Robustness in Next-Generation Networks

Gianmarco PANZA[1], Antonio CAPONE[2], Damiano PINARELLO[3], Pietro BELOTTI[4]
[1,2,3]CEFRIEL/Politecnico di Milano, via Fucini 2, Milano, 20133, Italy
Tel: +39 0223954326, Fax: +39 0223954526, Email:
[1]gianmarco.panza@cefriel.it, [2]capone@elet.polimi.it, [3]damiano.pinarello@cefriel.it
[4]Taper School of Business, Carnegie Mellon University, 5000 Forbes Ave, Pittsburgh PA, 15213, USA
Email: [4]belotti@lehigh.edu

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. 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 either the overall switching delay experienced by the user in case of failure or the bandwidth allocation, thanks to a shared protection, while limiting the recovery time to some tens of milliseconds as in SONET/SDH networks.

Keywords: Fault Resilience, FRR, ILP, MPLS, NGN, Shared Protection, Traffic Engineering.

1. 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 some tens of milliseconds 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 uses 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 [50][6][22][23].

This work has been carried out within the framework of the IST OPTIMIX project, partially supported by the European Commission under the contract FP7 n°INFSO-ICT-214625. The aim is to address in a straightforward manner the traffic engineering problem of a MPLS network which is fully recoverable against single link or node failure, 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 between them) 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 either the bandwidth allocation by applying the Shared Protection (SP) or the overall switching delay, as seen by the user in case of failure, while constraining the recovery time to a given limit (i.e. some tens of milliseconds).

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.

2. 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 in the event that multiple failures occur.

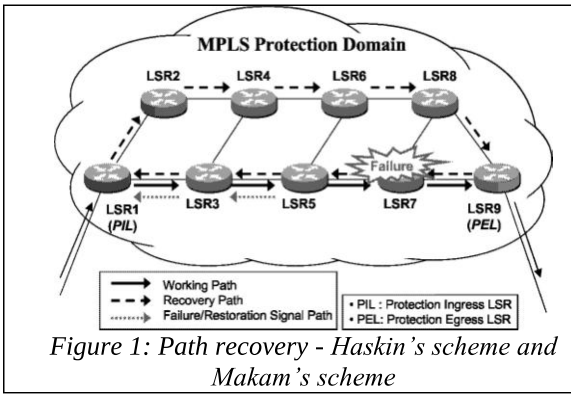

Figure 1: Path recovery - Haskin's scheme and Makam's scheme

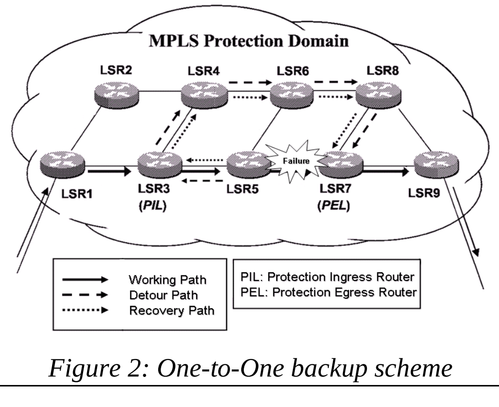

Figure 2: One-to-One backup scheme

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 Makam et al. [16], respectively. Both schemes are depicted in Figure 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 Figure 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 Figure 1, when the node LSR7 fails, the working traffic is rerouted along the RP LSR1-2-4-6-8-9.

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.

Figure 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 Figure 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 Figure 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.

3. 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 single 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 optimal 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 modelled 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,

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

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,

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

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:

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

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

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

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

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.

3.1 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 Figure 3. We can decompose the U path into four portions between the four delimiting routers A, B, C, and D, as shown in Figure 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 "fourth portion" of a detour path with the "second portion" of the following one, as shown in Figure 5.

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. 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 Figure 5). also, every node belonging to the WP has a protection DP, as natural with Haskin's method and by activating also the following chained U, when a fault happens in the node A of a given U with One-to-One Backup (in this case the restored traffic is to be routed from the "third portion" of the given U, directly to the "third portion" of the following U).

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 Figure 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 Figure 4.

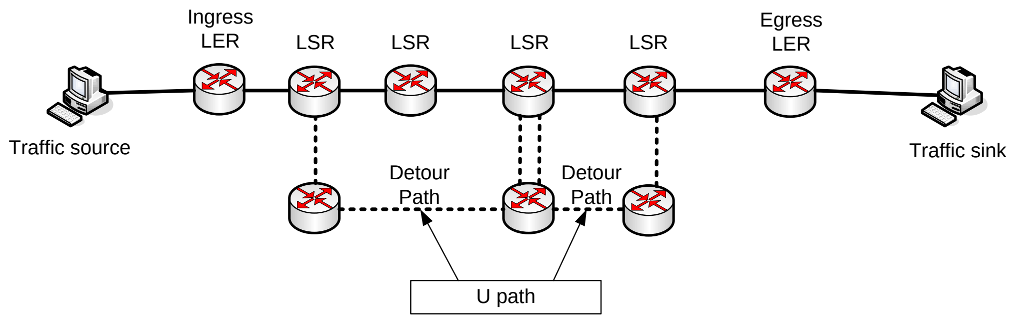

Figure 3: *U-path examples*

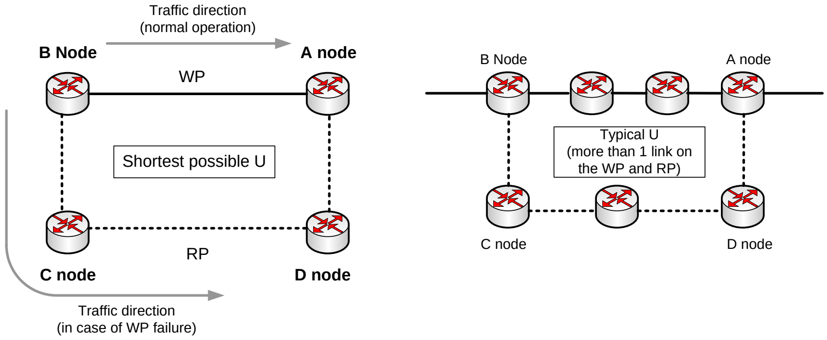
Traffic direction (normal operation)
B Node
A node
WP
Shortest possible U
C node
RP
D node
Traffic direction (in case of WP failure)
B Node
A node
Typical U (more than 1 link on the WP and RP)
C node
D node

Figure 4: U-path decomposition.

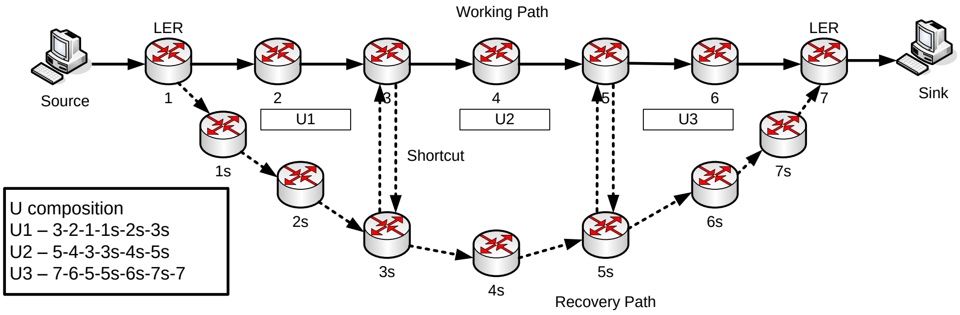

Figure 5: Example of resulting U chain.

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 Figure 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.

3.2 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 overall performance, as discussed in the next section about the analysis of some practical cases).
- 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 in practice.

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 [22], 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.

3.3 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. 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 Figure 6.

An efficient resource allocation is addressed by applying the shared protection. On each link the maximum bandwidth needed for all possible working conditions (i.e. also in case of single link/node failure) is allocated, rather than the sum of the bandwidth to be allocated for each satisfied traffic demand both in the WP and RP. The SP is fully integrated in the formulated ILP model in order to minimize the resulting resource allocation.

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.

4. Analysis of practical cases

As an evaluation process, we have fed the developed engineering tool with several networks and traffic demand sets. All tests were performed using a laptop PC equipped with a 1.73 GHz Intel Core Duo T2250 of 2 GB RAM, and the GLPSOL ver. 4.9 (the solver LP/MIP GPLK standalone [21], based on GNUMathProg, which is a subset of the well-known AMPL modelling language [20].

Hereafter, we report the results for topologies that resemble some realistic cases: partially mesh and resulting from either tangent or intersected rings (with a higher capacity of the core links for the latter ones).

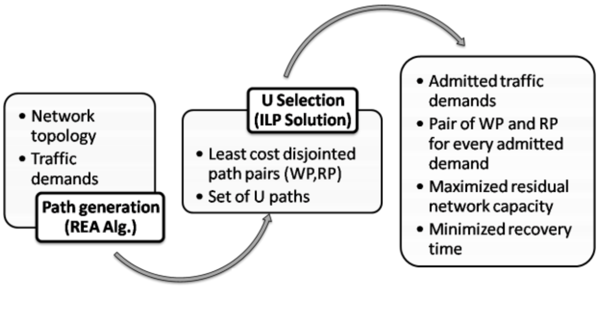
• Network topology
• Traffic demands
Path generation (REA Alg.)
U Selection (ILP Solution)
• Least cost disjointed path pairs (WP,RP)
• Set of U paths
• Admitted traffic demands
• Pair of WP and RP for every admitted demand
• Maximized residual network capacity
• Minimized recovery time

Figure 6: Scheme of the network engineering algorithm.

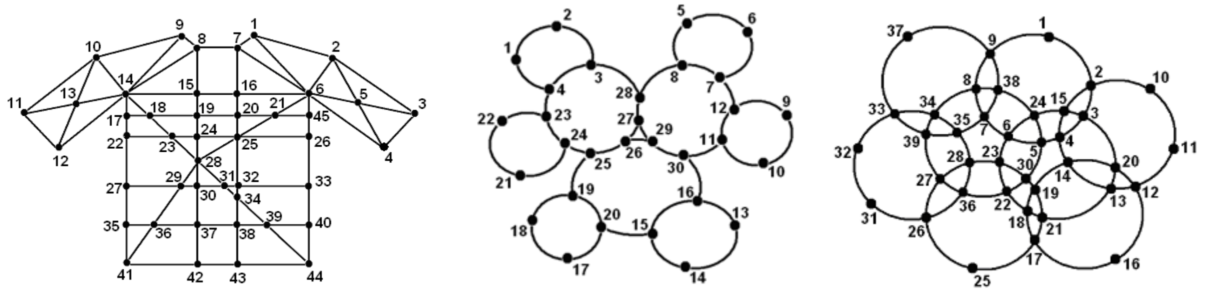

Figure 7: Investigated topologies (from left to right): Mesh, Molecule and Maze.

More often referring to metropolitan networks, where recovery time within some tens of milliseconds was first supported (by technologies as SONET/SDH). The aim is to show the simplicity, flexibility and correctness, as well as the good performance of our proposal, while providing numerical results. The set of traffic demands for Mesh network of Figure 7 (49 nodes, 83 links and link capacity of 100 Mbit/s) are shown in Figure 8. Every triplet of numbers in the table indicates ingress node identifier, egress node identifier and amount of required bandwidth for a given demand.

We applied our solution with the number K of initial minimum cost paths, obtained in the first phase with REA (as described in the previous section), equals to either 5 or 10. The designed tool computes the network configuration and the residual capacity for the two supported FRR methods (Haskin's and One-to-One backup), optimizing either the allocated capacity or the overall switching time (i.e. the extra-delay experienced by the user in case of link/node failure). In the process, the bandwidth is allocated for both the working and recovery paths with resource sharing (i.e. applying the SP), as for protection switching recovery techniques.

Regarding the capacity allocation optimization, Tables Table I and Table II report the outputs of our system for the different traffic demand sets of 10 requests each (A+B means that the two sets A and B were submitted together), with equal-cost links and related to Mesh topology only for lack of room (however, the same conclusions can be drawn out for all the investigated networks).

Starting from the upper left, the value of the residual capacity, the number of Us generated and the traffic request are shown. The first three rows refer to the results with the SP, while the last three rows refer to the results without it.

The first four columns encompasses both the issued protection methods, Haskin (HK) and One-to-One Backup (DP) for either "fixed-length" Us (FIX), i.e. Us that have three links in the WP portion, or "variable-length" Us (VAR), i.e. Us that have three links in the WP portion at most; the computational time ($T_{c,}$ in seconds) and memory RAM usage (M in Mb) are also included.

The fifth and sixth columns show the number of selected Us, respectively "fixed-length" and "variable-length" Us, related to the accepted traffic demands (whose number is indicated in the last column).

As clear from the tables, augmenting K increases the number of admitted traffic demands. Because with a too small set of initial paths 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 requests in two ways. First, fewer initial paths means fewer variables in the ILP model, therefore a restricted solution set. Second, couples of disjoint paths are more unlikely found for some demands with fewer initially calculated paths. In the case of 10 initial shortest 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.

The gain in terms of spare capacity by the SP grows with the increase of the number of accepted traffic requests. The peak is reached with Haskin FRR and variable-length Us, with 20 traffic demands and 10 initially calculated shortest paths (15.880 Mbit/s of residual capacity, 130 Mbit/s more than without the SP). Indeed, Haskin's doesn't involve the "fourth portion" of the US and "variable-length" Us allow for a greater flexibility, being a superset of the "fixed-length" Us (i.e. Us having a number of links in the first segment lower than three can be also selected to optimise the objective function).

Results have been computed in very few seconds with a memory usage of just some Mbytes, for both "fixed-length" and "variable-length" Us.

Whereas classical edge flow formulations usually fail to provide a provably good solution in short time.

SET A				SET B			
1)	10	28	10	1)	45	10	10
2)	42	5	10	2)	25	6	10
3)	21	2	10	3)	14	32	10
4)	20	18	10	4)	1	11	10
5)	16	12	10	5)	21	18	10
6)	33	17	10	6)	7	12	10
7)	19	13	10	7)	12	5	10
8)	27	12	10	8)	41	24	10
9)	13	16	10	9)	28	5	10
10)	18	42	10	10)	16	12	10

Figure 8: Traffic demand set specification for Mesh network.

Table I: Results for Mesh network, K*=5*

5 Path	DP_fix\Tc\M [Mbps/s/MB]	HK_fix\Tc\M [Mbps/s/MB]	DP_var\Tc\M [Mbps/s/MB]	HK_var\Tc\M [Mbps/s/MB]	#U fix	#U var	Req
A	16580\ 0.5\4.1	16590\0.7\4.2	16580\0.8\4.4	16590\1.2\4.6	4	9	5
B	16310\0.7\4.4	16350\0.9\4.5	16320\3.7\6.5	16370\4.5\7.8	14	50	2,3,5,8
A + B	16270\0.9\4.9	16290\1.2\5.1	16270\3.9\9.2	16300\4.7\9.6	17	58	
A	16520\0.2\2.2	16530\0.2\2.2	16520\0.4\4.3	16530\0.4\4.3	4	9	5
B	16240\0.3\2.6	16260\0.3\2.6	16240\1.6\4.7	16260\1.6\4.7	14	50	2,3,5,8
A + B	16180\0.4\3.6	16200\0.4\3.6	16180\1.8\6.5	16200\1.8\6.5	17	58	

Table II: Results for Mesh network, K*=10*

10 Path	DP_fix\Tc\M [Mbps/s/MB]	HK_fix\Tc\M [Mbps/s/MB]	DP_var\Tc\M [Mbps/s/MB]	HK_var\Tc\M [Mbps/s/MB]	#U fix	#U var	Req
A	16470\2\7.1	16490\2.3\7.3	16470\5.2\9.4	16480\6.4\9.7	38	99	2,4
B	15950\3\8.1	16000\3.4\8.4	15970\7.2\10.6	16010\8.9\11	43	140	1,2,3,4, 5,6,8
A + B	15790\4\9.2	15850\4.4\9.6	15810\12.3\12.1	15880\13.9\13.3	80	238	
A	16410\1\4.1	16420\1\4.1	16410\2.1\6.3	16420\2.1\6.3	38	99	2,4
B	15870\1.3\5.2	15920\1.3\5.2	15870\2.7\7.8	15920\2.7\7.8	43	140	1,2,3,4, 5,6,8
A + B	15690\1.6\6.6	15750\1.6\6.6	15690\4.7\8.8	15750\4.7\8.8	80	238	

Such data demonstrate the scalability of our approach for large real networks, being the processing time considerably low and the number of generated Us proportional to the finite cardinality of the set of the initial couples of disjoint shortest paths and input traffic demands (other than the dimension/diameter of the issued network).

As for path-based restoration, various solutions [23][25][26] have been proposed in the recent years. The advantage of such solutions over ours is a slightly higher network capacity efficiency actually due to the lack of shortcuts between WP and RP. However, they are much slower in restoration, because in such proposals the source node is responsible for the traffic switching to the backup path and cannot perform the operation 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 solutions (of some tens of ms) and having capacity efficiency close to that of path-based solutions.

Concerning the optimization of the overall switching time from a user standpoint, the issued delay can be calculated considering the different number of nodes to be crossed by the affected traffic to reach the egress border router in case of link/node failure. The reported results of Table III (again, collected in very few seconds and with some Mbytes of memory usage) are organized slightly as before.

The first row reports the values optimizing the bandwidth allocation and the second one the overall switching delay. For each cell of the first four columns, the residual capacity with or without the SP and the maximum value of the overall switching time are included. The fifth and sixth columns show the number of selected “fixed-length” and “variable-length” Us, respectively.

In the optimization of the overall recovery time, more Us with a shorter “first portion” are selected.

Table III: Comparison between the residual capacity and overall switching time optimizations for Mesh network

	HK_fix	HK_var	DP_fix	DP_var	#U fix	#U var
MESH	[Mbps/Mbps/ms]	[Mbps/Mbps/ms]	[Mbps/Mbps/ms]	[Mbps/Mbps/ms]		
Bandwidth Optimization	15850/15750/90	15880/15750/90	15790/15690/90	15810/15690/90	7	16
Switching Delay Optimization	15760/15600/80	15790/15600/80	15700/15500/80	15720/15500/80	10	19

Table IV: Residual capacity for Mesh network in the equal and non-equal link cost cases

10	DP_fix\Tc\M	HK_fix\Tc\M	DP_var\Tc\M	HK_var\Tc\M	#U fix	#U var	Req
Path	[Mbps/s/MB]	[Mbps/s/MB]	[Mbps/s/MB]	[Mbps/s/MB]			
A	16470\2\7.1	16490\2.3\7.3	16470\5.2\9.4	16480\6.4\9.7	38	99	2,4
B	15960\3\8.1	16010\3.4\8.4	15950\7.2\10.6	16000\8.9\11	43	140	1,2,3,4,5,6,8
A+B	15790\4\9.2	15860\4.4\9.6	15780\12.3\12.1	15850\13.9\13.3	80	238	
A	16410\1\4.1	16420\1\4.1	16410\2.1\6.3	16420\2.1\6.3	38	99	2,4
B	15870\1.3\5.2	15920\1.3\5.2	15870\2.7\7.8	15920\2.7\7.8	43	140	1,2,3,4,5,6,8
A+B	15690\1.6\6.6	15750\1.6\6.6	15690\4.7\8.8	15750\4.7\8.8	80	238	

Therefore, the residual capacity optimization (in particular with the SP) leads to a noticeable more efficient bandwidth allocation. On the other hand, the overall switching time is significantly improved by optimising it (up to 30 ms in Molecule network).

The results also show how there is not a substantial difference between Haskin's and One-to-One Backup methods regarding the overall switching delay, because the lengths of WPs and RPs for the same couple of ingress and egress border routers are quite similar (this is particularly true for Mesh topology, while for the other investigated practical cases a slight difference between the two issued FRR methods has been detected, in favour of the latter).

In the end, both the formulations of the objective function turn out to be effective in relation to their own targets. Moreover, it is worthwhile to underline that in both cases 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 minus one as many nodes as in the portion of WP protected by a U [22] before being routed into the recovery path. Assuming an average node traversal delay of 5 (10) ms, they experience a recovery time of 20 (40) ms at worst (the further delay due to the failure detection is typically of the order of 10 ms). Indeed, with the longest Us (i.e. the Us composed of three links in the "first portion") and a failure in the node A, four node interfaces must be crossed (the uplink node detects the failure and bounces back the traffic).

Furthermore, the maximum length of the "first portion" of the Us can be imposed as needed, depending on network performance (i.e. the node crossing delay of the high priority traffic to be protected) and the target recovery time. Therefore, 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, in particular 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 and memory usage are always of the same order of magnitude.

All the investigated networks have been tested for both One-to-One Backup (DP) and Haskin FRR (HK) methods, either with "fixed-length" Us or "variable-length" Us, also in the case of non-equal cost links. Having links of different cost has an impact on both the set of shortest path initially calculated by REA, and on the objective function to be optimized, in terms of weight assigned to the residual capacity of each link.

In particular, Mash topology was analyzed with 10 initial shortest paths on two sets of 10 traffic requests and then on the sum of them. The first 44 links set to a cost of either 0.2 or 0.8, while the remaining 39 links to a cost of 1.

The results show how the value of the residual capacity is significantly worse in the case of 0.2-1 than in the case of 0.8-1 (see Table IV). This is due to the increase in the number of Us selected to accept the same traffic requests in the latter.

The re-optimization issue has been also carefully considered. The objective is to find a possible relationship between the dynamics of traffic requests and the residual capacity. It is critical to understand when it is necessary to re-optimize globally the network during its life-cycle; this mean to determine the maximum number of re-applications of the system on additional traffic demands before a global re-optimization on the total number of traffic demands. Several tests have been run to outline some heuristics, taking into account the number of traffic requests added at the time, network topology, saturation level of the network, meshing facto and value of *K* for REA. The conclusions about such an analysis are directly reported in the next section.

5. Conclusion and future work

In this paper, a flexible system to effectively calculate working and recovery paths for the global repair technique with shortcuts designed by Haskin and the local repair technique of One-to-One backup, has been proposed and analyzed.

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 either the bandwidth allocation or the overall switching delay experienced by the user in case of failure, 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 and memory usage. As a result, the Network Operator has a simple, but powerful off-line tool to support robustness against single link or node failure even in large real networks, for highly demanding services (i.e. with recovery time of some tens of milliseconds).

An analysis on different types of networks and sets of traffic demands has been carried out reporting the maximum overall switching time and the bandwidth to be reserved. It has demonstrated the correctness and flexibility of our proposal, as well as its efficiency in the resource exploitation, achieved by the Shared Protection fully integrated in the ILP system.

Haskin FRR scheme performs better with respects to the bandwidth usage, not involving the “fourth portion” of the Us. While, One-to-One Backup could improve the overall switching time in some topologies, depending on the set of initial shortest paths. A higher number of such minimum cost paths calculated by REA could lead to accept more traffic requests at the cost of a slightly larger set of input variables to be managed by the solver. It has been highlighted as having variable length Us allows for a greater flexibility in determining the most optimal working and recovery paths for the accepted traffic requests, without a drawback in the system performance. Also, a driven distribution of the load within the network is possible but with a less efficient capacity allocation than in the case of equal cost links. The re-optimization issues has been carefully considered. A global optimization can take more advantage of a bigger number of input Us available (i.e. a higher number of shortest paths initially calculated). Furthermore, the higher the saturation level, the more critical the resource allocation; therefore, a stronger need for a global re-optimization. While, the mesh factor of the network has a positive effect, as more almost equal cost shortest paths are available, resulting in a less stringent necessity for a frequent global re-optimization.

Future work is about extending the designed system to accomplish also multicast traffic requests as typical of the application scenario addressed by the FP7 IST OPTIMIX project. In principle, a multicast distribution tree could be determined by the set of point-to-point paths from the root ingress router to each leaf egress router. Therefore, it could be managed as a set of unicast paths, for which the solution has already been provided. Of course, every path is to be separately protected against single link or node failure, and a deep analysis of the resulting topology for the multicast tree in case of failure must be carried out, also in order to correctly and accurately apply the Shared Protection, keeping The related ILP system constraints still linear. Furthermore, the joining or leaving of users, as well as the user mobility can induce a modification in the multicast tree topology. Such a matter should be managed run-time and transparently to the users, by retaining all the paths related to the old and still active leaf egress routers. In general, all the issues and solutions to support the QoS requirements of the users, possibly involved in real-time communications with multimedia data, must be carefully considered and properly solved, including the implementations aspects that in practice a Network Operator should deal with, in order to deploy our proposal.

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, pag. 58 – 62, Dec. 1999.

[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, pag. 629-644.

[9] L. Jorge, T. Gomes, Survey of Recovery Schemes in MPLS Networks. Dependability of Computer Systems DepCos-RELCOMEX '06, pag. 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): pag. 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): pag. 1012-1024, 1995.

[12] B. Fortz, M Labbé, and F. Maffioli, Solving the two-connected network with bounded meshes problem. Operations Research, 48: pag. 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, pag. 1-4, 2006.

[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, pag. 75-79, 2001.

[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, pag. 761-771, Oct. 2003.

[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, pag. 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: pag. 15-29, 1999.

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

[21] The GLPK (GNU Linear Programming Kit) package for solving large-scale linear programming (LP), mixed integer programming (MIP), and other related problems. http://www.gnu.org/software/glpk/.

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

[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, pag. 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, pag. 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, pag. 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, pag. 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, pag. 249-257, May 2007.