*Les Cahiers du GERAD* ISSN: 0711–2440

# A Method for the Integrated Design of Reliable GMPLS Networks

F. Mobiot, B. Sansò,
A. Girard

G–2003–01

January 2003
Revised: May 2005

Les textes publiés dans la série des rapports de recherche HEC n'engagent que la responsabilité de leurs auteurs. La publication de ces rapports de recherche bénéficie d'une subvention du Fonds québécois de la recherche sur la nature et les technologies.

# A Method for the Integrated Design of Reliable GMPLS Networks

**Florent Mobiot**
**Brunilde Sansò**

*GERAD and École Polytechnique de Montréal*
*C.P. 6079, Succ. Centre-ville*
*Montréal (Québec) Canada H3C 3A7*
florent.mobiot@gerad.ca; brunilde.sanso@polymtl.ca

**André Girard**

*GERAD and*
*INRS-EMT*
*Place Bonaventure, 800, rue de la Gauchetière ouest, Suite 6900*
*Montréal (Québec) Canada H5A 1K6*
andre@inrs-emt.uquebec.ca

January 2003
Revised: May 2005

*Les Cahiers du GERAD*

G–2003–01

## Abstract

The increase of bandwidth demand for new Internet applications suggests mapping directly IP over the WDM layer. Since reliability is such a critical issue in these broadband networks, we propose an integrated design method which addresses the problem of survivability as viewed from the IP/MPLS layers but taking into account the failure mechanisms in the optical layer. This approach is becoming practical because of the emergence of GMPLS as a multilayer control plane that can support the signaling required for coordinating the restoration mechanisms in multiple layers.

The model relies on network calculus to evaluate a QoS metric as actually perceived by end users and computes a preplanned restoration scheme to recover from failures. We discuss the numerical implementation, the convergence and the solutions produced by the algorithm and show that the resulting network can provide the prescribed QoS guarantees for all failure states.

**Key Words:** Survivability, IP, GMPLS.

## Résumé

La demande croissante de largeur de bande pour les applications du nouvel Internet suggère de disposer la couche IP directement au dessus de la couche WDM. La fiabilité étant un souci majeur pour ces réseaux large bande, nous proposons une méthode de synthèse intégrée pour résoudre ce problème tel que vu par les couches IP/MPLS mais en tenant compte des mécanismes de pannes dans la couche optique. Cette approche est rendue possible en raison de l'apparition de GMPLS en tant que plan de contrôle multi-couches qui peut assurer la signalisation requise pour coordonner les mécanismes de restauration sur plusieurs couches.

Le modèle se base sur le "network calculus" pour estimer la métrique de QS telle que perçue par les usagers et il fournit un schéma préétabli de restauration pour les cas de pannes. Nous discutons de l'implémentation numérique, de la convergence et des solutions obtenues par l'algorithme et nous montrons que le réseau issu respecte les contraintes de QS pour tous les états de panne.

**Acknowledgments:** This work was partly supported by NSERC Grant No. STPGP 246159-01 and FCAR Grant No. 00-ER-2654.

# 1 Introduction

The question of whether or not survivability has to be taken into account in the design of optical networks is not an issue anymore, since the failure of a network component may cause a huge amount of lost traffic. For this reason, numerous survivability schemes for protection and/or restoration have been developed for different network layers such as the IP/MPLS [13, 6], ATM, SONET and at the WDM layers [4, 8].

These mechanisms show some very important differences. They protect and restore entities at widely different granularities: packets at the IP layer, VC's or VP's at the ATM layer and optical channels at the WDM layer. They also react to different types of equipment failures. The failure of an optical channel will be directly recognized by the WDM optical cross-connect equipment but will also be felt at the IP layer as a reduction of the capacity of a number of links. In other cases, mechanisms in one layer, e.g., an OXC, are unaware of a failure in another layer, e.g., the failure of an IP router. And finally, the response times of these mechanisms are widely different, ranging from 50 ms for optical restoration to tens of seconds and even minutes for IP rerouting mechanisms.

In practice, users are totally unconcerned by these mechanisms. The users' notion of reliability is more closely tied to the Quality of Service they receive under normal and failure states. In this paper, we take the view that reliability objectives should be defined at the application layer in terms of the QoS delivered to the users. Because failures can occur at various layers, the design of networks must then be an integrated procedure that takes into account *all* layers and their failure mechanisms.

Even if networks could be designed with an integrated view of *all* the reliability mechanisms, there still remains the question of the actual implementation of these networks. One should be careful that the appropriate restoration mechanism is used for a given failure and the other layers are made aware that this procedure is being invoked so that they do not start a restoration procedure of their own. Up until recently, this kind of coordinated action was not possible in multilayer networks. This situation has changed since the IETF introduced *multiprotocol label switching* (MPLS) as a mechanism to provide QoS guarantees and traffic engineering (TE) on the Internet.

This has been extended to Generalized MPLS (GMPLS) which operates both in the IP and optical domains [2]. With GMPLS, a label is not only a simple identifier as in MPLS but it binds directly to a physical entity such as a frequency, a time slot or a wavelength for an optical layer. In other words, GMPLS defines a new *unified* control plane and signaling functions common to optical and electronic networks. Thus, GMPLS enables interoperability between network layers by providing a high level abstraction of the end-to-end connectivity. GMPLS relies on a peer model in which all networks elements are related to the same unified control and signaling plane that builds the bridge to operate the MPLS network with the optical network. It is because of the availability of GMPLS that the integrated design of reliable networks becomes a practical alternative since it makes possible the inter-layer coordination required by the assumptions of integrated design.

GMPLS is the key element that makes it possible to implement networks designed in this way. In current overlay networks (e.g., IP-over-ATM-over-SONET), each layer has its own control plane which may initiate restoration independently of what happens in other layers.

Even though our ultimate objective is to be able to design networks with an arbitrary number of layers, we consider in this paper only networks with the IP and WDM layers. This assumption is made to simplify the presentation and analysis of our results but it is not totally unrealistic either. With the deployment of optical core networks, it is expected that the architecture of the Internet of the future will migrate towards mapping directly IP layer 3 over wavelength-division multiplexing (WDM) layer with a proper control plane, e.g., GMPLS. In any event, the model presented here can easily be extended to multilayer networks if so desired.

The present work is built on the framework proposed in [17] for ATM by adapting it to MPLS networks. The traffic model proposed here defines the packet QoS in terms of the end-to-end delay rather than as cell loss rates. We assume that the queues are served according to *generalized processor sharing* (GPS) service policy and that traffic shaping is used to guarantee deterministic delay bounds for all packets [15, 16]. In that case, it is possible to define a simple effective bandwidth [10] that is uniform over all links of a connection and that depends on the source controllers only.

The other important improvement over [17] come essentially from a more accurate modeling of the QoS constraints and an improved solution procedure. In this way, we are able to meet fully all the QoS constraints rather than in a probabilistic manner as was the case with the model of [17].

The outline of the paper is as follows. The problem formulation is presented in Section 2. Section 3 describes the Lagrangian relaxation used to solve the model followed by some computational results and discussion in Section 4. Section 5 concludes this paper and gives the flavor for further work.

## 2 Problem Formulation

In this work, we consider only the synthesis of the logical network (routing and capacity assignment of the LSP's) for a given set of failure scenarios in the fiber network. We assume that a link-disjoint mapping of the lightpaths on the fiber network is given and that only single fiber link failure may happen (e.g., loss of signal or fiber cut). Finally, we assume that wavelength conversion is provided so that we can ignore the wavelength continuity constraints.

### 2.1 Network Model

Figure 1 shows the network model we consider here which consists in a *fiber network* (physical network) underlying an *optical layer* or *virtual topology* that carries MPLS label

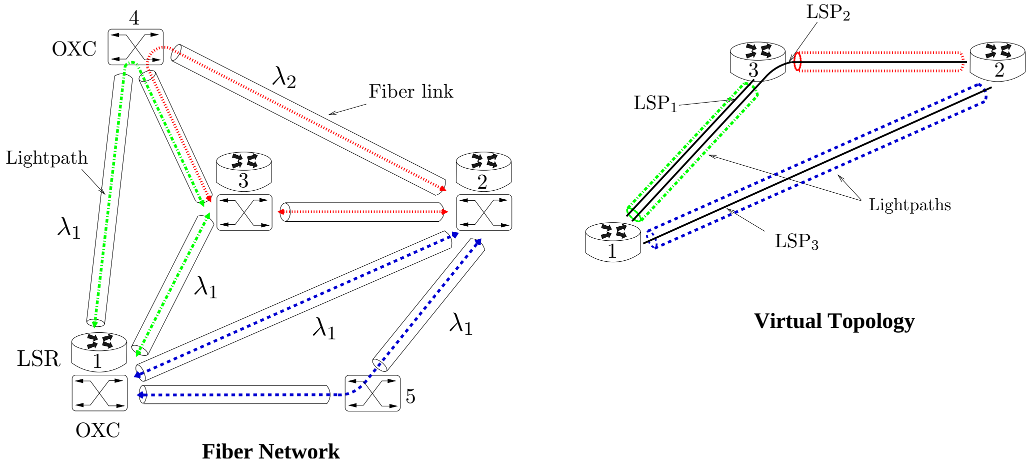


Figure 1: Lightpath routing

switched paths (LSP's) traffic. Here, we take the view that LSP's are semi-permanent big trunks that are set up in a network dimensioning stage and never torn down during a relatively long period. Viewed from the IP layer, these LSP's appear as IP tunnels or direct *logical links* between two IP nodes without any intermediate routers. These logical links, combined with the IP routers, make up the *logical network*. The fiber network is composed of WDM optical cross-connects (OXC's) and fiber links and the virtual topology involves label switched routers (LSR's) and lightpaths which are routed into the fiber layer. This figure displays a diverse lightpath routing where fiber link (3-4) is used to route a fraction of the lightpaths (1-3) and (2-3). At a higher level, LSP$_1$ is routed through lightpath (1-3) and LSP$_2$ through lightpaths (1-3) and (3-2). Thus, the failure of fiber (3-4) means a decrease of the total bandwidth available for both LSP$_1$ and LSP$_2$ and this actually affects the traffic distribution in the logical network. For this reason, it is important to look at an integrated synthesis where both IP/MPLS and optical layers are designed at the same time.

Restoration schemes co-exist in both layers. At this first stage of our work, we consider only the restoration of LSP's by rerouting them in the IP layer and don't look at restoration into the optical layer. We assume that some services (e.g., mail or ftp) can tolerate a relatively long restoration time so that we do not need to restore them via the fast mechanisms of the optical layer but can use traffic rerouting at the IP layer. The only assumption is that lightpaths connecting two LSR's are fiber-disjoint in the optical layer. This guarantees that the network remains connected in the event of a fiber link failure. Since we do not consider a dynamic reconfiguration (restoration) of the lightpaths after a failure, the question is then to compute the routes and bandwidths of the LSP's that have to be provisioned in the IP layer to meet *all* the QoS requirements in *all* the failure states. Because the traffic will be restored in the IP layer, these LSP's can be routed over

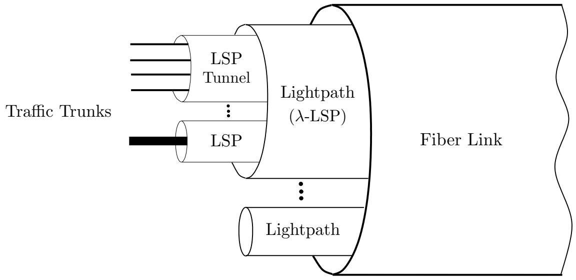


Figure 2: LSP Stacking

unprotected optical facilities thus reducing the amount of back-up facilities required in the fiber layer. This is essentially due to the bandwidth granularity which is coarser in the fiber layer than in the IP layer.

This approach is also attractive with respect to the restoration times it provides to the network. In the event of a fiber link failure detected at the optical layer, the ingress and egress LSR's are notified by GMPLS signaling messages to initiate the restoration process. Then, using CR-LDP [1] or RSVP-TE [3] with GMPLS, the ingress and egress LSR's can setup one or more backup LSP's to support the defected traffic (if the bandwidth is available) . The overall restoration time for this process is thus equal to fault detection times at the optical layer plus signaling, backup LSP's setup and traffic switching times. Since there is no routing algorithms invoked, the restoration times can be approximated to a few milliseconds, a little more the times required by optical layer restoration but certainly much less than an isolated IP/MPLS layer restoration.

The work presented here is an algorithm to compute the routing and capacity assignment of LSP's required in the IP network that meets all the reliability requirements.

### 2.2 Notation

The undirected graph $G(\mathcal{V}, \mathcal{E})$ represents the virtual topology where $\mathcal{V}$ is the set of LSR's and $\mathcal{E}$ the set of lightpaths. Likewise, $G^p(\mathcal{V}^p, \mathcal{E}^p)$ represents the fiber network where $\mathcal{V}^p$ represents the set of OXC's and $\mathcal{E}^p$ the set of optical fibers. We assume that there is no failure of cross-connect equipment and a state $k$ of the network is a binary vector which describes the condition (operational or failed) of all the fiber links.

Traffic demand in the logical network is represented by *traffic trunks* of various sizes that arrive at random instants and have random duration. A traffic trunk $i$ defines a path between an origin-destination (OD) pair followed by a particular class of traffic or forward equivalence class (FEC). In practice, we would have different types of LSP's corresponding to the types of traffic trunks, e.g., traffic trunks carrying voice or video sessions could be routed on voice- or video-dedicated LSP's. According to the LSP hierarchy used for the model (see Figure 2), traffic trunks correspond to dynamic LSP's which get through the semi-permanent LSP's (logical links or LSP tunnels). As we will see later, we can define an effective bandwidth for a traffic trunk and this bandwidth is expressed as an integer multiple of some basic bandwidth unit (BBU). We denote $\mathcal{I}$ the set of traffic trunks and $\mathcal{K}$ the set of network states. We want to compute the following variables:

$f_i(k)$ the traffic flow of traffic trunk $i$ for the network being in state $k$. We use $\mathbf{f_e}$ to denote the vector of traffic of all traffic trunks on link $e$;

$N_e$ the server rate i.e., the number of BBU's on link $e$. In general, a LSP requires more than one BBU.

The known parameters of the model are

$K_e$ be the cost of adding one unit of capacity to link $e$;

$N_e^0$ be the capacity already in place on link $e$;

$B_{e,i}$ be the blocking probability function for traffic from traffic trunk $i$ on link $e$;

$L_i(k)$ be the grade of service (GoS) of the traffic trunk $i$ in state $k$, i.e., the probability for that traffic trunk to be blocked. We refer to $\overline{L}_i(k)$ as the maximum GoS allowed.

### 2.3 Formulation

The general formulation of the IP design problem is:

$$\begin{aligned}
\min_{f,N} Z &= \sum_{e\in\mathcal{E}} K_e N_e && (1)\\
f_i(k) &\in \Omega(k) \qquad \forall\, i \in \mathcal{I}, k \in \mathcal{K} && (2)\\
L_i(k) &\le \overline{L}_i(k) \qquad \forall\, i \in \mathcal{I}, k \in \mathcal{K} && (3)\\
N_e^0 &\le N_e \qquad \forall\, e \in \mathcal{E} && (4)\\
N_e & \quad \text{integer,} && (5)
\end{aligned}$$

where $\Omega(k)$ is the set of traffic flows allowed by the traffic engineering in order to honour the QoS constraints for packet delivery in network state $k$.

In this model, we want to minimize the cost of the logical network (1) given that 1) traffic is feasible for the traffic engineering (routing) (2), 2) the GoS for all traffic trunks don't exceed the prescribed bounds (3), 3) there might be capacities already installed on some links (4) and 4) the capacities are integer variables (5). The GoS constraints (3) refer to the user prespecified QoS levels for all the considered states (normal and failure states). These traffic requirements are generally expressed as delay constraints at the packet level, but at the traffic trunk level this is more relevant when related to the GoS or *end-to-end blocking probability.*

This problem is difficult to solve because of the integer constraints (5) and the nonlinear GoS constraints (3). The problem is thus a nonlinear problem with integer constraints which is likely to be very hard to solve. Therefore, we consider some approximations to relax the original problem (1-5).

The first approximation will be to relax the lower bounds on installed capacities (4) and the integer constraints (5) since we are facing huge capacities in broadband networks.

## 2.4 Effective Bandwidth

The feasibility constraints (2) are simplified in a number of ways. First, the packet QoS constraints are replaced by an effective bandwidth, just as in the case of the design of ATM networks [17]. In the present work, a difference is that QoS constraints are not necessarily based on buffer overflow and packets or cell loss but rather on the delay experienced by packets as common in IP/MPLS networks.

Our work is based on network calculus which gives bounds on the maximum end-to-end delay experienced by traffic sources constrained by leaky-bucket [5]. It is generally felt that FCFS (First Come First Served) is not appropriate to achieve QoS where several traffic streams with different requirements are involved. Generalized Processor Sharing (GPS) is a widely studied model [15, 16] which can be used in the network calculus framework to yield the bounds.

The metric available is the maximum end-to-end delay $D_i^*$ faced by a packet in the traffic trunk $i$ and the measure of interest is the call blocking probability of the traffic trunk $i$.

Let $\rho_i$ be the average rate of arrival and $\sigma_i$ the maximum burst size of traffic trunk $i$, $\overline{D}_i$ the maximum end-to-end delay allowed for this traffic trunk, then according to [10], we can define an effective bandwidth required by $i$ so that the traffic trunk will satisfy the maximum delay constraint. An important point is that this effective bandwidth is the same over all links in the path. This is simpler than the case for ATM, where the effective bandwidth is link-dependent.

Let $w_i$ be the effective bandwidth required by traffic trunk $i$ and $\mathbf{W_e}$ be the vector of service rates of all traffic trunks passing through link $e$. The effective bandwidth of traffic trunk $i$ is:

$$w_i = \max\left\{\rho_i, \frac{\sigma_i}{\overline{D}_i}\right\}. \tag{6}$$

Based on (6) and [10], we can state that a traffic trunk $i$ is blocked if : $D_i^* > \overline{D}_i \iff g_i < \ w_i\,,\ \ g_i$ being the minimum rate allocated to $i$ by the servers in the path followed by $i$.

Given a traffic trunk $i$, computing the GoS returns to check whether the minimum available bandwidth on the path exceeds the effective bandwidth. Using this condition, we can see that the constraints (2) reduce to the calculation of carried traffic in a circuit-switched problem.

### 2.5 Flow Formulation

The QoS constraints (2) for a circuit-switched network can be solved by a fixed-point approach [9]. This procedure is time consuming for the size of problems we intend to deal with. Consequently, they are replaced by an uncapacitated multicommodity flow model where we assume 1) that all connection requests are accepted, 2) the flows representing the connections on the links are conserved at the network nodes and 3) the end-to-end loss probability on a path is given by the sum of the loss probabilities on the links of that path. Under these assumptions, the variables $f$ become a multicommodity flow and the set $\Omega(k)$ is a shorthand for the conservation equations for this flow at all the nodes in the network. In that case, the GoS of traffic trunk $i$ and network state $k$ is computed by

$$L_i(k) \simeq \sum_{e\in\mathcal{E}} B_{e,i}(k) \tag{7}$$

where $B_{e,i}(k)$ is the blocking probability function for the flow from traffic trunk $i$ on link $e$ in the network state $k$. $B_{e,i}$ depends on $\mathbf{f_e}$, $\mathbf{W_e}$ and $N_e$.

These approximations would appear quite strong but previous work [11] has shown that they were sufficiently accurate in the case of circuit-switched networks for the purpose of *network design*. Extending this technique to other networks raises two questions: first, whether the solution algorithm is still fast enough to be usable for large networks and if so, whether the solution is sufficiently accurate to be used with a real-time algorithm. This paper is an answer to the first question, the second one being left for further work.

The modified formulation of the integrated synthesis is then (Primal Problem):

$$\begin{aligned} \min_{f,N}\ Z &= \sum_{e\in\mathcal{E}} K_e N_e \\ \sum_{e\in\mathcal{E}} B_{e,i}(k) &\le \overline{L}_i(k) \qquad \forall\, i\in\mathcal{I}, k\in\mathcal{K}. \end{aligned} \tag{8}$$

where $\mathbf{f}$ is a multicommodity vector defined by Eq. (2). These equations can be written for a given state as

$$\sum_{l(k)} f_i^l(k) = A^i \tag{9}$$

where we have defined the index $l(k)$ to represent all the paths that are used by traffic of type $i$ in state $k$, $f_i^l(k)$ the amount of flow of type $i$ that is carried on path no $l$ in state $k$ and $A^i$ as the average connections demand for traffic of type $i$.

## 3 Solution Technique

We propose a Lagrangian relaxation as a dual solution method for the model (1), (2) and (8), so that the problem can be decomposed and be more tractable.

### 3.1 Lagrangian Relaxation

Let $\theta_i(k)$ be the multipliers associated to the QoS constraints (8). The Lagrangian function of the primal problem is given by

$$\begin{aligned}\mathcal{L}(\mathbf{f},\mathbf{N},\theta) &= \sum_{e\in\mathcal{E}} K_e N_e + \\ &\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}} \theta_i(k)\left(\sum_{e\in\mathcal{E}} B_{e,i}(k) - \overline{L}_i(k)\right).\end{aligned} \tag{10}$$

We want our method to be a service-oriented design. So we define $I_{e,i}(k)$ as the arc-path incidence matrix which takes the value 1 when arc $e$ belongs to the path of traffic trunk $i$ and 0 otherwise. Let $X_{e,i}(k) = f_i(k)I_{e,i}(k)$ be the total traffic from traffic trunk $i$ which is offered to link $e$ in state $k$. We set $\theta_i(k) = w_i\, A^i\, \alpha_i(k)$, then $\alpha_i(k)$ may be perceived as the cost of loosing flow of traffic trunk $i$ in network state $k$. Hence, the Lagrangian function becomes

$$\begin{aligned}\mathcal{L}(\mathbf{f},\mathbf{N},\alpha) &= \sum_{e\in\mathcal{E}} K_e N_e\ + \\ &\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}}\sum_{e\in\mathcal{E}} \alpha_i(k)\left\{w_i\left(X_{e,i}(k)\right)B_{e,i}(k)\right\}\ - \\ &\sum_{k\in\mathcal{K}}\sum_{i\in\mathcal{I}} \alpha_i(k) w_i A^i \overline{L}_i(k).\end{aligned} \tag{11}$$

The dual function is

$$\phi(\alpha) = \min_{\mathbf{f},\mathbf{N}} \mathcal{L}(\mathbf{f},\mathbf{N},\alpha)\Big|\ f_i(k) \in \Omega(k), \tag{12}$$

and the dual problem is then

$$\max_{\alpha\geq 0} \phi(\alpha). \tag{13}$$

### 3.2 Solving the subproblem

The minimization problem (12) yields two interrelated subproblems: the call routing and the dimensioning subproblems. This can be addressed with an iterative *capacity and flow assignment* (CFA) algorithm [7] in which the call routing problem is solved with fixed capacity variables and the dimensioning problem with fixed flow variables.

**3.2.1 Routing Sub-Problem** The call routing problem where we have fixed the capacity variables to their current values and with the constraints (2), (8) is a constrained nonlinear multicommodity flow problem. Several algorithms exist in the literature deal with this problem. We choose the Frank-Wolfe method also called *flow deviation method* [7] in which we try to find a direction of descent and a step size to move in that direction.

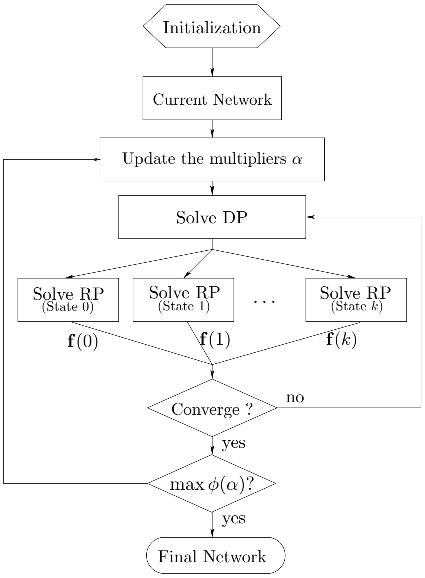


Figure 3: Algorithm

This method is interesting because it also provides candidate routes which can be used to operate the network as explained in [11].

Denote by $Y_{e,i}(k)$ the direction of descent. Then for each network state $k$, the call routing problem is

$$\min_{Y} \sum_{e\in\mathcal{E}} \sum_{i\in\mathcal{I}} \left[ w_i B_{e,i} + \sum_{j\in\mathcal{I}} w_j X_{e,j} \frac{\partial B_{e,j}}{\partial X_{e,i}} \right] Y_{e,i}. \tag{14}$$

subject to the flow constraints (2).

**3.2.2 Dimensioning Sub-Problem** When we fix the traffic carried by the LSP's $(f_i(k))$, the minimization of the resulting Lagrange function becomes separable by link. The first-order conditions are given for each link $e$ by

$$\begin{aligned} -K_e &= \\ & \sum_{k\in\mathcal{K}} \sum_{i\in\mathcal{I}} \alpha_i(k) w_i \left( f_i(k) \;\; I_{e,i}(k) \right) \left. \frac{\partial B_{e,i}(k)}{\partial N_e} \right|_{N_e = N_e(k)} . \end{aligned} \tag{15}$$

## 3.3 Dual Problem

To solve the dual problem, we propose to use a sub-gradient optimization method [14] where the Lagrangian multipliers $\alpha_i(k)$ for every traffic trunk and each network state

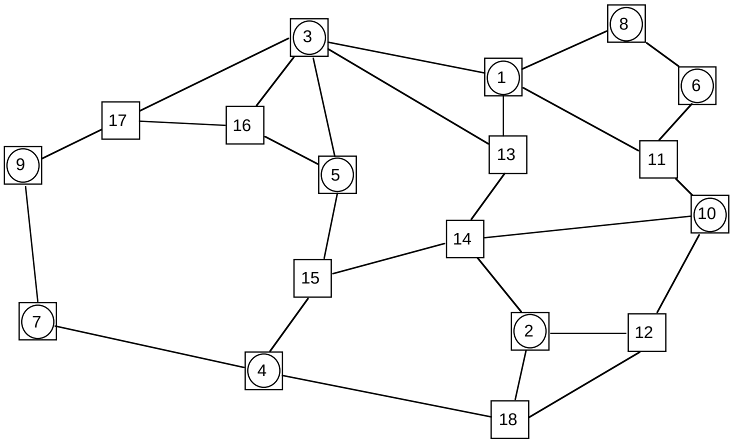


Figure 4: Sprint Network

(normal and failure states) are updated by

$$\alpha_i(k) \leftarrow \max\left\{0,\ \left(1+\xi(L_i(k)-\overline{L}_i(k))\right)\alpha_i(k)\right\}, \qquad (16)$$

where $\xi$ is a scaling factor for the constraint violation. The sub-gradient algorithm represented by the flow chart of Figure 3 works as follows. We begin with an initial solution for the flow variables $\mathbf{f}$. A one-hop routing scheme is a good starting point. At each iteration, we set or update the multipliers using (16) and then solve iteratively the dimensioning subproblem (DP) and the call routing subproblem (RP) for each network state (normal and failure states) until convergence. In order to speed up the algorithm, we only make one iteration of the CFA at each dual iteration. We check whether the overall dual problem converges, otherwise we iterate the whole process. The convergence of the dual problem is reached when the QoS constraints are met within 5% (in relative terms) in every state and the increase in the dual function becomes sufficiently small.

# 4 Results

In this section, we provide results for three test networks: a 3-node network (see Figure 1); a 19-node network of 37 physical nodes, 19 routers and 65 transmission links; and a real network composed of 10 routers, 18 physical nodes and 27 transmission links previously used by [12] (see Figure 4). The unit trunk costs structure $K_e$ is drawn from the data provided in [12] and the traffic demands for all traffic trunks were randomly generated.

The mean rate ($\rho$), maximum burst size ($\sigma$) and the GoS constraints in terms of maximum delay ($\overline{D}$) are given in Table 1 for 3 classes of service. The call blocking probabilities requirements for all connection requests are selected to be 1% in the normal state and 20% in each failure state. The computations were performed on a 2 GHz Pentium IV processor with the Linux Operating System.

Table 1: Classes of Service Parameters

| Class of service | 1 | 2 | 3 |
|---|---|---|---|
| $\rho$ (64 kb/s) | 1 | 4 | 12 |
| $\sigma$ (B) | $1.5 \times 10^4$ | $1.5 \times 10^4$ | $1.5 \times 10^4$ |
| $\overline{D}$ (s) | 0.1 | 0.1 | 0.1 |

We are interested in two kinds of results. The first set has to do with the behavior of the algorithm itself and in particular its convergence rate on relatively large networks. The second set is used to examine on the small network some properties of the solutions in terms of feasibility and the effect of the packet QoS requirement.

## 4.1 Convergence of the Algorithm

The speed of convergence of the algorithm is tied to two main factors: the size of the problem and the choice of initial multipliers. The choice of the initial multipliers determines the initial solution that starts the iterating process and the size of the physical network governs the number of states the algorithm has to go through. For example, the 19-node has about $37 \times 3 \times (19 \times (19 - 1))/2 = 18981$ QoS constraints with $65 \times 6327 = 1233765$ arc-flow variables.

The method performs well regarding the convergence of the algorithm for size of problems solved as illustrated by the graphs of Figures 5, 6 and 7. Table 2 shows the computation time and the total number of iterations performed for the different sets of tests.

Figures 6 show an example of slow convergence due to a poor choice of initial solutions (see sets 2 and 3). The same thing occurs with the set 2 of 19-node test network at Table 2. We have found that the initial value of the multipliers should be chosen such that the two terms of the Lagrange function (10) should be approximately equal. Otherwise, ill-conditioning often causes slow convergence and oscillations of the algorithm.

Altogether, these results indicate that the subgradient method can converge sufficiently fast to be able to compute networks of intermediate size. The next question is whether the solutions that are computed are adequate, especially in terms of the GoS constraints in the failed states.

## 4.2 Solutions

We now illustrate the structure of the solutions that are produced by the algorithm. Most of the results are summarized in Table 2. The table shows the network and dual costs, the duality gap, given by $(Z - \phi)/\phi$, the computation time (in min.) and the number of dual iterations (# it.) for each test network and for some initial values of the Lagrange multipliers ($\alpha_0$ for the normal state, $\alpha_k$ for all failure states).

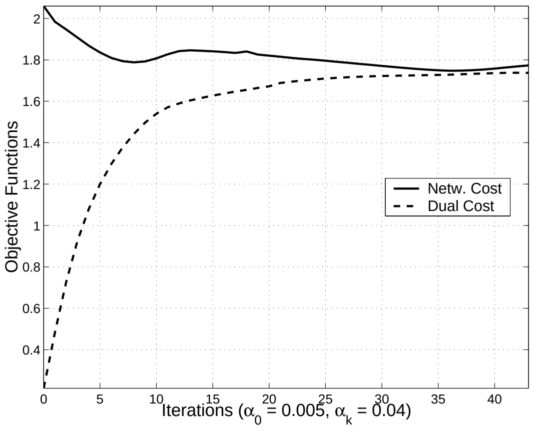


Figure 5: Convergence for the 3-node Network

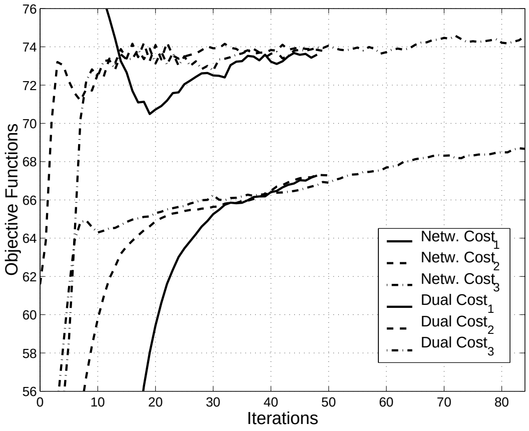


Figure 6: Convergence for the Sprint Network

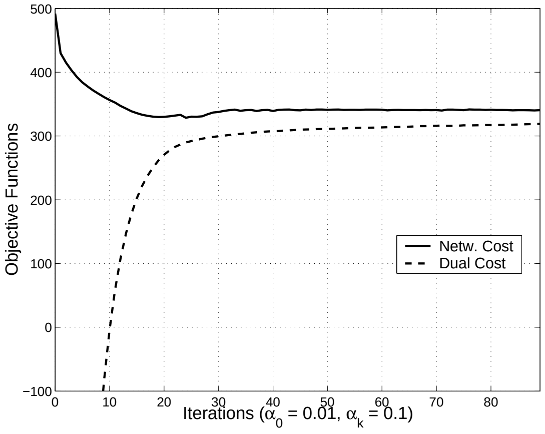


Figure 7: Convergence for the 19-node Network

Table 2: Costs and performances

| **Test Sets** | | **Initialization** | | **Network** | **Dual** | **Gap** | **CPU Time** | **#** |
|---|---|---|---|---|---|---|---|---|
| | | $\alpha_0$ | $\alpha_k$ | **Cost** | **Cost** | **(%)** | **(min)** | **It.** |
| 3-node | 1 | 0.005 | 0.04 | 1.77 | 1.74 | 1.7 | 0.07 | 43 |
| Sprint | 1* | 0.01 | 0.1 | 73.6 | 67.3 | 9.4 | 5.4 | 48 |
| Network | 2 | 0.01 | 0.01 | 73.9 | 67.3 | 9.8 | 5.4 | 50 |
| 10-node | 3 | 0.01 | 0.001 | 74.6 | 68.7 | 8.6 | 8.9 | 84 |
| 19-node | 1* | 0.01 | 0.1 | 340.5 | 319.0 | 6.7 | 109 | 89 |
| | 2 | 0.01 | 0.07 | 339.9 | 323.3 | 5.1 | 171 | 138 |

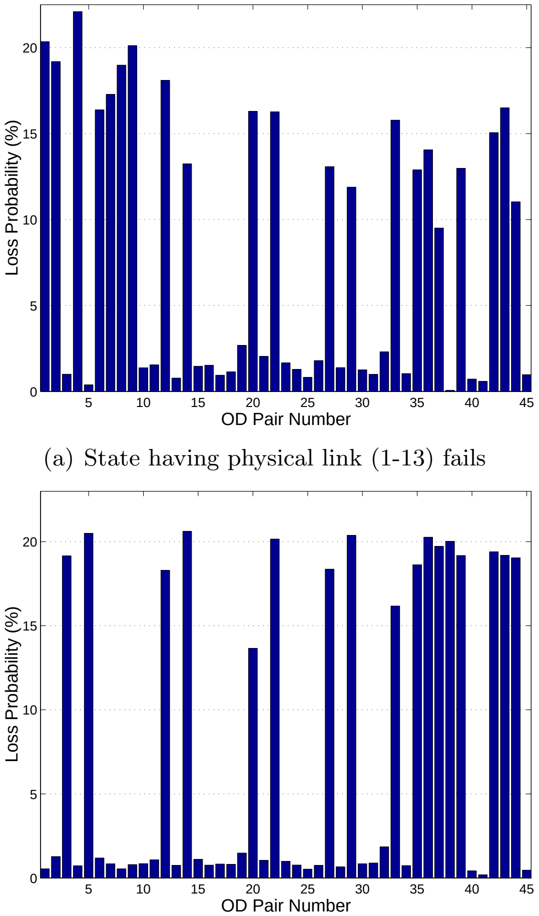
The different sets of results show that the method works well since we obtain solutions with duality gap under 10% and even about 5% for 19-node network and 1.7% for the 3-node network. These low duality gaps reveal that the local solutions produced are close to a local optimum.

The algorithm generates solutions that are feasible within a given small tolerance. For this purpose, Figures 8 and 9 display the actual values of the GoS experienced in some states for the test sets identified by a star in Table 2. Those graphs are presented here because they actually represent the states in which the GoS are maximum. We can see that the QoS constraints are met in all network states.

We can see on the graph that only a small number of GoS constraints are tight in a given failed state. This might suggest that the flows could be rearranged to take up the slack in the other constraints and thus allow a reduction of the overall network cost. In fact, this is not possible because when we consider the set of *all* failed states, all GoS constraints are tight in one state or the other. This is shown on Figure 10 which presents the GoS for the 3 OD pairs of the logical network under all failure states. We can see that there is very little slack available overall even though there can be some for any single failure state.

### 4.3 Effect of GoS

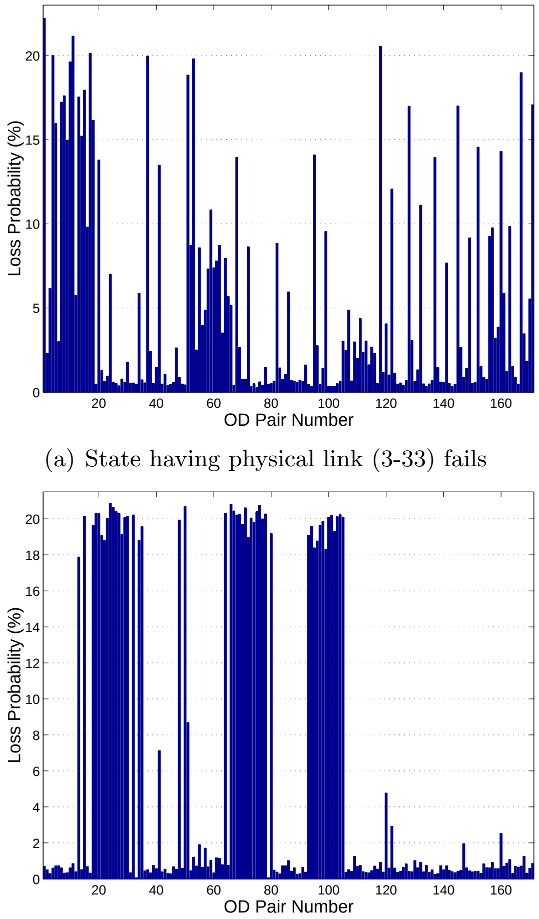
We can use the algorithm to investigate some features of reliable design. We present here some preliminary results for the 3-node network where we have examined the effect on the overall network cost of decreasing the packet-level delay constraint for each traffic class. The results are shown in Tables 3, 4 and 5. We see that there is a significant increase in the network cost.



(a) State having physical link (1-13) fails

(b) State having physical link (6-11) fails

Figure 8: GoS for Sprint Network



(a) State having physical link (3-33) fails

(b) State having physical link (7-34) fails

Figure 9: GoS for 19-node Network

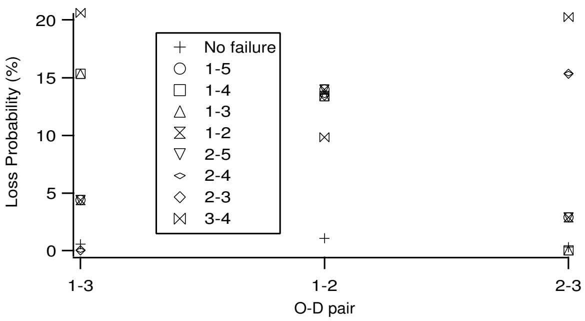


Figure 10: GoS for all states, 3-node network

Table 3: Varying $\overline{D}$ for class of service with $\rho = 1$

| $\overline{D}$<br>**(s)** | $w$<br>**(64 kb/s)** | **Network**<br>**Cost** | **Dual**<br>**Cost** | **Gap**<br>(%) |
|---|---|---|---|---|
| $\infty$ | 1 | 1.09 | 1.07 | 1.9 |
| 1 | 1.875 | 1.12 | 1.11 | 0.9 |
| 0.5 | 3.75 | 1.20 | 1.18 | 1.7 |
| 0.25 | 7.5 | 1.34 | 1.32 | 1.5 |

Table 4: Varying $\overline{D}$ for class of service with $\rho = 4$

| $\overline{D}$<br>**(s)** | $w$<br>**(64 kb/s)** | **Network**<br>**Cost** | **Dual**<br>**Cost** | **Gap**<br>(%) |
|---|---|---|---|---|
| $\infty$ | 4 | 1.21 | 1.19 | 1.7 |
| 0.1875 | 10 | 1.44 | 1.42 | 1.4 |
| 0.125 | 15 | 1.63 | 1.61 | 1.2 |
| 0.075 | 25 | 1.94 | 1.90 | 2.1 |

Table 5: Varying $\overline{D}$ for class of service with $\rho = 12$

| $\overline{D}$<br>**(s)** | $w$<br>**(64 kb/s)** | **Network**<br>**Cost** | **Dual**<br>**Cost** | **Gap**<br>(%) |
|---|---|---|---|---|
| $\infty$ | 12 | 1.52 | 1.49 | 2.0 |
| 0.125 | 15 | 1.63 | 1.61 | 1.2 |
| 0.075 | 25 | 1.94 | 1.90 | 2.1 |
| 0.050 | 37.5 | 2.25 | 2.18 | 3.2 |

## 5 Conclusion

This integrated design method explores a given failure scenario and proposes a robust dimensioning based on the physical and the logical network topologies. The approach takes into account the impact of failures from the end user perspective with the introduction of effective bandwidth. In case of failure, the solution offers a preplanned list of backup paths that can be used to reroute the traffic.

We have been able to test the model on big backbone networks with about 40 physical nodes and the method performs well since it provides quasi-optimal solutions which respect the different QoS constraints.

This method is a good tool to investigate different restoration strategies and evaluate their performances on network costs. We are going further in this work in order to compare this method to others with respect to solution costs and computation times.

## References

[1] P. Ashwood-Smith and L. Berger. Generalized Multi-Protocol Label Switching (GMPLS) signaling Constraint-based Route d Label Distribution Protocol (CR-LDP) extensions. RFC 3472, January 2003. Proposed standard.

[2] A. Banerjee, J. Drake, J.P. Lang, B. Turner, K. Kompella, and Y. Rekhter. Generalized multiprotocol label switching: an overview of routing and management enhancements. *IEEE Communications Magazine*, 39(1):144–150, January 2001.

[3] L. Berger. Generalized Multi-Protocol Label Switching (GMPLS) signaling Resource ReserVation Protocol-Traffic Engineering (RSVP-TE) extensions. RFC 3473, January 2003. Proposed standard.

[4] O. Crochat and J.-Y. Le Boudec. Design protection for WDM optical networks. *IEEE Journal on Selected Areas in Communications*, 16(7):1158–1165, September 1998. Special Issue on High-Capacity Optical Transport Networks.

[5] R.L. Cruz. A calculus for network delay. II. Network analysis. *IEEE Transactions on Information Theory*, 37(1):132–141, January 1991.

[6] R. Doverspike and J. Yates. Challenges for MPLS in optical network restoration. *IEEE Communications Magazine*, 39(2):89–96, February 2001.

[7] L. Fratta, M. Gerla, and L. Kleinrock. The flow deviation method : An approach to store-and-forward communication network design. *Networks*, 3:97–133, 1973.

[8] O. Gerstel and R. Ramaswami. Optical layer survivability: a services perspective. *IEEE Communications Magazine*, 38(3):104–113, March 2000.

[9] A. Girard. *Routing and Dimensioning in Circuit-Switched Networks*. Addison-Wesley, 1990.

[10] A. Girard and C. Rosenberg. Optimal rate allocation for maximum-delay GPS networks. In *COMCON8. 8th International Conference on Advances in Communications and Control: Telecommunications/Signal Processing*, June 2001.

[11] A. Girard and B. Sansò. Multicommodity flow models, failure propagation and reliable network design. *IEEE/ACM Transactions on Networking*, 6(1):82–93, February 1998.

[12] D. Medhi. A unified approach to network survivability for teletraffic networks: models, algorithms and analysis. *IEEE Transactions on Communications*, 42(2):534–548, February 1994.

[13] C. Metz. IP protection and restoration. *IEEE Internet Computing*, 4(2):97–102, March 2000.

[14] M. Minoux. *Mathematical Programming: Theory and Algorithms.* John Wiley, 1986.

[15] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the single-node case. *IEEE/ACM Transactions on Networking*, 1(3):344–357, June 1993.

[16] A.K. Parekh and R.G. Gallager. A generalized processor sharing approach to flow control in integrated services networks: the multiple node case. *IEEE/ACM Transactions on Networking*, 2(2):137–150, April 1994.

[17] B. Sansò, A. Girard, and F. Mobiot. Integrating reliability and quality of service in networks with switched virtual circuits. *Computers and Operations Research*, in press, 2003.