Chapter 1

Optimal Design of a Multi-Layer Network an IP/MPLS Over DWDM Application Case

Claudio Risso, Franco Robledo and Pablo Sartor

Additional information is available at the end of the chapter

http://dx.doi.org/10.5772/54340

1. Introduction

Some decades ago the increasing importance of the telephony service pushed most telecommunications companies (TELCOs) to deploy optical fibre networks. In order to guarantee appropriate service availability, these networks were designed in such a way that several independent paths were available between each pair of nodes, and in order to optimize these large capital investments several models and algorithms were developed.

Already the optimal design of a single layer network is a challenging task that has been considered by many research groups, see for instance the references: [1-3]. Throughout this work this optical network is referred to as the *physical layer*.

Some years afterwards, the exponential growth of Internet traffic volume demanded for higher capacity networks. This demand led to the deployment of dense wavelength division multiplexing (DWDM) technology. Today, DWDM has turned out to be the dominant network technology in high-capacity optical backbone networks. Repeaters and amplifiers must be placed at regular intervals for compensating the loss in optical power while the signal travels along the fibre; hence the cost of a lighpath is proportional to its length over the physical layer. DWDM supports a set of standard high-capacity interfaces (e.g. 1, 2.5, 10 or 40 Gbps). The cost of a connection also depends of the capacity but not proportionally. For economies of scale reasons, the higher the bit-rate the lower the per-bandwidth-cost. The client nodes together with these lightpath connections form a so-called *logical layer* on top of the physical one.

The increasing number of per-physical-link connections -intrinsic to DWDM- may cause multiple logical link failures from a single physical link failure (e.g., fibre cut). This issue led to the development of new multi-layer models aware of the stack of network layers. Most of these models share in common the 1+1 protection mechanism, that is: for every demand two independent lightpaths must be routed such that in case of any single physical link -or even node- failure, at least one of them survive. The following references: [4] and [5] are good examples of this kind of models. Those multi-layer models are suitable for certain families of logical layer technologies such as: synchronous optical networking (SONET) or synchronous digital hierarchy (SDH) since both standards have 1+1 protection as their native protection mechanism.

During many years the connections of IP networks were implemented over SONET/SDH -for simplicity we will only mention SDH from now on-. Most recently: multiprotocol label switching (MPLS), traffic engineering extensions for dynamic routing protocols (e.g. OSPF-TE, ISIS-TE), fast-reroute algorithms (FRR) and other new features were added to the traditional IP routers. This new *technology bundle* known as IP/MPLS, opens a competitive alternative against traditional protection mechanisms based on SDH.

Since IP/MPLS allows recovering from a failure in about 50ms, capital savings may come from the elimination of the intermediate SDH layer. Another improvement of this technology is that the number of paths to route demands between nodes is not pre-bounded; so it might exist in fact a feasible different configuration for most failure scenarios. Since IP/MPLS allows the elimination of an intermediate layer, manages Internet traffic natively, and makes possible a much easier and cheaper operation for virtual private network (VPN) services, it is gaining relative importance every day.

Setting aside technical details and for the purpose of the model presented in this work, we remark two important differences between SDH and IP/MPLS. The first one is the need of SDH to keep different demands between the same nodes. In IP/MPLS networks, all the traffic from one node to another follows the same path in the network referred to as *IP/MPLS tunnel*. The second remarkable difference is how these technologies handle the existence of parallel links in the logical layer. In SDH the existence of parallel links is typical but in IP/MPLS parallel links may conflict with some applications so we will avoid them.

In this paper we address the problem of finding the optimal -minimum cost- configuration of a logical topology over a fixed physical layer. The input data set is constituted by: the physical layer topology -DWDM network-, the client nodes of the logical layer -IP/MPLS nodes- and the potential links between them, as well as the traffic demand to satisfy between each pair of nodes and the per-distance-cost in the physical network associated with the bitrates of the lightpaths to deploy over it. The decision variables are: what logical links do we have to implement, which bitrate must be assigned to each of them and what path do these lightpaths have to follow in the physical layer. For being a feasible solution a configuration must be capable of routing every traffic demand over the remaining active links of the logical layer for every single physical link failure scenario.

The problem previously described is NP-hard and due to its complexity we developed a metaheuristic based on GRASP to find good quality solutions for real size scenarios. Actually, we analysed the performance of the proposed metaheuristic using real-world scenarios provided by the Uruguayan national telecommunications company (ANTEL).

The main contributions of this article are: i) a model to represent a common network overlay design problem; ii) the design of a GRASP metaheuristic to find good quality solutions for this model; iii) the experimental evaluation based on real-world network scenarios.

The remaining of this document is organized as follows. A mixed-integer programming model will be presented in Section-2. In Section-3 we will show some exact solutions found with CPLEX for small/simple but illustrative problems; in this section we also analyze the intrinsic complexity of the problem. In Section-4 a GRASP metaheuristic to solve this problem is presented. Finally, in Section-5 we will show the solutions found with the previous metaheuristic for real-world network scenarios.

2. Mathematical model

We will now introduce the basic mixed-integer programming model that arises from the detailed interaction of technologies.

2.1. Parameters

The physical network is represented by an undirected graph (V, P), and the logical network is represented by another undirected graph (V, L). Both layers share the same set of nodes. The links of the logical layer are potential -admissible logical links- while the links of the physical layer are definite. In both graphs the edges are simple since multigraphs are not allowed in this model.

For every different pair of nodes $p, q \in V$ is known the traffic volume d_{pq} to fulfil along the unique path (tunnel) this traffic follows throughout a logical layer configuration.

These paths are unique at every moment, but in case of link failures they may change to follow an alternate route. For simplicity we assume that the traffic volume is symmetric (i.e. $d_{pq} = d_{qp}$). Let $\hat{B} = \{b_1, \ldots, b_{\bar{B}}\}$ be the set of possible bitrate capacities for the lightpaths on the physical layer and therefore for the links of the logical one. Every capacity $b \in \hat{B}$ has a known per-distance cost c_b. For economies of scale reasons it holds that if $b' < b''$ then $(c_{b'} / b') > (c_{b''} / b'')$. Since both graphs of this model are simple and undirected, we will express links as pairs of nodes. For every physical link (ij) is known its length l_{ij}.

2.2. Variables

This model comprises three classes of variables. The first class is composed of the logical link capacity variables. We will use boolean variables τ_{pq}^{b} to indicate whether or not the logical link $(pq) \in L$ has been assigned with the capacity $b \in \hat{B}$. As a consequence the capacity of the link (pq) could be computed as: $\sum_{b \in \hat{B}} b \cdot \tau_{pq}^{b}$.

The second class of variables determines how are going to be routed the logical links over the physical network. If $\sum_{b \in \hat{B}} \tau_{pq}^{b} = 1$ then the logical link $(pq) \in L$ was assigned with a capacity, it is going to be used in the logical network and requires a lightpath in the physical one. y_{pq}^{ij} is a boolean variable that indicates whether or not the physical link $(ij) \in P$ is being used to implement the lightpath of $(pq) \in L$. Since lightpaths cannot automatically recover from a link failure, whenever a physical link (*ij*) fails all the logical links (*pq*) such that y_{pq}^{ij}=1 do fail as well. The only protection available in this model is that of the logical layer. For demands being protected against single physical link failures, it is necessary to have a feasible route through the remaining active logical links.

The third and final class of variables is that that determines how the IP/MPLS tunnels are going to be routed against any particular failure in the physical layer. ${}^{rs}x_{pq}^{ij}$ is a boolean variable that indicates whether the logical link $(pq) \in L$ is going to be used or not, to route traffic demand d_{rs}>0, under a fault condition in the physical link $(ij) \in P$.

NOTE: To keep the nomenclature of the variables as easy as possible we always placed: logical links subindexes at bottom right position, physical links subindexes at top right position and demands subindexes at top left position.

2.3. Constraints

This problem comprises three groups of constraints. The first group of constraints establishes the rules that the routes of the lightpaths must follow to be feasible.

$$\sum_{b \in \hat{B}} \tau_{pq}^{b} \leq 1 \qquad \forall (pq) \in L. \tag{1}$$

$$\sum_{j/(pj) \in P} y_{pq}^{pj} = \sum_{b \in \hat{B}} \tau_{pq}^{b} \qquad \forall (pq) \in L. \tag{2}$$

$$\sum_{i/(iq) \in P} y_{pq}^{iq} = \sum_{b \in \hat{B}} \tau_{pq}^{b} \qquad \forall (pq) \in L. \tag{3}$$

$$\sum_{j/(ij) \in P} y_{pq}^{ij} = 2\hat{\theta}_{pq}^{i} \qquad \begin{array}{l}\forall (pq) \in L, \forall i \in V, \\ i \neq p, i \neq q.\end{array} \tag{4}$$

$$\tilde{\theta}_{pq}^{i} + \hat{\theta}_{pq}^{i} = 1 \qquad \begin{array}{l}\forall (pq) \in L, \forall i \in V, \\ i \neq p, i \neq q.\end{array} \tag{5}$$

$$y_{pq}^{ij} - y_{pq}^{ji} = 0 \qquad \forall (pq) \in L, \forall (ij) \in P. \tag{6}$$

$$\forall(pq) \in L, \forall(ij) \in P. \qquad \begin{matrix}\forall(pq)\in L, \forall(ij)\in P, \\ \forall b\in \hat{B}, \forall i \in V.\end{matrix} \tag{7}$$

The meaning of the previous constraints is the following: (1) establishes that the number of capacities assigned to every logical link is at most 1 -it could be 0 if the link is not going to be used-; (2) and (3) guarantee that if any particular link $(pq) \in L$ was assigned with a capacity ($\sum_{b\in \hat{B}} \tau^{b}_{pq} = 1$) then there must exist one and only one outgoing -or incoming- physical link used for its lightpath.

Before going any further we have to introduce a set of auxiliary variables: $\tilde{\theta}^{i}_{pq}$ and $\hat{\theta}^{i}_{pq}$. These variables are defined for every combination of logical links $(pq) \in L$ and physical nodes $i \in V$. (5) guarantees that exactly one of the following conditions must meet: $(\tilde{\theta}^{i}_{pq}, \hat{\theta}^{i}_{pq}) = (1, 0)$ or $(\tilde{\theta}^{i}_{pq}, \hat{\theta}^{i}_{pq}) = (0, 1)$. Hence, (4) guarantees flow balance for routing the lightpaths through the remaining -not terminal- nodes. Finally (6) guarantees that the lightpaths go back and forth through the same path, while (7) stands the integrity of the variables.

The second group of constraints establishes the rules that the routes of the IP/MPLS tunnels must follow in the logical layer and their meaning is similar to the previous ones except for (1) and (8). The inequalities in (8) were added to guarantee that whatever the failure scenario is ($\forall\ (ij) \in P$), its associated routing configuration over the logical network keeps the aggregated traffic load below the link capacity for every data link ($\forall\ (pq) \in L$). Constrains (2) and (3) are equivalent to (9) and (10), except for the fact that in the latter the existence of a tunnel relies on the existence of demand and this is known in advance. Another remarkable point is that the second group of constraints has as many possible routing scenarios as arcs in P, so the number of variables is much greater.

$$\sum_{rs:d_{rs}>0} d_{rs} \cdot {}^{rs}x^{ij}_{pq} \leq \sum_{b\in \hat{B}} b \cdot \tau^{b}_{pq} \quad \forall(pq)\in L, \forall(ij)\in P. \tag{8}$$

$$\sum_{q/(rq)\in L} {}^{rs}x^{ij}_{rq} = 1 \qquad \forall d_{rs}>0, \forall(ij)\in P. \tag{9}$$

$$\sum_{p/(ps)\in L} {}^{rs}x^{ij}_{ps} = 1 \qquad \forall d_{rs}>0, \forall(ij)\in P. \tag{10}$$

$$\sum_{q/(pq)\in L} {}^{rs}x^{ij}_{pq} = 2 \cdot {}^{rs}\hat{\mu}^{ij}_{p} \qquad \begin{matrix}\forall d_{rs}>0, \forall(ij)\in P, \\ \forall p\in V, p\neq r, p\neq s.\end{matrix} \tag{11}$$

$${}^{rs}\tilde{\mu}^{ij}_{p} + {}^{rs}\hat{\mu}^{ij}_{p} = 1 \qquad \begin{matrix}\forall d_{rs}>0, \forall(ij)\in P, \\ \forall p\in V, p\neq r, p\neq s.\end{matrix} \tag{12}$$

$$ {}^{rs}x^{ij}_{pq} - {}^{rs}x^{ij}_{qp} = 0 \qquad \begin{array}{l} \forall d_{rs}>0, \forall(pq)\in L, \\ \forall(ij)\in P. \end{array} \tag{13} $$

$$ {}^{rs}x^{ij}_{pq}, {}^{rs}\tilde{\mu}^{ij}_{p}, {}^{rs}\hat{\mu}^{ij}_{p} \in \{0,1\} \qquad \begin{array}{l} \forall d_{rs}>0, \forall(pq)\in L, \\ \forall(ij)\in P, \forall p\in V. \end{array} \tag{14} $$

Variables sets ${}^{rs}\tilde{\mu}^{i}_{pq}$ and ${}^{rs}\hat{\mu}^{i}_{pq}$ are homologous to $\tilde{\theta}^{i}_{pq}$ and $\hat{\theta}^{i}_{pq}$; so are constraints from (4) to (7) with those from (11) to (14). Before proceeding any further we must notice that both groups are not independent. Many logical links may not be available for routing after a physical link failure. Which logical links are in this condition, relies on how the lightpaths were routed in the physical layer. Specifically, if some logical link (pq) uses a physical link (ij) for its lightpath implementation then this logical link cannot be used to route any tunnel under (ij) failure scenario.

$$ \left\{ {}^{rs}x^{ij}_{pq} \le 1 - y^{ij}_{pq} \right. \qquad \forall rs : d_{rs} > 0, \forall(pq) \in L, \forall(ij) \in P, \tag{15} $$

The group of constrains (15) prevents from using (pq) to route any traffic (${}^{rs}x^{ij}_{pq}=0$, $\forall\ rs : d_{rs}>0$) in any failure scenario which affects the link (when $y^{ij}_{pq}=1$).

2.4. Objective

The function to minimize is the sum of the cost of every logical link. According on what capacity was assigned to a logical link there is an associated per-distance-cost (c_b), and according on how the corresponding lightpath was routed over the physical layer there is an associated length ($\sum_{(ij)\in P} l_{ij} y^{ij}_{pq}$). The product of both terms is the cost of a particular logical link and the sum of these products for all of the logical links is the total cost of the solution. The direct arithmetic expression for the previous statement would be: $\sum_{(pq)\in L} \ (\sum_{b\in\hat{B}} c_b \tau^{b}_{pq})(\sum_{(ij)\in P} l_{ij} y^{ij}_{pq}) = \sum_{(pq)\in L \ ,(ij)\in P, b\in\hat{B}} c_b l_{ij} \cdot \tau^{b}_{pq} y^{ij}_{pq}$.

Although straightforward, this approximation is inappropriate because it is not linear. The following sub-problem expresses the objective value with an equivalent linear expression.

$$ \min \sum_{\substack{(pq)\in L \\ (ij)\in P \\ b\in\hat{B}}} c_b l_{ij} \cdot {}^{b}\eta^{ij}_{pq} \tag{16} $$

$$ {}^{b}\eta^{ij}_{pq} \ge \tau^{b}_{pq} + y^{ij}_{pq} - 1 \qquad \begin{array}{l} \forall(pq)\in L, \forall(ij)\in P, \\ \forall b\in\hat{B}. \end{array} \tag{17} $$

$$ {}^{b}\eta^{ij}_{pq} \geq 0 \qquad \begin{array}{l} \forall(pq)\in L, \forall(ij)\in P, \\ \forall b\in\hat{B}. \end{array} \tag{18} $$

We used the real variable ${}^{b}\eta^{ij}_{pq}$ instead of $\tau^{b}_{pq} y^{ij}_{pq}$ and added some extra constraints to guarantee the consistency. This consistency comes from the following observations: the result of $\tau^{b}_{pq} y^{ij}_{pq}$ is also a boolean variable, and since ${}^{b}\eta^{ij}_{pq}$ is being multiplied by a positive constant in a minimization problem it will take its lowest value whenever this is possible. This value would be zero because of constrains (3) of (D).

The only exception is when the values of τ^{b}_{pq} and y^{ij}_{pq} are both 1, in which case the value of ${}^{b}\eta^{ij}_{pq}$ should be 1 as well to keep consistency. This is guaranteed by constrain (17). The complete MIP is the result of merging (1) to (18).

3. Finding exact solutions

We will start by showing particular solutions for some simple example cases. The first example has four nodes $V = \{v_1, v_2, v_3, v_4\}$, the physical layer is the cycle (C^4) while the logical layer is the clique (K^4). The remaining parameters are: $\hat{B} = \{3\}$, $d_{pq} = 1 \ \forall\, 1 \leq p < q \leq 4$ and $l_{ij} = 1, \ \forall\, (ij) \in P$. c_b is irrelevant in this case because there is only one bitrate available. The optimal solution found for this case uses all of the logical links. Figure 1 shows with dashed lines the route in that solution followed for each lightpath over the physical cycle. This is an example where lightpaths routes are not intuitive, even for a very simple input data set.

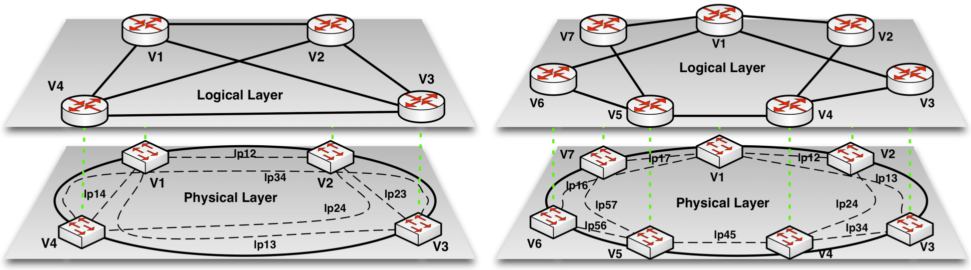

Figure 1. Optimal solutions found for: K^4 over C^4, with $d_{pq} = 1$ and $\hat{B} = \{3\}$, and for: K^7over C^7, with $d_{1q} = 1$ and $\hat{B} = \{3\}$.

The following example comprises seven nodes and explores again the clique-over-cycle case. The remaining parameters are analogous: $\hat{B} = \{3\}$, $l_{ij} = 1 \ \forall\, (ij) \in P$, except for demands that in this case are to/from one single node ($d_{1q} = 1, \forall\, 1 < q \leq 7$). Unlike the previous example, the optimal solution in this case (also sketched in Figure 1) does not make use of all the logical links. Although the route followed by each lightpath looks more natural in this example, it is not immediate why this set of logical links ought to be the appropriate to construct the optimal solution.

Through these two examples we attempted to show that solutions are not intuitive even for very simple cases. To find optimal solutions we used ILOG CPLEX v12.1. All computations were performed on a Linux machine with an INTEL CORE i3 Processor and 4GB of DDR3 RAM. Table 1 shows information for several test instances analogous to those represented in Figure 1, that is: K^n over C^n with $d_{pq}=1,\ \forall\, 1\leq p<q\leq n$ and $l_{ij}=1,\ \forall\,(ij)\in P$ over a range of integer b_1 values ($|\hat{B}|=1$).

$\|V\|$	b_1 range	#variables	#constrains	Elapsed time (hh:mm:ss)
5	2 – 6	1230	1640	00:00:00 – 000:00:11
6	3 – 9	3390	4035	00:00:02 – 000:19:31
7	2 – 12	7896	8652	(*) 00:00:05 – 087:19:05
8	3 – 16	16296	16772	(*) 00:00:02 – 100:10:17

(*)Note: The solver aborted for some intermediate cases

Table 1. Overall results for some particular cases

We proved that: it is always possible to find minimal feasible solutions for these particular topologies and demands when: $b_1=2$ and $|V|$ is odd, or when $b_1=3$ and $|V|$ is even. In the first situation the complete logical graph is needed, whereas in the second only diagonal links can be disposed of. The lowest computation times were found for these extreme cases.

We also proved that: the cycle configuration for the logical network -the simplest possible- is feasible for every b_1 greater or equal to: $|V|^2/4$ when $|V|$ is even, or $(|V|^2-1)/4$ when $|V|$ is odd. Very low computation times were found for these cases also. The time required for finding optimal solutions for non-extreme cases were much higher. CPLEX even aborted for many of them. Aside from a bunch of worthless exceptions, we couldn't find solutions for topologies other than K^n over C^n.

Keeping these physical and logical topologies while trying with simpler matrices of demand (e.g. $d_{1q}=1, \forall\, 1<q\leq n$) it was possible to increase the size of the problems to 15 nodes and yet being able to find optimal solutions. Suffices to say this size bound as well as the simplicity in the topologies and traffic matrices of the previous examples, are incompatible with real life network problems.

Proposition 1: The problem presented in this section is NP-Hard.

Demonstration lies under reduction of NPP (Number Partitioning Problem) to our particular problem that will be referred to as MORNP (Multi-Overlay Robust Network Problem) within this proof. NPP problem consist in finding two subsets with the same sum for a known multiset of numbers. Formally: given a list of positive integers: $a_1, a_2, \ldots, a_N$, a partition $A \subseteq \{1, 2, \ldots, N\}$ must be found so that discrepancy:

$$E(A) = \left| \sum_{i \in A} a_i - \sum_{i \notin A} a_i \right|, \tag{19}$$

finds its minimum value within the set $\{0, 1\}$. NPP is a very well known NP-Complete problem (see for instance [6]).

$(\Rightarrow)$ Given such a list of positive integers we create an instance of MORNP by taking: $B = \{b_1 = \lceil (\sum_{1 \le i \le N} a_i) / 2 \rceil\}$, logical and physical graphs with the same topology schematized in Figure 2, $l_{ij} = 1 \; \forall \, (ij) \in P$ and $d_{iD} = a_i, \forall \; 1 \le i \le N$.

Since logical and physical topologies are the same and all distances are equal, the logical layer projected over the physical one for any optimal solution must copy the underlying shape. So, if there exists a solution for such an instance of MORNP, this solution should have a feasible routing scenario when transport -and logical- link $(v_{A1} v_{A2})$ fails and therefore a way to accommodate traffic requirements over $(v_{H1} v_D)$ and $(v_{H2} v_D)$, due to the fact that both links are still in operational state and they are the only way to reach v_D.

Because there is only one capacity both links must have been assigned with b_1, this can only be done when discrepancy is not grater than one, so we indirectly found a solution for the original NPP problem.

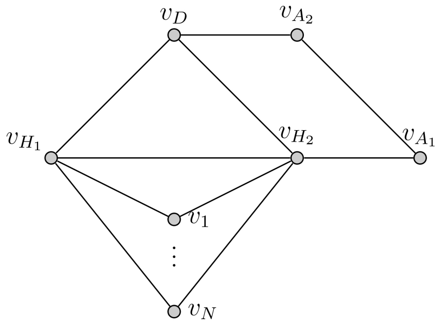

Figure 2. Graph used for NPP reduction to MORNP.

$(\Leftarrow)$ The complementary part of the proof is easier. Given a solution to an instance of NPP, this partition is used to distribute tunnels between $(v_{H1} v_D)$ and $(v_{H2} v_D)$. Once in v_{H1} or $v_{H2}$the tunnel is terminated directly in the corresponding node, except for some fault condition in one of these links, in which case a detour through the other v_{Hx} node is always possible. When the fault condition arises in $(v_{H1}v_D)$ or $(v_{H2}v_D)$, a detour may be taken through: (v_Dv_{A2}), $(v_{A2}v_{A1})$, $(v_{A1}v_{H2})$ or (v_Dv_{A2}), $(v_{A2}v_{A1})$, $(v_{A1}v_{H2})$, $(v_{H2}v_{H1})$.

Since all the transformation are of polynomial complexity it stands that $NPP \prec MORNP$ and MORNP is NP-Hard.

We proved the complexity of MORNP is intrinsic to the problem, since it is NP-Hard. Because of the previous result and like for most other network design problems, an exhaustive search for the optimal solution of the problem presented in this work is infeasible for real size problems.

4. Metaheuristics

We decided to use a metaheuristic algorithm based on GRASP to find good quality solutions for real instances of this problem. A very high level diagram of our algorithm is shown in Figure 3.

4.1. GRASP implementation

As for every GRASP implementation this algorithm has a loop with two phases. The *construction phase* builds a *randomized feasible solution*, from which a local minimum is found during the *local search phase*. This procedure is repeated *MaxIter* times, while the best overall solution is kept as the result. Further information and details in GRASP algorithms can be found in [7] or in [8].

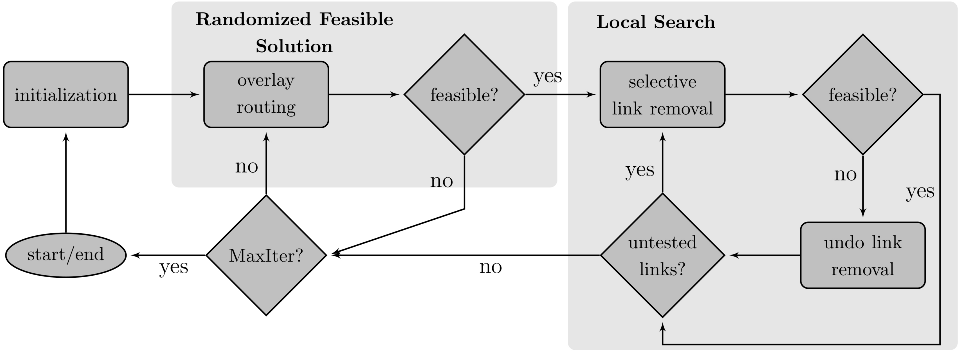

Figure 3. Block-diagram of the GRASP implementation used.

The *initialization phase* performs computations whose results are invariants among the iterations, like the shortest path and distance over the physical layer between each pair of nodes.

4.2. Construction phase

The *randomized feasible solution phase* performs a heuristic low cost balanced routing of the logical layer over the physical one. The exact solution for this sub-problem is also NP-Complete as it can be seen in [9]. The goal is to find a path for every lightpath, such that the number of physical link intersections is minimum. It is also desirable that the total cost be as low as possible but as a secondary priority. The strategy chosen in this heuristic is the following: nodes are taken randomly (.e.g.: uniformly), and for each node their logical links are also taken randomly but with probabilities in inverse ratio to the minimal possible distance of their lightpaths over the physical layer.

Instead of using the real distances of the physical links (l_{ij}), from this point on and until the next iteration pseudo-distances: l_{ij}, $\forall\,(ij) \in P$ will be used. Prior to start routing lightpaths, all these pseudo-distances are set to 1.

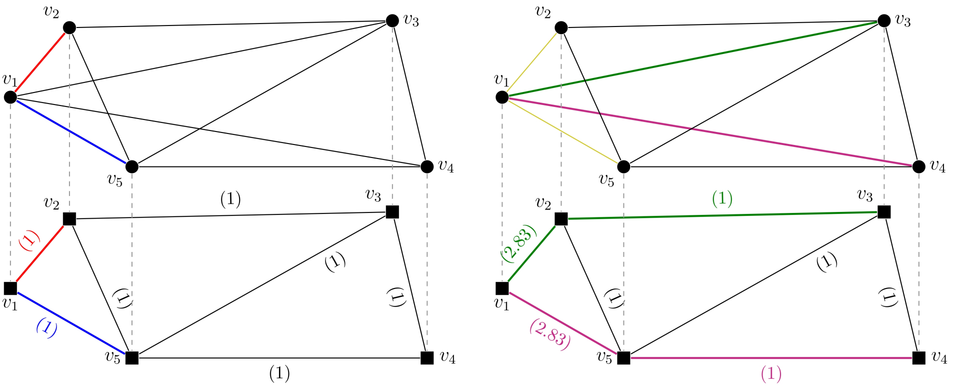

Figure 4. An example of the balanced routing heuristic.

According to these new weights, logical links are routed following the minimal distance without repeating physical links among them. Usually, after routing some lightpaths the set of not-yet-used physical links empties, and it is necessary to start over a new *control window* by filling again the not-yet-used set. Prior to do this, the pseudo-distances are updated using the following rule: $l_{ij} = (1 + n_{ij})^p$ for some fixed penalty p, where n_{ij} is the number of lightpaths that are making use of $(ij) \in P$ up to the moment.

For instance, let us guess our networks are like those sketched in Figure 4 and the links drawn are: (12), (15), (13), (14), (23), (35), and so on. The left half of Figure 4 shows with red and blue lines how are routed the lightpaths (12) and (15). At this point we need to update the pseudo-distances and restart the window. If $p = 1.5$ and since $n_{12} = n_{15} = 1$, then $l_{12} = l_{15} = 2^{1.5} \approx 2.83$ for the next window.

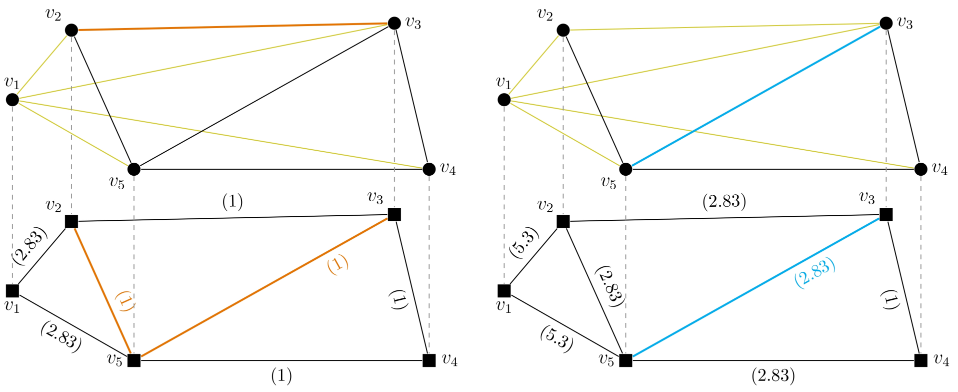

Figure 5. Lightpaths for logical links (23) and (35).

The next two logical links are (13) and (14). They are routed using the updated values. Their lightpaths are also represented with green and magenta lines in the right half of Figure 4. The link (23) is the following and it can be routed in two hops. A window restart is necessary to route the lightpath of (35), as it can be seen in Figure 5.

The elements of the input data in Algorithm-1 are: the logical graph (V, L), the physical graph (V, P), the minimum distance over the physical layer to connect each pair of nodes -computed in the *initialization phase*-. The output is an application between logical links and the subset of physical links used by their lightpaths.

The algorithm detailed in Algorithm-1 is the one depicted in Figure 4 and Figure 5. The outcome of the randomized feasible solution phase is a candidate configuration for the route of each lightpath over the physical network. We did not make use of capacity and traffic information yet; and before going any further we must state that -as in the exact examples- in our practical applications we limited the capacities set to only one capacity ($|\hat{B}| = 1$).

The main reason was that the telecommunications company we developed this application for, wanted the maximum possible bitrate for all the interfaces of its core network. The next issue is determining whether the configuration found is feasible or not. The answer to this question is far from being easy, since this sub-problem is NP-Complete. We have based on a heuristic to answer this question. The heuristic is the following: demands are taken in decreasing order of volume (d_{pq}) and each tunnel is routed over the logical layer following the minimal number of hops, but using only links with remaining capacity to allocate the new tunnel demand.

For instance, Figure 6 shows an example logical topology whose link capacities are 3. Let the demands be: $d_{24}=2$, $d_{12}=1$, $d_{13}=1$ and $d_{23}=1$. The path followed by every tunnel is sketched in Figure 6 using: violet, orange, red, and green curves respectively; so it is the remaining capacity in every link after routing each tunnel -two tunnels in the central image-.

Overlay routing (logical over physical). Algorithm 1

Input: $(V, L), (V, P), d : V \times V \to \mathbb{R}_0^+$. **Output:** $\Psi : L \to 2^P$.

```
Set $\Psi(e) = \emptyset, \forall e \in L$ and $pd : P \to \mathbb{R}_0^+, pd(e) = 1 \, \forall e \in P$.
while $\exists v \in V$ / not-processed$(v)$ do
    Select randomly $v \in V$ / not-processed$(v)$);
    Set $prob(vw) = \frac{1}{d(v,w)}, \forall (vw) \in L \,/\, \Psi(vw) = \emptyset$;
    Normalize $prob$ such that: $\sum_{e \in L} prob(e) = 1$;
    Create new control window;
    while $\exists w \in V \,/\, ((vw) \in L$ AND $\Psi(vw) = \emptyset)$ do
        Draw such $w \in V$ randomly weighted by $prob(vw)$;
        Find shortest-lp, a pd-distance shortest lightpath for $(vw)$ avoiding repeating physical
        links within this window;
        if (shortest-lp=$\emptyset$) AND (there are not unprocessed $(vw)$) then
            Update $\mathrm{pd}(v, w) = \mathrm{pd}(w, v) = (1 + \sum_{e \in L} |\Psi(e) \cap \{(vw)\}|)^p$;
            Create new control window;
        else
            $\Psi(v_e v_f) = \Psi(v_f v_e) =$ shortest-lp;
        end if
    end while
end while
return $\Psi : L \to 2^P$.
```

Algorithm 1.

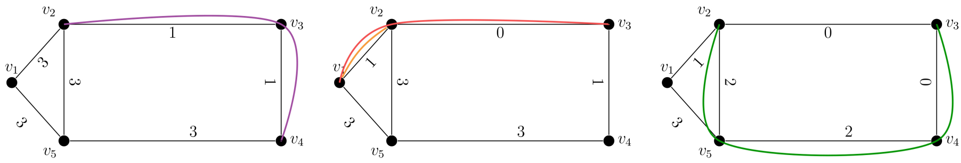

Figure 6. Routes for the tunnels (24), (12), (13), and (23)over a Logical Layer.

This constraint based routing algorithm is straightforward and it is based on Dijkstra's algorithm. Nevertheless an efficient implementation is quite complicated because of the following fact: to be sure a solution is feasible this algorithm must be repeated for each single failure scenario. In order to improve the efficiency: routes cache, optimized data structures and several others low-level programming techniques were used. This isFeasible function is used in both: construction and local search phases. The performance of this function is critic since it is used several times within the same iteration in the local search, as it is represented in Figure 3. Up to this point and before entering the local search phase, we have a feasible configuration for the routes of every lightpath; but we are still using all of the initial logical links and this input network is very likely to be over-sized. Moreover, in the construction phase we attempted to distribute the routes of the lightpaths uniformly over the physical layer, but it is still possible that many logical links fail simultaneously because of a single physical link failure. Therefore, it is very likely that many of these "redundant links" may be disposed of, if they are not really adding useful capacity. It is worth mentioning that from this point on and until the next iteration, lightpaths costs are revealed because we have their lengths -from the configuration for their routes- and there is only one possible capacity.

4.3. Local search

Through the local search phase we intend to remove the most expensive and unnecessary logical links for the current configuration. The process is the following: logical links are taken in decreasing order of cost for their lightpaths, each one is removed and the feasibility of the solution is tested again. If the solution remains feasible the current logical link is permanently removed, otherwise it is reinserted and the sequence follows for the remaining logical links. Once this processes is finished the result is a minimal solution. After *MaxIter* iterations the best solution found is chosen to be the output of the algorithm. Since the construction procedure we have used in this work privileges the nodes drawn earlier to shape the routes of the lightpaths, we presume that adding path-relinking to this algorithm could significantly improve the quality of the result, if the initial lightpaths routes of the elite solutions are prioritized to explore new solutions. We are planning to check this assumption in a future work. For further information in path-relinking enhancement to GRASP, please refer to: [7] and [10].

5. Application case context and results

We will focus now in the context of ANTEL -the telecommunication company we applied this metaheuristic to-. Prior to doing so we are giving some basic elements of the overall Internet architecture. Internet is actually a network that could be disaggregated into several separate smaller networks also known as *Autonomous Systems* (AS). Typically every AS is a portion of the global Internet owned/governed by a particular Internet Service Provider (ISP). Internet users access content residing in servers of: companies, universities, government sites or even from other residential customers (e.g. P2P applications). A portion of this content is located within the own network of the ISP this customer lease the service to -into some of the *Points Of Presence* (POP) of the ISP-, but most content is scattered over the Internet. Since traffic interchange is necessary among different ISPs, the Internet architecture needs special POPs known as *Network Access Points* (NAPs). Within these NAPs: Carriers, ISPs and important content providers (e.g.: Google, Akamai) connect to each other in order to interchange traffic.

This company had two different IP/MPLS networks referred to as: *aggregation network* and *public Internet network*. The *aggregation network* is geographically dispersed all over the country and it is responsible of gathering and delivering the traffic of the customers to the *public Internet network*. The *public Internet network* is where the AS of this ISP is implemented; centralizes the international connections with other ISPs as well as those to Datacenters of local content providers. The *public Internet network* is geographically concentrated and only has POPs in the Capital City and in an important NAP of the US territory (see grey clouds in Figure 7). In terms of the model covered in this article we may stand that the physical network has all of its nodes but one -the NAP- within the national boundaries.

There are four independent paths for international connections -leased to Carriers- between the NAP and the national boundaries. The *aggregation* and *public Internet* networks are both logical. The *public Internet network* only has presence in a few POPs of the Capital City and in the NAP; and although the *aggregation network* has full-national presence it does not span to the NAP. A Non-Disclosure Agreement (NDA) signed between the telecommunications company and our research institute protects more accurate information and details. The costs and traffic information shown in the rest of this article are only referential.

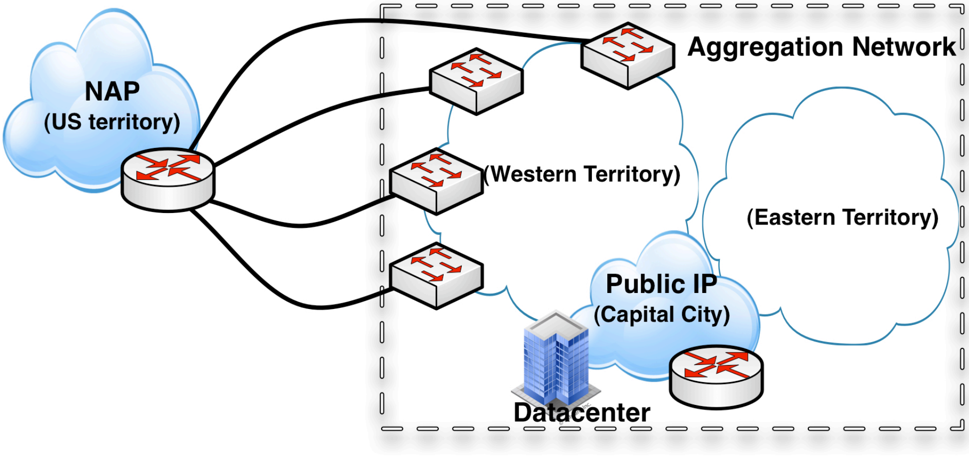

Figure 7. Remarkable aspects of the particular network architecture.

Several planning concerns arose from the situation exposed: Is it convenient the current architecture? Or it would be better to merge both IP/MPLS networks? Are profitable the IT infrastructure investments necessary to increase the percentage of local content? Which would be the optimal network to fulfil every demand requirements scenario? We helped to answer these questions by identifying representative scenarios and creating their associated data sets to feed the metaheuristic.

The overall performance of the algorithm described in Section-4 was very good -under the two hours of execution time in every scenario-.

We tried several scenarios based on the following considerations: traffic volume, network architecture and the percentage of locally terminated traffic. We selected eight remarkable scenarios to detail in Table 2.

According on traffic forecasts it is expected that some years from now the total volume of traffic to be placed somewhere between 57 and 100 (reference values). If some IT investments and agreements were made it is expected that the percentage of locally terminated traffic (national traffic) could be greater (High). These new potential sources of traffic would be placed in the Capital City, specifically in the same POPs where the public Internet network is present. Those scenarios where merged networks is set to False inherit the current network architecture.

scenario index	aggregated traffic demand	%local content	merged networks	number of nodes	required lightpaths	total cost
1	100	Low	False	56	81	10,000,000
2	100	Low	True	68	118	7,662,651
3	100	High	False	56	81	7,578,234
4	100	High	True	68	133	5,713,563
5	57	Low	False	56	75	6,319,470
6	57	Low	True	63	94	4,872,987
7	57	High	False	56	75	5,108,587
8	57	High	True	63	105	4,064,597

Table 2. Referential results for representative scenarios

Another remarkable aspect of this architecture is that whereas the aggregation network is deployed directly over the physical layer, there is an extra SDH layer between the public Internet network and the physical one. Since the public Internet network only has a few nodes and its protection relies on the 1+1 protection mechanism of SDH, its optimal value can be estimated easily. The only portion where we needed computer assistance is that of the aggregation network. The columns number of nodes and required lightpaths refers exclusively to the values for this last network.

On the other hand and in order to compare solutions fairly, the column *total cost* represents the combined cost of both networks -when they are not joined-. It is worth observing those scenarios: 1 and 3, as well as 5 and 7 require the same number of lightpaths. Moreover, their solutions use exactly the same lightpaths. This result should be expected because in both pairs of scenarios share the same traffic and non-merged network architecture; since Datacenters - the only difference- are connected to the *public Internet network*, the *aggregation network* is unaware of the percentage of local content. The only changes are in the *total cost* because of the saving of international capacity.

Less intuitive are those savings arising exclusively from the merging of both networks like: 1 and 2, 3 and 4, and so on. The reason is the following: “the routing search-space of the IP/MPLS technology is much greater than that of the SDH equivalent, so it is much more efficient”. For simplicity let us guess for a while that traffic does not need to be fitted in tunnels and instead can behave as a fluid. Since the length of international connections is measured in thousands of kilometres, this links are the most expensive of the physical network. As it was showed in Figure 7 there are four independent connections to the NAP; hence if we needed to guarantee 60Gbps of international traffic we could reserve 20Gbps in every one of these links, because a single failure could only affect one of them. Therefore the efficiency in the usage of international connections could rise to 75% if the efficiency of IP/MPLS would be available. The protection mechanism of SDH (1+1) cannot exploit this degree of connectivity. To protect 60Gbps of traffic using SDH active/stand-by independent paths, we always need other 60Gbps of reserved capacity, so the efficiency of SDH it is limited to 50%. The improved efficiency of IP/MPLS to exploit the extra connectivity degree between local and international traffic explains by itself most of the savings.

6. Conclusion

Perhaps the most remarkable result of this work relies on exposing through a real-world application, how much more cost-efficient could be networks deployed using IP/MPLS, than those based on traditional protection schemes like SDH. This efficiency comes not only from the savings linked to the elimination of an intermediate layer, but also from the extra degrees of freedom available to route the traffic.

We presume that the application this work dealt with is not an exception, and the potential savings might replicate from one ISP to the other. The results for the examples of Section-5 and their later analysis justify the convenience to update network design models this work introduced, in order to follow the new technology trends and exploit their benefits.

Regarding on the metaheuristic, we presume that applying path-relinking could significantly increase the computational efficiency of the proposed GRASP, so this is the line of our immediate future work.

Author details

Claudio Risso*, Franco Robledo and Pablo Sartor*

*Address all correspondence to: crisso@fing.edu.uy

Engineering Faculty – University of the Republic, Montevideo, Uruguay

References

[1] Okamura H. and Seymour P., (1981), “Multicommodity flows in planar graphs”, Journal of Combinatorial Theory 31(1), pp. 75–81.

[2] Stoer M., (1992), "Design of survivable networks", Lecture Notes in Mathematics.

[3] Kerivin H. et al., (2005), "Design of survivable networks: A survey", Networks 46(1), pp. 1–21.

[4] Orlowski S. et al., (2007), "Two-layer network design by branch-and-cut featuring MIP-based heuristics", Proceedings of the 3rd International Network Optimization Conference (INOC).

[5] Koster A. et al., (2008), "Single-layer Cuts for Multi-layer Network Design Problems", Proceedings of the 9th INFORMS Telecommunications Conference, Vol. 44, Chap. 1, pp.1-23.

[6] Mertens. S., (2006) "The Easiest Hard Problem: Number Partitioning", "Computational Complexity and Statistical Physics", pp.125-139, Oxford University Press, New York, http://arxiv.org/abs/cond- mat/0302536.

[7] Resende M., Riberio C., (2003), "Greedy randomized adaptive search procedures", ATT Research, http://www2.research.att.com/~mgcr/doc/sgrasp-hmetah.pdf.

[8] Resende M., Pardalos P., (2006), "Handbook of Optimization in Telecommunication", Springer Science + Business Media.

[9] Oellrich M., (2008), "Minimum Cost Disjoint Paths under Arc Dependence", University of Technology Berlin.

[10] Glover F., (1996), "Tabu search and adaptive memory programming - Advances, applications and challenges", Interfaces in Computer Science and Operations Research, pp.1-75.