

Optimal Design of a Multi-Layer Network An IP/MPLS over DWDM application case

Claudio Risso and Franco Robledo

Faculty of Engineering, University of the Republic,
Julio Herrera y Reissig 565, Montevideo, Uruguay.
e-mail: {crisso,frobledo}@fing.edu.uy

Abstract

In this paper we study a network design problem arising from the deployment of an IP/MPLS network over an existing transport infrastructure. The goal is to find a minimum cost installation of links such that traffic demands can resiliently be accomplished. We present an integer programming formulation for our problem and metaheuristics to find good quality solutions. This work is based on a real application case for a south american telecommunications company.

Keywords: telecommunications network; multi-layer network design; GRASP.

Topics: MH, OC, TEL & SI.

1 Introduction

Some decades ago the increasing importance of the telephony service pushed most telecommunications companies (TELCOs) to deploy optical fiber 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: Okamura (1981), Stoer (1998) and Kerivin (2005). 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 fiber; hence the cost of a lightpath 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., fiber 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, i.e., 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: Orłowski (2007) and Koster (2008) 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.

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. This 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; we will also analyze the intrinsic complexity of the problem. Due to the NP-Hardness of the problem, for finding solutions to real size problems we developed a metaheuristic based on GRASP that is presented in Section-4. Finally, in Section-5 we will show the solutions found through the previous metaheuristic.

2 Mathematical model

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

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 fulfill along the unique path 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 $\hat{B} = \{b_1, \dots, b_{\hat{B}}\}$ 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} .

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) . 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. 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 third and final class of variables is that that determines how the IP/MPLS tunnels are going to be routed against a failure in a physical link. ${}^{rs}x_{pq}^{ij}$ 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.

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.

$$\left\{ \begin{array}{ll} \sum_{b \in \hat{B}} \tau_{pq}^b \leq 1 & \forall (pq) \in L. \quad (1) \\ \sum_{j/(pj) \in P} y_{pq}^{pj} = \sum_{b \in \hat{B}} \tau_{pq}^b & \forall (pq) \in L. \quad (2) \\ \sum_{i/(iq) \in P} y_{pq}^{iq} = \sum_{b \in \hat{B}} \tau_{pq}^b & \forall (pq) \in L. \quad (3) \\ \sum_{j/(ij) \in P} y_{pq}^{ij} = 2\hat{\theta}_{pq}^i & \forall (pq) \in L, \forall i \in V, \\ & i \neq p, i \neq q. \quad (4) \\ \tilde{\theta}_{pq}^i + \hat{\theta}_{pq}^i = 1 & \forall (pq) \in L, \forall i \in V, \\ & i \neq p, i \neq q. \quad (5) \\ y_{pq}^{ij} - y_{pq}^{ji} = 0 & \forall (pq) \in L, \forall (ij) \in P. \quad (6) \\ \tau_{pq}^b, y_{pq}^{ij}, \hat{\theta}_{pq}^i, \tilde{\theta}_{pq}^i \in \{0, 1\} & \forall (pq) \in L, \forall (ij) \in P \\ & \forall b \in \hat{B}, \forall i \in V. \quad (7) \end{array} \right.$$

The meaning of constraints in group (2) 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 ($\sum_{b \in \hat{B}} \tau_{pq}^b = 1$) 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: $\tilde{\theta}_{pq}^i$ and $\hat{\theta}_{pq}^i$. 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: $(\tilde{\theta}, \hat{\theta}) = (1, 0)$ or $(\tilde{\theta}, \hat{\theta}) = (0, 1)$. 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. The meaning of the constraints in (B) is similar to those of (2) except for (1). The inequality in (1) 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) from (2) and (B) are equivalent, 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 (B) has as many possible routing scenarios as arcs in P , so the number of variables is much greater than those of (2).

$$\left\{ \begin{array}{ll}
 \sum_{rs:d_{rs}>0} d_{rs} \cdot r^s x_{pq}^{ij} \leq \sum_{b \in \hat{B}} b \cdot \tau_{pq}^b & \forall(pq) \in L, \forall(ij) \in P. \quad (1) \\
 \sum_{q/(rq) \in L} r^s x_{rq}^{ij} = 1 & \forall d_{rs} > 0, \forall(ij) \in P. \quad (2) \\
 \sum_{p/(ps) \in L} r^s x_{ps}^{ij} = 1 & \forall d_{rs} > 0, \forall(ij) \in P. \quad (3) \\
 \sum_{q/(pq) \in L} r^s x_{pq}^{ij} = 2 \cdot r^s \hat{\mu}_p^{ij} & \forall d_{rs} > 0, \forall(ij) \in P, \\
 & \forall p \in V, p \neq r, p \neq s. \quad (4) \\
 r^s \tilde{\mu}_p^{ij} + r^s \hat{\mu}_p^{ij} = 1 & \forall d_{rs} > 0, \forall(ij) \in P, \\
 & \forall p \in V, p \neq r, p \neq s. \quad (5) \\
 r^s x_{pq}^{ij} - r^s x_{qp}^{ij} = 0 & \forall d_{rs} > 0, \forall(pq) \in L, \\
 & \forall(ij) \in P. \quad (6) \\
 r^s x_{pq}^{ij}, r^s \tilde{\mu}_p^{ij}, r^s \hat{\mu}_p^{ij} \in \{0, 1\} & \forall d_{rs} > 0, \forall(pq) \in L, \\
 & \forall(ij) \in P, \forall p \in V. \quad (7)
 \end{array} \right. \quad (B)$$

Variables sets $r^s \tilde{\mu}_p^{ij}$ and $r^s \hat{\mu}_p^{ij}$ are homologous to $\hat{\theta}_{pq}^i$ and $\tilde{\theta}_{pq}^i$; so are constraints from (4) to (7). Before proceeding any further we must notice that (2) and (B) 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.

$$r^s x_{pq}^{ij} \leq 1 - y_{pq}^{ij} \quad \forall rs:d_{rs}>0, \forall(pq) \in L, \forall(ij) \in P. \quad (C)$$

The group of constrains (C) prevents from using (pq) to route any traffic ($r^s x_{pq}^{ij} = 0, \forall rs : d_{rs} > 0$) in any failure scenario that affects the link (when $y_{pq}^{ij} = 1$).

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 it has an associated length ($\sum_{(ij) \in P} l_{ij} y_{pq}^{ij}$). The product of both terms is the cost of a particular logical link and the sum of these products for all the logical links is the total cost of the solution. The direct arithmetic expression for the previous statement would be: $\sum_{(pq) \in L} (\sum_{b \in \hat{B}} c_b \tau_{pq}^b) (\sum_{(ij) \in P} l_{ij} y_{pq}^{ij}) = \sum_{(pq) \in L, (ij) \in P, b \in \hat{B}} c_b l_{ij} \cdot \tau_{pq}^b y_{pq}^{ij}$.

Although straightforward, this approximation is inappropriate because it is non-linear. The sub-problem (D) expresses the objective value with an equivalent linear expression. We used the real variable $b_{\eta_{pq}^{ij}}$ instead of $\tau_{pq}^b y_{pq}^{ij}$ and added some extra constraints to guarantee the consistency. This consistency arises from the following observations: the result of $\tau_{pq}^b y_{pq}^{ij}$ is also a boolean variable, and since $b_{\eta_{pq}^{ij}}$ is being multiplied by a positive constant in a minimization problem it will take its lowest value (zero) whenever this is possible.

$$\left\{ \begin{array}{l} \min \sum_{\substack{(pq) \in L \\ (ij) \in P \\ b \in \hat{B}}} c_b l_{ij} \cdot b_{\eta_{pq}^{ij}} \\ b_{\eta_{pq}^{ij}} \geq \tau_{pq}^b + y_{pq}^{ij} - 1 \\ b_{\eta_{pq}^{ij}} \geq 0 \end{array} \right. \quad \begin{array}{l} (1) \\ \forall (pq) \in L, \forall (ij) \in P, \\ \forall b \in \hat{B}. \\ (2) \\ \forall (pq) \in L, \forall (ij) \in P, \\ \forall b \in \hat{B}. \\ (3) \end{array} \quad (D)$$

The only exception is when the values of τ_{pq}^b and y_{pq}^{ij} are both 1, in which case the value of $b_{\eta_{pq}^{ij}}$ should be 1 as well to keep consistency. This is guaranteed by constrain (2) of (D). The complete MIP is the result of merging: (2), (B), (C) and (D).

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 (\mathcal{C}^4) while the logical layer is the clique (\mathcal{K}^4). The remaining parameters are: $B = \{3\}$, $d_{pq} = 1, \forall 1 \leq p < q \leq 4$ 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.

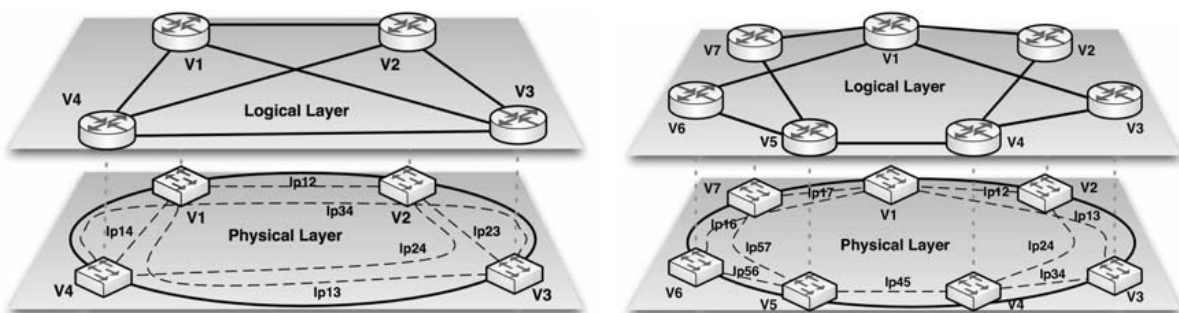


Figure 1: Optimal solutions found for: \mathcal{K}^4 over \mathcal{C}^4 , with $d_{pq} = 1$ and $B = \{3\}$, and for \mathcal{K}^7 over \mathcal{C}^7 , with $d_{1q} = 1$ and $B = \{3\}$

The following example comprises seven nodes and explores again the clique-over-cycle case. The remaining parameters are analogous: $B = \{3\}$, $l_{ij} = 1, \forall (ij) \in P$, 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 similar to those represented in Figure-1, that is: \mathcal{K}^n over \mathcal{C}^n with $d_{pq} = 1, \forall 1 \leq p < q \leq 4$ 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(*)

Table 1: Overall results for some particular cases

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

We proven 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 \mathcal{K}^n over \mathcal{C}^n . Keeping these physical and logical topologies but trying with simpler matrices of demand (e.g. $d_{1q} = 1, \forall 1 < q \leq |V|$) 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 network problems. We proved this complexity is intrinsic to the problem, since it is NP-Hard. The previous result shows that 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-2.

GRASP customization

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 Resende (2003) or in Resende (2006).

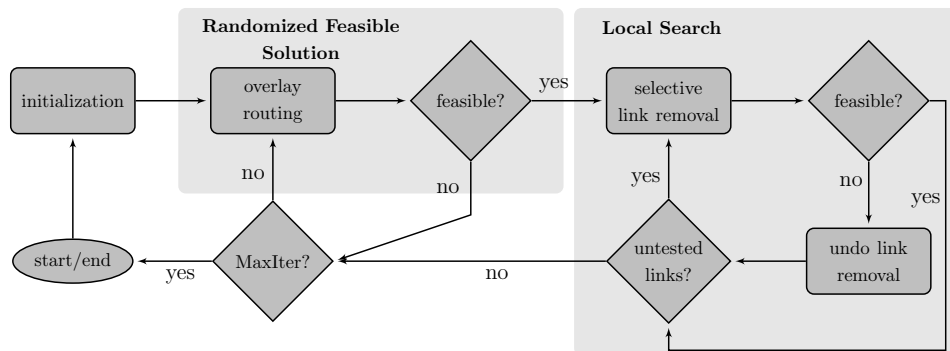


Figure 2: Block-diagram of the GRASP implementation used.

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

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 Oellrich (2008). The goal is to find a path for every lightpath, such that the number of physical link intersections be minimum. It is also desirable that the total cost be as low as possible but as a second 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: $\bar{l}_{ij}, \forall (ij) \in P$ will be used. Prior to start routing lightpaths, all these pseudo-distances are set to 1. According to these new weights, logical links are routed following the minimal distance without repeating physical links among them.

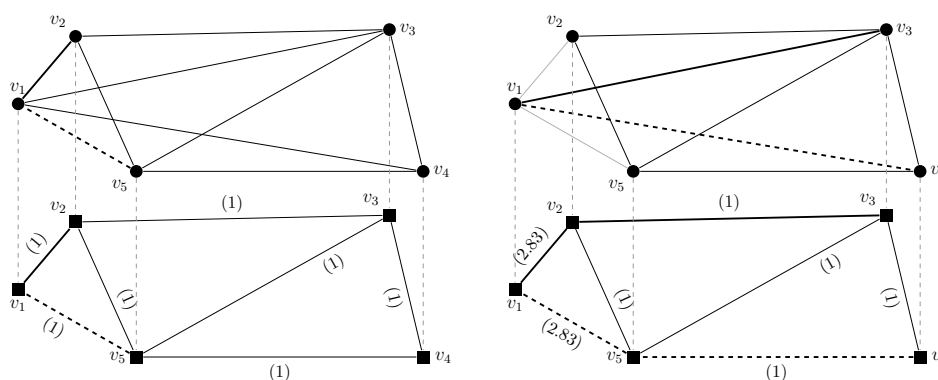


Figure 3: An example of the balanced routing heuristic.

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: $\bar{l}_{ij} = (1 + n_{ij})^p$ 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.

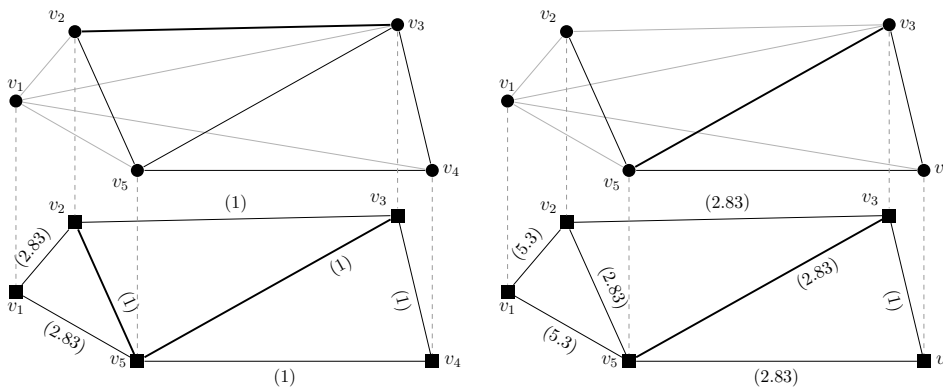


Figure 4: Lightpaths for logical links (23) and (35).

For instance, let us guess our networks are like those sketched in Figure-3 and the links drawn are: (12), (15), (13), (14), (23), (35), and so on. The left half of Figure-3 shows with solid and dashed 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 $\bar{l}_{12} = \bar{l}_{15} = 2^{1.5} \approx 2.83$ for the next window.

Overlay routing (logical over physical). Algorithm 1

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

- 1: Set $\Psi(e) = \emptyset, \forall e \in L$ and $pd : P \rightarrow \mathbb{R}_0^+, pd(e) = 1 \forall e \in P$.
 - 2: **while** $\exists v \in V / \text{not-processed}(v)$ **do**
 - 3: Select randomly $v \in V / \text{not-processed}(v)$;
 - 4: Set $prob(vw) = \frac{1}{d(v,w)}, \forall (vw) \in L / \Psi(vw) = \emptyset$;
 - 5: Normalize $prob$ such that: $\sum_{e \in L} prob(e) = 1$;
 - 6: Create new *control window*;
 - 7: **while** $\exists w \in V / ((vw) \in L \text{ AND } \Psi(vw) = \emptyset)$ **do**
 - 8: Draw such $w \in V$ randomly weighted by $prob(vw)$;
 - 9: Find *shortest-lp*, a *pd-distance* shortest lightpath for (vw) avoiding repeating physical links within this window;
 - 10: **if** (*shortest-lp* = \emptyset) **AND** (there are not unprocessed (vw)) **then**
 - 11: Update $pd(v, w) = pd(w, v) = (1 + \sum_{e \in L} |\Psi(e) \cap \{(vw)\}|)^p$;
 - 12: Create new *control window*;
 - 13: **else**
 - 14: $\Psi(v_e v_f) = \Psi(v_f v_e) = \text{shortest-lp}$;
 - 15: **end if**
 - 16: **end while**
 - 17: **end while**
 - 18: **return** $\Psi : L \rightarrow 2^P$.
-

The next two logical links are (13) and (14). They are routed using the updated values. Their lightpaths are also represented with solid and dashed lines in the right half of Figure-3. 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-4. 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-3 and Figure-4. 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-5 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-5 using grey curves, so it is the remaining capacity in every link after routing each tunnel -two tunnels in the central image-.

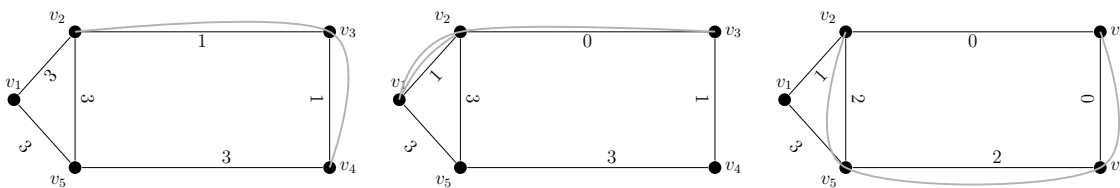


Figure 5: Routes for the tunnels (24), (12), (13), and (23) over a Logical Layer.

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-2. 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, light-paths costs are revealed because we have their lengths -from the configuration for their routes- and there is only one possible capacity.

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 process 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: Resende (2003) and Glover (2006).

5 Application case context and results

We will focus now in the context of the telecommunication company we applied this metaheuristic to, and 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-6). 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 the NAP. More accurate information and details are protected by a Non-Disclosure Agreement (NDA) signed between the telecommunications company and our research institute. The costs and traffic information shown in the rest of this article are only referential.

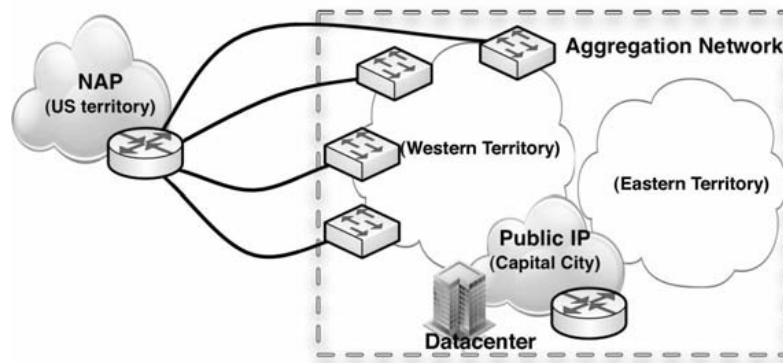


Figure 6: Remarkable aspects of the particular network architecture.

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 fulfill every demand requirements scenario? We helped to answer these questions by identifying representative scenarios and creating their associated data-set 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-.

<i>scenario index</i>	<i>aggregated traffic demand</i>	<i>% local content</i>	<i>merged networks</i>	<i>number of nodes</i>	<i>required lightpaths</i>	<i>total cost</i>
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

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 be placed somewhere between 56 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. 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. 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 that 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 kilometers, these links are the most expensive of the physical network. As it was showed in Figure-6 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 raise 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. 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 previous example and its later analysis justifies the convenience behind the update to existing models this work introduced.

References

- Glover F. (1996) “Tabu search and adaptive memory programming - Advances, applications and challenges”, *Interfaces in Computer Science and Operations Research*, pp.1-75.
- Kerivin H et al., (2005) “Design of survivable networks: A survey”, *Networks* 46(1), pp. 1–21.
- 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.
- Oellrich M. (2008) “Minimum Cost Disjoint Paths under Arc Dependence”, *University of Technology Berlin*.
- Okamura H, Seymour P, (1981) “Multicommodity flows in planar graphs”, *Journal of Combinatorial Theory* 31(1), pp. 75–81.
- 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)*.
- Resende M, Pardalos P. (2006) “*Handbook of Optimization in Telecommunication*”, Springer Science + Business Media.
- Resende M, Riberio C. (2003) “Greedy randomized adaptive search procedures”, *ATT Research*, <http://www2.research.att.com/~mgcr/doc/sgrasp-hmetah.pdf>.
- Stoer M, (1992) “Design of survivable networks”, *Lecture Notes in Mathematics*.