

Maximizing Restorable Throughput in MPLS Networks

Reuven Cohen Gabi Nakibly
Technion – Israel Institute of Technology
Computer Science
Haifa, Israel

Abstract—MPLS recovery mechanisms are increasing in popularity because they can guarantee fast restoration and high QoS assurance. Their main advantage is that their backup paths are established in advance, before a failure event takes place. Most research on the establishment of primary and backup paths has focused on minimizing the added capacity required by the backup paths in the network. However, this so-called Spare Capacity Allocation (SCA) metric is less practical for network operators who have a fixed capacitated network and want to maximize their revenues. In this paper we present a comprehensive study on restorable throughput maximization in MPLS networks. We present the first polynomial-time algorithms for the splittable version of the problem. For the unsplittable version, we provide a lower bound for the approximation ratio and propose an approximation algorithm with an almost identical bound. We present an efficient heuristic which is shown to have excellent performance. One of our most important conclusions is that when one seeks to maximize revenue, local recovery should be the recovery scheme of choice.

Index Terms—MPLS, Failure, Recoverability

I. INTRODUCTION

IP networks should support real-time applications that require stringent availability and reliability, such as Voice over IP and virtual private networks. Unfortunately, failures are still common in the daily operation of networks, for reasons such as improper configuration, faulty interfaces, and accidental fiber cuts [1], [2]. Therefore, mechanisms that restore the flow of traffic quickly and efficiently after a failure are essential.

The IP routing protocols are not suitable for fast restoration. Using these protocols, a node first detects a failure and then disseminates routing updates to other nodes. These updates are used for calculating new paths. This process takes several seconds before proper routing of data can resume [3], [1]. During this time, packets destined to some destinations may be dropped, and applications might be disrupted. Moreover, when QoS is supported, the routing protocol cannot guarantee that the alternate path will provide the same QoS as the failed one.

For these reasons, many network operators do not rely only on IP routing protocols to restore traffic, but also employ recovery mechanisms in Layer 1 and Layer 2 protocols such as WDM, SONET/SDH, and MPLS. These recovery mechanisms guarantee fast restoration and high QoS assurance because they establish backup paths in advance, before a failure event takes place. Such recovery mechanisms are usually referred to as “protection” mechanisms, as opposed to “rerouting”

Abbreviation	Meaning
LSP	Label Switched Path, an established route in an MPLS domain
LSR	Label Switch Router, a routing device in an MPLS domain
GR	Global Recovery, one of the studied MPLS recovery schemes (see Section I)
LR	Local Recovery, one of the studied MPLS recovery schemes (see Section I)
RLR	Restricted Local Recovery, one of the studied MPLS recovery schemes (see Section I)
FLR	Facility Local Recovery, one of the studied MPLS recovery schemes (see Section I)
EkFLR	Extended k Facility Local Recovery, one of the studied MPLS recovery schemes (see Section I)
UR	Unrestricted Recovery, one of the studied MPLS recovery schemes (see Section I)
SCA	Spare Capacity Allocation, a widely used optimization metric for restorable traffic (see Sections I and II)
S/U-PRFP	Splittable/Unsplittable Primary-restricted Restorable Flow Problem, a new optimization problem described in this paper (see Sections III-A and III-B)
S/U-RFP	Splittable/Unsplittable Restorable Flow Problem, a new optimization problem described in this paper (see Sections III-C)
S/U-FP	Splittable/Unsplittable Flow Problem, a well-known optimization problem [6], [7]
S/U-PFP	Splittable/Unsplittable Primary-restricted Flow Problem, a restricted version of S/U-FP (see Sections III-B)

TABLE I
ABBREVIATIONS AND ACRONYMS USED IN THE PAPER

mechanisms, which establish backup paths only after a failure occurs.

In this paper we focus on MPLS-based protection mechanisms [4], [5]. However, our results are also applicable to other Layer 1 and Layer 2 protection mechanisms. In keeping with MPLS terminology, we refer to the path that carries the traffic before a failure as a primary LSP, and the path that carries the traffic after the primary LSP fails as a backup LSP. Throughout the paper we consider only bandwidth guaranteed protection. For this kind of protection, the backup LSP must be able to provide the same amount of guaranteed bandwidth provided by the primary LSP. To this end, resources should be reserved upon the establishment of each backup LSP, to be used only when the protected element – link or node – fails.

Many MPLS recovery schemes have been proposed. We classify these schemes as follows (see Table II):

- 1) Global recovery (GR) schemes [5] (Figure 1(a)): In this class, each primary LSP has one backup LSP. The

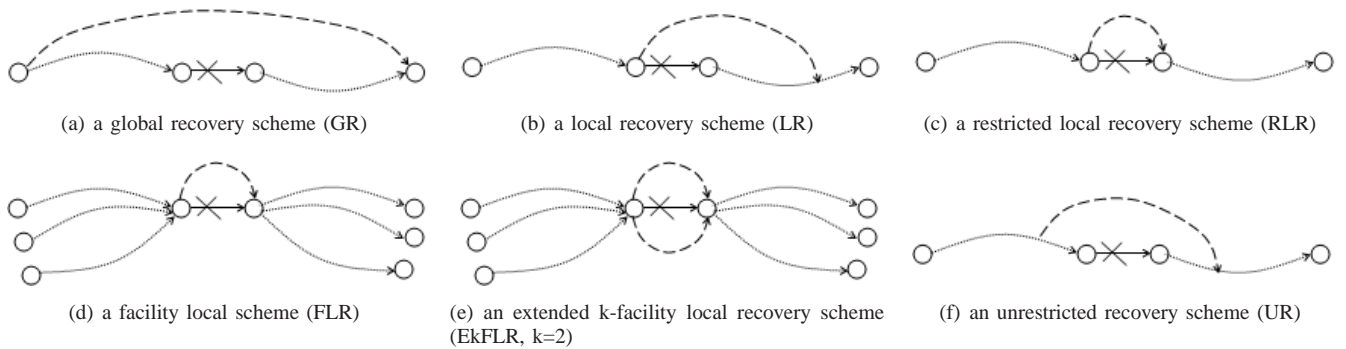


Fig. 1. Illustrations of the various recovery schemes

Recovery scheme class	Protected elements	Start node	End node	Protected primary LSPs
GR	all elements of primary LSP	head of primary LSP	tail of primary LSP	single LSP
LR	single element	immediately upstream of protected element	tail of primary LSP	single LSP
RLR	single element	immediately upstream of protected element	immediately downstream of protected element	single LSP
FLR	single element	immediately upstream of protected element	immediately downstream of protected element	all LSPs traversing the element
EkFLR	single element	immediately upstream of protected element	immediately downstream of protected element	a subset of LSPs traversing the element
UR	arbitrary	arbitrary	arbitrary	single LSP

TABLE II
CHARACTERISTICS OF THE RECOVERY SCHEMES

primary and backup LSPs share the same end nodes. The backup LSP protects against all link/node failures along the primary LSP, and it does not share any link/node with the primary LSP. A failure notification must propagate from the node that detects the failure to the head of the LSP. These recovery schemes are sometimes referred to as path recovery schemes.

- 2) Local recovery (LR) schemes [8], [5] (Figure 1(b)): In this class, a separate backup LSP is constructed to protect against a possible failure of each element along the primary LSP. Each backup LSP starts at the immediate upstream node of the protected element, and ends at the tail of the primary LSP. If such a local path does not exist, we assume that the LSP starts at the closest possible upstream node. A backup LSP may share links with the primary LSP upstream of the failure. The recovery in this class is faster than in the GR class, since the node that detects the failure is usually also the one that diverts the traffic to the backup LSP. However, more backup LSPs are needed to protect each primary LSP.
- 3) Restricted local recovery (RLR) schemes (Figure 1(c)): As in the LR scheme, a backup LSP starts at the immediate upstream node of the protected element but ends at the immediate downstream node. This makes the recovery process more local, since the route downstream of the failure does not change. Hence, unlike in the LR scheme, resources need not be released downstream of the primary LSP failure. The RLR schemes are sometimes referred to as link recovery schemes.
- 4) Facility local recovery (FLR) schemes [8] (Figure 1(d)): Backup LSPs are constructed as in the RLR schemes. However, a single backup LSP protects all the primary LSPs that traverse the protected element. This makes the process of restoring the traffic to the backup LSP simpler, using MPLS label stacking [4]. In addition, with this recovery scheme, the number of backup LSPs and the incurred state overhead are significantly reduced.
- 5) Extended k -facility local recovery (EkFLR) schemes [9] (Figure 1(e)): Backup LSPs are constructed as in RLR. However, there might be up to k backup LSPs that protect each element. Obviously, this scheme is more flexible than FLR, and permits the preferred trade-off between higher routing efficiency (k is larger) and lower administration overhead (k is smaller).
- 6) Unrestricted recovery (UR) schemes (Figure 1(f)): In this class, each primary LSP may be protected by any number of backup LSPs. Moreover, each backup LSP may start and end at any point along the primary LSP, and may protect against failures of any number of elements. This scheme incurs the highest administrative overhead while being the most flexible.

Table I summarizes the abbreviations and acronyms used throughout the paper.

A failure is frequently limited to a single network element – a link or a node. Hence, it is customary to compare recovery schemes by measuring their performance under the assumption that a failure may occur only after the network has recovered from the previous failure. An important implication of this assumption is that two backup LSPs protecting against different failures may share their reserved bandwidth. The single failure assumption may not hold in an optical WDM network, where the MPLS links are circuit-switched optical channels, called lightpaths [10]. Since a single physical fiber link may carry several lightpaths, a single failure in the optical layer may induce several MPLS link failures, an event known as “failure propagation”. Since the algorithms presented in this paper are complicated, we prefer to present them under the single failure assumption. However, in Appendix B we show how these algorithms can be extended to address MPLS tunnels over WDM lightpaths.

Most past research on the selection of backup LSPs is directed at minimizing the total bandwidth reserved for the backup LSPs. To this end, backup LSPs are routed to maximize their bandwidth sharing. This optimization metric is usually referred to as Spare Capacity Allocation (SCA). Models that seek to optimize SCA usually consider a network whose links have unbounded capacity, and a cost function associated with bandwidth usage. Their objective is to satisfy the entire set of flow demands while minimizing the cost of the backup LSPs. Such models are more suitable for network dimensioning or for the capacity planning stage, during which the network operator seeks to minimize the cost of the link capacities while satisfying future bandwidth demands. However, while minimizing the cost of building the backup LSPs is an important goal, network operators usually face a different optimization problem when they operate their networks: the networks have predetermined link capacities and the operators seek to maximize their revenue by maximizing the traffic the network can accommodate. In this model, it is assumed that link capacities have already been determined, and the goal is to maximize the revenue. In contrast to the SCA model, here not every demand should or can be satisfied. Another difference between our model and SCA is load balancing. The SCA model maximizes resource sharing and thus provides an incentive for load balancing. However, when resource sharing is not possible, SCA provides no incentive for load balancing since the cost associated with an established LSP does not depend on the current load of the selected route. This may result in traffic being concentrated on few highly loaded links. In contrast, a link in our model will not carry more than its predetermined capacity.

In light of the above, SCA is not the most practical criterion for running a given network during the operational stage. Hence, in this paper we present a comprehensive study of the problem of constructing primary and backup LSPs in a given network while *maximizing throughput*. The main contributions of paper are as follows:

- 1) We show that the splittable version of the problem is in \mathcal{P} and we offer the first polynomial time algorithm for it. In particular, we improve the results presented in [11], where only an FPTAS was shown.

- 2) We show that the unsplittable version of the problem is \mathcal{NP} -complete and has no approximation algorithm with a ratio of $|E|^{1/2-\epsilon}$.
- 3) We propose an approximation algorithm with the ratio of $O(|E|^{1/2})$ for the case where the traffic demand of an individual flow does not exceed half of the edge capacities.
- 4) We present efficient heuristic that is shown to have excellent performance.
- 5) We compare the various recovery schemes with respect to the throughput maximization criterion. We show that UR, GR and, LR differ only marginally in their performance. Since LR has the fastest restoration time of the three schemes, it should be the scheme of choice.
- 6) We show that EkFLR with $k = 2$ has almost the same performance as RLR and should be preferred over it for its lower administrative overhead (fewer backup LSPs).

The rest of the paper is organized as follows. Section II discusses related work. In Section III we formally define the problems addressed in the paper and discuss their computational complexity. Section IV presents algorithms for the problems. In section V we conduct a simulative comparison of algorithm performance for the various recovery schemes. In Section VI we extend the discussion to address node failures. Finally, Section VII concludes the paper.

II. RELATED WORK

Research on the performance of recovery schemes for virtual circuit routing has been conducted not only for MPLS, but also for ATM and optical networks. The recovery principles in most of these schemes are similar: a primary path is protected by one or more pre-established backup paths to which the traffic is restored. As mentioned in Section I, most previous work is directed at minimizing the Spare Capacity Allocation (SCA) metric. Many papers develop an Integer Linear Program (ILP) whose output is the set of backup paths that can fully restore the traffic on the primary paths. Since finding an optimal solution to an ILP is computationally hard, most of the papers focus on approximation algorithms or heuristics.

The heuristic algorithm in [12] chooses backup paths, one at a time, according to a GR scheme, and updates them iteratively. The algorithm in [13] uses a rounding process for the relaxed LP. It then uses hop-limited restoration routes to round the LP solution. Other algorithms that address the SCA problem using the ILP formulation can be found in [14], [15], [16], and [17]. In particular, [17] is concerned with the survivability of flows in ATM networks. As in our work, the authors consider several restoration schemes for splittable flows and formulates them as a linear program. However, there are some major differences between [17] and our work:

- The optimization model is different, as already discussed.
- In [17], an uncapacitated network model is used. Thus, its results are mostly relevant for network planning. In contrast, this paper considers strict capacity constraints. Therefore, our results are more relevant for the admission of connections in operational networks.
- In [17], the primary and backup LSPs are restricted to a predetermined set of routes. We have no such restriction.

- In [17], the unsplittable case is not addressed, while in this paper it is.

In [18], the work of [17] is extended by considering two ATM path restoration schemes: state independent, which is equivalent to our Global scheme, and state dependent, which is equivalent to our Unrestricted scheme. The optimization model of [18] is still SCA, and the unsplittable case is addressed.

Some works use non-ILP methods to address the SCA problem. The approach employed by [19] is based on a genetic algorithm. The algorithm utilizes crossover and mutation operators to evolve “good” solutions toward optimality. These operators force disjoint backup paths to share their bandwidth. The algorithm finds GR paths and can also deal with a non-linear cost function. In [20], a local search algorithm which adopts the tabu search technique is proposed. In [21], a two-phase approach is proposed. First, a set of link-disjoint paths for a given set of demands is found. Then, using this set, an ILP-based selection of primary and backup paths is made.

Only a few papers have proposed approximation algorithms with performance guarantees for the SCA problem or its variants. In [22], offline and online approximations for the FLR scheme are presented. These algorithms are based on approximation algorithms for the Steiner network problem. They present an $O(1)$ approximation for the case where the primary paths are predetermined. They also propose an algorithm for finding both primary and backup paths with an approximation ratio of $O(\log(n))$, where n is the number of nodes in the network. An $\frac{14}{15}$ -approximation algorithm that finds several backup paths for a single flow according to the GR scheme is presented in [23]. This algorithm reserves an integral part of the whole bandwidth for each of these backup paths, and it is shown to be the best possible for the considered problem.

Other works concentrate on non-SCA criteria. In [24], [25], several heuristics based on the UR scheme are proposed for minimizing restoration time and bandwidth consumption. In [26], two additional parameters are introduced into the online SCA problem. These parameters allow the length of the backup paths to be reduced and bandwidth consumption to be reduced further. In [27], the authors address the online SCA problem while guaranteeing an upper bound on the delay of both the primary and the backup paths. They use UR and show it to have better performance than GR.

A few papers compare some of the various recovery schemes presented in Section I. Most consider SCA as the optimization criterion. In [14], [28], [29], [30], [31], [16], [17], the main focus is on the GR and RLR schemes. Most of these papers use an ILP for optimizing SCA, and then approximate it. Their consensus is that GR outperforms RLR, because global backup paths are more flexible and generally traverse fewer links than RLR backup paths. Hence the GR scheme is more effective in decreasing the extra bandwidth that has to be reserved to ensure restorability.

Only a few papers use throughput maximization as an optimization criterion for the selection of primary and backup paths. In [11], an FPTAS based on the primal-dual approach is developed. An FPTAS (fully polynomial-time approximation scheme) is an approximation algorithm that takes a parameter

$\epsilon > 0$ and produces a solution that is within a factor ϵ of being optimal. The running time of an FPTAS is required to be polynomial in the problem size and $1/\epsilon$. The proposed algorithm in [11] deals with the splittable version of the problem, where each demand can be split into several primary and backup paths. This algorithm uses the RLR scheme. In [32], simple heuristics that find backup lightpaths for the restoration of IP over WDM are presented. This paper also compares the throughput and the restoration time of each algorithm.

III. PROBLEM DEFINITION AND COMPUTATIONAL COMPLEXITY

Two types of event failures can be considered: a link failure and a node failure. In [2] it was observed that link failures account for about 70% of the total network failures. Due to length constraints, we focus our discussion here on link failures. In Section VI we extend the discussion and explain how to apply our results to node failures as well.

Throughout the paper we deal with offline optimization. Namely, we assume that the operator knows the characteristics of the LSPs in advance. The justification for the offline model is that most of the LSPs are reserved in advance. For instance, a typical VoIP provider will reserve an LSP (VoIP trunk) with a certain bandwidth between two VoIP gateways for certain hours every day. Hence, the network operator can execute the offline algorithms once in a while, in order to optimize its revenue for the next hour(s).

In this paper we address two cases. In the first we assume that the primary LSPs are predetermined and only the backup LSPs need be established, while in the second case we need to establish both. Most of this paper is devoted to the former case, for two reasons. First, from a practical viewpoint, it is difficult for the network operator to retain joint optimization of primary and backup LSPs because of the implication that existing primary LSPs will require constant rearrangement to meet changing demands. Rerouting backup LSPs when necessary has far less impact on network operation than rerouting primary LSPs. It is therefore more practical to set up the primary LSPs, and then optimize the establishment of the backup LSPs separately. The second reason is that by setting up the primary LSPs in advance, we can focus our attention on comparing the performance of the various recovery schemes.

We refer to the problem where both primary and backup LSPs should be established as the *Restorable Flow Problem* (RFP). The problem where primary LSPs are given and only backup LSPs must be established is referred to as the *Primary-restricted Restorable Flow Problem* (PRFP). For each of the two problems we study the splittable and the unsplittable variants. In the splittable variant, a flow demand can be arbitrarily split over an unbounded number of LSPs. This variant is particularly applicable to MPLS networks, where traffic can be forwarded over multiple LSPs according to some classification rules. For WDM optical networks, this approach is less valid because the number of predetermined paths between two nodes is usually very limited. In the unsplittable variant, a flow can only be routed over a single LSP. In

the next subsections we formally define these problems and address their computational complexity. Table III summarizes our main results in this section. For every problem and every recovery scheme, the table indicates whether the problem is \mathcal{NP} -complete, or in \mathcal{P} , or its complexity is unknown (?). For some of the \mathcal{NP} -complete problems we also give a lower bound on their approximation ratio.

A. The Splittable Primary-restricted Restorable Flow Problem (S-PRFP)

We now define the *Splittable Primary-restricted Restorable Flow Problem (S-PRFP)* with respect to each recovery scheme. For simplicity, we assume throughout the paper that only one primary LSP is established for each flow. However, the results of the paper can be easily extended for the case where every flow has several primary LSPs. Let $G = (V, E)$ be a directed graph. Let u_e be the bandwidth capacity of edge $e \in E$. Let $F \subseteq V \times V$ be a set of source-destination pairs representing traffic flow demands. For every traffic flow $f = (s_f, t_f) \in F$, let s_f be the source node, t_f the target node, d_f the bandwidth demand, P_f the sequence of edges along the primary LSP, and w_f the revenue for f . A feasible solution is one that admits some of the traffic flows into the network while meeting the edge capacity constraints. Each admitted flow is routed on its primary LSP and must be fully restorable in the face of any single link failure. Hence, for every admitted flow f and edge $e \in P_f$, there must exist a set of backup LSPs that satisfies the constraints of the considered recovery scheme and can accommodate the admitted traffic of f when e fails. The objective is to maximize the total revenue of the admitted traffic flows.

Note, the traffic demand of each admitted flow need not be fully satisfied. Moreover, following a failure, the admitted traffic of a flow may be split among several backup LSPs. Therefore, this splittable version of the problem is more applicable to the case where the network can technically split each flow into smaller sub-flows. A good example for this is a VoIP trunk between two media gateways. Such a trunk should carry thousands of low bandwidth VoIP calls at any given time. Hence, when necessary, the flow carried by this trunk can be divided into smaller sub-flows, each carrying only a portion of the traffic. Allowing multiple LSPs to back up a single primary LSP is especially attractive when no single backup LSP can accommodate the entire flow demand. However, multiple backup LSPs incur more signaling and state overhead.

Theorem 1: S-PRFP is in \mathcal{P} for all recovery schemes discussed in Section I.

Proof

To show this, we formulate the problem as a linear program. We first present the constraints of the problem that are common to all recovery schemes. Then, we present additional constraints for each individual scheme. We define the following variables:

- $y_{fe}^{\bar{e}}$ – the fraction of d_f routed over edge e when edge \bar{e} fails; when no edge fails, $\bar{e} = \phi$.
- x_f – the total routed fraction of d_f .

The target function is to maximize the total gained revenue:

Maximize $\sum_f w_f \cdot x_f$
subject to the following constraints:

$$(C-1) \quad \sum_{e=(u,v)} y_{fe}^{\bar{e}} - \sum_{e=(v,u)} y_{fe}^{\bar{e}} = \begin{cases} x_f & v = t_f \\ -x_f & v = s_f \\ 0 & \text{else} \end{cases}$$

$$\forall v \in V, \forall f \in F, \forall \bar{e} \in \{E, \phi\}$$

$$(C-2) \quad \sum_f d_f \cdot y_{fe}^{\bar{e}} \leq u_e \quad \forall e \in E, \forall \bar{e} \in \{E, \phi\}$$

$$(C-3) \quad y_{fe}^{\phi} = 0 \quad \forall f \in F, \forall e \notin P_f$$

$$(C-4) \quad y_{fe}^e = 0 \quad \forall f \in F, \forall e \in E$$

$$(C-5) \quad 0 \leq x_f \leq 1, \quad 0 \leq y_{fe}^{\bar{e}} \leq 1 \quad \forall e \in E, \forall \bar{e} \in \{E, \phi\}, \quad \forall f \in F.$$

The set (C-1) of constraints ensures flow conservation. The set (C-2) ensures that no edge carries more than its capacity. The set (C-3) ensures that when no failure occurs, each flow is routed only along its primary LSP. The set (C-4) ensures that no flow is routed over a failed link. Finally, the set (C-5) of constraints ensures that the total routed bandwidth of each flow and the routed bandwidth on each backup LSP do not exceed flow demand.

The above constraints do not restrict the backup LSPs to be built according to a specific recovery scheme. The program selects, for every routed flow and for every failed link, a set of backup LSPs that can carry the flow's demand following a failure. Technically, according to these constraints, a flow may be diverted to a set of backup LSPs even if it is not routed through the failed edge. In fact, the above linear program solves $|E|$ different instances of the splittable flow problem, one for each failed link. The only connection between these $|E|$ instances is the requirement that the routed demand of each flow in every instance must be the same (i.e., x_f). It is clear that such a flexible "recovery scheme," which has almost no restriction, would yield the best performance.

However, each of the recovery schemes presented in Section I imposes a set of additional constraints on the backup LSPs. The more constraints a recovery scheme imposes, the less flexible and efficient it is. We now present the specific set of constraints for each recovery scheme.

The specific set of constraints for the LR scheme is:

$$(LR-1) \quad y_{fe}^{\bar{e}} \geq y_{fe}^{\phi} \quad \forall f \in F, \forall \bar{e} \in E, \{e | e \in E, e \neq \bar{e} \text{ and } e \text{ is not a downstream edge of } \bar{e} \text{ along } P_f\}.$$

The above set of constraints ensures that the backup LSP of f for $\bar{e} = (u, v)$, assuming $\bar{e} \in P_f$, will follow the primary LSP all the way from the source to u . From node u to the destination node, the backup LSP is not constrained.

The specific set of constraints for the RLR scheme is:

$$(RLR-1) \quad y_{fe}^{\bar{e}} \geq y_{fe}^{\phi} \quad \forall f \in F, \forall \bar{e} \in E, \{e | e \in E, e \neq \bar{e}\}.$$

RLR-1 is similar to LR-1, except that it also ensures that if a backup LSP protects against a failure of edge $\bar{e} = (u, v)$, it will follow the primary LSP not only from the source to u but also from v to the destination.

Since S-PRFP allows the traffic of the failed primary LSP to be split between several backup LSPs, RLR may use an

	Recovery schemes					
	GR	LR	RLR	FLR	EkFLR	UR
S-PRFP (Sec. III-A)	\mathcal{P} (Theorem 1)	\mathcal{P} (Theorem 1)	\mathcal{P} (Theorem 1)	\mathcal{P} (Theorem 1)	\mathcal{P} (Theorem 1)	\mathcal{P} (Theorem 1)
U-PRFP (Sec. III-B)	\mathcal{NP} -C (Theorem 2)	\mathcal{NP} -C (Theorem 2)	\mathcal{NP} -C (Theorem 2)	\mathcal{NP} -C (Theorem 2)	\mathcal{NP} -C (Theorem 2)	\mathcal{NP} -C (Theorem 2)
	no $ E ^{\frac{1}{2}-\epsilon}$ -apx. (Theorem 3)	no $ E ^{\frac{1}{2}-\epsilon}$ -apx. (Theorem 3)	?	?	?	no $ E ^{\frac{1}{2}-\epsilon}$ -apx. (Theorem 3)
S-RFP (Sec. III-C)	?	?	\mathcal{P} (Theorem 6)	\mathcal{P} (Theorem 6)	\mathcal{P} (Theorem 6)	?
U-RFP (Sec. III-C)	\mathcal{NP} -C (Theorem 4)	\mathcal{NP} -C (Theorem 4)	\mathcal{NP} -C (Theorem 4)	\mathcal{NP} -C (Theorem 4)	\mathcal{NP} -C (Theorem 4)	\mathcal{NP} -C (Theorem 4)
	no $ E ^{\frac{1}{2}-\epsilon}$ -apx. (Theorem 5)	no $ E ^{\frac{1}{2}-\epsilon}$ -apx. (Theorem 5)	?	?	?	no $ E ^{\frac{1}{2}-\epsilon}$ -apx. (Theorem 5)

TABLE III
SUMMARY OF THE COMPUTATIONAL COMPLEXITIES OF THE PROBLEMS

unbounded number of backup LSPs for each link failure. Hence, it is easy to see that an optimal solution for the FLR scheme and for the EkFLR scheme can be produced from an optimal solution for the RLR scheme. Hence, there is no need to specify special constraints for these two recovery schemes.

The specific set of constraints for the GR scheme is:

$$\begin{aligned}
\text{(GR-1)} \quad y_{f\bar{e}}^{\bar{e}} & \begin{cases} = 0 & \forall \bar{e} \in E, \{f|f \in F, \bar{e} \in P_f\} \\ & , e \in P_f \\ & \text{otherwise} \end{cases} \\
\text{(GR-2)} \quad y_{f\bar{e}}^{\bar{e}} - y_{f\bar{e}}^{\phi} & = \Delta y_{f\bar{e}}^{\bar{e}} \quad \forall f \in F, \forall \bar{e}, e \in E \\
\text{(GR-3)} \quad \Delta y_{f\bar{e}}^{\bar{e}_1} & = \Delta y_{f\bar{e}}^{\bar{e}_2} \quad \forall f \in F, \forall \bar{e}_1, \bar{e}_2 \in P_f \\
& \text{where } \bar{e}_2 \text{ immediately} \\
& \text{follows } \bar{e}_1 \text{ on } P_f, \forall e \in E.
\end{aligned}$$

The set (GR-1) of constraints ensures that the backup LSPs of every flow whose primary LSP crosses the failed edge must be edge disjoint with the primary LSP. The set (GR-2) introduces auxiliary variables $\Delta y_{f\bar{e}}^{\bar{e}}$. Each of these variables represents the difference between the bandwidth of f routed on the primary LSP, and the bandwidth of f to be routed on the backup LSPs that protect the flow against the failure of edge \bar{e} . This difference yields a circular traffic flow that traverses the backup LSPs from the source to the destination and the primary LSP in the reverse direction. The set (GR-3) of constraints ensures that for each flow the same set of backup LSPs is used to protect all the edges along the primary LSP.

Finally, the set of specific constraints for the UR scheme is:

$$\text{(UR-1)} \quad y_{f\bar{e}}^{\bar{e}} \geq y_{f\bar{e}}^{\phi} \quad \forall \bar{e}, e \in E, \forall f \in F, \bar{e} \notin P_f.$$

This set ensures that if the failed link is not included in the primary LSP of a flow, then the backup LSP is identical to the primary LSP. Otherwise, the set of backup LSPs has no constraint. \square

From the above discussion it is obvious that RLR imposes more constraints than LR. Hence, we expect LR to perform better. In addition, it is clear that UR outperforms all the other recovery schemes. However, it is not clear from the above discussion whether GR outperforms LR, or vice versa. On the one hand, LR is more restricted in that its backup LSPs must follow the route of the primary LSP all the way to the failed link, whereas the backup LSPs of GR must only avoid using

the links of the primary LSP. On the other hand, GR requires the same set of backup LSPs to protect against all possible failures on the primary LSP, whereas LR imposes no such restriction.

B. The Unsplittable Primary-restricted Restorable Flow Problem (U-PRFP)

We now address the *Unsplittable Primary-restricted Restorable Flow Problem (U-PRFP)*. There are two differences between U-PRFP and S-PRFP. First, in U-PRFP, a revenue can be obtained for a flow only when its entire demand is satisfied. Second, in U-PRFP, the traffic of each flow can be restored using only a single backup LSP. We now address the computational complexity of U-PRFP. In the decision variant of this problem, we are given a number K and ask whether there exists a feasible solution that yields a revenue equal to or larger than K .

Theorem 2: U-PRFP is \mathcal{NP} -complete for all recovery schemes.

This can be shown using a simple reduction from the knapsack problem [33].

Theorem 3: U-PRFP for GR, LR, or UR schemes cannot be approximated within $|E|^{1/2-\epsilon}$ unless $\mathcal{P} = \mathcal{NP}$.

Proof

We start by showing that U-PRFP for GR, LR, or UR schemes can be reduced from the *Unsplittable Flow Problem (U-FP)* [6], [7]. An instance of U-FP contains $G = (V, E)$, u_e , F , d_f , w_f and K , which are all the same as in U-PRFP. The target is to route a subset $F' \subseteq F$ of the flows such that their demands are fully satisfied and the edge capacity constraints are met. Each routed flow must use a single path. The question is whether there exists such a subset F' whose total revenue is greater than K .

We consider an instance of U-FP and show a reduction to an U-PRFP instance for the GR scheme. In the following we add a superscript r to all the parameters constructed for U-PRFP. Let $G^r = (V^r, E^r)$ be a directed graph with $V^r = V \cup \{u_1, u_2\}$ and $E^r = E \cup \{(v, u_1), (u_2, v) | v \in V\} \cup \{(u_1, u_2)\}$. For every $e \in E$, let $u_e^r = u_e$; otherwise, let $u_e^r = \infty$. Figure 2 depicts G^r .

Let $F^r = F$. For each flow $f \in F^r$, let $d_f^r = d_f$ and $w_f^r = w_f$. For every $f = (s_f, t_f) \in F^r$, let $P_f^r = (s_f, u_1, u_2, t_f)$.

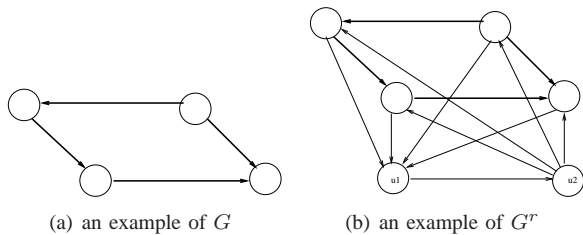


Fig. 2. An example of the reduction in the proof of Theorem 3

Namely, for every flow we construct a primary LSP that passes through the new vertices. Note that these paths do not pass through the original edges of G . Finally, we define $K^r = K$. It is easy to see that this U-PRFP instance can be constructed in polynomial time.

In the constructed U-PRFP instance, the failed edge (u_1, u_2) constrains the maximum revenue for any feasible solution. It is easy to see that any feasible solution for U-FP is equivalent to a feasible backup routing for U-PRFP under such failure. Hence, the existence of a solution for U-FP with a revenue of more than K implies the existence of a backup routing for the U-PRFP instance with a revenue of more than K^r , and vice versa.

A similar reduction can be used to show that U-PRFP is \mathcal{NP} -complete for the UR or LR schemes.

The above reduction is an L-reduction with the constants $\alpha = \beta = 1$ [34], because any feasible solution for a U-FP instance has the same value as for the constructed U-PRFP instance. In [34] it is shown that an L-reduction is an approximability preserving reduction. Hence, using the above reduction, every approximation algorithm for U-PRFP can be translated to an approximation algorithm for U-FP with the same approximation ratio. This implies that the best approximation ratio that can be achieved for U-PRFP is not worse than the best approximation ratio that can be achieved for U-FP. Since U-FP is \mathcal{NP} -hard to approximate within $|E|^{1/2-\epsilon}$ (see [6]), U-PRFP with GR, LR, or UR is also \mathcal{NP} -hard to approximate within $|E|^{1/2-\epsilon}$. \square

C. The Unsplittable and Splittable Restorable Flow Problems (U-RFP and S-RFP)

We now address the *Unsplittable Restorable Flow Problem* (U-RFP) and the *Splittable Restorable Flow Problem* (S-RFP). Recall that the goal of these problems is to establish not only the backup, but also the original (primary) LSPs. U-RFP establishes one primary LSP for every flow, and one backup LSP for every failure event along the selected primary LSP. A revenue is obtained for an admitted flow only if its entire demand is satisfied. In contrast, S-RFP can split the traffic over several primary LSPs. Every edge along these LSPs can be protected by several backup LSPs. The demand of every flow can be partially satisfied, in which case only part of the revenue is obtained.

Theorem 4: U-RFP is \mathcal{NP} -complete for all recovery schemes discussed in Section I. This is a trivial consequence of Theorem 2.

Theorem 5: U-RFP for GR, LR, or UR cannot be approximated within $|E|^{1/2-\epsilon}$ unless $\mathcal{P} = \mathcal{NP}$.

A similar reduction to the one presented in Theorem 2 for U-PRFP using GR, LR and UR can be used to reduce U-FP to U-RFP. The only difference is that the reduction does not set the primary LSPs for the flows. This is a suitable reduction, because if there exists a solution for U-FP with a revenue greater than K , then there must exist a solution for U-RFP with a revenue greater than K^r that routes all primary LSPs over (u_1, u_2) and all backup LSPs over the routes used in the U-FP solution. If there exists a solution for U-RFP that routes a set of flows whose revenue is greater than K^r , there must exist a routing that does not go through (u_1, u_2) for the same flows. Hence, there is a solution for U-FP with a revenue greater than K .

The above reduction is also approximability preserving. Therefore, as for U-PRFP, this implies that U-RFP is \mathcal{NP} -hard to approximate within $|E|^{1/2-\epsilon}$. \square

Theorem 6: S-RFP is in \mathcal{P} for RLR, FLR, and EkFLR schemes.

Proof sketch

The linear program constraints for S-PRFP with RLR do not depend on the primary LSPs. Thus, we can use this linear program for S-RFP without the set of constraints (C-3) that restricts the primary LSPs. Hence, we get that S-RFP with RLR can be solved in polynomial time. As noted in Section III-A, it can be easily shown that the optimal solutions for S-RFP with RLR, FLR and EkFLR are the same. This is because, in S-RFP, there is no bound on the number of backup LSPs. \square

Note that Theorem 6 improves the results presented in [11], where only an FPTAS was shown for S-RFP with RLR.

To formulate the S-RFP using the other recovery schemes (GR, LR, and UR) we need to use path-indexed variables, namely variables that indicate for each flow the routed bandwidth on every possible path in the graph. However, since the number of such paths is exponential in the size of the graph, it is not easy to solve this formulation in polynomial time. We therefore leave the complexity of S-RFP with GR, LR and UR open for future research.

IV. ALGORITHMS FOR U-PRFP AND U-RFP

A. Approximation Algorithms

We now present approximation algorithms for U-PRFP and U-RFP for the case where edge capacities are all equal and the bandwidth demand of each individual flow does not exceed half of the edge capacities. While these assumptions are not always realistic, these algorithms have theoretical value due to their worst case performance guarantee. The presented algorithms guarantee an approximation ratio of $|E|^{1/2}$. The algorithms are similar to those presented in [7] for U-FP, but have a different analysis. The algorithms can be used with every recovery scheme, and are based on the following simple observation.

Observation 1: If the load on each edge does not exceed half of the edge capacity, following a single failure every flow can be fully restored using arbitrary backup LSPs.

We first present the algorithm for U-PRFP. The algorithm begins by solving a similar problem, called the *Splittable Primary-restricted Flow Problem* (S-PFP). In this problem each flow can only be routed along a primary LSP given in advance, and restorability is not considered. Moreover, in this problem a flow can be only partially satisfied. A linear program for the problem follows. The edge capacities are denoted by u . We use the same variables as defined in Section III:

Maximize $\sum_f w_f \cdot x_f$

subject to the following constraints:

$$(S-PFP-1) \quad \forall v \in V, \forall f \in F$$

$$\sum_{e=(u,v)} y_{fe}^\phi - \sum_{e=(v,u)} y_{fe}^\phi = \begin{cases} x_f & v = t_f \\ -x_f & v = s_f \\ 0 & \text{else} \end{cases}$$

$$(S-PFP-2) \quad \sum_f d_f \cdot y_{fe}^\phi \leq u \quad \forall e \in E$$

$$(S-PFP-3) \quad y_{fe}^\phi = 0 \quad \forall f \in F, \forall e \notin P_f$$

$$(S-PFP-4) \quad 0 \leq x_f \leq 1, 0 \leq y_{fe}^\phi \leq 1 \quad \forall e \in E, \forall f \in F.$$

The approximation algorithm solves this linear program and applies a randomized rounding procedure that yields a solution for U-PRFP. Let x_f^* be the value given by the optimal solution of the linear program to the variable x_f . The algorithm then routes each flow f over the primary LSP with probability x_f^*/γ , where γ is a constant larger than 1.

Theorem 7: By choosing a proper value for γ , there exists a deterministic algorithm, based on the above randomized one, which produces a solution in which the load imposed on every link does not exceed half of the link capacity, and the total revenue is lower bounded by $\Omega(w_{S-PFP}^*/|E|^{1/2})$, where w_{S-PFP}^* is the value of the optimal solution for S-PFP.

The proof is presented in Appendix A.

By Observation 1, the flows routed by the algorithm can be fully restored in the face of a single link failure using any recovery scheme. It is obvious that w_{S-PFP}^* is greater than the optimal solution for S-PRFP with any recovery scheme. Hence, w_{S-PFP}^* is also greater than the optimal solutions for U-PRFP with any recovery scheme. We can therefore conclude that the algorithm can be used as an approximation algorithm for U-PRFP with an approximation ratio of $O(|E|^{1/2})$.

Next, we present an algorithm for U-RFP with an approximation ratio of $O(|E|^{1/2})$. The algorithm begins by solving the *Splittable Flow Problem* (S-FP). In this problem primary LSPs are not given in advance, and restorability is not considered. The linear program for this problem is the same as the one presented above for S-PFP, but without the set (S-PFP-3) of constraints. After finding an optimal solution for this linear program, every flow f is routed over a possibly empty set of Γ_f LSPs. Each LSP in Γ_f carries a fraction z_{fk}^* of the demand, where $1 \leq k \leq |\Gamma_f|$. The algorithm chooses to route the entire demand of flow f on the k -th LSP with probability z_{fk}^*/γ , independently of the other LSPs in Γ_f . If more than one LSP is chosen, only one is arbitrarily selected.

Theorem 8: By choosing a proper value for γ , there exists a deterministic algorithm, based on the above randomized one, which produces a solution in which the load imposed on every link does not exceed half of the link capacity, and the total revenue is lower bounded by $\Omega(w_{S-FP}^*/|E|^{1/2})$, where w_{S-FP}^* is the value of the optimal solution of S-FP.

The proof is similar to the one presented in Appendix A for Theorem 7. It is omitted for lack of space.

As for U-PRFP, by Observation 1, the flows admitted by the algorithm can be fully restored in the face of a single link failure using any recovery scheme. It can also be shown that w_{S-FP}^* must be greater than the optimal value for U-RFP with any recovery scheme. Therefore, the algorithm guarantees an approximation ratio of $O(|E|^{1/2})$ for U-RFP.

B. Heuristics

The algorithms presented earlier for U-PRFP and U-RFP have theoretical value due to their worst case performance guarantee. However, we expect that their average performance will not be satisfactory, since they only utilize up to half of the edge capacities. Moreover, since the algorithms do not depend on a specific recovery scheme, performance of the various schemes cannot be compared. Therefore, in this section we present heuristics that are based on the above approximation algorithms, but yield better average performance and allow us to compare the performance of the various recovery schemes. Due to lack of space we only present here the heuristic for U-PRFP. However, it can be extended to U-RFP in a straightforward manner.

The heuristic begins by solving the linear program presented in Section IV for S-PFP. Let x_f^* be the value given to the variable x_f by the optimal solution of the linear program. We then sort the flows in a non-increasing order of w_f/d_f . Then, for each flow, we apply the randomized rounding procedure presented above for the U-PRFP approximation algorithm. If the flow is selected by the randomized rounding procedure, we verify that (a) the flow can be routed on its primary LSP without violating the capacity constraints, and (b) for the chosen recovery scheme, feasible backup LSPs that do not violate the capacity constraints also exist. If both conditions hold, the flow is admitted along with its backup LSPs. If there are several feasible backup LSPs, the shortest one is selected.

The version for U-RFP is the same as the version for U-PRFP. However, for U-RFP the heuristic solves the linear program of S-FP (as presented in Section IV) and continues with the randomized rounding procedure presented above. If the flow is selected by the randomized rounding procedure, we first verify that there is a primary LSP over which the flow can be routed without violating its capacity constraints. In addition, we verify that for the primary LSP and chosen recovery scheme, feasible backup LSPs that do not violate the capacity constraints also exist. If both conditions hold, the flow is admitted along with its primary and backup LSPs. If there are several feasible primary or backup LSPs, the shortest ones are selected.

V. SIMULATION STUDY

In this section we evaluate the performance of the algorithms presented in Sections III and IV for the various recovery schemes. We use the BRITE simulator [35] to simulate MPLS domain topologies according to the Barabasi-Albert model [36]. This model captures two important characteristics of

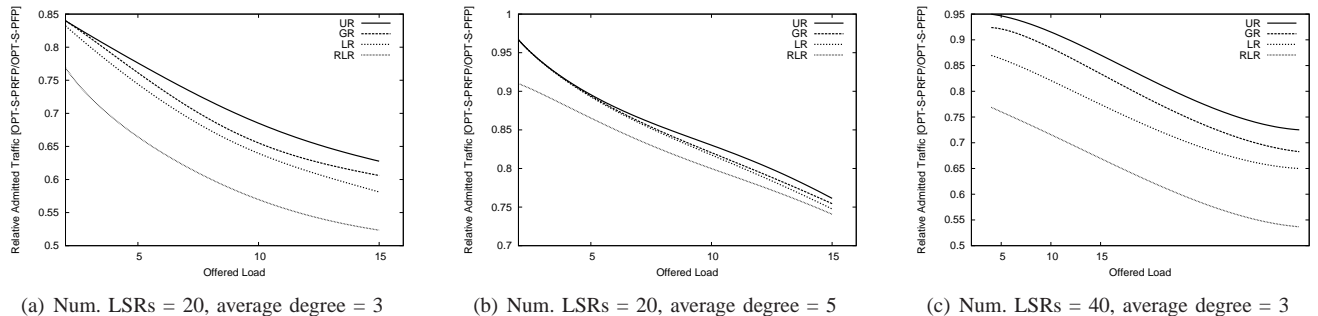


Fig. 3. The relative performance for OPT-S-PRFP for various MPLS domains

network topologies: incremental growth and preferential connectivity of a new label switch router (LSR) to well-connected existing LSRs. These characteristics yield a power-law degree distribution of the LSRs.

To validate our results, we also use actual ISP topologies, as inferred by the RocketFuel project [37]. The bandwidth for each link is based on the results reported in [38]. For each synthetic or real topology, we generate a set of flows according to a power-law distribution. A network topology and a set of flows form together one simulation instance. In the case of PRFP, the simulation instance also contains the primary LSP for each flow. For the primary LSP of each flow, we select the shortest path. For each simulation instance, we determine the backup LSPs using the following algorithms:

- 1) An optimal algorithm for S-PRFP that solves the linear program presented in Section III-A. In the following, this algorithm is referred to as OPT-S-PRFP.
- 2) An optimal algorithm for S-PFP that solves the linear program presented in Section IV. In the following, this algorithm is referred to as OPT-S-PFP.
- 3) The heuristic for U-PRFP presented in Section IV-B, based on the approximation algorithm for U-PRFP. In the following, this algorithm is referred to as HEU-U-PRFP.
- 4) An optimal algorithm for S-RFP that solves the linear program presented in Section III-C (using RLR). In the following, this algorithm is referred to as OPT-S-RFP.

To solve the various linear programs, we use the Lp_Solve software [39].

We start by evaluating the various recovery schemes using the OPT-S-PRFP algorithm. Figure 3 depicts the results for three types of MPLS domains: (a) with 20 LSRs whose average node degree is 3 links; (b) with 20 LSRs whose average node degree is 5; and (c) with 40 LSRs whose average node degree is 3. To compare the performance of the various recovery schemes, we use a relative performance metric: the ratio between the revenue of flows admitted by OPT-S-PRFP and the revenue of flows that can be admitted when no backup LSPs have to be established (OPT-S-PFP). This relative performance metric indicates the “penalty” incurred by the restoration requirement. This metric is represented by the y -axis of all the graphs in Figure 3, while the offered load is represented by the x -axis. The value of the offered load is the

average number of flows originated by each router.

As expected, it is evident from all three graphs that UR yields the best performance while RLR yields the worst. In addition, Figure 3 shows that GR yields higher revenue than LR. However, the performance of UR, GR and LR differs only marginally (5% on the average), whereas RLR lags behind by about 15%. It is also evident that the performance differences for the domain with 20 LSRs and average degree of 5 are small (Figure 3(b)). This is attributed to the very short length (usually one or two links) of the primary LSPs, when established over shortest paths, in such MPLS domains. Consequently, the differences between the various recovery schemes cannot really be expressed. Since the domain with 40 LSRs and average degree of 3 (Figure 3(c)) has longer primary LSPs, the differences between the recovery schemes are much more visible. For instance, there is a difference of up to 20% between UR and RLR. In addition, in larger domains there are more backup LSPs to choose from. This further widens the gap between the more flexible and the less flexible schemes.

It is also evident from all the graphs in Figure 3 that the penalty associated with building backup LSPs increases as the offered load increases. This can be intuitively explained by the fact that a more heavily loaded network requires that more flows be rejected in order for backup LSPs to be established. Moreover, the penalty of the backup LSPs is much higher for smaller domains, or for those with lower average degree. For example, the relative performance of UR in the domain with 20 LSRs and degree of 3 ranges between 0.67 and 0.85, where in the other domains it ranges between 0.73 and 0.95. This can be attributed to the fact that in the larger or denser domains the number of possible backup LSPs is higher, and the recovery scheme may be able to choose backup LSPs that allow more flows to be admitted.

Next, we evaluate the performance of the heuristic for U-PRFP. In this case, we use a different relative performance metric: the ratio between the revenue of flows admitted by the U-PRFP heuristic and the revenue of flows admitted by OPT-S-PRFP using the same recovery scheme. This relative performance metric indicates the penalty incurred due to the inability to split the traffic following a failure. The metric is represented by the y -axis of the graph in Figure 4, while the x -axis represents the offered load. Figure 4 depicts the results for the same three types of MPLS domains used for Figure 3.

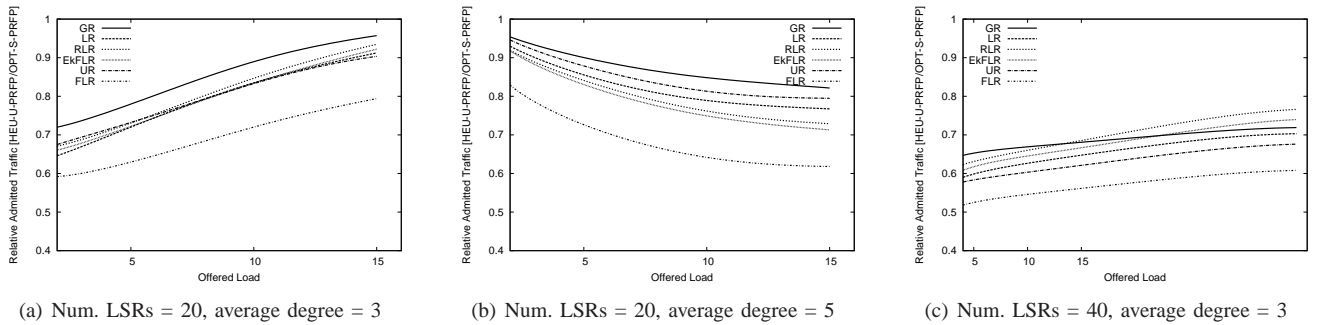


Fig. 4. The relative performance of the heuristic for U-PRFP for various MPLS domains

For the EkFRL scheme, we use $k = 2$.

Figure 4 suggests that in most cases the relative penalty of a single backup LSP while using the GR scheme is slightly smaller than for the other recovery schemes, including the UR scheme. This means that the performance difference between the unsplitable and splitable cases is smaller for GR than for the other recovery schemes. This may be explained by the fact that GR is the only scheme that needs to find a single unsplitable backup LSP for the entire primary LSP, whereas the other schemes need to cope with the harder task of finding several unsplitable backup LSPs.

For almost all recovery schemes the performance difference is marginal, and the heuristic exhibit excellent performance and yield an average of roughly 80% of the optimal revenue. This implies that the cost of using a recovery scheme with a single backup LSP rather than several is only about 20% of the revenue. In particular, the average relative performance of EkFRL with $k = 2$ is almost the same as the average relative performance of RLR. Since, as explained in Section III-A, both schemes have the same performance for OPT-S-PRFP, there is no real need to split the backup traffic into more than two LSPs. In contrast, the average relative performance of FLR is only about 0.65. The higher penalty implies that one backup LSP per link may be too strict.

For domains with 20 LSRs whose degree is 3 (Fig. 4(a)) we see that when the offered load increases, the penalty of the requirement not to split the flow decreases and the performance of the heuristic is comparable to the performance of OPT-S-PRFP. This result is surprising, because we expect that in a highly loaded network the ability to split the traffic of each flow across several paths would contribute both to the flexibility and to the performance. This result can be explained by the fact that for all flows we use the shortest path as a primary LSP. Due to the power-law distribution of the flows, many primary LSPs traverse a relatively small number of links. Hence, when the network becomes congested, the primary LSPs create bandwidth bottlenecks on these links. In a typical simulation instance, we found that roughly 80% of the load on these links is attributed to the primary LSPs. Thus, despite of the ability of the optimal S-PRFP algorithm to back up more flows, many of these flows cannot be admitted due to the lack of available bandwidth on their primary LSPs. Hence, somewhat surprisingly, the effectiveness of splitting the

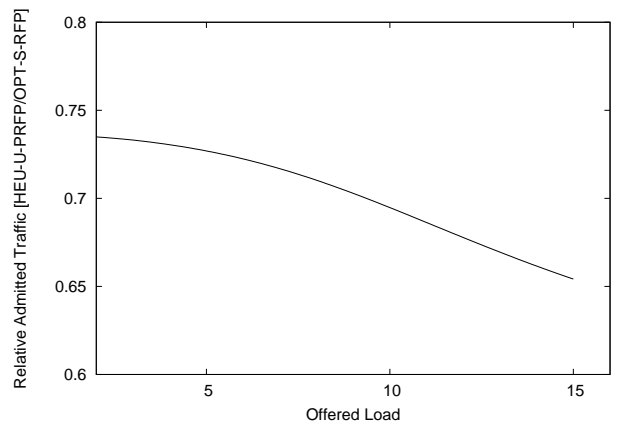


Fig. 5. The relative performance of U-PRFP using RLR in a network with 20 LSRs and an average degree of 3

backup LSPs diminishes as the load on the network increases.

The results for the higher degree domains, depicted in Fig. 4(b), demonstrate the opposite behavior: the penalty decreases as the offered load increases. This is attributed to the fact that the offered load induced a lighter congestion in the network compared to the former case. Therefore, the primary LSPs do not create bandwidth bottlenecks, and the increased flexibility of using several backup LSPs plays a more dominant role. The results for domains with 40 LSRs with an average degree of 3, depicted in Fig. 4(c), exhibit the same behavior as in Fig. 4(a). By significantly increasing the offered load, we created high congestion that leads to bottlenecks on the routes of the primary LSPs.

To give a more complete picture, we examine the performance of HEU-U-PRFP and OPT-S-RFP when the restriction on the primary LSP is eliminated and the flows can be split both for the primary and the backup LSPs. In such a case, the load imposed by the shortest-path primary LSPs on the bottleneck links is reduced. Fig. 5 depicts simulation results for a network with 20 LSRs and an average degree of 3. It shows the ratio between the performances of HEU-U-PRFP and OPT-S-RFP for the RLR scheme. As expected, as the offered load increases, the penalty for using unsplitable primary and backup LSPs increases, and splitting the flows is more profitable.

We now evaluate the penalty of using a single primary LSP for each flow. To this end we use the following relative performance metric: the ratio between the revenue of flows that are admitted by OPT-S-PRFP and the revenue of flows that are admitted by OPT-S-RFP using the RLR scheme. This relative performance metric indicates the penalty incurred when using a single primary LSP set in advance. This metric is represented by the y -axis of the graph in Figure 6, which depicts the results for two types of 20-LSR MPLS domains: one with average degree of 3 and another with average degree of 5. It is evident that as the offered load increases, so does the penalty for using a single primary LSP set in advance. This relation is not surprising since a highly loaded network requires the traffic to be split into several paths in order to maximize the admitted traffic. It is also evident that the penalty increases for a network with a higher average degree. This is because a higher network degree gives more options for splitting the traffic between two end nodes.

To validate the results from the synthetic graphs and power-law traffic distribution, we also present results for real AS topologies, inferred by the RocketFuel project [37], and for traffic demand distribution based on the Gravity model [40]. In this model, the offered load from node u to node v is proportional to the product of the total volume of traffic exiting u and the total volume of traffic entering v . The results for these simulations are presented in Figure 7. The topologies used are of the Exodus ISP, which consists of 80 routers and 147 links, and of the Telstra ISP, which consists of 115 routers and 153 links. In Figure 7 we compare the recovery schemes using the following relative performance metric: the ratio between the revenue of flows admitted by U-PRFP and the revenue of flows that can be admitted when no backup LSPs have to be established. This metric, represented by the y -axis of the graphs in Figure 7, reveals the penalty incurred by both the restoration requirement and the use of a single backup LSP. It is evident that the comparative performance of the various recovery schemes in both topologies is similar to what we found for the synthetic graphs and power-law traffic distribution (see Figures 3 and 4). Namely, GR outperforms LR, and EkFLR (with $k = 2$) is close to RLR. It is interesting to note that in the Telstra topology the performance of the various recovery schemes is worse than in the Exodus topology. Moreover, the difference in the performance gained by the various schemes is less noticeable. This can be attributed to the Telstra's AS lower link degree, which substantially reduces the path diversity in the network. This, in turn, reduces the ability to construct diverse backup LSPs.

To summarize, the main conclusions we draw from the simulation study are:

- The performance differences between UR, GR, and LR are only marginal (5% on the average) while RLR is considerably worse. Hence, LR should be the recovery scheme of choice due to its short restoration time.
- If low administrative overhead is the main goal, EkFLR with $k=2$ should be preferred over RLR.
- Heuristics U-PRFP achieves close to optimal revenue.
- When the primary LSPs are set in advance in congested networks, splitting the backup LSPs yields only a small

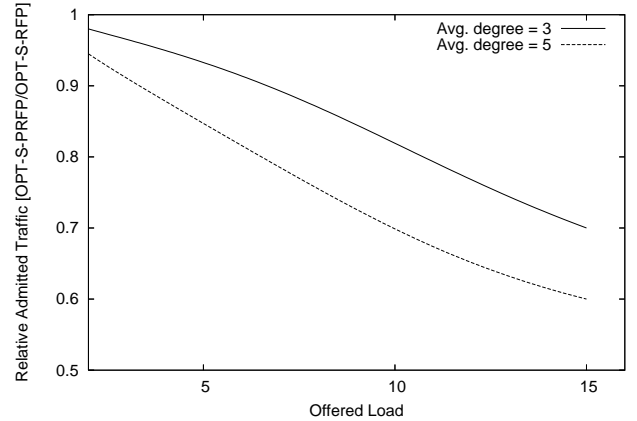


Fig. 6. The relative performance of S-PRFP using RLR

added revenue (less than 10%).

- In non-congested networks, the added revenue is small for joint optimization of primary and backup LSPs (less than 10%).

VI. NODE FAILURE

In this section we extend the results of the paper to the case of node failures. A node failure causes the failure of all edges incident to it. In particular, a failure of a single node may take down two links along an LSP.

A. The Splittable Problems (S-PRFP and S-RFP)

S-PRFP for node failures is in \mathcal{P} . To show this we change the linear program for link failures presented in Section III-A in the following ways. First, the variable $y_{fe}^{\bar{v}}$ indicates the fraction of demand of flow f routed over edge e after a failure of node \bar{v} . When no node fails, $\bar{v} = \phi$. The constraints presented in Section III-A are changed as follows:

$$(C-1) \quad \sum_{e=(u,v)} y_{fe}^{\bar{v}} - \sum_{e=(v,u)} y_{fe}^{\bar{v}} = \begin{cases} x_f & \text{if } v = t_f \\ -x_f & \text{if } v = s_f \\ 0 & \text{otherwise} \end{cases}$$

$$(C-2) \quad \sum_f d_f \cdot y_{fe}^{\bar{v}} \leq u_e \quad \forall e \in E, \forall \bar{v} \in \{V, \phi\}$$

$$(C-3) \quad y_{fe}^{\phi} = 0 \quad \forall f \in F, \forall e \notin P_f$$

$$(C-4) \quad y_{fe}^{\bar{v}} = 0 \quad \forall f \in F, \forall \bar{v} \in V, \\ e \text{ incident to } \bar{v}$$

$$(C-5) \quad 0 \leq x_f \leq 1, 0 \leq y_{fe}^{\bar{v}} \leq 1 \quad \forall e \in E, \forall \bar{v} \in \{V, \phi\}, \\ \forall f \in F$$

$$(LR-1) \quad y_{fe}^{\bar{v}} \geq y_{fe}^{\phi} \quad \forall f \in F, \forall \bar{v} \in V, \{e|e \in E, e \text{ is not} \\ \text{downstream upstream neighbor of } \bar{v} \\ \text{along } P_f\}$$

$$(RLR-1) \quad y_{fe}^{\bar{v}} \geq y_{fe}^{\phi} \quad \forall f \in F, \forall \bar{v} \in V, \{e|e \in E, e \text{ is not} \\ \text{incident to } \bar{v}\}$$

$$(UR-1) \quad y_{fe}^{\bar{v}} \geq y_{fe}^{\phi} \quad \forall \bar{v} \in V, \forall e \in E, \forall f \in F, \bar{v} \notin P_f$$

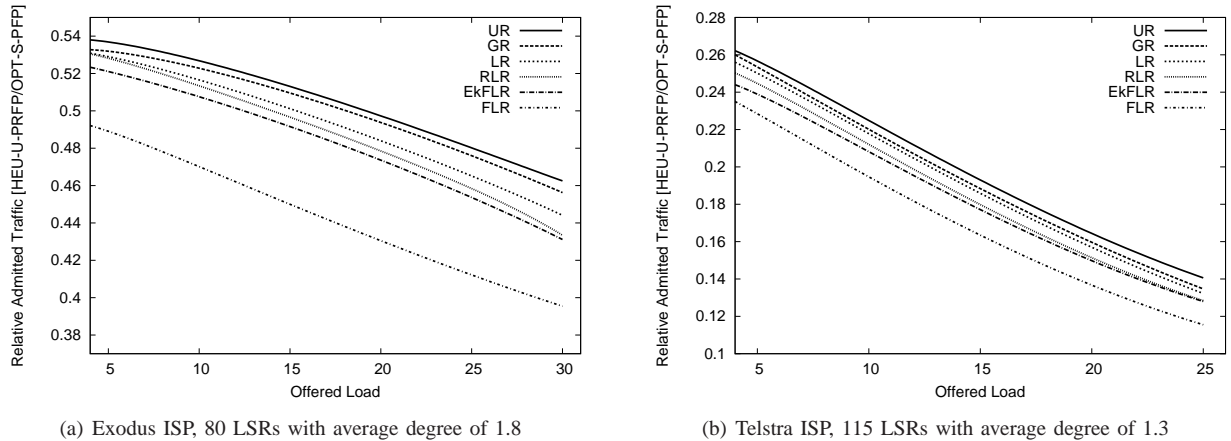


Fig. 7. The relative performance of U-PRFP using the various recovery schemes for real ISP topologies and for traffic demands generated according to the Gravity model

$$\begin{aligned}
 \text{(GR-1)} \quad y_{f\bar{v}}^{\bar{v}} & \begin{cases} = 0 & \forall \bar{v} \in V, \\ & \{f|f \in F, \bar{v} \in P_f\}, \\ & e \in P_f \\ \geq y_{f\bar{v}}^{\phi} & \text{otherwise} \end{cases} \\
 \text{(GR-2)} \quad y_{f\bar{v}}^{\bar{v}} - y_{f\bar{v}}^{\phi} & = \Delta y_{f\bar{v}}^{\bar{v}} \quad \forall f \in F, \forall \bar{v} \in V, \\ & \quad \forall e \in E \\
 \text{(GR-3)} \quad \Delta y_{f\bar{v}_1}^{\bar{v}_1} & = \Delta y_{f\bar{v}_2}^{\bar{v}_2} \quad \forall f \in F, \forall \bar{v}_1, \bar{v}_2 \in P_f \\ & \quad \text{where } \bar{v}_2 \text{ immediately} \\ & \quad \text{follows } \bar{v}_1 \text{ on } P_f, \\ & \quad \forall e \in E
 \end{aligned}$$

As for link failures, S-RFP for the RLR scheme can be solved using the constraints (C-1), (C-2), (C-4), (C-5), and (RLR-1).

B. The Unsplittable Problems (U-PRFP and U-RFP)

U-PRFP and U-RFP are \mathcal{NP} -complete. This can be shown using the same reduction presented in Section III-B from U-FP. The maximal revenue in the constructed instance of U-PRFP is upper bounded by the maximal revenue when one of the nodes added by the reduction fails. This revenue is bounded by the maximal revenue for the U-FP instance, and vice versa. The reduction is also approximability preserving, implying that, in the case of node failures, U-PRFP and U-RFP cannot be approximated within $|E|^{1/2-\epsilon}$.

Assume that all edge capacities are equal to u . Let us define $r^{\max} = \max\{r_{\text{in}}^{\max}, r_{\text{out}}^{\max}\}$, where r_{in}^{\max} and r_{out}^{\max} are the maximum in-degree and out-degree in G , respectively.

Observation 2: If the load on each edge does not exceed $u/(r^{\max}+1)$, then each flow can be fully restored using arbitrary backup LSPs.

This observation follows from the fact that if a node fails, then the total amount of bandwidth that must be restored does not exceed $u \cdot r^{\max}/(r^{\max}+1)$. This amount can be fully routed on any edge. Hence, any arbitrary backup LSP can be used to restore the traffic.

An approximation algorithm very similar to the one presented in Section IV can be adapted for node failures. However, here we assume that the bandwidth demand of each flow

does not exceed $u/(r^{\max}+1)$. Using analysis as in Appendix A, it can be shown that the load imposed by the algorithm does not exceed $u/(r^{\max}+1)$ and its maximum throughput is not lower than $\Omega(w^*/|E|^{1/2})$.

VII. CONCLUSIONS

We presented the first comprehensive study of maximizing restorable throughput in MPLS networks. We focused on the establishment of backup LSPs when the primary LSPs are already set. We showed that the splittable version of the problem is in \mathcal{P} while the unsplittable version is \mathcal{NP} -complete and cannot be approximated within $|E|^{1/2-\epsilon}$. We gave an algorithm with an approximation ratio of $|E|^{1/2}$ for the case where the bandwidth demand of an individual flow does not exceed half of the edge capacities. We developed two practical and efficient heuristics that were shown to achieve excellent performance. Using simulation, we compared the performance of the various MPLS recovery schemes. We showed that LR should be the scheme of choice since it has the fastest restoration time and almost the same performance as the best (UR) scheme. In addition, we showed that if reducing the administrative cost is the main concern, EkFLR with $k=2$ should be the recovery scheme of choice.

REFERENCES

- [1] G. Iannaccone, C.-N. Chuah, R. Mortier, S. Bhattacharyya, and C. Diot, "Analysis of link failures in an IP backbone," in *IMW*, November 2002.
- [2] A. Markopulu, G. Iannaccone, S. Bhattacharyya, C.-N. Chuah, and C. Diot, "Characterization of failures in an IP backbone," in *Proceedings IEEE INFOCOM*, March 2004.
- [3] C. Alaettinoglu, V. Jacobson, and H. Yu, "Toward millisecond IGP convergence," in *Proceedings NANOG 20*, October 2000.
- [4] E. Rosen, "Multiprotocol label switching architecture (MPLS)," RFC 3031, January 2001.
- [5] V. Sharma and F. Hellstrand, "Framework for multi-protocol label switching (MPLS)-based recovery," IETF RFC 3469, February 2003.
- [6] V. Guruswami, S. Khanna, R. Rajaraman, F. Shepherd, and M. Yannakakis, "Near-optimal hardness results and approximation algorithms for edge-disjoint paths and related problems," *Journal of Computer and System Sciences*, vol. 67(3), pp. 473–496, 2003.
- [7] A. Srinivasan, *Approximation Algorithms via Randomized Rounding: a Survey. Lectures on Approximation and Randomized Algorithms*. PWN, Warszawa: Polish Scientific Publishers, 1999.

- [8] P. Pan *et al.*, “Fast reroute extensions to RSVP-TE for LSP tunnels,” IETF RFC 4090, May 2005.
- [9] M. Alicherry and R. Bhatia, “Simple pre-provisioning scheme to enable fast restoration,” *IEEE/ACM Transactions on Networking*, vol. 15, no. 2, pp. 400–412, 2007.
- [10] I. Chlamtac, A. Ganz, and G. Karmi, “Lightpath communication: An approach to high bandwidth optical wans,” *IEEE Transactions on Communication*, vol. 7, pp. 1171–1182, 1992.
- [11] R. Bhatia, M. S. Kodialam, T. V. Lakshman, and S. Sengupta, “Capacity allocation and routing of locally restorable bandwidth guaranteed connections,” in *Proceedings of the IEEE INFOCOM*, vol. 4, 2005, pp. 2691–2701.
- [12] Y. Liu, p. Tipper, and D. Siripongwutikorn, “Approximating optimal spare capacity allocation by successive survivable routing,” *IEEE/ACM Transactions on Networking*, vol. 13(1), pp. 198–211, February 2005.
- [13] M. Herzberg, S. Bye, and U. Utano, “The hop-limit approach for spare capacity assignment in survivable networks,” *IEEE/ACM Transactions on Networking*, pp. 775–784, December 1995.
- [14] B. V. Caenegem, N. Wauters, and Demeester, “Spare capacity assignment for different restoration in mesh survivable networks,” in *Proceedings IEEE ICC*, vol. 1, 1997, pp. 288–292.
- [15] B. V. Caenegem, W. V. Parys, F. D. Turck, and P. M. Demeester, “Applying tabu search to spare capacity planning for network restoration,” *IEEE Journal on Selected Areas in Communications*, vol. 16(7), pp. 1146–1157, September 1998.
- [16] T. Frisanco, “Optimal spare capacity design for various switching methods in ATM networks,” in *Proceedings IEEE ICC*, vol. 1, 1997, pp. 293–298.
- [17] Y. Xiong and L. Mason, “Restoration strategies and spare capacity requirements in self-healing ATM networks,” *IEEE/ACM Transactions on Networking*, vol. 7, no. 1, pp. 98–110, 1999.
- [18] —, “Comparison of two path-restoration schemes in self-healing networks,” *Computer Networks*, vol. 38, no. 5, pp. 663–674, 2002.
- [19] A. Al-Rumaih, D. Tipper, Y. Liu, and B. A. Norman, “Spare capacity planning for survivable mesh networks,” in *Proceedings IFIP-TC6 Networking, Lecture Notes in Computer Science (LNCS) 1815*, May 2000, pp. 957–968.
- [20] C.-C. Shyur, T.-C. Lu, and Ue-PyngWen, “Applying tabu search to spare capacity planning for network restoration,” *Computers and Operations Research*, vol. 26(10), pp. 1175–1194, October 1999.
- [21] E. Yetginer and E. Karasan, “Robust path design algorithms for traffic engineering with restoration in MPLS networks,” in *Proceedings of the Seventh ISCC*, 2002, pp. 933–938.
- [22] C. Chekuri, A. Gupta, A. Kumar, S. Naor, and D. Raz, “Building edge-failure resilient networks,” *Integer Programming and Combinatorial Optimization (IPCO)*, pp. 439–456, 2002.
- [23] G. Brightwell, G. Oriolo, and F. B. Shepherd, “Reserving resilient capacity in a network,” *SIAM J. Discret. Math.*, vol. 14, no. 4, pp. 524–539, 2001.
- [24] A. Gupta, B. N. Jain, and S. Tripathi, “QoS aware path protection schemes for MPLS networks,” in *Proceedings of the 15th ICC*, 2002, pp. 103–118.
- [25] L. Li, B. M.M., C. Chekuri, and K. Guo, “Routing bandwidth guaranteed paths with local restoration in label switched networks,” in *Proceedings of 10th IEEE International Conference on Network Protocols*, February 2002, pp. 110–120.
- [26] Y. Xiong, D. Xu, and C. Qiao, “Achieving fast and bandwidth efficient shared-path protection,” *Journal of Lightwave Technology*, vol. 21(2), pp. 365–371, February 2003.
- [27] Y. Bejerano, Y. Breitbart, A. Orda, R. Rastogi, and A. Sprintson, “Algorithms for computing QoS paths with restoration,” in *Proceedings of IEEE INFOCOM*, April 2003.
- [28] R. Doverspike and B. Wilson, “Comparison of capacity efficiency of DCS network restoration routing techniques,” *Journal of Network and Systems Management*, vol. 2, no. 2, pp. 1064–1070, 1994.
- [29] R. R. Iraschko, M. H. MacGregor, and W. D. Grover, “Optimal capacity placement for path restoration in STM or ATM mesh-survivable networks,” *IEEE/ACM Trans. Netw.*, vol. 6, no. 3, pp. 325–336, 1998.
- [30] M. Patel, R. Chandrasekaran, and S. Venkatesan, “A comparative study of restoration schemes and spare capacity assignments in mesh networks,” in *Proceedings of ICCCN*, 2003, pp. 399–404.
- [31] S. Ramamurthy and B. Mukherjee, “Survivable WDM mesh networks, part I - protection,” in *Proceedings of IEEE INFOCOM*, vol. 2, 1999, pp. 744–751.
- [32] N. S. C. Correia and M. C. R. Medeiros, “Protection schemes for IP-over-WDM networks: Throughput and recovery time comparison,” *Journal Photonic Network Communications*, vol. 11, no. 2, pp. 127–149, 2006.
- [33] M. R. Garey and D. S. Johnson, *Computers and Intractability: A Guide to the Theory of NP-Completeness*. San Francisco: W. H. Freeman and Co., 1979.
- [34] C. H. Papadimitriou, *Computational Complexity*. Addison-Wesley Publishing, Co., 1994.
- [35] A. Medina, A. Lakhina, I. Matta, and J. Byers, “BRITE: An approach to universal topology generation,” in *Proceedings of MASCOTS*, 2001.
- [36] A. Barabasi, R. Albert, and H. Jeong, “Scale-free characteristics of random networks: the topology of the world-wide web,” in *Physica A: Statistical Mechanics and Its Applications*, vol. 281, June 2006, pp. 69–77.
- [37] N. Spring, R. Mahajan, and D. Wetherall, “Measuring ISP topologies with rocketfuel,” in *Proceedings of the ACM SIGCOMM*, August 2002.
- [38] R. Mahajan, N. Spring, D. Wetherall, and T. Anderson, “Inferring link weights using end-to-end measurements,” in *ACM SIGCOMM Internet Measurement Workshop (IMW)*, November 2002.
- [39] M. Berkelaar, “Lp_solve software,” ftp.es.tue.nl/pub/lp_solve.
- [40] J. Kowalski and B. Warfield, “Modeling traffic demand between nodes in a telecommunications network,” in *Proceedings of ATNAC*, 1995.

APPENDIX A A PROOF FOR THEOREM 7

The proof is similar to the proof of the approximation ratio of $|E|^{1/2}$ for U-FP presented in [7] except that our aim is to bound the load on every edge to $1/2$ rather than 1. We analyze the randomized rounding process in the approximation algorithm for U-PRFP. The purpose of this analysis is to show that by choosing an appropriate γ , the load imposed on every edge does not exceed half the edge capacity, and the total revenue is not less than $\Omega(w_{S-PFP}^*/|E|^{1/2})$.

Let \hat{x}_f be a Bernoulli random variable for choosing flow f . The success probability of \hat{x}_f is x^*/γ . The expected total load imposed on every link $e \in E$ by the end of this process is

$$E[L_e] = E\left[\sum_{i \in P_f} d_f \hat{x}_f\right] = \sum_{i \in P_f} d_f x_f^*/\gamma \leq u/\gamma. \quad (1)$$

The inequality follows from constraints (S-PFP-2) in the linear program. For the sake of simplicity, we scale the demands and capacities such that $u = 1$.

For every $e \in E$, let \mathcal{E}_f denote the “bad event” that $L_e > 1/2$. We now construct well-behaved estimators for these events. Consider first the case, where $d_f \leq 1/4$ holds for every f . Define $d'_f = 4d_f$ and $L'_e = \sum_{i \in P_f} d'_f \hat{x}_f$ for all $e \in E$. From (1) it follows that $\mu \doteq E[L'_e] \leq 4/\gamma$. From [7] we know that a possible well-behaved estimator for \mathcal{E}_f is χ_e^1 :

$$\chi_e^1 = \frac{\prod_{i \in P_f} (1 + \delta)^{d'_f \hat{x}_f}}{(1 + \delta)^{\mu(1 + \delta)}},$$

where $\delta = 2/\mu - 1$. From [7] we also know that

$$E[\chi_e^1] \leq G(4/\gamma, \gamma/2 - 1),$$

where $G(\mu, \delta) \doteq \left(\frac{e^\delta}{(1 + \delta)^{\mu(1 + \delta)}}\right)^\mu$.

Next, consider the case where $d_f \geq 1/4$ for every f . In this case, the event \mathcal{E}_f holds, if and only if more than 2 flows whose primary LSP traverses e are chosen. From [7] it can be shown that a well-behaved estimator for \mathcal{E}_f is χ_e^2 :

$$\chi_e^2 = \psi_2\left(\{\hat{x}_f | e \in P_f\}\right),$$

$$\Psi_2(z) = \sum_{1 \leq i_1 < i_2 \leq n} z_{i_1} z_{i_2}, \text{ for } (z_1, z_2, \dots, z_n) \in \mathcal{R}^n$$

Since in this case Eq. (1) implies that $\sum_{i|e \in P_f} E[\hat{x}_f] \leq 4/\gamma$, $E[\chi_e^2] = 8/\gamma^2$, must hold.

Next, we construct a proper estimator for the bad event “ $w(T) < w^*(T)/(2\gamma)$ ”, where $w(T) \doteq \sum_{i \in T} w_f \hat{x}_f$, and $w^*(T) \doteq \sum_{i \in T} w_f x_f^*$. We know that $\mu_1 \doteq E[w(T)] = w^*(T)/\gamma$. From [7] follows that a proper estimator for the above bad event is

$$\chi_w(T) = \frac{\prod_{i \in T} (1 - \delta_1)^{w_f \hat{x}_f}}{(1 - \delta_1)^{\mu_1 (1 - \delta_1)}}.$$

where $\mu_1(1 - \delta_1) = w^*(T)/(2\gamma)$. From [7], it is also known that

$$E[\chi_w] \leq H(w^*(T)/\gamma, 1/2)$$

where $H(\mu, \delta) \doteq e^{-\mu\delta^2/2}$.

Before proceeding to the approximation ratio of the algorithm, we present the following theorem from [7]:

Theorem 9: Let E_1, E_2, \dots, E_t be events and r, s be non-negative integers with $r + s \leq t$ such that:

- E_1, E_2, \dots, E_r are all increasing, with respective well-behaved estimators g_1, g_2, \dots, g_r ;
- E_{r+1}, \dots, E_{r+s} are all decreasing, with respective well-behaved estimators g_{r+1}, \dots, g_{r+s} ;
- E_{r+s+1}, \dots, E_t are arbitrary events, with respective proper estimators $g_{r+s+1}, g_2, \dots, g_t$;
- For every i , E_i and g_i are completely determined by \vec{X} .

Then, if

$$1 - \left(\prod_{i=1}^r (1 - E'[g_i]) \right) + 1 - \left(\prod_{i=r+1}^{r+s} (1 - E'[g_i]) \right) + \sum_{i=r+s+1}^t E'[g_i] < 1$$

holds, where $E'(\cdot) \doteq \min\{E(\cdot), 1\}$, we can efficiently construct a deterministic assignment for \vec{X} under which none of E_1, E_2, \dots, E_t holds. Note that: (a) empty products are taken to be 1; (b) If there is g_i such that $E[g_i] > 1$, then the entire product is equal to 0.

We continue by showing the $\Omega((w^*)^2/|E|)$ bound. Let I_0 be the subset of flows for which traffic demand is at most $1/4$ and $I_1 = I/I_0$. We first assume that $w_{I_1}^* \leq w_{I_0}^*$, i.e., $w_{I_0}^* \geq w^*/2$. Since the events \mathcal{E}_e are increasing, avoiding the bad events would require

$$1 - \left(\prod_e (1 - E'[\chi_e^1]) \right) + E'[\chi_w(I_0)] \leq 1. \quad (2)$$

For a suitably large constant c , $\gamma = c|E|/w^*$ satisfies the inequality of Eq. (2), since:

$$E[\chi_w(I_0)] \leq H(w^*(I_0)/\gamma, 1/2) \leq H(w^*/(2\gamma), 1/2).$$

Thus, from Theorem 9 follows that if $w^*(I_1) \leq w^*(I_0)$, we can efficiently select feasible paths for I_0 with objective function value $\Omega((w^*(I_0))^2/|E|) = \Omega((w^*)^2/|E|)$.

Next, we consider the case where $w^*(I_1) \geq w^*(I_0)$. As in the previous case, we must have

$$1 - \left(\prod_e (1 - E'[\chi_e^2]) \right) + E'[\chi_w(I_1)] \leq 1. \quad (3)$$

if we are to avoid all the bad events. Again, for a suitably large constant c , and from the same considerations given above, $\gamma = c|E|/w^*$ satisfies the inequality of Eq. 3. Thus, from Theorem 9 follows that if $w^*(I_1) \geq w^*(I_0)$, we can efficiently select feasible paths for I_1 with objective function value $\Omega((w^*(I_1))^2/|E|) = \Omega((w^*)^2/|E|)$.

To conclude the proof we need to show the $\Omega(w^*/\sqrt{|E|})$ bound. If $w^* \geq \sqrt{|E|}$, this bound immediately follows from the $\Omega((w^*)^2/|E|)$ bound. If $w^* < \sqrt{|E|}$, we can simply choose to admit a flow f for which $w_f = 1$. \square

APPENDIX B

EXTENSION TO MPLS OVER WDM LIGHTPATHS

Due to space constraints, we show here only how to extend the linear program of Section III-A. The extension of the other linear programs is similar. When a flow enters a lightpath, it can leave it only at the lightpath's egress node. Let l_i denote the set of (physical) links composing lightpath i , and $y_{fe}^{i\bar{e}}$ denote the part of $y_{fe}^{\bar{e}}$ traffic that traverses lightpath i . The following constraints should be added to the linear program:

$$\begin{aligned} \text{(O-1)} \quad & \sum_i y_{fe}^{i\bar{e}} = y_{fe}^{\bar{e}} \quad \forall f \in F, \forall \bar{e}, e \in E \\ \text{(O-2)} \quad & y_{fe_1}^{i\bar{e}} = y_{fe_2}^{i\bar{e}} \quad \forall f \in F, \forall \bar{e} \in E, \forall i, \forall e_1, e_2 \in l_i \\ \text{(O-3)} \quad & y_{fe}^{i\bar{e}} = 0 \quad \forall f \in F, \forall \bar{e} \in E, \forall i, \forall e \notin l_i \end{aligned}$$

Constraints (O-1) ensure that the sum of the traffic carried over all the lightpaths on a certain link is equal to the total traffic traversing this link. Constraints (O-2) ensure that the amount of traffic carried by a lightpath is equal on all the links composing this lightpath. Finally, constraints (O-3) ensure that no traffic traversing a lightpath is carried over links that are not included in that lightpath.

PLACE
PHOTO
HERE

Reuven Cohen (M'93, SM'99) received the B.Sc., M.Sc. and Ph.D. degrees in Computer Science from the Technion - Israel Institute of Technology, completing his Ph.D. studies in 1991. From 1991 to 1993, he was with the IBM T.J. Watson Research Center. Since 1993, he has been a professor in the Department of Computer Science at the Technion. He has also been a consultant for numerous companies, mainly in the context of protocols and architectures for broadband access networks. Dr. Cohen has served as an editor of the IEEE/ACM

Transactions on Networking, and the ACM/Kluwer Journal on Wireless Networks (WINET). He is also the chair of the Technical program Committee for Infocom 2010, and the head of the Israeli chapter of the IEEE Communications Society.

PLACE
PHOTO
HERE

Gabi Nakibly received the B. Sc. in Information Systems engineering (summa cum laude) and PhD in Computer Science from the Technion - Israel Institute of Technology, Haifa, Israel, in 1999 and 2008, respectively. His dissertation addressed QoS routing and traffic engineering. He is currently a network security researcher at Israel's National EW Research & Simulation Center where he is involved in the security analysis of network protocols and the secure deployment of network services.