLS protocol [5, 19] allows for a richer (general) specification of routings and more fine-tuned traffic engineering. Our optimization is with respect to routings of the more general form that can be implemented via MPLS but not necessarily via OSPF or IS-IS.

An example of a routing specification for a single OD pair is given in Figure 1(A). The complete specification involves such a flow for all OD pairs. Note that using standard flow techniques, such a flow can always be decomposed to a set of "path" flows from origin to destination (see e.g., [1]), which can be implemented under MPLS. It is known that $m$ path flows are always sufficient (where $m$ is the number of links in the network), but in the routings we obtain, a much smaller number of paths (say 5-10 paths) can be used (with no significant loss in routing performance.)

## 2.2 Routing performance

A common metric for the performance of a given routing with respect to a certain traffic matrix (TM) is the *maximum link utilization*. This is the maximum, over all links, of the total flow on the link divided by the capacity of the link (see e.g. [11, 12]). Formally, the maximum link utilization of a routing **f** on TM **D** (where $d_{ab}$ is the demand from $a$ to $b$) is

$$\text{MLU}_G(\mathbf{f},\mathbf{D}) = \max_{(i,j)\in \text{links}(G)} \frac{\sum_{a,b} d_{ab} f_{ab}(i,j)}{\text{cap}_{ij}},$$

where links$(G)$ is the set of links in the network $G$ and $\text{cap}_{ij}$ is the capacity of the link $(i,j)$. In the example of Figure 1(A), if there is demand of 40Mbps between $a$ and $b$, then the flow placed by the $a \rightarrow b$ demand on each of the two outgoing links from $a$ is 20Mbps ($0.5$ times the demand). Note that each of the other OD pair demands may also contribute to the flow on these two links. If the capacity of the two outgoing links from $a$ is 100Mbps, the flow placed by $a \rightarrow b$ demand contributes $0.2$ to the utilization on these links. The total flow on a link is the sum, over all OD pairs, of the flow contributed by the demand of that pair. Similarly, the utilization of the link is the sum, over all OD pairs, of the contribution of each pair to the utilization.

An *optimal* routing for a TM $\mathbf{D}$ on a network $G$ is a routing which minimizes the maximum link utilization. Formally, the optimal utilization for a TM $\mathbf{D}$ is given by

$$\textsc{optU}_G(\mathbf{D}) = \min_{\mathbf{f} | \mathbf{f} \text{ is a routing on } G} \textsc{MLU}_G(\mathbf{f}, \mathbf{D}) \, .$$

The *performance ratio* of a given routing $\mathbf{f}$ on a given TM $\mathbf{D}$ measures how far $\mathbf{f}$ is from being optimal on the TM $\mathbf{D}$. It is defined as the maximum link utilization of $\mathbf{f}$ on $\mathbf{D}$ divided by the minimum possible maximum link utilization on this TM. Formally,

$$\textsc{perf}_G(\mathbf{f}, \{\mathbf{D}\}) = \frac{\textsc{MLU}_G(\mathbf{f}, \mathbf{D})}{\textsc{optU}_G(\mathbf{D})} \, .$$

Note that the performance ratio is always at least 1; it is exactly 1 if and only if the routing is optimal for $\mathbf{D}$.

It is well known that the optimal routing for a given TM can be computed by solving a corresponding multi-commodity flow linear program (this routing was looked at in [16]). Since this routing is optimized for a specific TM, however, it does not provide performance guarantees for other TMs. The definition of the performance ratio of a routing $\mathbf{f}$ can be extended as follows to be with respect to a *TM-set* (a set of TMs), $D$:

$$\textsc{perf}_G(\mathbf{f}, D) = \max_{\mathbf{D} \in D} \textsc{perf}_G(\mathbf{f}, \{\mathbf{D}\}) \, .$$

A routing $\mathbf{f}$ is optimal for the set $D$ if and only if it minimizes the performance ratio; that is, $\textsc{perf}(\mathbf{f}, D)$ is minimal. The performance ratio $\textsc{perf}(\mathbf{f}, D)$ on the TM-set $D$ is always at least 1 – but note that the best possible performance ratio can be strictly larger than 1; since generally, a single routing that is optimal for all TMs in the set may not exist. When the TM-set $D$ includes all possible TMs, we refer to the performance ratio as the *oblivious performance ratio* of a routing. The oblivious ratio is the worst performance ratio a routing can obtain over any TM. A routing with a minimum oblivious ratio is an *optimal oblivious routing*, and its oblivious ratio is the *optimal oblivious ratio* of the network.

The definition of the performance ratio follows the "competitive analysis" framework where performance guarantees of a certain solution are provided relative to the best possible solution. In [4] we constructed and evaluated routings that provide performance guarantees for a range of TMs. We extend our definitions and metrics to measure performance of routing strategies under failures.

Maximum link utilization is obviously not the only performance metric one could evaluate in a study of optimal routing. We have chosen it because it has been used in many previous routing studies (for example, [11, 12]), it is used in capacity planning decisions by ISPs, and it may well be correlated with other metrics of interest, such as loss rate. However, we also believe that the techniques we have introduced here can be extended to incorporate other relevant performance metrics, such as end-to-end delay.

# 3. RESTORATION MODELS

We will consider link and node failures, but computationally all failures can be treated as a failure of a set of links. A node failure is simply a failure of all of its adjacent links. For a network $G$ and a set of links $L$, we denote by $G \setminus L$ the "failed" network with all links in $L$ removed. We measure the performance of restoration routing by comparing it to the best possible routing $\textsc{optU}_{G \setminus L}(\mathbf{D})$ on $G \setminus L$.

Failures can be viewed as forced temporary modification of the network, where the capacity of certain links decreases, possibly to 0.[1] When failures occur, the current routing may no longer be optimal (or even well defined) as some of the flow is destined to traverse a failed link.

Failures introduce an inherent tradeoff: On one hand, one would like to use a routing which is optimal for the "new" network. On the other hand, one would like to limit modifications to the current routing, since changes can potentially cause disruptions in service due to path shifts and route convergence time while the system reaches a consistent state. Thus, restoration performance has two axes. One is the complexity of performing the restoration. The other is link utilization compared to that of the best possible routing on the "failed" network.

Restoration models specify how routing is modified when failures occur. The three basic models we evaluate differ in complexity of implementation and in the disruption they introduce to other traffic flows. What is common in all cases is that we compute and evaluate the optimal restoration solution (in terms of its performance ratio) under the constraints of the model. The cost of the complexity and disruption of the strategies depends on specific characteristics of the networks involved. Further, these costs are not directly comparable to the performance advantages of the algorithms. Therefore, we make no attempt in this work to quantify these costs, and leave that as an item for future study.

- OPT: In this model, the restoration strategy is allowed to reroute traffic arbitrarily in order to obtain an optimal utilization on the modified network. Thus, the new routing $\mathbf{f}'$ minimizes the performance ratio $\textsc{perf}_{G \setminus L}(\mathbf{f}', D)$. The model captures one end of the spectrum where network utilization can be optimized, but the disruption can be large. We use OPT as a reference point for evaluating the performance of more restrictive (and practical) restoration models.

- AUG: In this model, the restoration strategy is end-to-end. It is allowed to reroute *arbitrarily* all *affected* flow paths (i.e., those flows that traverse a failed part of the network.) Our algorithms compute the optimal routing under the restriction that it must preserve all unaffected flow paths.

- BYPASS: In this model, the restoration strategy again reroutes only affected flows, but under more stringent constraints. The re-routing, which must be performed as a *bypass* around the failed link, is local rather than end-to-end. All original flow paths are preserved, with the exception that the portion of each path that utilizes the failed link is rerouted through the bypass. (Thus, BYPASS applies to failures of a single link at a time).

We now provide formal definitions of AUG and BYPASS restoration performance.

## 3.1 Routing augmentation restoration (AUG)

For a routing $\mathbf{f}$ and a set of links $L$, we define the $L$-truncation $\mathbf{f}'$ to be such that all affected routes (that is, routes that utilize a link in $L$) are removed. A computational definition of $\mathbf{f}'$, along with an algorithms to compute it, are provided in Section 6.1. For each OD pair $a \to b$, the truncated routing $\mathbf{f}'_{a,b}$ constitutes a flow of value at most 1 from $a$ to $b$. For each link $\ell \in L$ we have $f'_{a,b}(\ell) = 0$ and more generally, for any link $\ell$ we have $f'_{a,b}(\ell) \leq f_{a,b}(\ell)$.

A routing $\bar{\mathbf{f}}$ on $G \setminus L$ *augments* a truncation $\mathbf{f}'$ if for all OD pairs $(a, b)$ and links $\ell \notin L$, $\overline{f}_{a,b}(\ell) \geq f'_{a,b}(\ell)$. We refer to $h_{a,b}(\ell) = \overline{f}_{a,b}(\ell) - f'_{a,b}(\ell)$ as the *augmentation* of $\mathbf{f}'$ that forms $\bar{\mathbf{f}}$.

[1]We motivate the failure scenarios we address in Section 4.4. In particular, we discuss why we consider both complete and partial reduction of capacity when links fail.

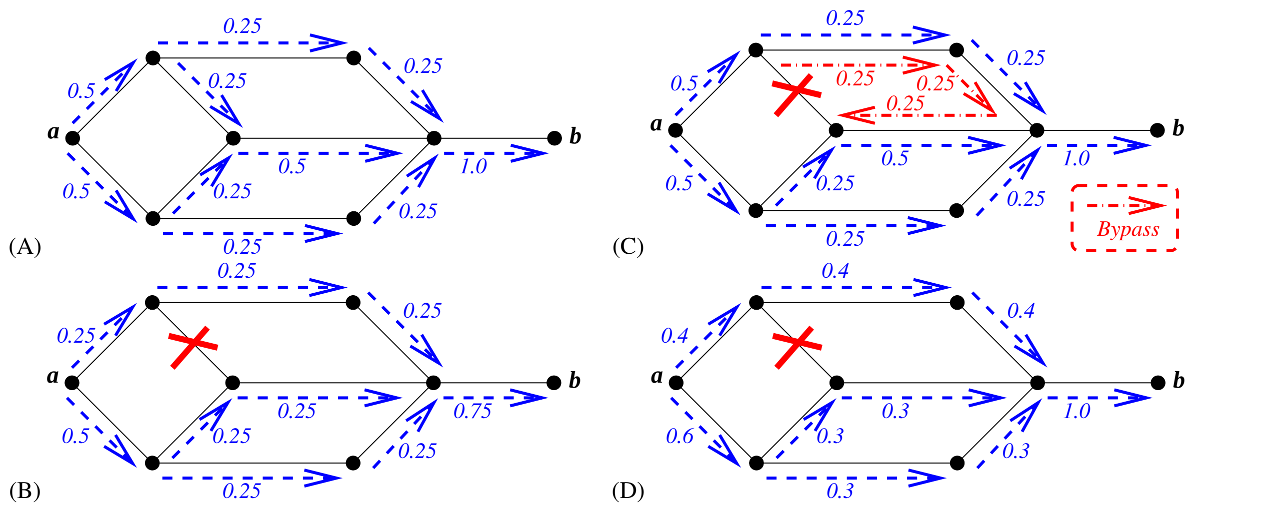


**Figure 1: (A) Example of a routing for a single OD pair in a network. (B) The truncation of the routing with respect to a link failure. (C) Link failure and a corresponding bypass routing. (D) A routing that augments the truncation.**

We now look for the best-performing routing for the TM-set $D$ on $G \setminus L$, which is one that minimizes $\text{PERF}_{G\setminus L}(\bar{\mathbf{f}}, D)$ subject to the constraint that $\bar{\mathbf{f}}$ augments $\mathbf{f}'$.

Figure 1(A) shows an example of a routing $\mathbf{f}_{a,b}$ for a single OD pair $a \rightarrow b$. Figure 1(B) shows the truncation $\mathbf{f}'_{a,b}$ as a result of a link failure. The original routing specified that the failed link carries $0.25$ of the total $a \rightarrow b$ demand. The truncated routing accounts for the removal of all end-to-end routes which traverse this link, and thus removed routes for $0.25$ portion of the end-to-end demand. Figure 1(D) shows a routing $\bar{\mathbf{f}}_{a,b}$ which augments $\mathbf{f}'_{a,b}$.

## 3.2 Link-bypass restoration (BYPASS)

Under BYPASS restoration, when a link $k = (a', b')$ fails, traffic which traverses the link is routed along alternate paths which do not utilize the failed link. A *bypass* $\mathbf{b}$ of a link $k$ is specified by values $b(a,b) \geq 0$ (for all edges $(a,b) \neq k$ which constitute a flow of value $1$ from $a'$ to $b'$. Each value $b(a,b)$ specifies the fraction of the flow that traversed the failed link $k$ under the original routing that the bypass routes along the edge $(a,b)$.

Under bypass restoration, the flow on a given link is equal to the original flow in the non-failed network plus the additional flow placed by the bypass routing. The maximum link utilization and respective performance ratio of a bypass $\mathbf{b}$ around a link $k$, for a given routing $\mathbf{f}$ on $G$ and demands $\mathbf{D}$ are

$$\text{MLUB}_{G\setminus\{k\}}(\mathbf{b},\mathbf{f},\mathbf{D}) = \max_{(i,j)\in \text{links}(G\setminus\{k\})} \frac{\sum_{a,b} d_{ab}(f_{ab}(i,j) + b(i,j) f_{ab}(k))}{\text{cap}_{ij}}$$

$$\text{PERFB}_{G\setminus\{k\}}(\mathbf{b},\mathbf{f},\{\mathbf{D}\}) = \frac{\text{MLUB}_{G\setminus\{k\}}(\mathbf{b},\mathbf{f},\mathbf{D})}{\text{OPTU}_{G\setminus\{k\}}(\mathbf{D})} .$$

Similarly, the performance with respect to a TM-set $D$ is defined as

$$\text{PERFB}_{G\setminus\{k\}}(\mathbf{b},\mathbf{f},D) = \max_{\mathbf{D}\in D} \text{PERFB}_{G\setminus\{k\}}(\mathbf{b},\mathbf{f},\{\mathbf{D}\}) .$$

For the original routing shown in Figure 1(A), Figure 1(C) shows a bypass and the fraction of the $a \rightarrow b$ demand routed along each edge as a result of the failure and routing along the bypass. Note that with bypass restoration, the same flow can be routed along both directions of the same link.

# 4. EVALUATION SETUP

In this section we describe our experimental setup, including the network topologies, link capacities, traffic matrices and failure scenarios used to evaluate our restoration strategies. The topologies and TMs are the same as those used in [4]. We review them below; additional details can be found in [4].

## 4.1 Topologies

We used five ISP maps from the Rocketfuel dataset [14] and collapsed the topologies so that "nodes" correspond to cities to obtain approximate PoP to PoP (Point of Presence) topologies. The resulting topologies are listed in Table 1. The topologies provided by Rocketfuel did not include the capacities of the links, which were needed for our study. We used a heuristic based on link weights to assign capacities. Since links and weights were symmetric (equal on both directions of each link), we obtained symmetric capacities.

## 4.2 Traffic matrices (TMs)

Accurate traffic matrices were not available for the topologies we studied. Therefore, we used two families of synthetic traffic matrices, which we refer to as *Bimodal* and *Gravity* TMs:[2]

*Bimodal TMs.* Bimodal TMs are motivated by the observation that only a fraction of Origin-Destination (OD) pairs has very large flows [7, 15]. The random bimodal distribution samples randomly a fraction of OD pairs and then assigns a demand for the pair uniformly at random from some range.

*Gravity TMs.* Since networks are designed with some expectation of traffic demands in mind, it is desirable to evaluate the performance of different routings with respect to such traffic demands. We used a Gravity model, similar to that suggested in [20], to generate demands that "correspond" to the network. The work

[2]Results in Section 5 are fairly consistent across the two traffic matrices, indicating that the specific choices of models may not be critical.

| AS | routers | orig-links | cities | links | reduced cities | reduced links |
|---|---|---|---|---|---|---|
| Telstra (Australia) 1221 | 108 | 306 | 57 | 59 | 7 | 9 |
| Ebone (Europe) 1755 | 87 | 322 | 23 | 38 | 18 | 33 |
| Tiscali (Europe) 3257 | 161 | 656 | 50 | 88 | 28 | 66 |
| Exodus (Europe) 3967 | 79 | 294 | 22 | 37 | 21 | 36 |
| Abovenet (US) 6461 | 141 | 748 | 22 | 42 | 17 | 37 |

**Table 1: Topologies from Rocketfuel (with AS number and name)**
The table lists the number of routers and links, the number of cities and inter-city links which we refer to as PoPs. The last two columns (reduced cities and links) list the number of remaining cities and links if 1-connected components are removed (these components do not change the performance ratio).

in [20] suggested a way to extrapolate a complete TM from measurements of incoming-outgoing flow into each PoP from the backbone. The extrapolation then assumed that the fraction of traffic sourced from a PoP is sinked at other PoPs proportionally to the total sinked flow at these PoPs. Since we did not have OD flow aggregates, we used a capacity-based heuristic, which assumes that the incoming/outgoing flow from each PoP is proportional to combined capacity of connecting links. We then applied the gravity model as in [20] to extrapolate a complete TM.

## 4.3 TM-sets

For each network, we considered the performance of different routings with respect to 3 sets of TMs (*D*), which refer to as TM-sets. The first set consisted of all TMs. The other two sets were obtained from a bimodal and a gravity "base" TM, respectively, with an "error margin" of $w = 2$: The base TM, $\mathbf{D}$ can be thought of as our best "guess" of the actual TM. The set of applicable TMs, $D_w$, includes each $\mathbf{D}'$ such that for all OD pairs $(i, j)$, $d_{ij}/w \leq d'_{ij} \leq wd_{ij}$. This set can be thought of as including all TMs with respect to which we want a performance guarantee. We used a fixed margin value of $w = 2$ (performance as a function of the margin was performed in [4] but is not a focus of the current work.)

## 4.4 Failure scenarios

We evaluated several types of failure scenarios, which we describe and motivate below. For each network, we generated 10 random instances of each failure scenario.

- *Single link failure (l)* A link (selected uniformly at random) fails (loses all its capacity). Thus all traffic through the link is affected. ISPs often use the notion of *span* for restoration planning. Two routes are *span disjoint* if they don't share any physical channel. Physical co-location implies correlation in the likelihood of failure due to malicious action, the elements, or human error. Single link failure simulates a scenario where all links (or the only link) between two PoPs are affected.

- *Partial link failures (hl)* A link (selected uniformly at random) loses half of its capacity. We assume that in such a case, a proportional amount of the traffic traversing the link is affected. The motivation for considering this failure type is that the links on our PoP-to-PoP topologies are aggregations of several different physical links. ISPs often deploy parallel links to increase resilience, and often there are multiple direct paths between two cities or PoPs.

- *Single node failure (n)* A node of the network (selected uniformly at random) fails. Thus, all traffic to, from and through the node is affected. The nodes of our networks correspond to cities or PoPs and a node failure captures a scenario where, due to a natural disaster, technical failure, or some malicious action, a whole geographic region is affected.

- *Multiple node failure (mn)* 20% of the nodes (selected uniformly at random) fail. This may happen as a result of a large-scale attack.

- *Multiple link failure (ml)* 20% of the links (selected uniformly at random) fail. This may happen as a result of a large-scale attack.

Some failures may leave the network disconnected. In particular, node failures always isolate a single node, and multiple link or node failures can result in multiple disconnected sections. When $G$ is disconnected and $\mathbf{D}$ has demands between disconnected parts we define $\text{OPTU}_G(\mathbf{D}) \equiv \infty$ and for all routings $\mathbf{f}$, $\text{PERF}_G(\mathbf{f}, D) \equiv 1$. This is consistent with our "competitive analysis" type definition of the performance ratio, where each routing is compared to the best possible routing: If the optimal routing incurs infinite maximum utilization, then our routing has performance ratio of "1" (is as good as the optimal one).

The oblivious performance ratio considers performance across all possible demand matrices. When the network is disconnected, some matrices will have demand between disconnected parts, while others will have demands only within connected parts. In this case, the oblivious performance ratio of each routing is determined by its performance on demand matrices that have non-zero demands only within components.

When considering restricted sets of demands, where *D* is defined with respect to a base TM and a margin, we used a different calculation. Note that node failures always disconnect the network in the sense that if there was demand to or from the failed node, then this TM can not be routed with finite congestion by any routing (and thus trivially, all routings have performance ratio of 1.). To obtain meaningful results, when the network is disconnected we compute the performance ratio with respect to a modified base TM, where demand values between disconnected components are eliminated. Note that this puts the optimal routing computed for the original network at some disadvantage, since it is expected to handle TMs for which it was not optimized.

## 4.5 Base routings

For each network and set of TMs, we computed an optimal routing (i.e., one that minimizes $\text{PERF}(\mathbf{f}, D)$). The optimal performance ratio for each setting is given in Table 2. The computation was performed using the tools and algorithms of [4]. Note that there are in general many optimal routings. We used an interior-point LP solver that generally tends to yield "average" rather than "basic" solutions, and we applied a flow "cleaning" routine to remove flow-cycles (which while redundant, may not affect the performance ratio). For each failure scenario, the AUG and BYPASS restoration were computed with respect to that optimal routing on the original network. We also computed the optimal routing for the failed network (corresponding to the OPT restoration model).

| AS / $D$ | All TMs | bimodal/$w = 2$ | gravity/$w = 2$ |
|---|---|---|---|
| 1221 | 1.43 | 1.00 | 1.00 |
| 1755 | 1.78 | 1.06 | 1.00 |
| 3257 | 1.80 | 1.27 | 1.30 |
| 3967 | 1.62 | 1.25 | 1.16 |
| 6461 | 1.91 | 1.14 | 1.30 |

**Table 2: Optimal performance ratio for each network and set of TMs (without failures).**

# 5. EVALUATION RESULTS

The three models vary in the degree to which they constrain the restoration process. Specifically, OPT is the least constrained model, followed by AUG, and BYPASS is the most constrained. We compared the performance of these models and attempted to answer the following questions. How much is lost (in terms of network utilization) if we restrict routing to AUG or BYPASS restoration? How does performance depend on the failure scenario (extent of failure), the size of the network, and the type of TM-set?

For each network and corresponding TM-sets we performed the following computations. For each single-link failure (l), we computed the performance ratio of the optimal restoration under the three restoration models: OPT, AUG, and BYPASS. For all other failure scenarios (lh, n, ml, and mn) we computed the performance ratio of optimal restoration under the OPT and AUG restoration models.[3]

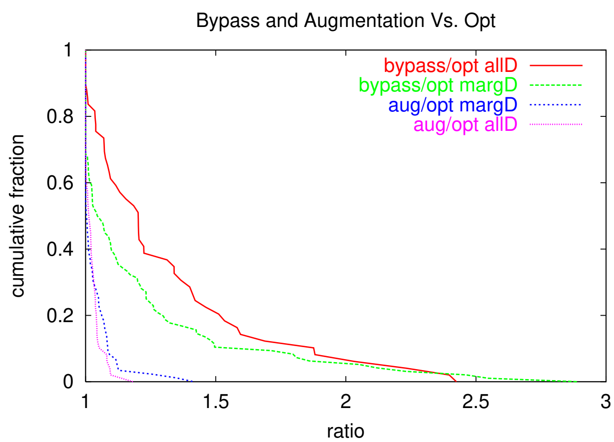


**Figure 2: Cumulative distribution of the ratio of the performance ratio BYPASS or AUG restoration to that of OPT restoration. The ratio shows by how much performance deviates from the optimal possible with arbitrary reroutes of traffic. (For all TMs and for gravity/bimodal TMs with $w = 2$ margin).**

For single-link failures (l), Figure 3 shows the cumulative distribution of the optimal performance ratios (across networks) under optimal restoration in each restoration model. The left figure shows it for routings designed for all TMs, whereas the right side shows it for routings designed for a base TM (which is gravity or bimodal TM) with a margin. For each instance, OPT restoration obtains the best utilization, followed by AUG and BYPASS, and this order is thus reflected in the cumulative distributions. The optimal performance ratio of both OPT and AUG restoration on all of the networks and TM-sets we studied is less than 2. The performance ratio is lower on the more restricted TM-sets (that is, a base TM with a margin) than for the TM-set containing all TMs. Not surprisingly, BYPASS restoration resulted in higher performance ratios (over 4 in some instances for the TM-set containing all TMs and over 2.5 in some instances for the TM-sets containing a base TM and error margin), but the performance ratio is below 2 in the majority of instances. A direct comparison of the performance of the different restoration models is provided in Figure 2. The figure shows the distribution of the ratio of the performance ratio of AUG and BYPASS to that of OPT, over all instances, separately for the TM-set containing all TMs and for the TM-sets that include a base TM and margin. The figure shows that on the vast majority of cases, AUG is very close to OPT and BYPASS is within 50% of OPT. For example, in over 90% of the cases, the ratio of AUG to OPT is less than 1.1. Therefore, the performance impact of allowing the restoration process to reroute only those flows that traversed the affected link is minimal.

We next examine the performance of AUG and OPT under all of the failure scenarios. Figure 4 shows a breakdown of the ratio of the performance ratios of AUG to OPT, aggregated across all networks. We can again see that for all failure scenarios and TM-sets, the performance of optimal AUG restoration is still close to OPT. Interestingly, we can also see that the gap in performance between AUG and OPT tends to be larger for the more severe failure scenarios. (Ranked by severity, the failure scenarios are lh, l, n, ml, mn, with hl being the least severe.) This is because as the failed network diverges more significantly from the original one, the optimal routing for the original network is less likely to be close to optimal on the modified one. This dependence is more evident for the TM-set containing all TMs; for the base-TM-with-margin TM-sets, the curves are clustered close together (except for hl failures) and the relation is not as evident.

We next consider the dependence of performance on the network topology. Figure 5 shows no obvious correlation between network size and restoration quality. However, smaller networks with lower connectivity can be more sensitive to failures (in particular the hl, l, and n scenarios that are not scaled with the size of the network.) These failures can drastically change the structure for a smaller network. This is even more significant with the TM-sets that includes a base-TM-with-margin, since they are not optimized for cases where the network becomes disconnected and demand between disconnected components is dropped. This explains the poorer performance of network 1221 (which has two 2-connected components, connected by a single link.) Note that under the ml and mn failure scenarios, the failure "damage" is scaled with the size of the network. Therefore, with these failure scenarios, larger networks did not have uniformly better ratios.

When links fail, the optimal performance ratio over a TM-set can increase or decrease. The minimum possible maximum link utilization always increases, but so does the optimal utilization for any specific TM. Since the performance ratio is measured relative to the optimum for the specific TM, the ratio can either decrease or increase. Figure 6 shows how the performance ratio changes for specific networks and TM-sets, aggregated across failure scenarios. The figure shows the cumulative distribution of performance ratios for AUG and OPT restoration. The figure also shows the performance ratio of the original network (that is, the network with no failures.) The performance ratio under failures is clustered around that of the original network for both AUG and OPT restoration, but is below the original optimal ratio much less frequently with AUG than with OPT restoration.

[3]Recall from Section 3 that BYPASS is only applied to single link failures.

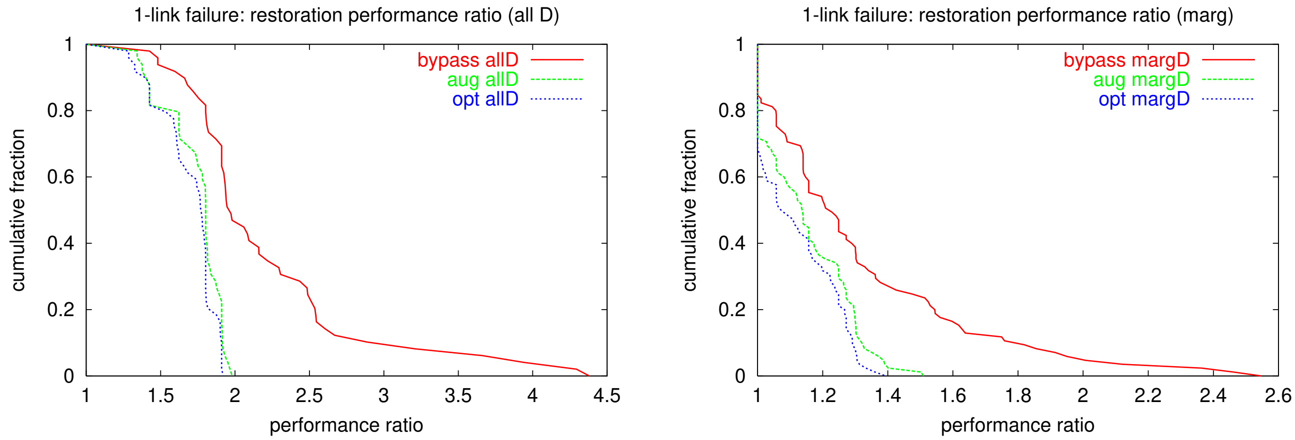


**Figure 3: Cumulative distribution of the performance ratio of the different restoration methods for single link failures: BYPASS, AUG, and OPT restoration, aggregated across all networks. Separately for all TMs (left) and for gravity/bimodal TMs with $w = 2$ margin (right).**

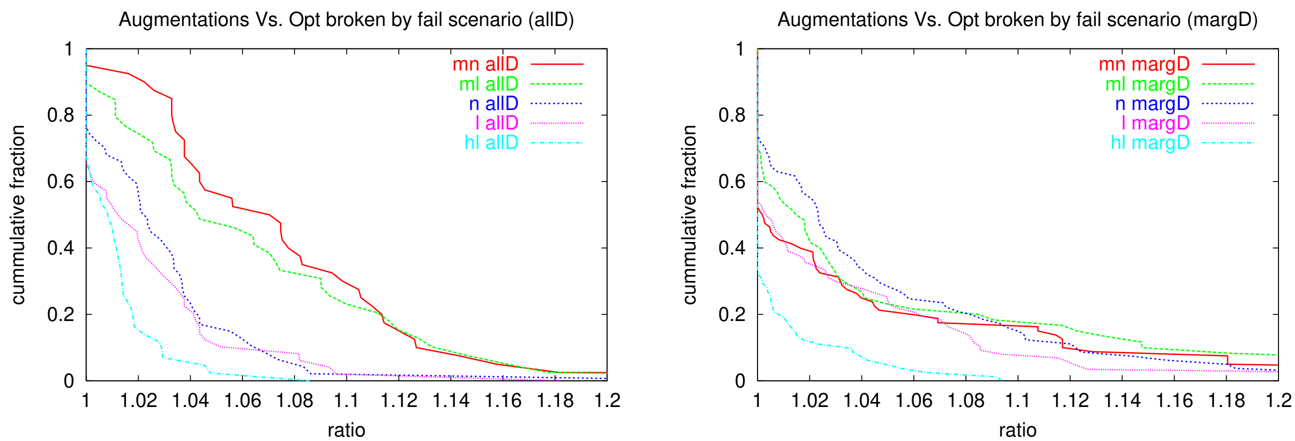


**Figure 4: Cumulative distribution of the ratio of performance ratios of AUG restoration to OPT across networks broken down by failure scenarios and demand sets, for all TMs (left) and for gravity/bimodal TMs with $w = 2$ margin (right).**

# 6. LP MODELS AND ALGORITHMS

We provide details on the algorithms and LP models we developed and applied.

## 6.1 Computing a truncated flow

The flow truncation algorithm inputs a flow and a failed link and outputs a truncation of the original flow in which the flow going on, into, and from the failed link is removed. Formally, the input constitutes of a network, an $s$-$t$ flow $\mathbf{g}_{s,t}$ (a flow from the node $s$ to the node $t$ in $G$ where $g(e) \geq 0$ denotes the flow value on an edge $e$), an edge $(a, b)$ (which corresponds to a "failed" link), and a *removed flow* value $R(a, b) \leq g(a, b)$.

The output of the algorithm are values $0 \leq R(e) \leq g(e)$ for all edges $e \neq (a, b)$. The values $R(e)$ (for $e \in G$) specify an $s$-$t$ flow of value $R(a, b)$. The truncated flow is then defined as $\mathbf{g}'_{s,t}$ where $g'(i, j) = g(i, j) - R(i, j)$. There are many possible truncations of $\mathbf{g}_{s,t}$. Our algorithm computes the one that corresponds to removal of the $s$-$t$ flow paths traversing $(a, b)$ under *stateless* flow splitting. A routing protocol supports stateless splitting for an OD pair flow, if in each node $u$, the link selected for outgoing packets is independent of the path the packet was routed on to get from $s$ to $u$. Stateless splitting can be realized, e.g., by splitting according to a random hash function (different at each node) applied to some parameters of the flow (e.g., source IP address and port). While OSPF attempts to approximate stateless splitting, MPLS, from its definition of end-to-end paths, does not: There are many possible translations of a flow-specified routing to a decomposition into $s$-$t$ routes. The truncation can always be computed with respect to the particular decomposition used. In a sense, our stateless truncation corresponds to an "average" decomposition of the flow-specified routing into paths.

The statement of the algorithms assumes that the input flow $\mathbf{g}_{s,t}$ is *clean* (does not contain flow cycles). Unclean flows do not make sense as routings, and it is always possible to remove flow cycles without increasing flow values on any edge. Since the flow is clean, the subgraph induced on the edges $\{e | g(e) > 0\}$ is acyclic. Thus, it is possible to order its nodes in a topological order that is consistent with the flow directions. We can assume that $a < b$ in this topological order (otherwise, the flow value and the removed flow values are 0 and the truncated flow is equal to the original flow).

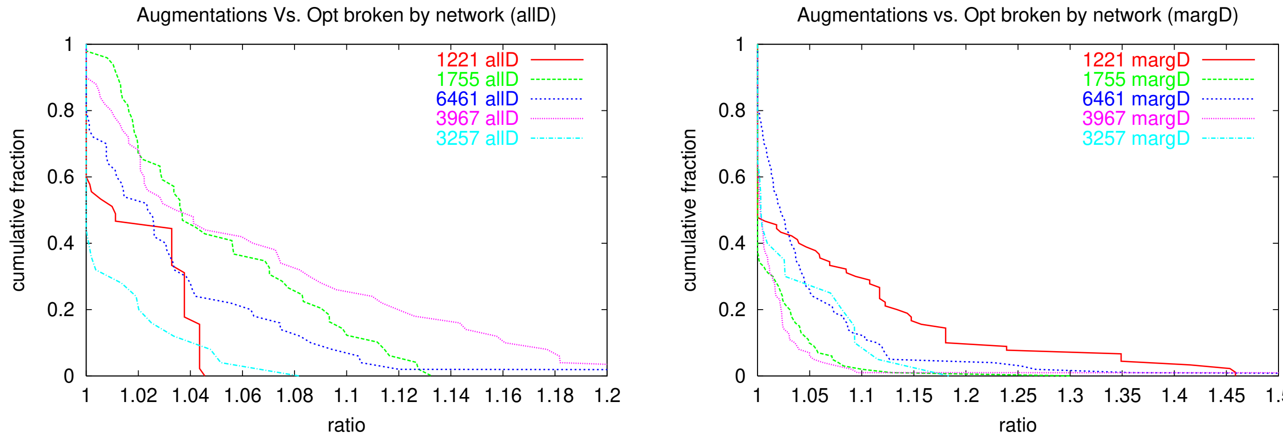


**Figure 5: Cumulative distribution of the ratio of performance ratios of AUG restoration to OPT across failure scenarios broken down by network and demand sets, for all TMs (left) and for gravity/bimodal TMs with $w = 2$ margin (right).**

*Algorithm truncate*$(\mathbf{g}_{s,t}, (a,b), R(a,b))$

- Compute a topological order of the nodes spanned by $\mathbf{g}$ (where edges with positive flow are directed in the direction of the flow).
- Initialize $R(i,j) \leftarrow 0$ for all edges $(i,j) \neq (a,b)$.
- Consider nodes $v$ sequentially, starting at $b$ and continuing according to increasing topological order:
  - Let $T_r(v)$ be the sum of $R()$ values on all incoming edges to $v$.
    Let $T(v)$ be the sum of $g()$ values on all outgoing edges from $v$.
  - For each outgoing edge from $v$, $(v,u)$, set $R(v,u) \leftarrow T_r(v)g(v,u)/T(v)$.
- Consider nodes $v$ sequentially, starting at $a$ and continuing according to decreasing topological order:
  - Let $T_r(v)$ be the sum of $R()$ values on all outgoing edges from $v$.
    Let $T(v)$ be the sum of $g()$ values on all incoming edges to $v$.
  - For each incoming edge to $v$, $(u,v)$, set $R(u,v) \leftarrow T_r(v)g(u,v)/T(v)$.

In our simulations the truncation is applied to all OD pairs (observe that the computation is independent for different pairs). When simulating a link failure, we remove all flow on that link and truncate accordingly. In a partial link failure (where part of the capacity fails), we remove flow proportionally to the capacity failure. For multiple link failure we apply this procedure iteratively, link by link, to compute the truncated flow with respect to all failed links.

## 6.2 Computing an optimal BYPASS restoration

For the purposes of presentation in this section, we use the term *link* for an undirected edge, *edge* for a directed edge, and denote by link-of$(e)$ the link corresponding to edge $e$. We use the notation

$$f_{ij}(l) = \sum_{e:\text{link-of}(e)=l} f_{ij}(e)\ ,$$

the flow on a link is defined to be the sum of flows on the two directed edges of the link. As in [4], we present LP formulations for networks that are *symmetric* (the two directions of each link have equal capacities) and *undirected* (where link capacities apply to the sum of flows on both directions). Straightforward modifications yield the respective models for directed (and possibly asymmetric) networks (with a 2-fold increase in the size of the LPs).

Let $\mathbf{f}$ be a routing and let $k = (a,b)$ be a failed link. A bypass $\mathbf{b}_{a,b}$ is defined by values $0 \leq b_{a,b}(e)$ for $e \in G \setminus \{k\}$ which specify a flow of value 1 from $a$ to $b$. The *reverse* bypass $\mathbf{b}_{b,a}$ routes flow from $b$ to $a$. We say that $\mathbf{b}_{b,a}$ and $\mathbf{b}_{a,b}$ are *symmetric* if for all edges $(i,j)$, $b_{a,b}(i,j) = b_{b,a}(j,i)$. By extending arguments from [4], we obtain that when the network and TM-set are symmetric, there is an optimal pair of bypasses for the two directions of the failed link which are symmetric. We denote $b_k(\text{link-of}(i,j)) = b_{a,b}(i,j) + b_{a,b}(j,i)$.

We start by describing a simpler LP formulation for computing an optimal bypass for a single link failure with respect to a TM-set that includes all TMs.

THEOREM 1. *Given a routing on $G$ and a failed link $k = (a,b)$, a bypass that minimizes the performance ratio can be computed by an LP with $O(mn)$ variables and $O(mn)$ constraints.*

PROOF. The variables in our formulation are $b_{a,b}(e)$ (for edges $e \in G \setminus \{k\}$) to specify the bypass, $d_{ij}$ (for pairs $i \to j$) to specify demand values, $g_i(e)$ (for nodes $i$ and $e \in G \setminus \{k\}$) to specify a multi-sink flow [4] in $G \setminus \{k\}$ that is sourced at $i$, and $s$ and $r$ (which as we shall see essentially can be treated as a constant where we perform a binary search on its value). We first argue that the following formulation works:

$$\min r$$

$b_{a,b}(e)$ for $e \in G \setminus \{k\}$ is a bypass

$$\text{The TM } \mathbf{D} \text{ has } \text{OPTU}(\mathbf{D}) \leq s \qquad (1)$$

$$\sum_{ij} f_{ij}(k) d_{ij} = 1 \qquad (2)$$

$\forall$ links $l \in G \setminus \{k\}$,

$$\frac{\sum_{ij} f_{ij}(l) d_{ij} + b_k(l)}{\text{cap}(l)} \leq sr \qquad (3)$$

[4] A muti-sink flow is one with a single source and multiple sinks.

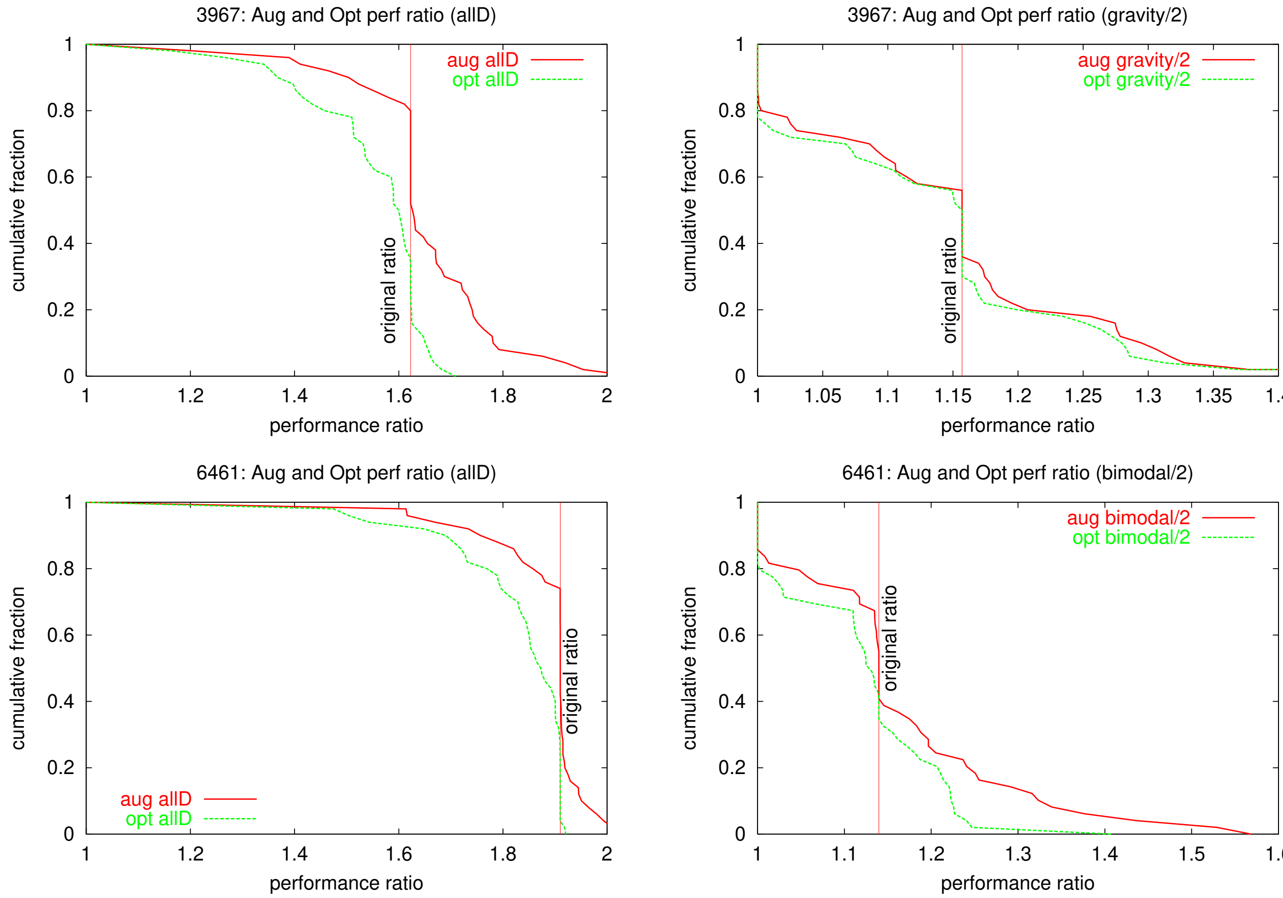


**Figure 6: Cumulative distribution of the performance ratios of AUG restoration to OPT for network 3967, all demands (left), and gravity TM/$w = 2$ (right), and network 6461, all demands (left), and bimodal TM/$w = 2$ (right). The plot also shows the optimal performance ratio of the original network (that is, the network without failures.)**

Observe that the performance ratio of a routing is invariant under scaling of the TM. Thus, when computing an optimal routing for a TM-set $D$, it is sufficient to consider a set that includes a scaled "representative" TM for each TM in the original TM-set. Constraint (2) which says that the total flow on the failed link is equal to 1 essentially determines a scaling factor for the TM, since for any TM $\mathbf{D}$, the TM $\mathbf{D}' = \alpha\mathbf{D}$ with $\alpha = (\sum_{ij} f_{ij}(k) d_{ij})^{-1}$ has $\sum_{ij} f_{ij}(k) d'_{ij} = 1$. The TM, $\mathbf{D}'$ satisfies the constraints, with $s = \mathrm{OPTU}(\mathbf{D}')$.

We next consider Constraint (3). By combining Constraints (2) and (3) we obtain that an equivalent constraint to (3) is

$$\sum_{ij} \frac{f_{ij}(l) + b_k(l) f_{ij}(k)}{\mathrm{cap}(l)} d_{ij} \le sr \ ,$$

which states that the flow on link $l$ when routing demands $d_{ij}$ using the bypass is at most $s$ times $r$. (This constraint, however, is not linear, and thus the need for this little manipulation.)

To conclude the proof, note that Requirement (1) can be encoded via the following linear constraints (which introduce $O(mn)$ additional variables $g_i(e)$ that specify $n$ single-source multi-sink flows [3]):

$$\begin{aligned}
&\forall \text{ nodes } i, \forall \text{ nodes } j \neq i: \qquad\qquad (4)\\
&\qquad \sum_{e \in \mathrm{OUT}[j]} g_i(e) - \sum_{e \in \mathrm{IN}[j]} g_i(e) + d_{ij} \le 0\\
&\forall \text{ links } m \in G \setminus \{k\}: \sum_i g_i(m) \le \mathrm{scap}(m)\\
&\forall \text{ nodes } i, \text{ links } m \in G \setminus \{k\}: g_i(m) \ge 0\\
&\forall \text{ pairs } i \to j\text{:} \;\; d_{ij} \ge 0
\end{aligned}$$

□

*Bypasses for multiple link failures.* We next derive a more powerful formulation using techniques developed in [4]. This formulation extends to bypasses for multiple link failures. Given a network $G$, a routing $\mathbf{f}$, and a set $L$ of failed links, the goal is to obtain an optimal set of bypasses $\mathbf{b}_e$ (for every $e$ such that link-of$(e) \in L$). Each bypass $\mathbf{b}_e$ is a flow of value 1 on $G \setminus L$ between the endpoints of $e$. A set of optimal bypasses is defined as a solution of the following LP (which has an infinite number of constraints.)

$$\min r$$
$$\forall \{(u,v) | \text{link-of}(u,v) \in L\} \quad (5)$$
$$\mathbf{b}_{u,v} \text{ is a } (u,v)\text{-flow of value 1 in } G \setminus L$$
$$\forall \text{ links } l \in G \setminus L \quad (6)$$
$$\forall \text{ TMs } \mathbf{D} \text{ with } \text{OPTU}_{G \setminus L}(\mathbf{D}) = 1:$$
$$\sum_{ij} \frac{f_{ij}(l) + \sum_{k \in L} b_k(l) f_{ij}(k)}{\text{cap}(l)} d_{ij} \leq r \quad (7)$$

Given a set of bypasses, Constraints (6) can be tested by solving, for each link $l$, the following "slave LP," and testing if the objective is $\leq r$ or not[5]

$$\max \sum_{ij} \frac{f_{ij}(l) + \sum_{k \in L} b_k(l) f_{ij}(k)}{\text{cap}(l)} d_{ij} \quad (8)$$
$$\mathbf{D} \text{ is a TM with } \text{OPTU}_{G \setminus L}(\mathbf{D}) \leq 1 \quad (9)$$

Thus, the LPs (8) can be used as a separation oracle for Constraints (6), giving polynomial solvability using the Ellipsoid algorithm [13]. Using similar manipulations and consideration as in [4] we obtain the following polynomial size LP formulation with $O(mn^2)$ variables and $O(nm^2)$ constraints:

$$\min r \quad (10)$$
$$\forall \{(u,v) | \text{link-of}(u,v) \in L\}$$
$$\mathbf{b}_{u,v} \text{ is a } (u,v)\text{-flow of value 1 in } G \setminus L$$
$$\forall \text{ links } l \notin L:$$
$$\sum_m \text{cap}(m) \pi_l(m) \leq r$$
$$\forall \text{ pairs } i \to j: \quad (11)$$
$$p_l(i,j) - \sum_{k \in L} \frac{f_{ij}(k)}{\text{cap}(l)} b_k(l) \geq \frac{f_{ij}(l)}{\text{cap}(l)}$$
$$\forall \text{ nodes } i, \forall \text{ edges } e = (j, j'):$$
$$\pi_l(\text{link-of}(e)) + p_l(i,j) - p_l(i,j') \geq 0$$
$$\forall \text{ links } m \notin L: \ \pi_l(m) \geq 0$$
$$\forall \text{ nodes } i: \ p_l(i,i) = 0$$
$$\forall \text{ nodes } i,j: \ p_l(i,j) \geq 0$$

The LP which computes the optimal performance ratio on TMs when for each pair $i \to j$ $d_{ij}$ is restricted to the range $[a_{ij}, b_{ij}]$, differs from the LP (10) as follows. We use additional sets of variables $s_l^-(i,j)$ and $s_l^+(i,j)$. For each link $l \notin L$, Constraint (11) is replaced by

$$\forall \text{ pairs } i \to j:$$
$$p_l(i,j) - \sum_{k \in L} \frac{f_{ij}(k)}{\text{cap}(l)} b_k(l) + s_l^+(i,j) - s_l^-(i,j) = \frac{f_{ij}(l)}{\text{cap}(l)}$$

and the following constraints are added

$$\sum_{ij} (b_{ij} s_l^+(i,j) - a_{ij} s_l^-(i,j)) \leq 0$$
$$\forall \text{ nodes } i,j: \ s_l^-(i,j) \geq 0$$
$$\forall \text{ nodes } i,j: \ s_l^+(i,j) \geq 0$$

[5]To get an LP formulation recall that Requirement (9) can be encoded via the linear constraints 4 (substituting 1 for the variable $s$).

## 6.3 Computing an optimal AUG restoration

Given a routing $\mathbf{f}$ and a set $L$ of failed links, we can compute an $L$-truncation $\mathbf{f}'$ of $\mathbf{f}$ using the truncation algorithm in Section 6.1.

An optimal augmentation $\mathbf{h}$ of $\mathbf{f}'$ amounts to solving the following problem on the network $G \setminus L$: Given a flow $\mathbf{f}' \geq 0$ on $G \setminus L$, we are looking for an augmentation $\mathbf{h} \geq 0$ such that $\mathbf{f}' + \mathbf{h}$ is a routing with minimum performance ratio on $G \setminus L$ among all routings that augment $\mathbf{f}'$. This problem is similar to computing an optimal oblivious routing on $G \setminus L$ except that we constrain the routing via some lower bounds on the flow values on some edges:

$$\min r$$
$$\forall \text{ pairs } i \to j, \text{ edges } e: h_{ij}(e) \geq 0$$
$$\mathbf{f}' + \mathbf{h} \text{ is a routing in } G \setminus L$$
$$\forall \text{ links } l \quad (12)$$
$$\forall \text{ TMs } \mathbf{D} \text{ with } \text{OPTU}_{G \setminus L}(\mathbf{D}) = 1:$$
$$\sum_{ij} (h_{ij}(l) + f'_{ij}(l)) d_{ij} / \text{cap}(l) \leq r$$

Given $\mathbf{h}$ such that $\mathbf{h} + \mathbf{f}'$ is a routing, the constraints (12) can be tested by solving, for each link $l$, the following "slave LP," and testing if the objective is $\leq r$ or not.

$$\max \sum_{ij} \frac{h_{ij}(l) + f'_{ij}(l)}{\text{cap}(l)} d_{ij} \quad (13)$$
$$\mathbf{D} \text{ is a TM with } \text{OPTU}_{G \setminus L}(\mathbf{D}) \leq 1$$

Thus, the LPs (13) can be used as a separation oracle for the constraints (12), giving polynomial solvability using the Ellipsoid algorithm. By manipulating as in [4] we obtain that an optimal augmentation $\mathbf{h}$ can be obtained via the following LP with $O(mn^2)$ variables and $O(nm^2)$ constraints:

$$\min r \quad (14)$$
$$\mathbf{f}' + \mathbf{h} \text{ is a routing in } G \setminus L$$
$$\forall \text{ links } l \notin L:$$
$$\sum_m \text{cap}(m) \pi_l(m) \leq r$$
$$\forall \text{ pairs } i \to j: \quad p_l(i,j) - \frac{h_{ij}(l)}{\text{cap}(l)} \geq \frac{f'_{ij}(l)}{\text{cap}(l)} \quad (15)$$
$$\forall \text{ nodes } i, \forall \text{ edges } e = (j, j'):$$
$$\pi_l(\text{link-of}(e)) + p_l(i,j) - p_l(i,j') \geq 0$$
$$\forall \text{ links } m \notin L: \ \pi_l(m) \geq 0$$
$$\forall \text{ nodes } i: \ p_l(i,i) = 0$$
$$\forall \text{ nodes } i,j: \ p_l(i,j) \geq 0$$

When for each pair $i \to j$, the demand value $d_{ij}$ is restricted to a range $[a_{ij}, b_{ij}]$, we obtain a modified version of the LP (14), with additional variables $s_l^-(i,j)$ and $s_l^+(i,j)$. For each $l \notin L$, constraint (15) is replaced by

$$\forall \text{ pairs } i \to j: p_l(i,j) - \frac{h_{ij}(l)}{\text{cap}(l)} + s_l^+(i,j) - s_l^-(i,j) = \frac{f'_{ij}(l)}{\text{cap}(l)}$$

and we add the constraints:

$$\sum_{ij} (b_{ij} s_l^+(i,j) - a_{ij} s_l^-(i,j)) \leq 0$$
$$\forall \text{ nodes } i,j: \ s_l^-(i,j) \geq 0$$
$$\forall \text{ nodes } i,j: \ s_l^+(i,j) \geq 0$$

## 7. WORST CASE BOUNDS

We provide upper bounds on the performance of AUG and BYPASS restoration with respect to the optimal routing. Our simulations indicate that in practice we can expect considerably better performance than obtained from these bounds. We first bound the oblivious performance ratio of an AUG restoration derived on top of an optimal oblivious routing on the original network by the sum of the optimal oblivious ratios on the original and failed networks.

LEMMA 7.1. *Consider a routing* $\mathbf{f}$ *on* $G$*, a TM-set* $D$*, and a set of failed links* $L$*. Let* $\hat{\mathbf{f}}$ *be an optimal routing for* $D$ *on* $G \setminus L$*. Let* $\mathbf{f}' + \mathbf{h}$ *be an optimal AUG restoration routing on* $G \setminus L$*. Then, the performance ratio of* $\mathbf{f}' + \mathbf{h}$ *on* $G \setminus L$ *is at most the sum of the performance ratio of* $\mathbf{f}$ *on* $G$ *and the optimal performance ratio on* $G \setminus L$*, that is,*

$$\text{PERF}_{G\setminus L}(\mathbf{f}' + \mathbf{h}, D) \leq \text{PERF}_G(\mathbf{f}, D) + \text{PERF}_{G\setminus L}(\hat{\mathbf{f}}, D) .$$

PROOF. Let $\mathbf{h}'$ be an augmentation of the truncation $\mathbf{f}'$, constructed according to the routing $\hat{\mathbf{f}}$ as follows: For the OD pair $a, b$ and an edge $e \in G \setminus L$, we have $h'_{a,b}(e) = (1 - |f'_{a,b}|)\hat{f}_{a,b}(e)$ (where $|f'_{a,b}|$ is the magnitude of the flow from $a$ to $b$ on the truncation $\mathbf{f}'$.) We will show that the inequality holds for the oblivious performance ratio of the routing $\mathbf{f}' + \mathbf{h}'$. The claim for the optimal augmentation $\mathbf{h}$ follows since by optimality of $\mathbf{h}$,

$$\text{PERF}_{G\setminus L}(\mathbf{f}' + \mathbf{h}, D) \leq \text{PERF}_{G\setminus L}(\mathbf{f}' + \mathbf{h}', D) .$$

Consider a particular TM $\mathbf{D} \in D$, an OD pair $(a, b)$, and an edge $e$. The flow induced on $e$ by $(a, b)$ when routed according to $\mathbf{f}' + \mathbf{h}'$ is at most $d_{a,b}(f_{a,b}(e) + \hat{f}_{a,b}(e))$. The total flow induced by all OD pairs is thus at most $\sum_{a,b} d_{a,b}(f_{a,b}(e) + \hat{f}_{a,b}(e))$ and hence the following inequality holds for the maximum link utilization

$$\text{MLU}_{G\setminus L}(\mathbf{f}' + \mathbf{h}', \mathbf{D}) \leq \text{MLU}_G(\mathbf{f}, \mathbf{D}) + \text{MLU}_{G\setminus L}(\hat{\mathbf{f}}, \mathbf{D}) .$$

Dividing both sides by $\text{OPTU}_{G\setminus L}(\mathbf{D})$ and using the fact that $\text{OPTU}_G(\mathbf{D}) \leq \text{OPTU}_{G\setminus L}(\mathbf{D})$ we obtain that

$$\text{PERF}_{G\setminus L}(\mathbf{f}' + \mathbf{h}', \{\mathbf{D}\}) \leq \text{PERF}_G(\mathbf{f}, \{\mathbf{D}\}) + \text{PERF}_{G\setminus L}(\hat{\mathbf{f}}, \{\mathbf{D}\}) .$$

Since the inequality holds for each particular TM $\mathbf{D}$, it also holds for the maximum over $\mathbf{D} \in D$. ☐

We next Lemma bound the performance ratio under the optimal bypass restoration on a *symmetric* network. (It is easy to see that symmetry is required for obtaining such bounds.)

LEMMA 7.2. *Consider a routing* $\mathbf{f}$ *on* $G$*, a failed link* $k$*, and a TM-set* $D$*. Let* $\mathbf{b}$ *be an optimal bypass* $\mathbf{b}$ *with respect to* $\mathbf{f}$*,* $D$*, and* $k$*. Let* $\hat{\mathbf{f}}$ *be an optimal routing for* $D$ *on* $G \setminus \{k\}$*. Then*

$$\text{PERFB}_{G\setminus\{k\}}(\mathbf{b}, \mathbf{f}, D) \leq 2\text{PERF}_G(\mathbf{f}, D) + \text{PERF}_{G\setminus\{k\}}(\hat{\mathbf{f}}, D) .$$

PROOF. Let the bypass $\mathbf{b}^{st}$ be a (value 1) $s$-$t$ flow on $G \setminus \{k\}$ with minimum maximum link utilization. For any routing $\mathbf{f}$ and TM, the bypass $\mathbf{b}^{st}$ has the property that it minimizes the maximum contribution of the bypass flow to the utilization of each link. Since $\mathbf{b}$ is optimal for $\mathbf{f}$ we have $\text{PERFB}_{G\setminus\{k\}}(\mathbf{b}, \mathbf{f}, D) \leq \text{PERFB}_{G\setminus\{k\}}(\mathbf{b}^{st}, \mathbf{f}, D)$. It thus suffices to show that

$$\text{PERFB}_{G\setminus\{k\}}(\mathbf{b}^{st}, \mathbf{f}, D) \leq 2\text{PERF}_G(\mathbf{f}, D) + \text{PERF}_{G\setminus\{k\}}(\hat{\mathbf{f}}, D) .$$

Consider a TM $\mathbf{D}$ and a "tailored" bypass routing $\mathbf{b}^{\mathbf{D}}$ defined so that flow that reaches an endpoint of $k$ when routed using $\mathbf{f}$ is sent "backwards" to its respective origin and destination using the reverse directions of each link. This flow is then routed from its origin to its intended destination according to $\hat{\mathbf{f}}$. Thus, the tailored bypass is such that for each OD pair, $f_{ab}(k)d_{ab}/\sum_{i,j} f_{i,j}(k)d_{i,j}$ units are routed "back" to $a$ and "from" $b$ according to $f_{a,b}$ and then sent according to $\hat{f}_{a,b}$ from $a$ to $b$.

Consider routing $\mathbf{D}$ with $\mathbf{f}$ on $G \setminus \{k\}$ using the tailored bypass $\mathbf{b}^{\mathbf{D}}$. From the definition of $\mathbf{b}^{\mathbf{D}}$ we obtain that for each edge $e$, the contribution of the bypass flow to the congestion on the edge $e$ is at most the sum of the flow on the reverse edge of $e$ when routing $\mathbf{D}$ using $\mathbf{f}$ on $G$ and the flow on $e$ when routing $\mathbf{D}$ using $\hat{\mathbf{f}}$ on $G \setminus \{k\}$. The maximum utilization of a link when using the bypass $\mathbf{b}^{\mathbf{D}}$ is thus at most the maximum utilization under $\mathbf{f}$ plus the maximum contribution to the utilization under the bypass flow. Since links are symmetric, the maximum contribution to utilization of the bypass flow is at most the sum of the maximum utilizations under $\mathbf{f}$ on $G$ and $\hat{\mathbf{f}}$ on $G \setminus k$. Formally we have

$$\text{MLUB}_{G\setminus\{k\}}(\mathbf{b}^{st}, \mathbf{f}, \{\mathbf{D}\}) \leq \max_{(i,j)\in\text{links}\setminus\{k\}} \frac{\sum_{a,b} d_{ab} f_{ab}(i,j)}{\text{cap}_{ij}} + \max_{(i,j)\in\text{links}\setminus\{k\}} \frac{\sum_{a,b} d_{ab} b^{st}(i,j) f_{ab}(k)}{\text{cap}_{ij}}$$

$$\leq \text{MLU}_G(\mathbf{f}, \{\mathbf{D}\}) + \max_{(i,j)\in\text{links}\setminus\{k\}} \frac{\sum_{a,b} d_{ab} b^{\mathbf{D}}(i,j) f_{ab}(k)}{\text{cap}_{ij}}$$

$$\leq \text{MLU}_G(\mathbf{f}, \{\mathbf{D}\}) + \max_{(i,j)\in\text{links}\setminus\{k\}} \frac{\sum_{a,b} d_{ab}(f_{ab}(i,j) + \hat{f}_{ab}(i,j))}{\text{cap}_{ij}}$$

$$\leq 2\text{MLU}_G(\mathbf{f}, \{\mathbf{D}\}) + \text{MLU}_{G\setminus\{k\}}(\hat{\mathbf{f}}, \{\mathbf{D}\}) .$$

Dividing both sides by $\text{OPTU}_{G\setminus\{k\}}(\mathbf{D})$ and using the fact that $\text{OPTU}_{G\setminus\{k\}}(\mathbf{D}) \geq \text{OPTU}_G(\mathbf{D})$. We obtain

$$\text{PERFB}_{G\setminus\{k\}}(\mathbf{b}^{st}, \mathbf{f}, \{\mathbf{D}\}) \leq 2\text{PERF}_G(\mathbf{f}, \{\mathbf{D}\}) + \text{PERF}_{G\setminus\{k\}}(\hat{\mathbf{f}}, \{\mathbf{D}\})$$

and therefore

$$\text{PERFB}_{G\setminus\{k\}}(\mathbf{b}^{st}, \mathbf{f}, D) \leq 2\text{PERF}_G(\mathbf{f}, D) + \text{PERF}_{G\setminus\{k\}}(\hat{\mathbf{f}}, D) .$$

☐

## 8. CONCLUSION

Traffic engineering is becoming increasingly important as network operators strive to utilize network resources efficiently and to provide good performance to their end users. Routing is one component of traffic engineering, and a key challenge for routing is to adapt to changes in network topology. In this paper, we investigated optimal demand oblivious route restoration. These restoration strategies are able to compute new end-to-end routes following a failure, and they perform well even when the traffic matrix is uncertain. We explored a fundamental tradeoff between performance and complexity, and found that a strategy which only reroutes traffic flows directly affected by link or node failures can perform nearly as well as one that recomputes all routes, while reducing the disruption on unaffected traffic. Thus, the impact of the restoration process can be limited to a subset of flow paths without significantly sacrificing performance. The algorithms and techniques presented here, as well as the results of our evaluation, should aid network designers as they consider strategies for routing in the face of network failures.

## 9. REFERENCES


[1] R. K. Ahuja, T. L. Magnanti, and J. B. Orlin. *Network flows*. Prentice-Hall, Inc., Englewood Cliffs, NJ, 1993.

[2] J. Anderson, B. T. Doshi, S. Dravida, and P. Harshavardhana. Fast restoration of ATM networks. *IEEE journal on selected areas in communications*, 12:128–138, 1994.

[3] D. Applegate and E. Cohen. Making routing robust to changing traffic demands: Algorithms and evaluation. Manuscript.

[4] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: understanding fundamental tradeoffs. In *Proceedings of the ACM SIGCOMM'03 Conference*, 2003.

[5] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. RFC 2702: Requirements for Traffic Engineering over MPLS, September 1999.

[6] D. Awduche, A. Chiu, A. Elwalid, I. Widjaja, and X. Xiao. RFC 3272: Overview and Principles of Internet Traffic Engineering, May 2002.

[7] S. Bhattacharya, C. Diot, J. Jetcheva, and N. Taft. Geographical and temporal characteristics of inter-POP flows: view from a single POP. In *European transactions on telecommunications*, 2002.

[8] J. Cao, D. Davis, S. V. Wiel, and B. Yu. Time-varying network tomography: router link data. *J. Amer. Statist. Assoc.*, 95:1063–1075, 2000.

[9] N. G. Duffield and M. Grossglauser. Trajectory sampling for direct traffic observation. *IEEE/ACM Transactions on Networking*, 9:280–292, 2001.

[10] A. Feldmann, A. Greenberg, C. Lund, N. Reingold, J. Rexford, and F. True. Deriving traffic demands for operational IP networks: methodology and experience. *IEEE/ACM Transactions on Networking*, 9:265–279, 2001.

[11] B. Fortz and M. Thorup. Internet traffic engineering by optimizing OSPF weights. In *Proceedings of INFOCOM*, pages 519–528. IEEE, 2000.

[12] B. Fortz and M. Thorup. Optimizing OSPF/IS-IS weights in a changing world. *IEEE journal on selected areas in communications*, 20(4), 2002.

[13] B. Grötschel, L. Lovasz, and A. Schrijver. *Geometric algorithms and combinatorial optimization*. Springer-Verlag, New York, 1988.

[14] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson. Inferring link weights using end-to-end measurements. In *Proceedings of the 2nd Internet Measurement Workshop*. ACM, 2002.

[15] A. Medina, N. Taft, K. Salamatian, S. Bhattacharyya, and C. Diot. Traffic matrix estimation: Existing techniques and new directions. In *Proceedings of the ACM SIGCOMM'02 Conference*. ACM, 2002.

[16] D. Mitra and K. G. Ramakrishna. A case study of multiservice, multipriority traffic engineering design for data networks. In *Proceedings of IEEE GLOBECOM*, pages 1077–1083. IEEE, 1999.

[17] J. Moy. RFC 1583: OSPF version 2, 1994.

[18] K. Murukami and H. Kim. Optimal capacity and flow assignment for self-healing ATM networks bsed on line and end-to-end restoration. *IEEE/ACM Transactions on Networking*, 6(2):207–221, 1998.

[19] E. C. Rosen, A. Viswanathan, and R. Callon. RFC 3031: Multi Protocol Label Switching architectures, 2001.

[20] M. Roughan, A. Greenberg, C. Kalmanek, M. Rumsewicz, J. Yates, and Y. Zhang. Experience in measuring backbone traffic variability: models, metrics, measurements, and meaning. In *Proceedings of the 2nd Internet Measurement Workshop*. ACM, 2002.

[21] Internet Traffic Engineering Working Group, 2003. `http://www.ietf.org/html.charters/tewg-charter.html`.

[22] Y. Xiong and L. G. Mason. Restoration strategies and spare capacity requirements in self healing ATM networks. *IEEE/ACM Transactions on Networking*, 7(1):98–110, 1999.