Addressing robust Next-Generation Networks

G. Panza[1], G. Bigini[2], M. Prudente[3] and P. Belotti[4]
1, 2, 3 – CEFRIEL, Politecnico di Milano, via Fucini 2, 20133 Milano, Italy
4 - Tepper School of Business, Carnegie Mellon University, 5000 Forbes Ave, 15213 Pittsburgh PA

***Abstract* — Next-Generation Networks (NGNs) employ the Internet Protocol (IP) over a wide variety of packet-switching technologies, which often lack in fault resilience enabling features. features. An overlay MPLS infrastructure with its fast-reroute mechanisms can be deployed to overcome such an issue.**
Addressing NGNs robust to single link and node failures, an off-line method to effectively calculate working and recovery paths for highly demanding services, is proposed and analyzed.
The strength of our work is the ability to address two recovery techniques in a very simple manner, by formulating an Integer Linear Programming (ILP) problem, optimizing the bandwidth allocation while limiting the recovery time.

***Index Terms* — Fault resilience, FRR, ILP, MPLS, NGN, Traffic Engineering.**

I. Introduction

Nowadays, the Internet scenario is more and more characterized by value added services based on applications (e.g. VoIP, videoconferencing, video streaming) with stringent QoS (Quality of Service) requirements.
A critical issue to cope with is the robustness to network faults: link or node failures should be recovered quickly (as fast as 50 ms to be comparable with current SONET/SDH networks) and transparently to users.
Multi-Protocol Label Switching (MPLS) [1] is an advanced forwarding technology which uses the control plane of the IP routing protocol. The main idea of MPLS is to map packets to a specific QoS class, called Forwarding Equivalence Class (FEC) at the edge of the MPLS domain (at the ingress router) and use only FEC labels to process and forward the packets inside the domain.
Packet forwarding in an MPLS domain is performed only by using the data contained in the label, resulting in an increase in forwarding speed. The label is removed at the egress router of the MPLS domain. The issued route is named LSP (Label Switched Path), the ingress and egress routers LERs (Label Edge Routers) and the other routers LSRs (Label Switched Routers).
MPLS can react rapidly to faults by switching failed connections to secondary paths. General specifications and bandwidth reservation for traffic engineering and path recovery are discussed in [2], [3] and [4]. Providing reliable services in MPLS is studied and fast rerouting techniques are proposed in [5][6][22][23].
In this paper, we target in a straightforward manner the traffic engineering problem of a MPLS network which is fully recoverable against all single-link/node failures, relying on fast-reroute mechanisms. We assume that the MPLS network has a two-connected topology (i.e. each node pair admits at least two link-disjoint paths) with given link capacities and traffic demand set. For such a demand set, we calculate the working and recovery paths for all the requested connections, subject to capacity and recovery time constraints.
To cope more efficiently with any subsequent additions to the traffic demand set, we have designed an algorithm that can assign larger spare capacity to the links with higher likelihood of being crossed by traffic. The algorithm is based on a simple and computationally efficient Integer Linear Programming (ILP) model, which optimizes the bandwidth allocation while constraining the maximum recovery time within a given limit.
The remainder of this paper is organized as follows. The next section overviews and analyzes the existing path restoration schemes. Then a description of our method and the discussion of some practical cases are provided. Finally, the last section summarizes the paper and outlines the main conclusions together with the future developments.

II. Robustness with MPLS

Path recovery consists in rerouting traffic around a failure, i.e. all packets routed through a link that has failed are rerouted along an alternative path (called recovery path) [5].
There are two basic models for path recovery: *rerouting* and *protection switching*. Rerouting is a model that establishes a recovery path (RP) after a failure on its working path (WP). Protection switching is a model that establishes a RP prior to any failure on the WP, therefore it can support fast restoration schemes. Depending on how the repairs are carried out upon the occurrence of a failure on the WP, we deal with global repair or local repair. In global repair, an alternative path is established from the source to the end of the WP, protection is always activated on an end-to-end basis, irrespective of where a failure occurs. In local repair, a portion of WP is protected by a dedicated alternative path and protection is activated by each LSR that has detected a failure.
Local and global repair have different advantages [9], in practice most of the former schemes are based on rerouting with local repair [6]. While rerouting a demand with global repair can handle concurrent link failures more easily, it needs more network resources as the path has to be established from source to destination. On the other hand, local repair consumes less extra network capacity, but it is harder to reroute demands

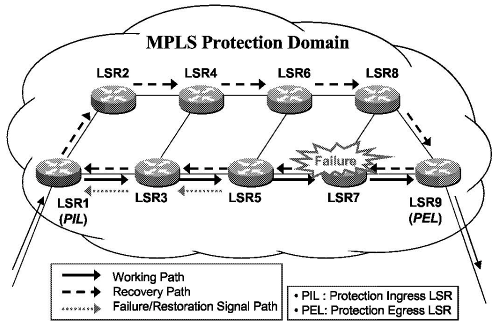

Fig. 1. Path recovery - Haskin's scheme and Makam's scheme.

in the event that multiple failures occur.

For this last reason, global repair schemes have been proposed [15][16], although those schemes have a drawback in solving problems such as resource utilization, which can be addressed by a shared restoration [17][18].

Two schemes using protection switching have been designed by Haskin and Krishnan [15], and Huang et al.[16], respectively. Both schemes are depicted in Fig. 1.

The straight line between LSR1 and LSR9 is the working path. In Haskin's scheme, a recovery path is established as follows:

1) The initial segment of the RP is established between PEL and PIL in the reverse direction of the WP.
2) The second and final segment of the RP is established between PIL and PEL along a transmission path that does not utilize any working path segment.
3) The initial and final segments of the RP are linked to form an entire recovery path.
4) When a failure occurs, the node detecting it reroutes incoming traffic by linking the upstream portion of the WP to the downstream portion of the RP. In Fig. 1, when the node LSR7 fails, the working traffic is rerouted along the RP LSR5–3–1–2–4–6–8–9.

The merit of the Haskin's scheme is that almost no packet loss occurs during link/node failure. However, it introduces re-ordering of packets in the event that traffic is switched back from the recovery path to the working path after a link/node goes up. To solve this issue, a buffering technique [22] can be employed.

In the Makam's scheme, a recovery path is established as follows:

1) A RP is established between PIL and PEL that does not use any link of the WP.
2) When a failure occurs, the node that has detected the failure sends a failure notification message toward its PIL. Upon receiving the message, PIL reroutes the incoming traffic through the RP. In Fig. 1, when the node LSR7 fails, the working traffic is rerouted along the RP LSR1–2–4–6–8–9.

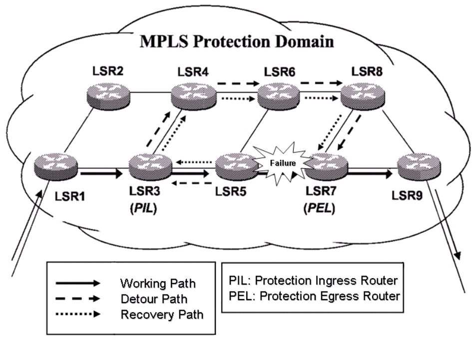

Fig. 2. One-to-One backup scheme.

The merit of the Makam's scheme is that almost no problems occur in reordering of packets during link/node failure. However, it introduces packet loss because the PIL does not execute the protection switching until it receives the failure notification message from a node detecting a link/node failure.

Another recovery mechanism with local repair, called One-to-One backup [6], can be used to reduce the recovery time. The idea is to deploy an alternative path, called detour path, between the ends of each protected portion of the working path. This way the length of the recovery path is decreased and consequently, the recovery time shortened. For fast reroute, the detour path needs to be pre-setup for each portion, i.e. link or node, of the working path. Therefore, to protect a working path composed of N nodes, N-1 detour paths are required.

Fig. 2 shows the One-to-One backup technique. The straight line between LSR1 and LSR9 is the working path. In the One-to-One backup scheme, a recovery path is established as follows:

1) The initial segment of the RP is established between PEL and PIL in the reverse direction of the protected portion of the WP. In Fig. 2, the link between LSR5 and LSR3 illustrates such a segment of the RP.
2) The second and final segment of the RP is established between PIL and PEL along a detour path that does not utilize any working path segment. In Fig. 2, the dashed line which crosses LSR 3-4-6-8-7 illustrates such a segment of the RP.
3) The initial and final segments of the RP are linked to form an entire RP.
4) When the link between LSR5 and LSR7 fails, the working traffic is rerouted along the RP LSR3–4–6–8-7.

III. WORK DESCRIPTION

The aim of our work is a method to compute a set of WP-RP pairs for global repair, and a set of detour paths (DPs) for the local protection of portions of a working path in a network, such that the whole working path is protected against faults. The computed WPs, DPs, and RPs must minimize the overall resource allocation (and thus maximize the residual network capacity).

As a starting point, we define a generic network topology with a set of ingress, transit and egress routers and a set of traffic demands. The network links may have different capacities and costs. The problem we tackle is to select, for every traffic demand, working and recovery paths that satisfy the following requirements:

- Capacity constraint: for every link, the overall required capacity on the link must not exceed its capacity.
- Protection constraint: each WP must always be protected by one recovery path (or set of DPs).
- Recovery time constraint: the recovery time should be as in a SONET/SDH network. (i.e. some tens of milliseconds)

The input data can be formalized as follows: a graph *G*, defined by a set of nodes *N* and a set of links *L*, is given with a set *Q* of traffic demands. Each demand *q*, in turn, is defined by an ingress node ($I_q, q \in Q$), an egress node ($E_q, q \in Q$), where $I_q, E_q \in N$, and the amount of bandwidth required, B_q. Finally, the link capacity is defined as $C_l, l \in L$.

From a theoretical standpoint, network design problems such as the one at hand fall in the class of *multi-commodity minimum cost flow problems* [7]. In particular, survivability in telecommunication networks has been a hot topic in the Operations Research community in the past decade, with several contributions that have shed some light on the structure of these difficult problems and provide a wealth of efficient solution algorithms.

Significant works (e.g. [8] and [10]) have been done to model and solve network engineering problems with survivability constraints.

The structure of the two-connected network design problem has also been studied in [11][12]. An ILP formulation for a problem similar to ours has been introduced in [27]. The authors describe a segment protection scheme and an ILP model which can only be applied to small networks. A Dynamic Programming scheme is then used to obtain a solution in more realistic cases.

Heuristics are a more common choice than exact algorithms, in solving real-world network engineering problem, given the difficulty of the problem at hand.

The novelty of our approach is that we use the ILP model, and an exact ILP solver, to obtain an initial solution, even for large real networks. The problem we study is NP-hard, as it contains the minimum cost two-connected graph [12] as a special case.

At first glance, such a routing problem can be modeled by defining one variable for each link *l* and for each traffic demand *q* – the well known *edge-flow formulation*. Consider a set of variables f_{ij}^q, defined for each directed arc $l = (i,j) \in L$ and each traffic demand $q \in Q$. Consider another set of variables y_{ij}, defined on each arc in L as the residual capacity. The objective function, to be maximized, is then the sum of all residual capacities $\sum_{(i,j)\in L} y_{ij}$, subject to a conservation of flow on variables f_{ij}^q:

$$\sum_{j\in N:(i,j)\in L} f_{ij}^q - \sum_{j\in N:(j,i)\in L} f_{ji}^q = b_i^q \quad \forall i \in N, \forall q \in Q$$

where we define, for all nodes *i* and all demands *q*,

$$b_i^q = \begin{cases} 1 & i = I_q \\ -1 & i = E_q \\ 0 & otherwise \end{cases}$$

and a constraint that links flow variables to the residual capacities in the objective function,

$$\sum_{j\in V:(i,j)\in E} B_q f_{ij}^q + y_{ij} = C_{ij} \quad \forall (i,j) = l \in L$$

This simple formulation can be adapted to a problem with survivability by either (a) simply duplicating the right-hand side of the flow conservation constraint (as flow variables are binary they will be forced to define two link-disjoint paths) or (b) adding an extra set of flow variables g_{ij}^q, with similar constraints as f_{ij}^q, and which would then be added to define the residual capacity,

$$\sum_{j\in V:(i,j)\in E} B_q (f_{ij}^q + g_{ij}^q) + y_{ij} = C_{ij} \quad \forall (i,j) = l \in L,$$

However, since both f_{ij}^q and g_{ij}^q are binary variables, the resulting Mixed-Integer Linear Program is barely tractable. In other words, such a formulation can be hardly tailored to a problem of working and recovery path calculation, and could only be solved within reasonable time for networks of limited size (10-15 nodes), even with state-of-the-art ILP solvers.

A more usual approach is an elaborated formulation with a potentially very large set of variables, of which only a subset is employed, that allows for a smaller formulation and thus can be solved in reasonable time even for large real networks. More specifically, let us define a set of paths *P* into subsets P_q containing all paths from I_q to E_q. The binary *path variable* z_p^q is then defined for all $p \in P_q$. Variable z_p^q is equal to one if path *p* is used as a working path for demand *q*. Similarly, we define variable w_p^q equal to one if path *p* is the recovery path for demand *q*. A new formulation, called *path formulation*, follows:

$$\max \sum_{(i,j)\in L} y_{ij}$$

$$\sum_{p\in P_q} z_p^q \ge 1 \quad \forall q \in Q$$

$$\sum_{p \in P_q} w_p^q \geq 1 \quad \forall q \in Q$$

$$\sum_{q \in Q} B_q \left(\sum_{p \in P_q : (i,j) \in p} z_p^q + \sum_{p \in P_q : (i,j) \in p} w_p^q \right) + y_{ij} = C_{ij} \quad \forall (i,j) \in L$$

where the first two constraints are equivalent to conservation of flow and require that at least one (or equivalently, exactly one) path is used for each demand q. The third constraint is also similar to the one used in the edge flow formulation, as the quantity $\sum_{p \in P_q : (i,j) \in p} z_p^q$ equals the amount of working flow through link (i,j).

It is worth noting that (i) the number of constraints for this formulation drops from $2|Q||N|+|L|$ to $2|Q|+|L|$; (ii) the number of variables, instead, can grow exponentially with the size of the network. The latter disadvantage can be overcome by defining a restricted formulation where only a subset of paths is used at the beginning, and a so-called *column generation* scheme is used to introduce new path variables in the formulation. This approach has been successfully used in the past in network design but also for a several classes of problems such as airline crew scheduling.

A. Recovery path decomposition – the U-shaped paths

In order to search for recovery paths, we point out that for a given WP we can find a U-shaped detour path between the ingress and egress node of the WP portion to be protected. Let us call "U-path" this basic detour path, as depicted in Fig. 3. We can decompose the U path into four portions between the four delimiting routers A, B, C, and D, as shown in Fig. 4. From this basic decomposition we see that every U has three <portions, with the following meaning in a network topology.

- First portion: reverse WP segment (A → B);
- Second portion: a shortcut (B → C);
- Third portion: a RP segment (C → D).

An extra portion, consisting of a return path from the RP to the WP (D → A), is used only in the case of One-to-One backup (i.e. for DP determination). It is worthwhile to highlight that every U implements both the previously discussed target Fast Re-Routing (FRR) techniques at the same time.

In order to form a complete WP-RP path pair where every portion of a given WP is protected, we only need to chain several U by collapsing the destination of a detour path with the source of the following one, as shown in Fig. 5.

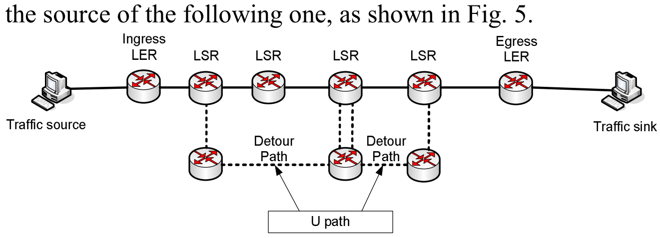

Fig.3. U path examples

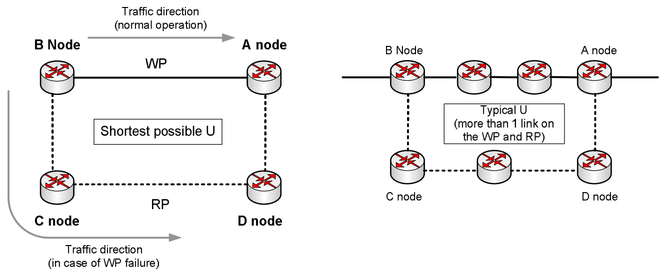

Fig. 4. U-path decomposition.

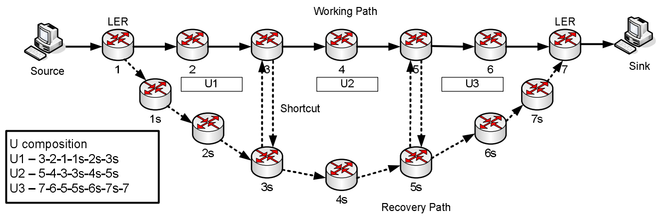

Fig. 5. Example of resulting U chain.

With this scheme, every link belonging to the WP has a protection DP (essentially, the U) and the "third portions" of the chained U form a complete recovery path, analogous to the one calculated with the classical FRR model. Our model implements the shortcut mechanism as defined by Haskins, since the "second portion" of the U path can be considered as the shortcuts to the RP (see Fig.5).

A similar approach, proposed by Hong et al. [13], called *segment-based protection*, also considers re-routing paths that restore connectivity around a failing link. However, the authors choose a sub-optimal two-step method, where first the working paths are selected, and then the backup paths are generated taking into account of the available capacity of the network. Some other proposals of this type (see for instance [14]) assume that the capacity for the working path is allocated only, while the capacity for the recovery paths is sought upon failure of a link in the working path, which does not assure the same recovery time as in SONET/SDH networks.

An ILP formulation needs a further tweak to introduce the U-paths. Instead of a pure protection scheme, involving only one protection path, disjoint from the working one, we developed a restoration scheme where a link failure is dealt with a dedicated recovery path, as in Fig. 3. Hence, we cannot resort to an extra class of variables such as g_{ij}^q (or w_p^q) as there are different recovery paths used for different failure scenarios. The aggregation paradigm that holds in the path formulation above can however be applied to our problem as follows: instead of using flow variables for each directed arc, and rather than associating one variable to each path from origin to destination, we define *U-path* variables. Consider a set of directed arcs defined between points A, B, C, and D of Fig. 4. We associate a portion of the working and the restoration resource to a single variable and a solution will be defined by a set of binary variables u_{AB}^{CD} that represent the concatenated *U-paths* and that are equal to one if the U-path is used in the solution. Although we only specify four nodes A,B,C, and D, in our implementation all variables u are associated with a U-path. As appears from Fig. 4, variables u need to be "connected" to each other in such a way that, for all U-path variables of each demand q, for each variable u_{AB}^{CD} equal to one:

1) there is another variable $u_{A'B'}^{C'D'}$ such that $B' = A$;

2) there is another variable $u_{A'B'}^{C'D'}$ such that $C' = D$.

Moreover, there must be at least one $u_{A'M}^{C'M}$ where $M = E_q$, that is, there must be one U-path closing the structure onto the egress node.

Consider the set U of all U-paths and the set U_q of all U-paths that can be used to route demand q. Conditions 1) and 2) above are treated as pseudo-flow conservation constraints. Finally, the flow conservation constraints of the edge-flow and the path formulations are replaced by the following two constraints:

$$\sum_{u \in U_q:(X,*,*,*) \in u} u_{XB}^{CD} = \sum_{u \in U_q:(*,X,*,*) \in} u_{AX}^{CD} \quad \forall X \in N \setminus \{I_q, E_q\}$$

$$\sum_{u \in U_q:(*,*,X,*) \in u} u_{AB}^{XD} = \sum_{u \in U_q:(*,*,*,X) \in u} u_{AB}^{CX} \quad \forall X \in N \setminus \{I_q, E_q\}$$

It is worthwhile to highlight that the only nodes "hooked" to each other are node A of one semi-path to node B of the successive one, and node D of one semi-path to node C of the successive one. Pseudo-conservation constraints for the source and destination of each traffic demand are defined analogously. The capacity constraint is replaced by

$$\sum_{q \in Q} B_q \sum_{s \in U_q:(i,j) \in s} u_{AB}^{CD} + y_{ij} = C_{ij} \quad \forall (i,j) \in L.$$

It is worth noting that a sort of "flow conservation" constraints is implied when using U variables rather than flow variables: a chain of U can only be established if the proper U are placed in the right sequence, from ingress to egress.

To sum up, we introduce special graph structures (U-paths) and use them in an ILP model instead of a flow ILP model, where one variable is defined for each arc in the graph and the protection mechanisms are described by constraints in the ILP. Indeed, we embed these mechanisms in the model variables, obtaining a model with fewer complicating constraints and hence easier to manage.

B. Least cost path determination

The first step of our method consists in determining a set of minimum cost paths between each pair of ingress and egress routers. Then, the Us to protect the portions of all possible WPs are identified.

The minimum cost paths are generated through Recursive Enumeration Algorithm (REA) described by Jiménez and Marzal [19].

The set of paths is created as follows.

- First, determining a set of K minimum cost paths between each pair of border routers (K value has an impact on the results accuracy).
- Second, from the K paths finding all the completely disjoint pair of paths that can form the WPs and RPs between every pair of border routers.
- Third, for every WP-RP pair, obtaining all the minimum cost shortcuts that connect the routers of the WP to the routers of the RP, thus identifying all the Us. As for building the U paths, REA firstly looks for paths including a three-link WP segment on the first portion of U (we called this "fixed U"). If none is found, a path with two links is searched, and if none exists, then a single link path is considered (we called this "variable-length U"). If no paths with less than 3 links are found, a U with more than 3 WP links will be used, although this hardly happens.

This approach avoids loops and does not use the same link in both the WP and RP (hence, robustness against single link/node failure is achieved). The recovery time constraint implies some restrictions on the length of the U. In particular, the portion of the WP in the U (named "first portion") has at most 3 links. Therefore, assuming an average node traversal delay of 10 ms, the packets, crossing twice as many nodes as in the protected WP portion before being routed into the recovery path [21], experience a recovery time as in SONET/SDH networks.

Finally, the result of the last step is a set of U paths to which our formulation is restricted. This is equivalent to establishing an initial set of variables in a Column Generation approach.

C. Selecting the best Us

The next step of our algorithm is to calculate the optimal WP-RP pairs, or equivalently, the optimal chains of Us. An Integer Linear Programming (ILP) problem has been formulated for the purpose.

Let us define $L_u \subset L$ the set of links contained in a specific s, for all $s \in U$. As we have specified before, this set has three portions if we are running our algorithm for engineering the network with Haskin's method and 4 portions in the case of the One-to-One backup method.

We also point out that we allow demands not to be accepted , but for each unsatisfied demand we introduce a high penalty in the objective function that is given to the ILP solver.

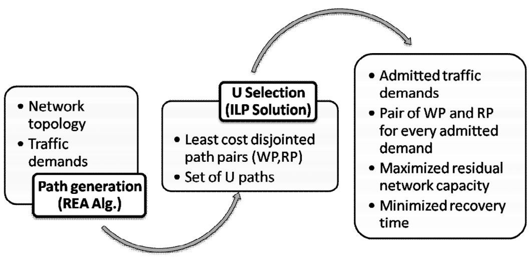

Fig. 6. Scheme of the network engineering algorithm.

This avoids unfeasibility of the routing problem and obtains a solution with a minimum number of unsatisfied traffic demands. Recovery time is instead guaranteed by the constraint on the length of the first portion of U and the a priori generation of U variables. The optimal solution of the ILP problem leads to a network engineering with working and recovery paths for every accepted traffic demand, which maximizes the overall residual capacity, allowing the maximum possible number of traffic demands to be satisfied. The algorithm is shown in Fig. 6.

The resource sharing issue (shared protection) is addressed by a simple post-processing. It allocates on each link the maximum bandwidth needed in all possible working conditions (i.e. also in case of single link/node failure), rather than the sum of the bandwidth to be allocated for each satisfied traffic demand.

We point out that we do use real optimal techniques in our algorithm, since an ILP solver is employed. This allows to obtaining solutions that are optimal within the restricted initial set of shortest paths.

IV. Analysis of Practical Cases

As an evaluation process, we fed the network engineering tool we developed with several network topologies and traffic demand sets. All computational tests were performed using a computer equipped with a Pentium Xeon 2.8 GHz processor and 2 GB of RAM memory. We used the AMPL modelling language [20] to specify our Optimization model and solved it through Cplex's commercial ILP solver [21]. Hereafter, we report the results for topologies that resemble some realistic cases, more specifically referring to metropolitan networks, resulting from interconnecting various optical rings. It is just in such context where fault resilience within tens of milliseconds was first provided (by technologies as SONET/SDH).

The aim is primarily to show the simplicity, flexibility and correctness of our proposal, while providing numerical results. The set of traffic demands we defined for the different networks of Fig. 7 are shown in Fig. 8. Every triplet of numbers in the table indicates ingress node identifier, egress node identifier and amount of required bandwidth.

We applied our method with the number *K* of initial paths, obtained in the first phase with REA, ranging from 5 to 10. The algorithm also computes the network configuration and the residual capacity for the two supported FRR models (Haskin's and One-to-One backup). In the analysis, the bandwidth is allocated for both the working and recovery paths with resource sharing.

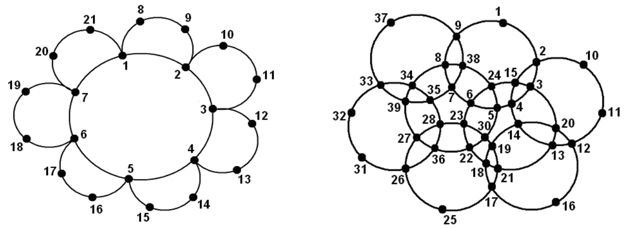

Fig. 7. From left to right: daisy network with main ring and secondary extension leaves, multiple ring networks with intersections.

Set A	Set B
1) 10 12 10	1) 14 17 10
2) 12 5 10	2) 1 11 10
3) 21 2 10	3) 21 18 10
4) 20 18 10	4) 6 2 10
5) 16 12 10	5) 7 24 10
6) 11 17 10	6) 12 19 10
7) 19 13 10	7) 21 24 10
8) 7 22 10	8) 20 14 10
9) 13 4 10	9) 16 9 10
10) 18 8 10	10) 1 24 10

Fig. 8. Example of traffic demand set descriptions.

Tables I and II report the outputs of our algorithm for the different demand sets (A+B means that the two sets were submitted together). The first four columns represent, respectively, the residual capacity (Mbps) of the network for the One-to-One backup (DP) and Haskin's (SC) methods, differentiating between the case where we consider either fixed Us only (i.e. Us that have three links in the WP portion) or variable length Us. The residual network capacity was calculated as the sum of the residual capacity of all the links of the network. The last three columns represent the number of fixed Us, the number of variable Us, and the number of admitted traffic demands, respectively. As appears from the tables, augmenting the number of available path pairs (i.e. augmenting K value) increases the number of admitted traffic demands. This is due to the fact that, with a too small K, REA may fail in finding a big enough number of disjoint paths to form WP-RP pairs for every traffic demand. This affects the chance to admit traffic demands in two ways. First, fewer paths means fewer variables in the ILP model, therefore a restricted solution set. Second, with fewer paths we more unlikely find couples of disjoint paths for some demands. In the case of 10 paths, the number of computed Us is obviously higher and thus the ILP solver can lead to a more optimized solution (a larger set of options is available), although a higher number of variables in the model increases the processing time.

TABLE I. RESULTS FOR THE DAISY NETWORK TOPOLOGY, 5 REA-CALCULATED LEAST COST PATH (10 TRAFFIC DEMANDS PER SET)

5 Path	DP_FIX	SC_FIX	DP_VAR	SC_VAR	N° U FIX	N° U VAR	REQ
A	7910	7920	7910	7920	14	22	2,3,6, 7,8
B	7560	7570	7560	7560	25	33	1,2,3,4,5, 6,8,9,10

TABLE II. RESULTS FOR THE DAISY NETWORK TOPOLOGY, 10 REA-CALCULATED LEAST COST PATH (10 TRAFFIC DEMANDS PER SET)

10 Path	DP_FIX	SC_FIX	DP_VAR	SC_VAR	N° U FIX	N° U VAR	REQ
A	7470	7480	7460	7470	97	65	Any
B	7490	7500	7490	7490	89	102	Any

This considerations apply to the daisy network topology case (see table I and II), where a larger set of paths allows to accept all the traffic demands. This topology consists of a double ring (a main ring and a secondary one formed by the leaves only) and it is very common for example in MANs. In the second and more complex topology case (see Table III and IV), we set *K* to either 10 or 15. Again, the higher the number of path pairs, the higher the number of accepted traffic demands. It is worthwhile to point out that in this case, the number of initial paths does not need to be increased further, because the constraint on the link capacities itself prevents admitting additional traffic due to lack of spare bandwidth. Our target objective function minimizes the network capacity usage, therefore the ILP solver selects the shortest paths that satisfy the traffic demands. An input set of such demands able to saturate the network was intentionally considered in order to have an effective assessment of our proposal.

As for path-based restoration, various schemes [23][25][26] have been proposed in the recent years. The advantage of such schemes over ours is a slightly higher network capacity efficiency mainly due to the lack of shortcuts between WP and RP. However, they are much slower in restoration, because it is the source node responsible for switching the traffic to the backup path and cannot perform the switching until it receives a failure notification message from the node that has detected the failure. While, a key feature of our method is that it can address the same recovery time as in link-based schemes (of some tens of ms as in SONET/SDH networks) and having capacity efficiency close to that of path-based schemes.

For each issued topology, Table V reports the number of nodes, links, and demands, the value of K, and the number of Us and variables (most of them are 'z' variables) of the ILP model.

Although we do not apply exact optimization algorithms in all steps of our approach, we do take into account the ILP model in the crucial path selection step, where the Wps and RPs are selected from a given set.

Moreover, results have been computed by our solution in less than 5 seconds with both fixed and variable length Us by a regular Pentium IV with a 2.8 GHz clock and using just some Mbytes of memory space, whereas classical edge flow formulations usually fail to provide a provably good solution in short time. Such data also demonstrate the scalability of our approach for large real networks, being the processing time relevantly low and the number of generated Us proportional to the finite cardinality of the set of the starting shortest paths and their lengths, as well as input traffic demands.

TABLE III. RESULTS FOR THE MULTIPLE RING NETWORK TOPOLOGY, 10 REA-CALCULATED LEAST COST PATH (10 TRAFFIC DEMANDS PER SET)

10 Path	DP_FIX	SC_FIX	DP_VAR	SC_VAR	N° U FIX	N° U VAR	REQ
A	21170	21190	21150	21190	81	182	2,3,7,8
B	21220	21250	21210	21250	13	39	2,3,8

TABLE IV. RESULTS FOR THE MULTIPLE RING NETWORK TOPOLOGY, 15 REA-CALCULATED LEAST COST PATH (10 TRAFFIC DEMANDS PER SET)

15 Path	DP_FIX	SC_FIX	DP_VAR	SC_VAR	N° U FIX	N° U VAR	REQ
A	21040	21070	21020	21070	207	534	2,3,4,7,8
B	20990	21020	20970	21020	109	273	2,3,4,7,8

TABLE V. NETWORKS AND DATA OF THE OPTIMIZATION PROCESS

Networks	N° Nodes	N° Links	N° Demands	N° *K* paths	N° U Fix	N° U Var	N° Variables Fix	N° Variables Var
Grid	24	76	20	10	369	1062	7456	21316
Daisy	21	56	20	10	186	167	3776	3396
Multiple ring	39	136	20	15	316	807	6456	16276

Tests were also carried out with the objective function minimizing the maximum recovery time. More specifically, the extra-delays for the traffic due to a fault were evaluated for the purpose. Such values can be calculated by considering the different number of nodes to be crossed to reach the egress border router in case of failure.

The results, still collected in few seconds, confirmed the consistency of our ILP models for this variant of the problem. As expected, we obtain solutions with a less efficient capacity usage, but with improvements up to 30 ms in maximum recovery times.

Both formulations for the objective function turn out to be effective in relation to their own targets. actually, in both models the constraint of some tens of milliseconds for the recovery time)i.e., the time necessary to switch the traffic from the primary to the backup path) is always satisfied because implicitly integrated into the model variables. Just to make an example, in case of fault the packets of the concerned traffic at worst cross twice as many nodes as in the portion of WP protected by a U [21] before being routed into the recovery path. Assuming an average node traversal delay of 5 (10) ms, they experience a recovery time of 25 (50) ms at worst. Indeed, with the longest Us (i.e. the Us composed of three links) and a failure downlink to the issued U, five node interfaces must be crossed.

It is worthwhile to underline that the maximum length of the first portion of the Us can be imposed as needed, depending on network performance (i.e. node crossing delay of the high priority traffic to be protected) and the target recovery time. Our proposal is flexible and of general validity, being just a matter of generating the appropriate set of input variables)(i.e. the Us) to the ILP model. Of course, with variable length Us, the higher the allowed maximum number of links in the first portion of the U, the larger the set of input variables, but the test results have demonstrated that the processing time is always of the same order of magnitude.

V. CONCLUSIONS AND FUTURE WORK

In this paper, an off-line system to effectively calculate working and recovery paths for Haskin's method with shortcuts and One-to-One backup, has been proposed and analyzed.

Haskin's method is also able to recover from node failure and is a global repair technique, while One-to-One backup is a local repair technique.

In order to take advantage of modern exact solvers for discrete optimization problems, we have formulated the proposal as an Integer Linear Programming (ILP) system, with bandwidth allocation as the objective function. Since common optimization models, such as the edge-flow formulation, are hardly viable for practical-size networks, we have introduced a different class of variables based on a decomposition of the working and recovery path. This approach significantly reduces the complexity of the model and hence, the optimization processing time. As a result, our system is also applicable to large networks.

An analysis on different types of network topologies and sets of traffic demands has been carried out reporting the recovery time and the bandwidth to be reserved. Although a sub-optimal solution has been designed, the analysis shows the correctness, flexibility and good performance of our proposal.

Although resource sharing is tackled as a post-processing step, our two-phase optimization algorithm has shown to give consistent results for sizeable networks. However, we are planning to include shared-protection constraints in the ILP model in order to obtain a more integrated and optimal solution.

Once working and recovery paths for a given set of traffic demands have been calculated, further service requests can be accomplished by applying the same method but considering the residual (i.e., not yet allocated) capacity of the network. It would be interesting to analyze the relationship between the overall traffic demands and the spare bandwidth into the network in order to select proper time for a recalculation of all the working and recovery paths from the beginning. Concerning this issue, a new constraint should be included into the ILP problem, binding the lengths of each couple of old and newly calculated working paths to allow a switching between them transparent to the user.

Moreover, the proposed solution will be extended to also deal with point-to-multipoint traffic demands, as foreseen within the framework of IST FP7 OPTIMIX project, where fault resilience mechanisms are to be devised for video streaming applications.

REFERENCES

[1] E. Rosen et al.. Multi-Protocol Label Switching Architecture. IETF MPLS WG. RFC 3031, Jan. 2001.

[2] D. Awduche et al.. Overview and Principles of Internet Traffic Engineering. IETF TE WG. RFC 3272, May 2002.

[3] D. Awduche et al.. Requirements for Traffic Engineering Over MPLS. IETF TE WG. RFC 2702, Sept. 1999.

[4] Chen, T.M.; Oh, T.H. Reliable services in MPLS, Communications Magazine, IEEE, Volume 37, Issue 12, Dec. 1999 Page(s):58 - 62 .

[5] V. Sharma et al. Framework for Multi-Protocol Label Switching (MPLS)-based Recovery. IETF MPLS WG. RFC 3469, Feb. 2003.

[6] P. Pan et al. Fast Reroute Extensions to RSVP-TE for LSP Tunnels. IETF MPLS WG. RFC 4090, May 2005.

[7] R.K. Ahuja, T.L. Magnanti, J.B. Orlin. Network flows: theory, algorithms, and applications. Prentice-Hall, Inc., Upper Saddle River, NJ, USA, 1993.

[8] J. L. Kennington, E. V. Olinick and G. Spiride. Basic mathematical programming models for capacity allocation in mesh-based survivable networks. The Omega International Journal of Management Science 35 (6) 2007, pp. 629-644.

[9] L. Jorge, T. Gomes. Survey of Recovery Schemes in MPLS Networks. Dependability of Computer Systems DepCos-RELCOMEX '06, pages 110-118, 2006.

[10] G. Dahl and M. Stoer. A cutting plane algorithm for multicommodity survivable network design problems. INFORMS Journal on Computing, 18(1):1-11, 1998.

[11] M. Grötschel, C. Mona, and M. Stoer. Polyhedral and computational investigations for designing communication networks with high survivability requirements. Operations Research, 43(6):1012-1024, 1995.

[12] B. Fortz, M Labbé, and F. Maffioli. Solving the two-connected network with bounded meshes problem. Operations Research, 48:866-877, 2000.

[13] Hong, D.W., Choong Seon Hong, Woo-Sung Kim, A Segment-based Protection Scheme for MPLS Network Survivability. Network Operations and Management Symposium, 2006, pp. 1-4.

[14] S. Yoon, H. Lee, D. Choi, Y. Kim, G. Lee, M. Lee. An efficient recovery mechanism for MPLS-based protection LSP. Joint 4th IEEE International Conference on ATM (ICATM 2001) and High Speed Intelligent Internet Symposium, 2001, pp. 75-79.

[15] D. Haskin and R. Krishnan. A method for setting an alternative label switched paths to handle fast reroute. IETF MPLS WG. Internet draft, May 2000.

[16] C. Huang, V. Sharma, K. Owens, S.Makam "Building Reliable MPLS Networks Using a Path Protection Mechanism" IEEE Communications Magazine March 2002.

[17] Guangzhi Li; Dongmei Wang; Kalmanek, C.; Doverspike, R.; Efficient distributed recovery path selection for shared mesh restoration; Networking, IEEE/ACM Transactions on, Volume 11, Issue 5, Oct. 2003 Page(s):761 – 771.

[18] F. Aslam, S. Raza, Z. Uzmi, Y. Kim.. Bandwidth Sharing with Primary Paths for Protection Routing in an MPLS Network, In Proc. of INFOCOM 2006, pages 1-6, 2006.

[19] VM Jiménez and A. Marzal. Computing the *K* shortest paths: A new algorithm. and an experimental comparison. WAE 1999, LNCS 1668: 15-29, 1999.

[20] R. Fourer, D.M. Gay and B.W. Kernighan. A Modeling Language for Mathematical Programming. Management Science 36 (1990), pp.519-554. http://www.ampl.com .

[21] Ilog Cplex Division. Cplex Optimization Package. http://www.cplex.com .

[22] Hundessa, L.; Domingo-Pascual, J.; Reliable and fast rerouting mechanism for a protected label switched path. Global Telecommunications Conference, 2002. GLOBECOM '02. IEEE Volume 2, 17-21 Nov. 2002 Page(s):1608 - 1612 vol.2.

[23] S.Petrov. An approach for MPLS Recovery. ACM International Conference Proceeding Series; Vol. 285 , Proceedings of the 2007 international conference on Computer systems and technologies, 2007.

[24] C. Qiao and D. Xu. Distributed Partial Information Management (DPIM) Schemes for Survivable Networks - Part I. In Proc. of IEEE INFOCOM' 02, pages 302 - 311, June 2002.

[25] M. Kodialam and T. V. Lakshman. Dynamic routing of bandwidth guaranteed tunnels with restoration. In Proc. of IEEE INFOCOM'00, pages 902-911, 2000.

[26] G. Li, D. Wang, C. Kalmanek, and R. Doverspike. Efficient distributed path selection for shared restoration connections, In Proc. of IEEE INFOCOM'02, pages 140 - 149, June 2002.

[27] D. Xu, Y. Xiong, and C. Qiao,"Novel Algorithms for Shared Segment Protection," IEEE Journal on Selected Areas in Communications, Vol. 21, No. 8, pp. 1320-1331, Oct. 2003.

[28] S. B. Zheng, B. Yang, M. Yang and J. Wang. Finding Minimum-Cost Paths with Minimum Sharability, In Proc. of IEEE INFOCOM'07, pages 249-257, May 2007.