Reliability as an Interdomain Service

Hao Wang* Yang Richard Yang* Paul H. Liu*
Jia Wang† Alexandre Gerber† Albert Greenberg§
AT&T Labs – Research† Microsoft Research§ Yale University*

ABSTRACT

Reliability is a critical requirement of the Internet. The availability and resilience of the Internet under failures can have significant global effects. However, in the current Internet routing architecture, achieving the high level of reliability demanded by many mission-critical activities can be costly. In this paper, we first propose a novel solution framework called reliability as an interdomain service (REIN) that can be incrementally deployed in the Internet and may improve the redundancy of IP networks at low cost. We then present robust algorithms to efficiently utilize network redundancy to improve reliability. We use real IP network topologies and traffic traces to demonstrate the effectiveness of our framework and algorithms.

Categories and Subject Descriptors: C.2.1 [Computer Communication Networks]: Network Architecture and Design – *Network communications*; F.2.2 [Analysis of Algorithms and Problem Complexity]: Nonnumerical Algorithms and Problems

General Terms: Algorithms, Performance, Design.

Keywords: Reliability, Traffic Engineering, Fast Rerouting

1. INTRODUCTION

Reliability is a major concern on the Internet. As it becomes a critical infrastructural component of the global information-based society, the availability and resilience of the Internet under failures can have significant global and social effects. Recognizing the importance of Internet reliability, the GENI initiative [16] states that "any future Internet should attain the highest possible level of availability, so that it can be used for *mission-critical* activities, and it can serve the nation in times of crisis." The Internet service providers (ISPs) also recognize the importance of reliability. In a recent survey of major network carriers including AT&T, BT, and NTT, Telemark [40] concluded that "The 3 elements which carriers are most concerned about when deploying communication services are network reliability, network usability and network fault processing capabilities. The top 3 elements all belong to the reliability category."

Unfortunately, due to accidents, maintenance mistakes, natural disasters, and even malicious attacks, failures are part of the everyday life of an IP network (*e.g.*, [19, 32]). Cable cuts, the most common physical-layer failures, occur virtually every few days in extensive networks, despite many measures for the physical protection of cables [19].

One of the major goals of network engineers and operators is to eliminate or minimize the impacts of failures on customers. Indeed, IP network owners have clear incentives to manage these failures well, because unmanaged failures can cause severe service disruptions and lead to significant financial and reputation damages.

However, achieving high reliability by tolerating failures well generally requires significant investments. A major challenge is obtaining *redundancy* at a reasonable cost. By redundancy, we mean both the diversity of physical connectivity and the over-provisioning of bandwidth to carry traffic originally passing through any failed equipment. Optical layer protection on a SONET ring provides redundancy, but at a cost that limits its deployment [17]. Therefore, IP networks rely also on IP routing to detect and reroute traffic around failures. However, IP rerouting still depends on available redundancy. With the cost of over-provisioning and, in particular, the expenses to obtain rights of way to install alternative paths (*e.g.*, along natural gas pipelines, highways or railways), many IP networks, in particular ISP networks, face the challenge of adding redundancy in a cost-effective way to stay competitive in the highly competitive ISP market.

In this paper, we propose a novel framework called reliability as an interdomain service (REIN) to increase the redundancy (*i.e.*, physical diversity and bandwidth) available to an IP network at low cost, thereby increasing the failure processing capability of IP networks. Specifically, previous studies consider redundancy only in a single network, which is limited by its budget and rights of way. Our key observation is that there can be redundancy across multiple networks. For example, large IP networks in the US cover the same geographic regions and place their routers at similar sites (*e.g.*, major cities). If we overlay two networks, for two sites in both networks, when one network does not have direct links between these two sites, the other network may have. Even when both networks have direct links between these two sites, the links may be placed at different locations (*e.g.*, one along highway and the other along railway). Thus, when there is a failure inside one network, the other network can provide redundancy. By designing a framework that allows neighboring networks to use the resources of each other as backup, we may be able to improve network reliability at low social and network cost. This may be more beneficial to smaller networks with limited resources than to larger networks with more resources to manage physical diversity and capacity internally.

Our idea of pooling the resources of multiple networks together for mutual backup provides a totally new avenue towards improving network reliability at low cost. A similar idea is already being practiced in some other contexts. For example, multiple participants contribute to a common pool of resources to form a shared insurance. This can be far more affordable than individual provisioning. The airline industry also uses a similar idea. When an

SIGCOMM'07, August 27–31, 2007, Kyoto, Japan.

airline's aircraft for some flight segment is not available due to mechanical problems or late arrival, the airline may use other airlines' flights to provide transport for its customers. This is far cheaper than each airline having a large number of reserved aircrafts. Such sharing of resources improves the reliability of individual carriers, reduces the backup costs of individual carriers, and also reduces social cost (*i.e.*, less waste of overall resources).

We caution that the benefits of our scheme depend on several practical factors. In particular, it depends on the agreement between neighboring networks to participate in the scheme and share more information than they currently do. For instance, for best performance, REIN would require cross-provider shared risk link group (SRLG) data which may not be readily available. Also, the potential physical diversity improvement using the scheme may be limited among a group of networks, such as those that have their fibers laid in the same conduit. In this case, a single event such as an excavation accident could impact all networks in the group. These practical factors must be taken into account while weighing the cost and benefits of the scheme.

Complementing the REIN framework, we design algorithms to efficiently utilize both the intradomain and the interdomain redundancies to improve network reliability. It is our observation that, even if a network has enough redundancy to handle a failure scenario under optimal usage of the redundancy, the implemented algorithm may not use the redundancy efficiently enough, leading to reduced reliability. For example, the widely implemented fast rerouting (FRR) algorithm uses only the shortest paths for rerouting. This limited form of rerouting may not efficiently utilize all available network redundancy [22]. For example, in our evaluations, we show that there are single-link failure scenarios in the Abilene network where shortest-path-based rerouting would require a bandwidth nearly 300% of actual link capacity at a bottleneck link, while the bandwidth requirement of an optimal rerouting algorithm is far below actual link capacity. Although TCP congestion control or priority-based schemes could alleviate the impact, overloaded links can cause long delays, high packet loss rates, reduced network throughput (*e.g.*, TCP flows), BGP session resets, and/or even router crashes.

Our algorithms address two practical challenges. First, although various optimal rerouting algorithms have been studied before (*e.g.*, [5, 26]), they are typically based on *flow-based* routing, which is not readily implementable in the Internet. We introduce the general notion of the coverage of a set of paths, and devise a general technique to convert a flow-based routing to a path-based routing that is implementable in the Internet. This technique can be applied in a broader context, such as traffic engineering (TE) in general. Second, as virtual private networks (VPNs) become a fast growing segment of IP business and a major source of revenue, there is growing practical need for integrating VPNs. Although there are previous studies on bandwidth provisioning for VPNs (*e.g.*, [10, 12, 30]), we present the first formulation and algorithms on integrating traffic engineering and failure protection with VPNs.

We demonstrate the effectiveness of our framework and algorithms using real IP network topologies and traffic traces. We show that network reliability may be substantially improved by using a small number of interdomain bypass paths. For example, we evaluate the number of interdomain bypass paths required to reduce the percentage of links with low *link connectivity* (*i.e.*, less than or equal to 2) in a network. We show that we can reduce this percentage to 0 for Abilene using only 3 paths, and from 29% to 8% for Sprint using only 5 paths, assuming that these interdomain bypass paths are not part of any intradomain shared risk link groups. We also evaluate the effectiveness of our framework and algorithms to reduce traffic intensity at bottleneck links during failure scenarios. For Abilene, we show that there exist failure scenarios where we can reduce the traffic intensity on the bottleneck by 270% of the bottleneck link capacity, when Abilene uses the COPE traffic engineering technique [42] as the basis. For a major IP network, we show that there exist failure scenarios where we can reduce the normalized traffic intensity on the bottleneck link by 118% and 35% of the bottleneck link capacity, when the network uses oblivious routing [6] and COPE as basis, respectively. The reduction is more substantial if the networks use standard shortest path routing. Furthermore, the impacts of the diverted traffic on the neighboring IP networks that provide the bypass paths are small.

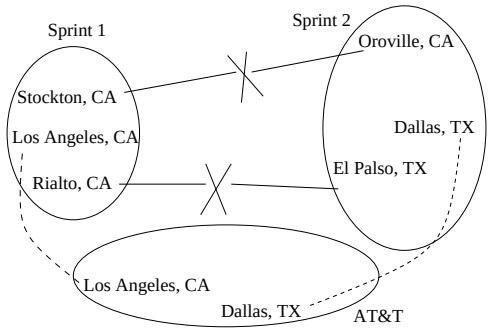


Figure 1: Illustration of using interdomain bypass for a partitioned Sprint backbone.

The rest of this paper is organized as follows. Section 2 presents the overall architecture of REIN. Next, we present our algorithms for integrated traffic engineering, VPN provisioning, and fast rerouting in Section 3 and our algorithms for selecting interdomain bypass paths in Section 4. In Section 5, we present evaluation results. In Section 6, we discuss related work. Our conclusions and future work are in Section 7.

2. ARCHITECTURE

In this section, we outline the basic system architecture of REIN. It is further specified in the following sections.

2.1 Overview

An IP network needs to be protected against failures both inside and outside of the network. In this paper, we focus on protecting intradomain links and directly connected interdomain (peering) links.

To motivate and illustrate the basic idea of the REIN architecture, we consider one recent event [43] on January 9, 2006, when the Sprint backbone network, a major US backbone network, was partitioned into two disconnected components by two fiber cuts. Figure 1 illustrates the event. This event led to the disconnection of long-distance service for millions of Sprint PCS and Nextel wireless customers west of the Rockies, network partitions for corporations that relied on the carrier to link office networks, and substantially decreased throughput of transcontinental Internet traffic routed over Sprint.

The key observation of REIN is that under network partitioning, it is possible to route traffic between disconnected components through neighboring IP networks. We refer to such routes through neighboring IP networks as *interdomain bypass paths*. In the preceding example, both of the two disconnected components of the Sprint network have peers connected to AT&T. For instance, Figure 1 shows one peering between Sprint and AT&T at Los Angeles, and another peering of the two networks at Dallas. If Sprint were able to use AT&T's network as a backup, it could be connected and thus have greatly reduced the impacts of the partitioning.

The Sprint incident is one example of network partitioning. In Section 5, we have evaluated the connectivity of many IP networks and found, in each case, that there are backbone links that have connectivity less than or equal to two (as high as 60% of links in some smaller networks that are more vulnerable). These networks may improve their connectivity through REIN.

REIN is also useful when an IP network is not partitioned, but nevertheless does not have enough redundant bandwidth to reroute traffic around failures. Such a network can benefit from the additional bandwidth made available through REIN. For example, in our evaluations, we identify failure scenarios in the Abilene network where, when two links are down, a single link becomes a bottleneck and the total traffic demand on the link could be almost 3 times its capacity even under optimal rerouting. However, using REIN, the network can handle the failure scenarios without overloading any links.

To realize the benefits of REIN, we need to address the following practical issues:

- Why would IP networks share interdomain bypass paths?
- What is the signaling protocol to share these paths?
- How can the imported interdomain bypass paths be used in the data path?
- After an IP network imports these paths, how does it effectively utilize them in improving reliability?

In the next three subsections we discuss the first three issues respectively. We address the fourth issue in Sections 3 and 4.

2.2 Business Models

Similar to traditional Internet interdomain business relationships, there can be multiple business models for the sharing of interdomain bypass paths. We are particularly interested in models that involve only two neighboring IP networks, because the two networks can reach an agreement and provision their peering links to support the agreed model independent of other IP networks. Since there is no need for global coordination, such models allow incremental deployment in the Internet.

- The first is a *peering* model where networks *A* and *B* provide mutual backup without financial settlement. This model improves the reliability of both networks at low cost, and thus provides both networks with incentives. Similar to the traditional Internet peering relationship which depends on symmetry in traffic, there should be some enforcement of symmetry in bypass path capacity provisioning and usage. A potential advantage of mutual backup through peering is that the two networks involved tend to have similar geographic coverage and thus the bypass paths are less likely to have long detour delay.
- The second is a *cost-free* model without the requirement for symmetry. For example, the backbone of the educational Abilene network overlaps with many commercial IP networks. Although in typical cases Abilene will not carry any commercial traffic, it is possible that Abilene can provide interdomain bypass paths for commercial networks in emergencies, as these commercial networks are part of a critical national infrastructure.
- The third is a *provider-customer* model. This is similar to the traditional provider-customer relationship in the Internet; that is, network *A* will pay network *B* to provide bypass paths. The cost model can be either a fixed pricing model or a usage-based pricing model. The usage of the bypass paths (*e.g.*, in terms of amount of time and/or traffic volume) may be limited to avoid potential abuse. A bypass path provider might charge lower prices just as some ISPs charge lower prices for backup BGP links (*e.g.*, shadow links of UUNet).

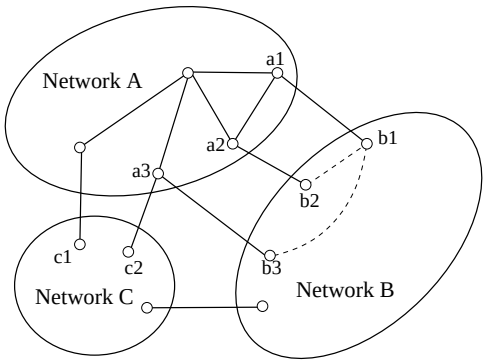


Figure 2: An example illustrating REIN interdomain bypass paths signaling.

2.3 Signaling Interdomain Bypass Paths

We consider the case where network *B* provides interdomain bypass paths to network *A*. We assume *A* peers with *B* at multiple locations referred to as points of presence (PoPs).

We emphasize that there can be multiple choices of protocols or mechanisms for *A* and *B* to signal the interdomain bypass paths. One possibility is manual configuration. Since, as we will show later, it typically needs only a small number of interdomain bypass paths between two neighboring networks, manual configuration is possible and may be the preferred configuration mechanism in many setups. Below we present signaling based on BGP as it does not require introducing a new protocol. In a cleaner signaling mechanism, each network could setup a dedicated REIN server, and the protocol we present below could be run over a TCP connection between the two REIN servers.

Figure 2 shows the networks used in the description. To discover interdomain bypass paths re-entering at border router $a1$ of network *A* through neighboring network *B*, $a1$ makes a special BGP announcement to its corresponding peer $b1$, over the existing eBGP session between $a1$ and $b1$. The destination address of the BGP announcement is $a1$. This BGP announcement can be considered as a request for bypass paths in *B* through $b1$ back to $a1$. The message could include additional attributes such as desired starting points of the bypass paths (*e.g.*, starting from $a2$ to *B* and then to $a1$) and desirable bandwidth. The additional attributes are carried as *opaque attributes* in the BGP message. The message carries a unique BGP community tag REIN_PATH_REQUEST to enable special treatment within each network.

This BGP announcement goes through standard BGP export/import policies and is imported into the routing information base of $b1$. Periodically, inside *B*, REIN extracts from border routers such request announcements using the tag REIN_PATH_REQUEST, and computes the interdomain bypass paths that it can provide, subject to its local policy. Note that one objective of the local policy is to mitigate the operational difficulties involved in the planning for carrying another network's traffic. For instance, *B*'s local policy could dictate that bypass paths are provided to *A* only through lightly-loaded links.

If *B* can provide bypass paths from some border router $b2$, REIN will configure $b2$ to announce a BGP update message carrying a unique BGP community tag REIN_PATH_AVAILABLE to its peer $a2$. The message from $b2$ to $a2$ is shown in Table 1.

The bypass path attribute in the REIN_PATH_AVAILABLE message does not include the complete router path inside *B*, to protect *B*'s private information. The exported values of bandwidth should be relatively stable to avoid frequent re-computation. Note that the bandwidths are allocated bandwidths instead of the total bandwidth of a bypass path. In addition, the bandwidth(s) may be constrained by the bandwidths of the peering links. However, since it can be cheaper to over-provision the bandwidth of a peering link than that

Destination:	$a1$
AS path:	the AS path traversed by this bypass path. It will be *AB*.
Bypass path:	$a2$, $b2$, opaque ID identifying the internal path inside B from b_2 to b_1, $b1$, $a1$.
Path metrics:	these include aggregated delay from $b2$ to $b1$, best-effort bandwidth $b2$ can provide to $b1$, and guaranteed bandwidth $b2$ can provide to $b1$, etc.
Shared risk link groups:	the IDs of the shared risk link groups involved in the path from $b2$ to $b1$.

Table 1: An example REIN_PATH_AVAILABLE message.

of a link connecting two faraway locations, this might be a lesser concern. The delay value will be used by network *A* when there is delay requirement. The path metrics may also include pricing information in a more flexible system.

The shared risk link groups (SRLGs) information needs coordination between the neighboring networks to assign consistent SRLG IDs to links or use a global information database. Two links belong to the same SRLG if they are considered to be likely to fail together. An example is two links that share some common conduit at some segment. A potential practical issue is that the networks may not be able to provide complete SRLG information. There can be several ways to address this issue. For example, some previous studies have shown that much of such information can be inferred using public records [18]. We emphasize that if this information is not provided, the benefits of REIN may be reduced. For example, a single event such as an excavation accident could impact multiple IP networks, damaging both interdomain bypass paths and intradomain links.

Periodically, inside *A*, using the tag REIN_PATH_AVAILABLE, REIN extracts interdomain bypass paths announced by neighboring networks. It then computes how to use these paths to improve reliability. For those paths that it chooses to use, it sends a BGP update message with a unique BGP community tag REIN_PATH_COMMIT to inform the neighbor. The neighbor then configures its data forwarding path to allow usage of the path (see below).

Note that we can extend this protocol to allow interdomain bypass paths to traverse more networks. For simplicity and also to avoid bypass paths with long delays, we focus on interdomain bypass paths that involve only one direct neighbor in this paper. In this case, BGP announcements for bypass paths are not propagated further, which avoids global BGP updates.

2.4 Data Forwarding Using Interdomain Paths

The main data-path capability needed by REIN is to allow traffic to leave and re-enter a network. This is not possible in the current Internet due to the separation of intradomain and interdomain routing. Specifically, a major problem is potential forwarding loops inside a neighboring network. Forwarding loops cannot arise in the hierarchical Internet routing, because that would imply a loop in AS paths. However, direct usage of interdomain bypass paths may cause forwarding loops. Consider the preceding example when the interdomain bypass path $a2 \rightarrow b2 \rightsquigarrow b1 \rightarrow a1$ is used. When $a2$ uses the bypass path, it encapsulates a packet using source address $a2$ and destination address $a1$, and sends the encapsulated packet to $b2$. However, a router inside *B* close to $b2$ may look up the destination address $a1$ and send the packet back to $b2$, causing a forwarding loop. There are several solutions to address this issue [36]. One is to use interdomain GMPLS to setup an interdomain label switched path (LSP) for the whole interdomain bypass path. As a second solution, $b2$ can configure an intradomain LSP from $b2$ to $b1$, and notify $a2$ about the LSP. Then $a2$ can use IP tunneling to forward packets to $b2$, where the tunnel header (*e.g.*, shim header) indicates that the LSP from $b2$ to $b1$ should be used.

3. OPTIMAL FAST RE-ROUTING

The preceding section has presented our overall architecture. The interdomain bypass paths can be utilized in multiple ways. In this section, we present a fast rerouting algorithm to efficiently utilize these paths. Note that our algorithm applies both with and without interdomain bypass paths. For simplicity, we also refer to such interdomain bypass paths as interdomain bypass links or virtual links. The coverage-based path generation technique developed in this section is also a general tool that can be used to implement other traffic engineering related algorithms.

3.1 Design Decisions

Network operators configure their networks for different objectives. We study one specific formulation for routing under failures.

We make the following design decisions. First, we consider a link failure as the basic failure unit, as we observe more frequent link failures in network failure logs; previous studies have also identified cable damage as the most common type of physical-layer failure [19]. Furthermore, it generally takes longer to fix a link failure than a node failure since it requires sending a crew to the field. Also, a node failure can be treaded as a failure scenario involving multiple links.

Second, we implement protection, which pre-computes rerouting paths to use upon failure detection, instead of restoration, which depends on routing re-convergence. There are two basic protection mechanisms [21]: *link protection* (*i.e., fast rerouting*), and *path protection*. In fast rerouting, a detour around the failed link is created. In path protection, the sources of all flows using the failed link are notified and detour to avoid the failed link. An advantage of path protection is that, since detours are computed for each source, it can avoid potential bottlenecks around the head end of the failed link, and thus achieve better rerouting performance. In this paper, we focus on link protection since it has faster response time and better scalability. Our scheme extends naturally to path protection or a hybrid of the two.

We compute fast rerouting for only a subset of all failure scenarios, as the total number of failure scenarios is exponential. We refer to the scenarios for which we compute fast rerouting as the *high-priority failure scenarios*. In practice, these scenarios are determined by the operator of an IP network. In our evaluations, we use single- or two-link failure scenarios. Specifically, each high-priority failure scenario consists of one or more SRLGs, where an SRLG is a set of links that are likely to fail simultaneously. When a high-priority failure scenario consists of a single SRLG and this SRLG appears alone in the high-priority scenarios, the head-end router of each link in the SRLG can detect the failure scenario locally and switch to the bypass paths. To improve response time for critical failure scenarios (*e.g.*, failure of two SRLGs that can cause a major network partitioning as in the Sprint example), we generalize fast rerouting to allow a failure scenario to consist of more than one SRLG. Then, a detection mechanism is needed to detect the exact failure scenario, and the response time may be slower than that of the single SRLG case, but is still faster than that of a failure scenario outside of the high-priority set, in which the recovery depends on re-computation by the routing protocol or the traffic engineering component. In the rest of this section we focus on handling the high-priority failure scenarios.

3.2 Algorithm Overview

Given our design decisions, our algorithm naturally consists of two steps. In the first step, we compute optimal routing using traffic engineering when there are no failures. In the second step, we compute fast rerouting for high-priority failure scenarios on top of traffic engineering. When we compute fast rerouting, we distinguish important traffic (*e.g.*, voice and VPN) and select intradomain links, if possible, to protect such traffic.

A common issue in both steps is how to handle implementation feasibility and computational complexity. In terms of implementation feasibility, since about half of the ISPs have already deployed

MPLS in their core [23] and more ISPs are following suit, we seek to implement optimal traffic engineering and fast rerouting using IP/MPLS. However, computation of optimal traffic engineering and fast rerouting directly using *path-based routing* (*i.e.*, routing specified by how traffic is split among LSPs) can be intractable, since there can be exponential number of candidate LSPs between each origin-destination (OD) pair. There are previous studies considering selecting LSPs. For example, one previous proposal is to select K-shortest paths (*e.g.*, [23]). However, this method does not consider traffic dynamics in the network and can result in poor path selection.

On the other hand, it is computationally effective to use a representation called *flow-based routing*, in which the routing is specified at each link by the fraction of traffic of each OD pair that is routed on this link. However, flow-based routing is fundamentally different from MPLS routing, and is not readily implementable in the Internet.

Our methodology is a two-phase process. In the first phase, we use the flow-based routing representation to make computation tractable. In the second phase, we use a *path generation* technique to convert a flow-based routing into practical implementation. In Section 3.3, we present traffic engineering and fast rerouting with VPNs using flow-based routing. The path generation phase will be presented in Section 3.4.

3.3 Integrated TE/FRR with VPNs

We first present our formulation and algorithm for integrated TE/FRR with VPNs using flow-based routing.

3.3.1 Robust TE with VPN Support

The IP network first conducts traffic engineering to determine base routing without failures. The uncertainty to handle in this case is traffic volume variations. We base our TE formulation using either the traditional oblivious routing technique developed by Applegate and Cohen [6], or the COPE technique developed by Wang *et al.* [42]. We extend their techniques to address VPN support.

We represent a network by a graph $G = (V, E \cup E')$, where V is the set of routers, E is the set of intradomain links, and E' is the set of interdomain bypass links. The capacity of link l is denoted by $cap(l)$.

Let $\mathcal{X}$ denote the set of all possible traffic demand matrices. Each traffic demand matrix is a set of traffic demands $d = \{d_{ab} | a, b \in V\}$, where d_{ab} is the traffic demand from a to b. For traffic with destination outside the network, we always use the technique in [42] to convert interdomain traffic demand to intradomain traffic demand.

Denote by $o(f, d)$ the performance of flow-based routing f under traffic demand matrix $d \in \mathcal{X}$, where the *flow-based routing* f is specified by a set of values $f = \{f_{ab}(i, j) | a, b \in V, (i, j) \in E\}$ and $f_{ab}(i, j)$ specifies the fraction of demand from a to b that is routed over the link (i, j). Note that this formulation assumes all traffic demand will be routed by traffic engineering. We have extended to the case that most OD pairs are routed using a default routing (*e.g.*, OSPF/ISIS), and only selected, major OD pairs (*e.g.*, heavy hitters [28, 35]) are involved in defining f. Furthermore we may need to aggregate routers inside a PoP for scalability. For clarity, we omit these extensions. Let $o(f, \mathcal{D})$ be the aggregated performance of routing f on the set $\mathcal{D}$, where $\mathcal{D} \subset \mathcal{X}$ is the set of common-case traffic demands. The aggregation can be done, for example, by taking the maximum, or a weighted average.

Let $c(f, d)$ be the penalty (cost) of routing f under traffic demand d. Then the objective of the basic robust TE problem is to search for a base routing f that optimizes $o(f, \mathcal{D})$, subject to a worst-case penalty bound $\bar{r}$ on $c(f, d)$ for all $d \in \mathcal{X}$.

As VPNs are particularly important to ISPs, we add additional constraints to the preceding robust TE problem formulation. We use the popular hose model [10, 30] to specify VPN demand. For each source (or destination) $a \in V$, we denote by $ECR(a)$ (resp. $ICR(a)$) the total egress (resp. ingress) committed rate, which is the guaranteed total demand to (resp. from) all other nodes inside the network for VPNs. Then the additional constraints guarantee bandwidth provisioning for VPNs. Specifically, they require that the base routing f be able to route, without overloading any intradomain link $l \in E$, an arbitrary VPN traffic demand matrix d^w that conforms to the *ECR* and *ICR* specification.

3.3.2 Robust Fast Rerouting

Let the routing computed by the preceding formulation be f^*. The network proceeds to compute fast rerouting f^h on top of f^*, to protect against each high-priority link failure scenario h, where $h \subset E$ represents the failure of a set of links belonging to one or more SRLGs. The fast rerouting computation would use not only intradomain links in E, but also interdomain bypass links in E'. To be robust to traffic variations when a failure scenario happens, we compute fast rerouting that minimizes the oblivious ratio on all possible total traffic demands [5].

Due to the high priority and sensitivity of VPN traffic, we compute separate fast reroutings, $f^{h,B}$ for best-effort traffic and $f^{h,V}$ for VPN traffic, with the requirement that all VPN traffic be completely rerouted using intradomain links only. However, this requirement may not be satisfiable (*e.g.*, under network partitioning). When this happens, we simply compute a common fast rerouting, f^h, for both best-effort and VPN traffic. The detailed formulation and algorithm are presented in Appendix.

3.3.3 Extensions to Handle Peering Link Failures

We can extend our algorithm to better handle directly connected interdomain peering links and take advantage of the point to multipoint flexibility for interdomain traffic. This appears both in the normal routing case and in the fast rerouting case. For the fast rerouting case, when an intradomain link i to j fails, the detour is a flow from i to j. As a contrast, for an interdomain link from i to a neighboring network B, it can use multiple peering points at B: b_1, b_2, ..., b_B, where the b's are border gateway routers between A and B. We can thus compute multiple flows $(i \to b_1), (i \to b_2), \ldots, (i \to b_B)$. We can further extend to allow multiple egress networks.

3.4 Path Generation Based on Flow Routing

The algorithm that we have just developed computes base routing and fast rerouting using linear programming and generate flow-based routing. However, a flow-based routing is not readily implementable in the current Internet. We now provide a general technique to convert a flow-based routing to a path-based routing with bounded performance penalty.

Clearly, there is a tradeoff between the number of paths and the performance we can achieve. Using flow decomposition [2], we can convert any flow-based routing to a path-based routing using up to $|E|$ paths per OD pair. However, in an IP network, $|E|$ could be large. One way to control the number of paths is the path selection approach proposed by Li *et al.* [31], which adds penalty terms in the optimization objective to implicitly limit the path diversity and thus the number of paths. However, this approach is computationally expensive and provides no explicit control on either the number of paths or the performance degradation. There are approaches to incrementally generate paths, but they are based on the specific optimization problem [37].

Our approach, based on the notion of the coverage of a set of paths, does not depend on how the flow-based routing is derived and selects effective paths guided by the given flow-based routing. The algorithm explicitly considers the tradeoff between the number of paths and the performance gain, and enables one to choose the paths based on preferences between performance and scalability.

Below, we first formalize our notion of selecting effective paths to approximate a flow-based routing. We then propose an algorithm to carry out this approximation. Our algorithm has two configurable parameters, with different effects on performance and scalability.

3.4.1 Q-coverage Path Set

We first define the concept of coverage of a set of paths.

Consider a flow-based routing $f = \{f_{ab}(i,j) | a,b \in V, (i,j) \in E\}$. For each OD pair $a \to b$, we construct a graph where each edge (i,j) has a capacity of $f_{ab}(i,j)$. Without loss of generality, we assume that all cycles in f have already been removed, and thus the graph is a directed acyclic graph (DAG).

Let $P_{ab} = \{P_{ab}^k | k = 1,\ldots,K\}$ be a given set of K paths from a to b. A *path-based routing* over P_{ab} specifies the fraction of traffic to be carried by each path in P_{ab}. Specifically, a path-based routing over P_{ab} can be represented by a vector $p_{ab} = \{p_{ab}^k > 0 | k = 1,\ldots,K\}$, where p_{ab}^k denotes the fraction of demand from a to b that is routed on path P_{ab}^k. The *value* of p_{ab}, denoted by $|p_{ab}|$, is defined as

$$|p_{ab}| = \sum_{k=1}^{K} p_{ab}^k. \quad (1)$$

A path-based routing p_{ab} is *valid* if its value is 1.

DEFINITION 1. *A set P_{ab} of paths from a to b is a* Q-percentage coverage path set *(or* Q-percentage path set *for short) for flow-based routing f_{ab} if there exists a path-based routing p_{ab} over P_{ab} that satisfies the following two conditions:*

$$|p_{ab}| = Q; \quad (2)$$

$$\sum_{k:(i,j)\in P_{ab}^k} p_{ab}^k \le f_{ab}(i,j), \forall (i,j) \in E. \quad (3)$$

Moreover, a set $P = \cup_{a,b\in V} P_{ab}$ is called a Q-percentage coverage path set *for flow-based routing f if, for each OD pair $a \to b$, P_{ab} is a Q-percentage path set of f_{ab}.*

With the coverage of a set of paths, we can measure how well a set of paths approximate a given flow-based routing. This can be stated formally as the following lemma:

LEMMA 1. *Given a flow-based routing f and a Q-percentage path set P for f, a* valid *path-based routing $p = \{p_{ab} | a,b \in V\}$ over P can be constructed such that for any demand d, the routed traffic on any link $l \in E$ under p is upper bounded by $1/Q$ of the routed traffic on l under f.*

PROOF. Please see Appendix. □

In general, consider any network performance metric m which is a function of $|E|+1$ variables: the utilization u_l of link $l \in E$, and a function $z(d)$ of a traffic demand matrix d; that is, $m = m(u_1,u_2,\ldots,u_{|E|}; z(d))$. Here, $z(d)$ can be any function, as long as it depends only on d. One example $z(d)$ is the optimal link utilization of the network under d. If m is monotonic increasing with respect to $u_l (l \in E)$, we have

PROPOSITION 1. *Given a flow-based routing f and a Q-percentage path set P for f, a* valid *path-based routing p over P can be constructed such that for any demand d, the performance metric m under p is upper bounded by $m(1/Q \cdot u_1,\ldots,1/Q \cdot u_{|E|}; z(d))$, where u_l is the utilization of link l under f.*

For example, assume that $m(u_1,u_2,...,u_{|E|}; z(d)) \triangleq \max_{l\in E} u_l$, which is a popular TE performance metric referred to as the *bottleneck traffic intensity* or *maximum link utilization* (MLU). Then the constructed valid path-based routing p guarantees that, for any demand d, its bottleneck traffic intensity is at most $1/Q$ times that of the original flow-based routing f.

3.4.2 Path Generation for a Flow-based Routing

With the notion of the coverage of a path set, we can present our algorithm for finding a small number of paths P guided by a flow-based routing f. The algorithm to generate paths P_{ab} from a to b based on f_{ab} is presented in Figure 3. To generate the complete path set P, the same algorithm is repeated for each OD pair.

```
1. construct a DAG from flow-based routing f_ab
2. i ← 1, P_ab ← ∅
3. while (termination condition not met)
4.    compute max unsplittable flow p_ab^i satisfying SLA
6.    add flow path P_ab^i to path set P_ab
7.    deduct flow rate p_ab^i from link capacities along P_ab^i
8.    remove links with capacity 0 from DAG
9.    i ← i + 1
```

Figure 3: The coverage-based algorithm for generating paths P_{ab} from a to b based on flow-based routing f_{ab}.

There can be two approaches to the termination condition. The first is to generate no more than a fixed number, K, of paths per OD pair. We call such an approach K-path coverage. A network may adopt this approach if it knows the maximum number of paths it wants to select for any OD pair. The network can then evaluate the performance of the selected path set by computing its coverage. The second approach terminates only after a certain coverage is achieved for every OD pair, and can thus bound the performance. We call this approach *Q-percentage coverage*.

A key step of the algorithm is line 4: to compute the maximal unsplittable flow between a and b that satisfies the service level agreement (SLA) delay constraint. This can be done in polynomial time based on the observation that the link with the lowest capacity on the maximal unsplittable flow path should be saturated. Specifically, we partition links according to their capacities. For a certain capacity value C, we construct a subgraph by removing all links with capacity less than C. We compute the lowest delay path from source a to destination b in this subgraph. If the delay of the computed path satisfies the SLA delay requirement, we have identified that there is a unsplittable flow satisfying the SLA constraint with flow rate at least C. Therefore, we can conduct a binary search over all capacity values to identify the maximum unsplittable flow rate . Given this algorithm, at line 8, we can remove at least one link in the network. Thus, in the worst case, the path set will consist of $|E|$ paths. However, as we will see in Section 5, we typically need a much smaller number of paths than $|E|$.

4. INTERDOMAIN BYPASS SELECTION

The preceding section assumes interdomain bypass paths to be used are already chosen. In this section, we address the issue that an IP network may receive many interdomain bypass paths and selectively use a subset of these paths. This can reduce configuration overhead and/or cost for bypass paths with non-zero cost.

4.1 Overview

We select interdomain bypass paths in two steps. In the first step, we select interdomain bypass paths to improve the physical connectivity of the network. In the second step, we augment this selection with additional interdomain bypass paths to improve the performance of optimal fast rerouting for high priority failure scenarios.

4.2 Bypass Selection for Connectivity

We first select interdomain bypass paths such that the link connectivities of all intradomain links are above a certain level (*e.g.*,

greater than 2 or 3). Formally, the *link connectivity* of a link is defined as follows.

DEFINITION 2 (LINK CONNECTIVITY). *The link connectivity of a link is the minimal number of links (including the link itself) that must be removed in order to disconnect the two endpoints of this link.*

For any link $l \in E$, we denote by $\mathrm{EC}(l)$ the link connectivity of l. EC is referred to as the link connectivity function.

Since each interdomain bypass path has associated (allocated) bandwidth(s) and aggregated delay, we first prune those bypass paths with low bandwidths and long delays. The thresholds used in this pruning process should depend on the SLA requirements of the IP network. Among the interdomain bypass paths that survive the pruning, we select a subset that minimizes the total cost while achieving the target connectivities.

This selection problem can be stated formally as follows. Given

- a *multigraph* $G = (V, E)$ that represents the network, similar to that defined in Section 3.3, except that G may contain parallel links due to the existence of multiple physical links between some pair of nodes;
- a set BYPASS of interdomain bypass links, each of which represents a different available interdomain bypass path. For a link $l \in \mathrm{BYPASS}$, $\mathrm{cost}(l)$ denotes the cost of using the corresponding interdomain bypass path. Note that there may be parallel links in BYPASS, because there may be multiple interdomain bypass paths between the same pair of intradomain nodes from multiple neighboring networks.
- a link connectivity requirement function *req* for a selected (low connectivity) link set $L \subseteq E$;

our goal is to choose a subset $E' \subseteq \mathrm{BYPASS}$ such that, in the augmented graph $G' = (V, E \cup E')$, the link connectivity $\mathrm{EC}_{G'}(l) \geq req(l), \forall l \in L$, and the total cost, as defined by $\mathrm{cost}(E') = \sum_{l \in E'} \mathrm{cost}(l)$ is minimized.

Several comments on this problem statement follow.

- Not all link connectivities can be improved even when one considers the entire set of available bypass paths, *i.e.*, when $E' = \mathrm{BYPASS}$. One must choose a realistic function *req* for a realistic set of links $L \in E$. For instance, *req* can be systematically generated according to the priorities of links whose connectivities need improvement. For each link $l \in E$, the feasibility of $req(l)$ can be checked by computing the link connectivity of l in $\widetilde{G} = (V, E \cup \mathrm{BYPASS})$.
- In addition to minimizing total cost, one may wish to also limit the number of neighboring networks involved. Since the total number of neighboring networks is usually small, this can be handled by trying all combination of no more than a certain number of neighbors.
- We handle SRLGs by treating the failure of all links in a SRLG as a "single link failure" in Definition 2, and setting target connectivity values accordingly. For instance, let *CL* be the minimal set of links that must be removed to disconnect the two endpoints of a link $l = (i, j)$. If *CL* includes 3 links in the same SRLG and 1 link in a different SRLG, we take the link connectivity of l to be $1 + 1 = 2$ instead of $3 + 1 = 4$. Therefore, if we want to keep i and j connected under two simultaneous failures, we shall set the target connectivity $req(l) = 5$.

The above problem is a generalization of the *connectivity augmentation* problems studied in [7]. In [7], the link connectivity requirement function $req(l) = 2$ for all $l \in E$; while in our problem, we may need to set $req(l) > 2$ to guard against SRLG failures. For a general survey of the connectivity augmentation problem, we refer interested readers to [15]. Our problem differs in that the set of available links that can be added to the original graph is constrained. Our problem also resembles the *network flow improvement* problem [29], where one incurs a cost for increasing the capacity of an link and the goal is to achieve a maximum flow through the network under a given budget. Our problem differs in that we seek to minimize the cost while achieving certain levels of connectivity. Also, we seek to improve a set of flows, each independently, instead of just one.

We formulate this selection problem as a Mixed Integer Program (MIP). Specifically, let $\widetilde{G} = (V, E \cup \mathrm{BYPASS})$ be a flow network with unit capacity on all links. Let $x(l) \in \{0, 1\}, l \in \mathrm{BYPASS}$ be the indicator variables of interdomain bypass link selection, such that $x(l) = 1$ if bypass link l is selected, and 0 otherwise. The MIP can be formulated as follows:

$$\min \quad \sum_{l \in \mathrm{BYPASS}} \mathrm{cost}(l) \cdot x(l) \tag{4}$$

subject to $\forall (s,t) = l \in L$, $f_{(s,t)}$ is a s-t flow such that:

$$0 \leq f_{(s,t)}(l) \leq 1, \forall l \in E \tag{5}$$

$$0 \leq f_{(s,t)}(l) \leq x(l), \forall l \in \mathrm{BYPASS} \tag{6}$$

$$\sum_{k \in V} f_{(s,t)}(s,k) \geq req(s,t) \tag{7}$$

Note that in the above MIP, we have used the Maximum-Flow Min-Cut Theorem [2] to implicitly encode the link connectivity requirement. We solve this MIP using ILOG CPLEX [8].

4.3 Bypass Selection for Fast Rerouting

We further augment the set of interdomain bypass paths to ensure desired performance level during fast rerouting. Note that bypass selection is involved in both of the two steps of our optimal fast rerouting algorithm. First, bypass selection determines part of the input set of links for optimal fast rerouting. Second, the coverage-based path generation phase of our fast rerouting algorithm needs to select paths that provide good coverage. Some of such paths may need to traverse interdomain bypass paths.

One strategy would be to take all possible interdomain bypass paths as input to our optimal fast rerouting algorithm, and leave it to coverage-based path generation to determine which bypass paths are really necessary. A serious drawback of this strategy is that optimal fast routing may use more bypass paths than necessary.

Another strategy is to formulate the problem as a mixed integer program, which seeks to minimize the number of bypass paths selected under the constraint that optimal fast rerouting has acceptable performance. This strategy, however, requires the coordination of optimal fast rerouting among different failure scenarios, which increases the computational overhead of the problem formulated in Section 3 dramatically, let alone the complexity of MIP.

Our solution to this problem is a simple sequential strategy. We first sort all available interdomain bypass paths from best to worst according to a scoring function. This scoring function could be cost, unit cost per bandwidth, or some combination of cost and bandwidth constraints. For each k, we select the first k paths and test the performance of fast rerouting based on this set of bypass paths. The selection process stops once we achieve performance target.

5. EVALUATIONS

In this section, we evaluate our REIN framework and algorithms using real network topologies and traffic traces.

5.1 Evaluation Methodology

Network	Aggregation level	#Nodes	#Links
Abilene	router-level	11	28
Abovenet	PoP-level	15	60
AOL	PoP-level	21	64
Cogent	PoP-level	20	60
Level-3	PoP-level	46	536
Qwest	PoP-level	33	166
Sprint	PoP-level	32	128
US-ISP	PoP-level	-	-
UUNet	PoP-level	47	336

Table 2: Summary of network topologies used.

Dataset description: We use the real topologies of Abilene and a major IP network which we call US-ISP. The topology of Abilene is a router-level topology, while for US-ISP, we use a PoP-level topology, which differs from the real router-level topology, but still illustrates the scope and power of the framework and algorithms proposed here. The topologies of Abilene and US-ISP are complete without inference errors. In addition, we use the PoP-level topologies of 7 IP networks as inferred by RocketFuel [39]. We recursively merge the leaf nodes of the topologies with their parents until no nodes have degree one, so that we have the backbone of the networks. We assume that the available bandwidth of an interdomain bypass path is 20% of the typically link bandwidth of the network (*e.g.*, 2 Gbps in Abilene). Table 2 summarizes the topologies that contribute or use interdomain bypass paths. The data for US-ISP are not shown due to privacy concerns.

We use real traffic demand matrices of Abilene, made available from [1]. We assume a random portion (up to 50%) of the traffic of Abilene are VPN traffic. We captured, in Jan. 2007, both IP traffic and VPN traffic matrices for US-ISP.

To generate failure scenarios, for Abilene, we enumerate all single and two-link failure scenarios; for US-ISP, we process its system logs and identify failure events.

Algorithms: We consider the following algorithms.

- Robust path-based TE/FRR (*robust*): The normal case routing, fast rerouting, and path generation are computed using the algorithms in Section 3.
- Oblivious routing/bypassing (*oblivious*): The normal case routing, fast rerouting are computed using the algorithms in [6] and [5]. Note that *oblivious* does not use path generation.
- CSPF: This is the constrained shortest-path-first (CSPF) algorithm popularly used in IP/MPLS networks for traffic engineering. We use the actual weights of Abilene and US-ISP to compute the normal case routing. The fast rerouting for a set of failed links is computed by running the CSPF algorithm with the failed links removed. Note that the normal case routing can be implemented by standard IP forwarding, while the bypass routing generally would require MPLS forwarding using LSPs.
- Flow-based optimal routing (*optimal*): This is the optimal routing for each given traffic and failure scenario. It is unrealistic to implement, as it is a flow-based routing scheme. Furthermore, under a failure scenario, it would require complete change of routing, and thus could cause large disruption to network traffic. We use it as a lower bound for evaluating the performance of other algorithms.

Performance metric: We measure the performance of the network by the traffic to capacity ratio at the bottleneck link (*i.e.*, the link with the highest traffic to capacity ratio). We referred to this metric as the bottleneck traffic intensity or traffic intensity for short. This metric is also referred to as maximum link utilization (MLU) in the literature.

Network	Period	Data description
Abilene	03/01/04 - 09/10/04	5-minute traffic traces
US-ISP	Jan. 2007	hourly traffic traces

Table 3: Summary of real traffic traces used.

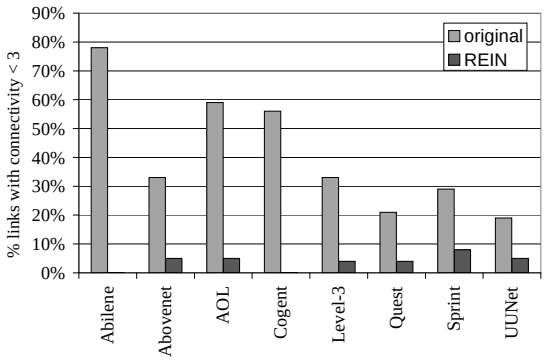


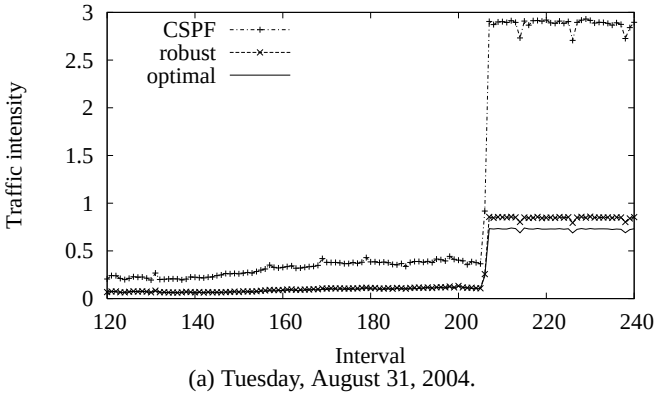
Figure 4: Connectivity improvement of REIN with no more than 7 links from at most 3 neighboring networks. Note that all topologies are inferred by Rocketfuel except for Abilene. Inferred topologies may have missing links.

5.2 Evaluation Results

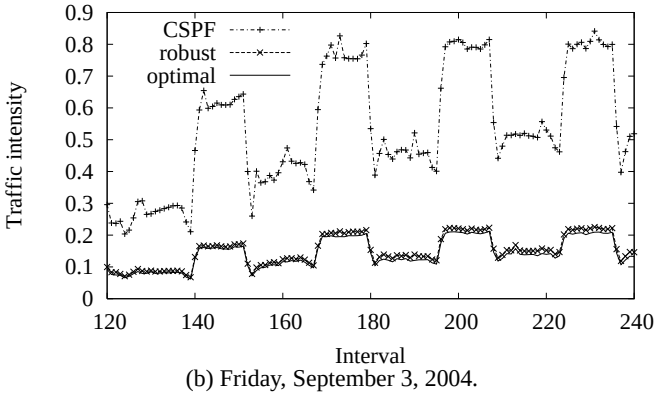
Connectivity: We start with demonstrating the effectiveness of REIN by improving the reliability of networks through connectivity improvement. We measure the reliability of an network by the link connectivities of the links in its network; for definition and algorithm on link connectivity see Section 4.2. An network with links with low link connectivity is less reliable. For example, during the aforementioned Sprint incident, the Sprint network is partitioned due to the existence of links with link connectivity 2. Due to the lack of SRLG data, we assume all links, both intradomain and interdomain, belong to their own and distinct SRLGs.

Figure 4 shows the link connectivity of the 8 network topologies shown in Table 2. The *y*-axis is the percentage of links with low connectivity (*i.e.*, connectivity less than or equal to 2). We make two observations. First, without interdomain bypass paths, all evaluated networks have at least 19% links with link connectivity less than or equal to 2. Some can be as high as close to 60%. The percentage for the Sprint network is 29%. Thus, it is not totally surprising that the network could partition. Note that the topologies inferred by Rocketfuel may have missing links, and adding missing links may reduce the percentage of links with low connectivity. However, for the Abilene and US-ISP topologies, both with no inference errors, the percentage of low connectivity links is still non-negligible. Specifically, for the Abilene topology, 11 out of the 14 links have connectivity 2.

Second, interdomain bypass paths significantly reduce the percentage of links with low connectivity. Using the algorithm presented in Section 4.2, for each topology, we select interdomain bypass paths so that the link connectivity is improved to at least 3, if possible, assuming a same-city peering relationship at PoP level, without using more than 7 interdomain bypass paths. From Figure 4, we observe that, REIN has substantially reduced the percentage of links with low connectivity. For example, with just 3 paths from US-ISP, REIN has completely eliminated from Abilene any link with connectivity less than 3; with 5 paths from two neighboring networks, REIN has reduced the percentage of Sprint from 29% to 8%; with 5 paths from three neighboring networks, REIN has reduced the percentage of US-ISP by a factor of 5 (exact numbers are not shown in the figure due to privacy concerns). Detailed

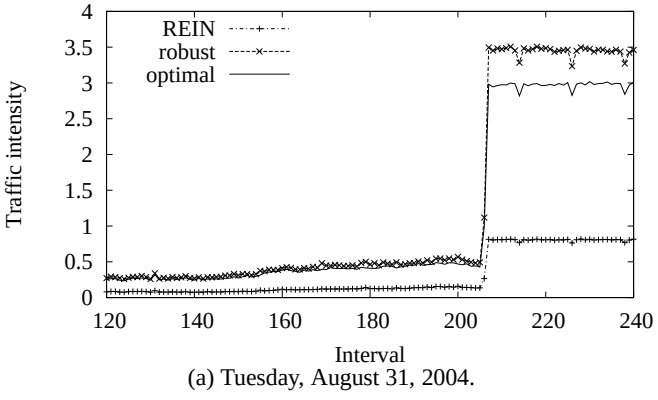


(a) Tuesday, August 31, 2004.

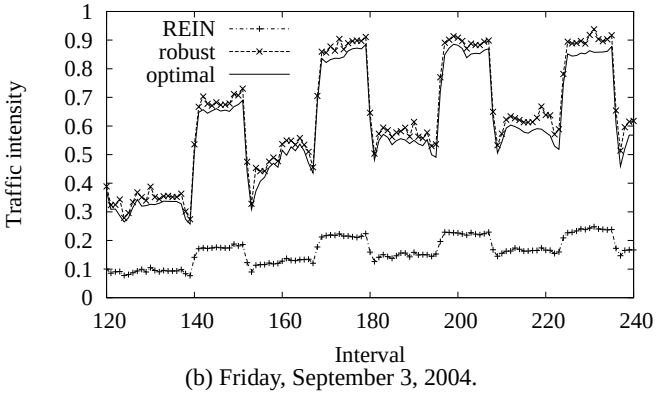


(b) Friday, September 3, 2004.

Figure 5: Time series plots of bottleneck link traffic intensity: Abilene traces, one-link failure scenarios.



(a) Tuesday, August 31, 2004.



(b) Friday, September 3, 2004.

Figure 6: Time series plots of bottleneck traffic intensity: Abilene traces, two-link failure scenarios.

investigation shows that the reason we can improve connectivity with a small number of interdomain bypass paths is that the topologies of most IP networks have several well connected components, between which the connecting links are scarce. This is a good optimization target for connectivity improvements, and also the reason for the substantial gain with only a few extra paths added. Note that although it is also possible for an IP network to install additional physical links at these locations by itself to improve connectivity, the cost can be much higher.

Next, we evaluate the effectiveness of REIN using real Internet topologies and traffic traces.

Abilene: First, we report the results using Abilene. We evaluate two categories of failure scenarios: single-link failure scenarios, and two-link failure scenarios. For these two categories, for each interval, we enumerate each scenario in the category and report the worst-case traffic intensity on the bottleneck link. We select two days to show traces: one with high traffic demand (August 31, 2004) and one with relatively low traffic demand (September 3, 2004).

We first introduce single-link failure scenarios without using REIN. Figure 5 shows the results. It is clear that CSPF is less efficient in dealing with failures in the network, as there are single-link failure scenarios where CSPF can drive traffic demand to the bottleneck link to be as high as nearly 300% link capacity, and three times that of robust path-based TE/FRR. A closer examination of the Abilene topology shows a 2.5 Gbps link between Atlanta and Indianapolis, whereas the rest of the links are 10 Gbps. When a link fails, CSPF diverts a large amount of traffic into the low-bandwidth link, creating severe congestion. For ease of presentation, we will not show the performance of CSPF under two-link failure scenarios. As a comparison, our robust path-based TE/FRR performs well under all single-link failure scenarios, being close to optimal for both days. Thus, with an effective TE/FRR algorithm such as the robust path-based TE/FRR, there may not be a need to introduce interdomain bypass paths under single link failure scenarios.

Now we stress the Abilene network with two-link failures. We do not include two-link failure scenarios which may partition the network. Figure 6 shows the results. On August 31, 2004, after interval 208, there are failure scenarios where our robust path-based TE/FRR drives the traffic demand to be almost 3.5 times the bottleneck link capacity. However, the figure also shows that at these severe failure scenarios, even the optimal routing using intradomain links will drive the traffic demand to be almost 3 times bottleneck link capacity, leaving it impossible for any fast rerouting algorithm to recover without significantly reduced throughput.

Figure 6 also shows the effectiveness of REIN. Specifically, it shows network performance when the 10 interdomain bypass paths selected by our path selection algorithm are added. These 10 paths include both the 3 needed for connectivity and additional 7 for fast rerouting. The performance of our robust path-based TE/FRR with additional interdomain bypass paths is labeled as REIN. On August 31, 2004, we observe that REIN reduces traffic intensity from 350% to only 80% of bottleneck link capacity: the interdomain bypass paths reduce the traffic intensity by 270% on the bottleneck link.

The preceding traces illustrate the effectiveness of REIN and the robust path-based algorithm at several intervals during two days. We next summarize the overall performance during the whole week period in which we have evaluated REIN and robust path-based TE/FRR. Figure 7(a) shows the reduction of traffic intensity by REIN for one-link failure scenarios. We observe that the reduction is relatively small. However, this is expected since the TE/FRR algorithm spreads traffic well and the network is not stressed under one-link failures, as illustrated by Figure 5. REIN is effective when there are severe network failures. Figure 7(b) shows the effectiveness of REIN under two-link failure scenarios. For around 90% of the intervals, REIN reduces traffic intensity on the bottle-

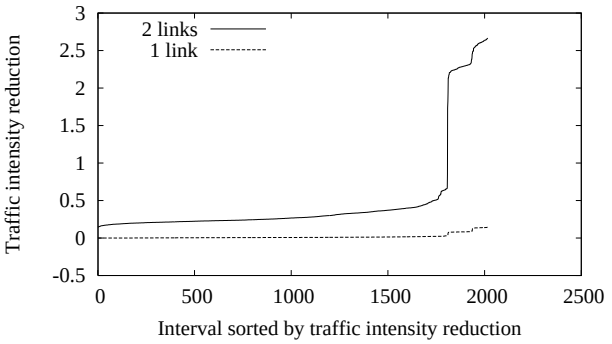


Figure 7: Sorted interval of the improvement of bottleneck traffic intensity by REIN from Saturday, August 28, 2004 to Friday, September 3, 2004 for Abilene.

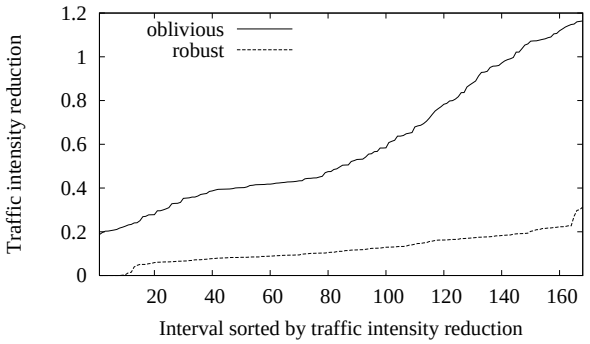


Figure 8: Sorted interval of the improvement of traffic intensity by REIN from a week in January 2007 for US-ISP.

neck link by 20% to 60% of the link capacity of the bottleneck link. This is significant reduction of traffic. For the remaining 10% intervals, the reduction of traffic on the bottleneck link by REIN is even more significant: 220% to 270% of the capacity of the bottleneck link. These intervals include those starting from interval 208 in Figure 6(a). They show clearly the benefits and effectiveness of REIN.

US-ISP: The evaluation of performance under failures on Abilene is mainly worst-case analysis, due to the lack of accurate failure logs. Next, we evaluate the performance of REIN and the robust path-based TE/FRR algorithm on US-ISP, with a replay of both traffic demands and the failure logs. In our experiments, our interdomain bypass selection algorithm has selected 12 bypass paths from 3 neighboring networks of US-ISP.

For confidentiality, we do not report the absolute traffic intensity on the bottleneck link for US-ISP. Instead, we report relative bottleneck traffic intensity normalized by the highest bottleneck traffic intensity under the optimal routing without failures over the evaluation period.

Figure 8 shows the reduction of the normalized traffic intensity on the bottleneck link by REIN, for both our robust path-based TE/FRR algorithm and the oblivious routing/bypass algorithm of Applegate *et al.* [5]. We observe that the distribution of traffic reduction by REIN for US-ISP is more uniform than that for Abilene. Also, the maximum reduction of normalized traffic intensity on the bottleneck link is also smaller than that of Abilene (*i.e.*, from 270% to 118% and 35%). One possible reason for these differences is that US-ISP has an over-provisioned network with high redundancy. In addition, the failure logs we have collected may not include all serious, but possible failure scenarios. Despite these differences, we nevertheless note that REIN may still reduce traffic to the bottleneck link significantly. We observe that there exist failure cases where REIN can reduce the normalized traffic intensity on the bottleneck link by 118% and 35% when the network uses *oblivious* and *robust* respectively.

One potential concern of the REIN scheme is its impact on the peering links and the links in the neighboring networks when an network diverts traffic to them during a failure event. In all of our evaluations, the traffic diverted on an interdomain bypass path have never exceeded 60% of the exported capacity of the path for Abilene, and 70% for US-ISP. Thus, the benefit of REIN is achieved at a moderate cost to the neighboring networks.

5.3 Tradeoffs in REIN

The preceding results have demonstrated the effectiveness of our framework and the robust path-based TE/FRR algorithm. A key implementation technique that we have used is the coverage-based path generation algorithm. It is a general path generation technique that can be applied to any flow-based routing and allows explicit tradeoff between optimality and scalability. Below we demonstrate

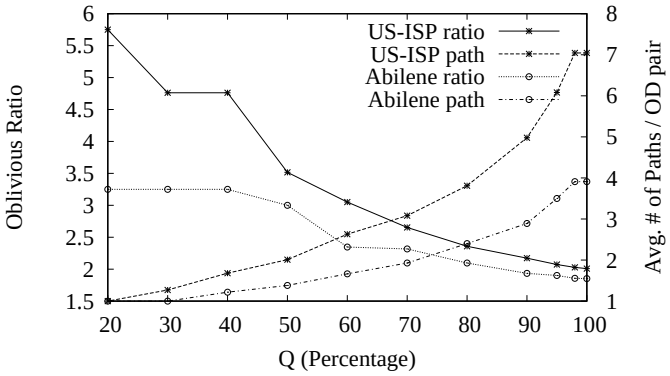


Figure 9: Tradeoff between optimality and scalability.

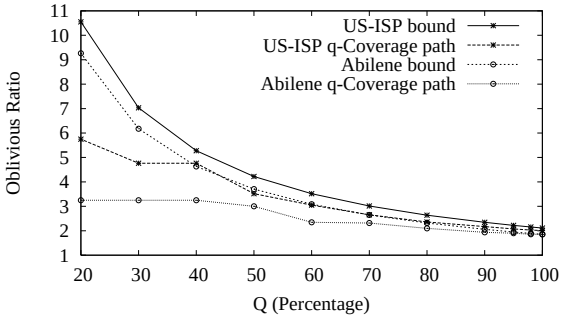


Figure 10: Performance of achieving Q-percentage coverage and theoretical bound.

how to control the tradeoff by varying the target percentage value of the adaptive path generation algorithm presented in Section 3.4.2. In order to measure optimality, we use computation of the optimal oblivious routing [6] as the benchmark problem, and measures the optimality of a routing by the oblivious ratio that it achieves. We show results for Abilene and US-ISP.

Figure 9 shows the oblivious ratio achieved by different path sets with different target coverage percentages used during path generation. It also shows the average number of paths per OD pair required to achieve each target coverage percentage. We can see that higher coverage leads to higher number of paths required per OD pair and better performance. Using 3 LSP paths in Abilene and 5 in US-ISP can achieve near-optimal performance. Note that the flat part of the performance-ratio curve at the low coverage region is due to the fact that all those coverage ratios can be achieved using only one path for each OD pair.

We also evaluate the tightness of the bound provided by Proposition 1, and the results are shown in Figure 10. We can see that after 70% coverage, our bound is tight and is a good prediction for the performance using those paths.

6. RELATED WORK

The importance of reliability in communication networks has long been recognized, and there is a large body of related work. REIN is part of the efforts to improve network reliability at low cost.

In the past, most networks rely on link layer techniques such as SONET rings to protect against failures (*e.g.*, [17]). Later, due to the relatively high cost of SONET protection and the lower cost and better flexibility of IP, many networks turned to the IP layer to handle failures [21].

Many techniques for optimizing routing adaptively under potential failures have been developed (*e.g.*, [11, 14, 17, 23, 33, 34]). These techniques are referred to as *restoration* techniques.

A potential drawback of the restoration techniques is their relatively slow response time, which may not be able to meet the requirements of some mission-critical applications (*e.g.*, VPN networks carrying VoIP traffic). Thus, the restoration techniques are supplemented by MPLS-based *protection* techniques, which precompute the rerouting paths and quickly reroute traffic upon failure detection (and before routing convergence) [38]. Such techniques are major features of modern routers.

Our REIN framework can improve the effectiveness of both restoration and protection techniques, by utilizing them over an augmented intradomain topology with virtual links that correspond to the additional interdomain bypass paths. The added virtual links increase the redundancy available to these techniques, and therefore can improve their performance in most cases.

A key requirement of protection techniques is to compute the rerouting paths. Many algorithms have been proposed (*e.g.*, [24, 25, 26, 27]). In particular, Applegate *et al.* [5] presented the optimal oblivious fast rerouting algorithms. However, the formulation in [5] uses flow-based routing, which is not readily implementable in the current Internet. In this paper, our coverage based path generation technique achieves implementable optimal fast rerouting.

The REIN framework is complementary to those that focus on protecting interdomain links (*e.g.*, [41]). For example, multihoming has been introduced as a way to improve network performance (*e.g.*, [3]) and reliability (*e.g.*, [9]). The difference between multihoming and our approach is that multihoming is used to protect links at the border and outside the IP network, while our architecture focuses on protecting intradomain links. In [13], the general capability of setting up interdomain label switched paths (LSPs) is discussed along with many potential applications. However, the deployment of interdomain LSPs appears to be challenging.

Another technique to improve reliability is overlay networks (*e.g.*, [4, 20]). Although overlays provide flexibility and can detect failures faster than BGP response time, they rely on transport layer timeouts to detect network failures, which can be much slower than the link layer detection used in REIN. Additionally, since it is operated by IP networks, REIN has much better visibility of the network and ability to conduct traffic engineering than overlays.

7. CONCLUSIONS AND FUTURE WORK

In this paper, we have proposed a novel framework called reliability as an interdomain service to improve the redundancy of IP networks at low cost. We have also developed robust algorithms to efficiently utilize network resources under failures. We have demonstrated the performance of our framework and algorithms using real network topologies and traffic traces.

There are multiple directions for future work. One is a thorough quantification of the effects of cross-provider SRLG data. Another one is to evaluate the effectiveness of our approach on smaller IP networks instead of the backbone networks that we evaluated. Further improvement on TE robustness under dynamic topology is also an interesting issue.

8. ACKNOWLEDGMENTS

Hao Wang and Y. Richard Yang were supported in part by grants from the U.S. NSF. The coverage-based path generation framework is designed and implemented by Zheng Ma. We are grateful to Anees Shaikh, our shepherd, for his extensive help in revising the paper. Richard Alimi, David Applegate, Lee Breslau, Charles Kalmanek, Arvind Krishnamurthy, Michael Merritt, Han Nguyen, K. K. Ramakrishnan, Jennifer Rexford, Dave Wang, Haiyong Xie, and Jennifer Yates have given us very valuable suggestions. We are also grateful to the anonymous reviewers whose comments improve the paper.

9. REFERENCES

[1] Abilene topology and traffic dataset. `http://www.cs.utexas.edu/~yzhang/research/AbileneTM/`.

[2] R. Ahuja, T. Magnanti, and J. Orlin. *Network Flows*. Prentice Hall, 1993.

[3] A. Akella, B. Maggs, S. Seshan, A. Shaikh, and R. Sitaraman. A measurement-based analysis of multihoming. In *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003.

[4] D. G. Andersen, H. Balakrishnan, M. F. Kaashoek, and R. Morris. Resilient overlay networks. In *Proceedings of the 18th Annual ACM Symposium on Operating Systems Principles*, Banff, Canada, Oct. 2001.

[5] D. Applegate, L. Breslau, and E. Cohen. Coping with network failures: Routing strategies for optimal demand oblivious restoration. In *Proceedings of SIGMETRICS*, New York, NY, June 2004.

[6] D. Applegate and E. Cohen. Making intra-domain routing robust to changing and uncertain traffic demands: Understanding fundamental tradeoffs. In *Proceedings of ACM SIGCOMM*, Karlsruhe, Germany, Aug. 2003.

[7] M. Conforti, A. Galluccio, and G. Proietti. Edge-connectivity augmentation and network matrices. In J. Hromkovic, M. Nagl, and B. Westfechtel, editors, *WG*, volume 3353 of *Lecture Notes in Computer Science*, pages 355–364. Springer, 2004.

[8] ILOG CPLEX: optimization software. http://www.ilog.com/products/cplex/.

[9] A. Dhamdhere and C. Dovrolis. ISP and egress path selection for multihomed networks. In *Proceedings of IEEE INFOCOM*, Barcelona, Spain, Apr. 2006.

[10] N. G. Duffield, P. Goyal, A. Greenberg, P. Mishra, K. K. Ramakrishnan, and J. E. van der Merive. A flexible model for resource management in virtual private networks. In *Proceedings of ACM SIGCOMM*, Cambridge, MA, Aug. 1999.

[11] A. Elwalid, C. Jin, S. Low, and I. Widjaja. MATE: MPLS adaptive traffic engineering. In *Proceedings of IEEE INFOCOM*, Anchorage, AK, Apr. 2001.

[12] T. Erlebach and M. Rüegg. Optimal bandwidth reservation in hose-model VPNs with multi-path routing. In *Proceedings of IEEE INFOCOM*, Hong Kong, China, Apr. 2004.

[13] A. Farrel, J.-P. Vasseur, and A. Ayyangar. *A Framework for Inter-Domain Multiprotocol Label Switching Traffic Engineering, RFC 4726*, Nov. 2006.

[14] B. Fortz and M. Thorup. Robust optimization of OSPF/IS-IS weights. In *Proceedings of INOC*, Oct. 2003.

[15] A. Frank. Augmenting graphs to meet edge-connectivity requirements. *SIAM J. Discrete Math.*, 5(1):25–53, 1992.

[16] GENI Planning Group. GENI: Conceptual design, project execution plan. GENI Design Document 06-07, Jan. 2006.

[17] F. Giroire, A. Nucci, N. Taft, and C. Diot. Increasing the robustness of IP backbones in the absence of optical level protection. In *Proceedings of IEEE INFOCOM*, San Francisco, CA, Mar. 2003.

[18] S. P. Gorman. *Networks, Complexity, and Security: The Role of Policy in Critical Infrastructure Protection*. PhD thesis, George Mason School of Public Policy, May 2004.

[19] W. Grover and D. Tipper. Design and operation of survivable networks. *Journal of Network and System Management*, 13(1), Mar. 2005.

[20] K. P. Gummadi, H. Madhyastha, S. D. Gribble, H. M. Levy, and D. J. Wetherall. Improving the reliability of Internet paths with one-hop source routing. In *Proceedings of OSDI*, San Francisco, CA, Dec. 2004.

[21] I. Hussain. *Fault-Tolerant IP and MPLS Networks*. Cisco Press, 2005.

[22] S. Iyer, S. Bhattacharyya, N. Taft, and C. Diot. An approach to alleviate link overload as observed on an IP backbone. In *Proceedings of IEEE INFOCOM*, San Francisco, CA, Apr. 2003.

[23] S. Kandula, D. Katabi, B. Davie, and A. Charny. Walking the tightrope: Responsive yet stable traffic engineering. In *Proceedings of ACM SIGCOMM*, Philadelphia, PA, Aug. 2005.

[24] K. Kar, M. S. Kodialam, and T. V. Lakshman. Routing restorable bandwidth guaranteed connections using maximum 2-route flows. *IEEE/ACM Transactions on Networking*, 11(5):772–781, 2003.

[25] M. Kodialam and T. V. Lakshman. Dynamic routing of locally restorable bandwidth guaranteed tunnels using aggregated link usage information. In *Proceedings of IEEE INFOCOM*, Anchorage, AK, Apr. 2001.

[26] M. Kodialam, T. V. Lakshman, and S. Sengupta. A simple traffic independent scheme for enabling restoration oblivious routing of resilient connections. In *Proceedings of IEEE INFOCOM*, Hong Kong, China, Apr. 2004.

[27] M. S. Kodialam and T. V. Lakshman. Dynamic routing of restorable bandwidth-guaranteed tunnels using aggregated network resource usage information. *IEEE/ACM Transactions on Networking*, 11(3):399–410, 2003.

[28] S. Köhler and A. Binzenhöfer. MPLS traffic engineering in OSPF networks - a combined approach. In *Proceedings of ITC*, Berlin, Germany, Sept. 2003.

[29] S. Krumke, H. Noltemeier, S. Schwarz, H. Wirth, and R. Ravi. Flow improvement and network flows with fixed costs, 1998.

[30] A. Kumar, R. Rastogi, A. Silberschatz, and B. Yener. Algorithms for provisioning virtual private networks in the hose model. In *Proceedings of ACM SIGCOMM*, San Diego, CA, Aug. 2001.

[31] Y. Li, J. Harms, and R. Holte. A simple method for balancing network utilization and quality of routing. In *Proceedings of ICCCN*, San Diego, CA, 2005.

[32] A. Markopoulou, G. Iannaccone, S. Bhattacharyya, C. Chuah, and C. Diot. Characterization of failures in an IP backbone network. In *Proceedings of IEEE INFOCOM*, Hong Kong, China, Apr. 2004.

[33] S. Nelakuditi, S. Lee, Y. Yu, and Z. Zhang. Failure insensitive routing for ensuring service availability. In *Proceedings of International Workshop on Quality of Service (IWQoS))*, 2003.

[34] A. Nucci, B. Schroeder, S. Bhattacharyya, N. Taft, and C. Diot. IGP link weight assignment for transient link failures. In *Proceedings of ITC*, 2003.

[35] K. Papagiannaki, N. Taft, and C. Diot. Impact of flow dynamics on traffic engineering design principles. In *Proceedings of IEEE INFOCOM*, Hong Kong, China, Apr. 2004.

[36] C. Pelsser. *Interdomain Traffic Engineering with MPLS*. PhD thesis, UCL/INGI, Belgium, Oct. 2006.

[37] M. Pióro and D. Medhi. *Routing, Flow, and Capacity Design in Communication and Computer Networks*. Morgan Kaufmann, 2004.

[38] V. Sharma, B. M. Crane, S. Makam, K. Owens, C. Huang, F. Hellstrand, J. Weil, L. Andersson, B. Jamoussi, B. Cain, S. Civanlar, and A. Chiu. *Framework for MPLS-Based Recovery. RFC 3469*, Feb. 2003.

[39] N. Spring, R. Mahajan, and D. Wetherall. Rocketfuel: An ISP topology mapping engine. Available from `http://www.cs.washington.edu/research/networking/rocketfuel/`.

[40] Telemark. Telemark survey. http://www.telemarkservices.com/, 2006.

[41] The Totem Project. Available at: `http://totem.info.ucl.ac.be//`.

[42] H. Wang, H. Xie, L. Qiu, Y. R. Yang, Y. Zhang, and A. Greenberg. COPE: Traffic engineering in dynamic networks. In *Proceedings of ACM SIGCOMM*, Pisa, Italy, Sept. 2006.

[43] Wired News. The backhoe: A real cyberthreat. `http://www.wired.com/news/technology/1,70040-0.html`, Jan. 2006.

APPENDIX

A. Robust TE/FRR with VPN Support

Our algorithm in Section 3.3 can be described by the following three optimization problems. We first compute a base routing with hose-model VPN bandwidth provisioning using (8):

$$\begin{aligned} \min_{f} \quad & o(f,\mathcal{D}) && (8)\\ \text{s.t.} \quad & f \text{ is a routing;} \\ & \forall d \in \mathcal{X} : c(f,d) \le \overline{r}; \quad \text{(penalty envelope)} \\ & \forall \text{link } l \in E \text{ and VPN traffic demand matrix } d^w \text{ such that} \\ & \forall a \in V, \sum_{b\in V} d^w_{ab} \le ECR(a), \sum_{b\in V} d^w_{ba} \le ICR(a): \\ & \sum_{a,b\in V} d^w_{ab} f_{ab}(l) \le cap(l). \quad \text{(VPN provisioning)} \end{aligned}$$

Let f^* be a solution to (8). For each high-priority failure scenario $h \subset E$, our algorithm first tries to solve (9):

$$\begin{aligned} \min_{f^{h,B},\, f^{h,V}} \quad & r && (9)\\ \text{s.t.} \quad & f^{h,B} \text{ is a fast rerouting in } E \cup E' \\ & \text{bypassing links } e \in h \text{ for best-effort traffic;} \\ & f^{h,V} \text{ is a fast rerouting in } E \\ & \text{bypassing links } e \in h \text{ for VPN traffic;} \\ & \forall \text{link } l \in E \cup E', l \notin h \text{ and } d \text{ such that } OU_{G\setminus h}(d) = 1: \\ & \sum_{a,b\in V} d_{ab}(f^*_{ab}(l) + \sum_{e\in h} f^*_{ab}(e) f^{h,B}_e(l)) \le r \cdot cap(l), \text{ and} \\ & \sum_{a,b\in V} d_{ab}(f^*_{ab}(l) + \sum_{e\in h} f^*_{ab}(e) f^{h,V}_e(l)) \le r \cdot cap(l); \\ & \forall \text{link } l \in E, l \notin h \text{ and } d^w \text{ such that} \\ & \forall a \in V, \sum_{b\in V} d^w_{ab} \le ECR(a), \sum_{b\in V} d^w_{ba} \le ICR(a): \\ & \sum_{a,b\in V} d^w_{ab}(f^*_{ab}(l) + \sum_{e\in h} f^*_{ab}(e) f^{h,V}_e(l)) \le cap(l). \quad (10) \end{aligned}$$

In (9), constraint (10) requires that all VPN traffic be completely rerouted using intradomain links only.

If (9) has no feasible solutions (*i.e.*, it cannot reroute VPN traffic using intradomain links only), we solve (11):

$$\begin{aligned} \min_{f^h} \quad & r && (11)\\ \text{s.t.} \quad & f^h \text{ is a fast rerouting in } E \cup E' \text{ bypassing all links in } h; \\ & \forall \text{link } l \in E \cup E', l \notin h \text{ and } d \text{ such that } OU_{G\setminus h}(d) = 1: \\ & \sum_{a,b\in V} d_{ab}(f^*_{ab}(l) + \sum_{e\in h} f^*_{ab}(e) f^h_e(l)) \le r \cdot cap(l). \end{aligned}$$

In (11), there is no distinction between best-effort and VPN traffic.

The VPN bandwidth provisioning constraint in (8) involves an exponential number of constraints, but can be converted into a polynomial number of constraints by applying linear programming duality:

$$\begin{aligned} & \forall a \in V : \mu_l(a) \ge 0, \nu_l(a) \ge 0; \\ & \forall a,b \in V : f_{ab}(l) \le \mu_l(a) + \nu_l(b); \\ & \sum_{a\in V} (\mu_l(a) ECR(a) + \nu_l(a) ICR(a)) \le cap(l). \end{aligned}$$

The constraint (10) can be handled in a similar way.

B. Proof of Lemma 1

PROOF. First consider an OD pair $a \to b$ and the corresponding path set $P_{ab} = \{P^1_{ab}, \ldots, P^K_{ab}\}$. By definition of a Q-percentage coverage path set, there exists a path-based routing $\widetilde{p}_{ab}$ with value $|\widetilde{p}_{ab}| = Q$. Let p_{ab} be another path-based routing over P_{ab} such that $p^k_{ab} = \frac{1}{Q}\widetilde{p}^k_{ab}$, for all $k = 1, \ldots, K$. We have

$$|p_{ab}| = \sum_{k=1}^{K} p^k_{ab} = \sum_{k=1}^{K} \frac{1}{Q}\widetilde{p}^k_{ab} = \frac{1}{Q}|\widetilde{p}_{ab}| = \frac{1}{Q} \cdot Q = 1.$$

Thus, p_{ab} is a valid path-based routing from a to b over the path set P_{ab}. Applying the same scaling for all OD pairs, we obtain a valid path-based routing p for the network over the Q-percentage coverage path set P.

Now for any demand d and any link $l \in E$, we have

$$\sum_{a,b\in V} \sum_{k:l\in P^k_{ab}} d_{ab} p^k_{ab} = \sum_{a,b\in V} \sum_{k:l\in P^k_{ab}} \frac{1}{Q} d_{ab} \widetilde{p}^k_{ab} \le \frac{1}{Q} \sum_{a,b\in V} d_{ab} f_{ab}(i,j),$$

where the last inequality holds due to Inequality (3). This completes the proof. □