Protection Interoperability for WDM Optical Networks *

O. Crochat J.-Y. Le Boudec† O. Gerstel

ICA
EPFL - DI
CH - 1015 Lausanne
Tel: +41 21 693 6601
Fax: +41 21 693 6610
olivier.crochat@epfl.ch *leboudec@epfl.ch*

Tellabs Operations
15 Skyline Drive
USA - Hawthorne, NY 10532
Tel: +01 914 593 4393
Fax: +01 914 593 4350
ori@ieee.org

Abstract

The failure of a single optical link or node in a Wavelength Division Multiplexing (WDM) network may cause the simultaneous failure of several optical channels. In some cases, this simultaneity may make it impossible for the higher level (SONET or IP) to restore service. This occurs when the higher level is not aware of the internal details of network design at the WDM level. We call this phenomenon "failure propagation". We analyze three types of failure propagation, called "bottleneck", "connectivity" and "multiple groups". Then we present a solution based on (1) the definition of appropriate requirements at network design and (2) a WDM channel placement algorithm (Protection Interoperability for WDM, PIW). Our method does not require the higher level to be aware of WDM internals, but still avoids the three types of failure propagation mentioned above. We finally show the result on various network examples.

Keywords: WDM, optical network, interoperability, protection, routing, taboo search.

1 Introduction

Wavelength Division Multiplexing (WDM) is about to play a major role in the expansion of photonic networks. One of the main reasons is that WDM has the advantage of not forcing the end-users to run at the aggregate data rate, and does not require any synchronization between channels. It is also the only multiplexing technique which allows the full use of the low-attenuation bandwidth regions of an optical fiber [1, 2, 3, 4].

The existence of many independent data channels over the fiber infrastructure could lead to problems in case of failure, as the amount of bandwidth lost by a resource failure is now much larger than what would have been lost in a traditional network. This problem can be alleviated to a large extend by building survivable WDM networks, namely in which a node or link failure does not cause the interruption of any communication [5, 6]. One can see an example of a survivable WDM network using protection by ring in the COBNET project [7]. In the case of WDM, reconfiguration of complete paths at the optical level is sometimes possible, but not always; see [7] for a discussion of which restoration is possible at the WDM level. In deployed networks today, however, most WDM installations rely on protection at the SDH and SONET layers to support failures of the optical layer. This is the case for many technical and non-technical reasons; please refer, for example, to [8] for a real deployment example.

We thus assume in this paper that failures causing the disruption of optical channels may still occur. We further assume that the *higher level networks* using the WDM network (such as SONET, ATM or IP) implement *intra-level*[1] protection and restoration strategies in order to survive failures of the optical channels. Traditional intra-level solutions are based on reserving spare capacity to re-route blocked higher level links; see for example SONET self-healing

*This work is supported in part by the ACTS project COBNET and the Swiss government (OFES). It represents the view of the authors only

†Corresponding author

[1]Intra-level refers here to procedures that rely only on the representation of the network visible at one single level, without the details of lower levels.

rings [9, 10] (where SONET is the higher level) or meshed networks [11]. Other solutions find multiple disjoint paths, in the higher level network [12], assuming that there would be no simultaneous failures of disjoint paths.

With this in mind, we start by pointing out that the failure of a single optical resource may generate the simultaneous failure of several optical channels. In some cases, this may make the restoration of service by the higher level network impossible, even if the intra-level solutions mentioned above are implemented, for example because the higher level network is no longer connected. To illustrate this, consider the following example. An IP network, made of IP routers, is built on top of a WDM infrastructure. The IP network requests connectivity from the WDM network in order to build adjacencies between routers; at the IP level, two adjacent routers are seen as being connected by a direct link (we call it a higher level link). Next, the IP network builds its own routing tables by means of a routing algorithm; in case of failures, the IP routing algorithm will try and find alternate paths around the failed area. However, the higher level links are mapped onto concatenations of lower level (physical) links in the WDM network. A single resource failure at the WDM network may cause multiple, simultaneous higher level links failures; if care is not taken, it may happen that the IP network after the failure is partitioned into several islands, and restoration of connectivity by the IP routing protocol becomes impossible (see the second example in Section 2 for a more details on this example). We call this phenomenon *failure propagation*, as in [7, 13].

The risk of failure propagation exists whenever the links of a higher level network are mapped onto multihop paths of a lower level network. There are two types of methods for avoiding failure propagation.

(1)*Make failures invisible to higher level links.* This is implemented by having the lower level network remap higher level links onto new lower level paths after a failure. This assumes that the lower level has its own protection mechanism. In the case of a lower level network based on WDM or SDH, a common strategy is to duplicate resources in order to achieve resilience [14, 10]. Automatic switchover procedures are then activated upon failure detection, making the failures invisible to the higher level. Alternatively, higher level links can be mapped statically to several disjoint, parallel lower level paths using a disjoint path finding algorithm; a small piece of software in the lower level network is then able to detect whether a path is operational and, if required, to switch from one failed path to an operational one [15, 16].

(2) *Joint network design or routing at both the higher and lower level networks.* Here, the design of higher level end-to-end paths takes into account the topology of the lower level. Consider for example a SONET network built on top of an optical network.The higher level links are the links between SONET active resources (add-drop multiplexers); they are mapped to optical paths; the optical paths may span multiple physical links. The SONET network is used to establish end-to-end SONET paths; a typical organization is to duplicate such paths, in order to offer failure resilience to the ultimate end-user (the user of the SONET network). A joint network design would place the end-to-end SONET paths on higher level links *and* higher level links onto optical paths such that end-to-end SONET paths use disjoint sets of resources. Such an approach works well because here the lower level network has little combinatorial complexity. If it would use optical cross-connects, then the joint design becomes more involved. A second example is IP over WDM cross-connects; here, the method of joint design would require that the IP routing protocol be aware of optical cross-connects and of their connectivity restrictions (such as limited wavelength conversion). The benefit would be to avoid failure propagation So far, no such protocol is known to us. A third example is IP over ATM; here, joint routing algorithms are being developed, which are aware of both the ATM and IP level topologies; this is the "peer model" used by the MPLS developments at the IETF [17, 18]. Each of the two types of methods has its own merits, and play an important role in the design of complex, multi-level networks. The first type of methods has the advantage of simplifying design and operation. It effectively keeps levels independent. It is adapted if the lower level can be easily reconfigured dynamically, as with a SONET or ATM lower level network. In the case of WDM, full dynamic reconfiguration around failed areas is not always possible [5, 6, 7]. Methods are susceptible of finding optimal solutions, since they use a global knowledge. However, in the case of a lower level network based on WDM, it is precisely the requirement for global knowledge which may make a problem. It is often impractical to require higher level software (IP routing, SONET management) to be able to use all the details of the WDM level.

In this paper, we propose and explore an intermediate way for avoiding failure propagation. We show that it is possible to avoid, or at least limit, failure propagation, without modifying the operation of the higher level network. (1) We define requirements on the demands placed by the higher level network onto the WDM network, with respect to protection. We call *demand* a request for establishing a higher level link as a path in the lower level network. (2) Then we use a network design algorithm which carefully places the demands on the optical infrastructure in order to minimize the usage of the infrastructure, while satisfying the constraints defined in the first phase. The effect of our method is that the failure of a single optical node, or of a fiber cable between two nodes, always leaves the higher level in a position where service restoration is possible. This means that the higher level network remains connected and, furthermore, that re-routing of the broken higher level links does not overflow the spare capacity planned by the higher level network [11, 19] when it restores the traffic. Contrary to the method of joint design mentioned above, our method does not require any change to the higher level routing software.

Our method is a complement to intra-level failure restoration mechanisms. Thus, we consider here only failures of optical resources which have a visible impact on the higher level, and exclude failures that are automatically repaired by the protection mechanisms of the WDM network itself. Note that in the case of automatic optical protection, care should be taken to adjust the protection speed to avoid race conditions between the active optical and the higher level protections. Our method should not be viewed as a complete failure avoidance method; rather, it is one component in a global, cross-level network protection strategy. Such global strategies are outside our scope. Section 2 analyzes three types problems that failure propagation could cause to the higher level. Section 3 is the first part of our method. A semi-formal framework is given in order to avoid ambiguities; then it defines (Section 3.2) the requirements that the higher level should issue in order to avoid failure propagation. Section 4 describes our algorithm, which uses taboo search for placing higher level links in order to minimize failure propagation while trying to enforce link capacity constraints. Section 5 discuss the results obtained by the algorithm in various cases.

2 Examples

In this section we point out the problems that occur in the higher level network in case of an unrecovered failure in the optical network. This is done through three examples which show how a higher level network uses the WDM optical infrastructure. Those examples also show the problems that should be addressed to protect the higher level network against single link or node failures.

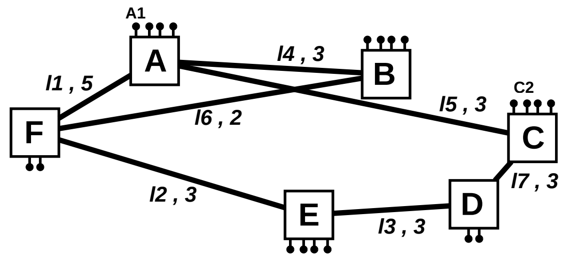

Figure 1: An example of a physical topology of a WDM optical network. It is composed of cross-connect nodes linked through optical fibers. The capacity of each optical fiber is shown next to the fiber number. A port is an access point to the transport capabilities of the optical network.

Figure 1 represents the topology of a simple WDM network with the cross-connect nodes and the optical fibers between them (Node A is connected to nodes B, C and F through fibers $l4$, $l5$ and $l1$). Each cross-connect node has multiple ports. A port is an access point to the WDM transport capabilities.

We call *network mapping* the process of finding a route in the topology of the physical network for each of the higher level network links. A route is always set up between two ports (for example a higher level link between $A1$ and $C2$ could have as route ($l5$), ($l1, l2, l3, l7$) or ($l4, l6, l2, l3, l7$)).

We assume in this paper that nodes perform full wavelength conversion, which means that nodes are capable of routing any incoming wavelength channel on any of the incoming fibers, to any wavelength channel on any of the outgoing fibers. Full wavelength conversion is feasible using opto-electronic conversion in the nodes for large scale networks. The COBNET project [20] is building a low-cost demonstrator using such opto-electronic conversion. All-optical full wavelength conversion is also possible, as can be seen in [21], but is still expensive.

We call *protected group with protection level k* a group of higher level links constituting a network that can support up to k (higher level) link failures. Typically, the higher level network repairs failures from the optical network by rerouting traffic that uses broken links of the group to some other links inside the group. We assume that the *protection strategy* of the higher level network (or collection of networks) involves orga-

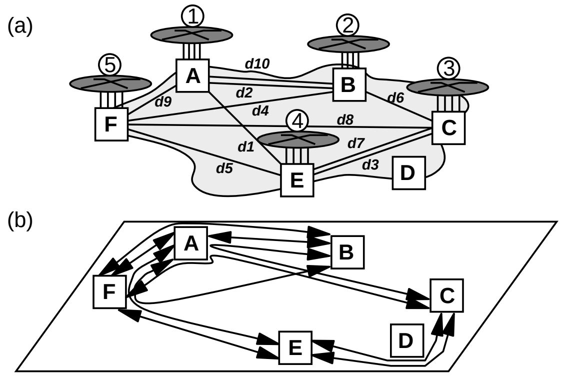

Figure 2: (a) The higher level network view and (b) a network mapping example for the bottleneck problem. High level link $d6$ uses optical links $l4$ and $l5$. There is a single-hop path $p1 = (d6)$ between higher level nodes 2 and 3, as well as alternate, multi-hop paths, like $(d4, d8)$, $(d4, d5, d3)$ and $(d2, d1, d3)$. The physical network is shown in Figure 1.

nizing the higher level links into groups. An example is a SONET network organized as a collection of rings.

We will see in the next paragraphs three different kinds of problems that may happen in the higher level network. Those could occur if the network mapping is done without taking into account the protection strategy of the higher level networks that use the optical infrastructure. The first two examples involve a network with a single protected group, while the last one involves more than one group.

Bottleneck Problem Consider for the first example the higher level network and network mapping shown in Figure 2. The links that the higher level network would like to set up are shown in (a), and a possible network mapping on the physical network of Figure 1 in (b). The higher level nodes are linked by bidirectional higher level links that need to be set up through the optical network (high level node 2 is linked to high level nodes 1, 3 and 5 with higher level links $d2$, $d10$, $d6$ and $d4$). For each higher level link the high level node is connected to a port which is the access point to the WDM transport capabilities (high level node 2 is connected to four ports of node B, as it has four higher level links going out of this node).

Let us assume that in this case the protection strategy consists of one protected group containing all the higher level links, and that the higher level has enough spare capacity to maintain the service even after the failure of three higher level links (protected group with protection level 3). One can see that if physical link $l1$ fails, the protected group is broken in five points ($d1$, $d4$, $d8$, $d9$ and $d10$ were using link $l1$). The higher

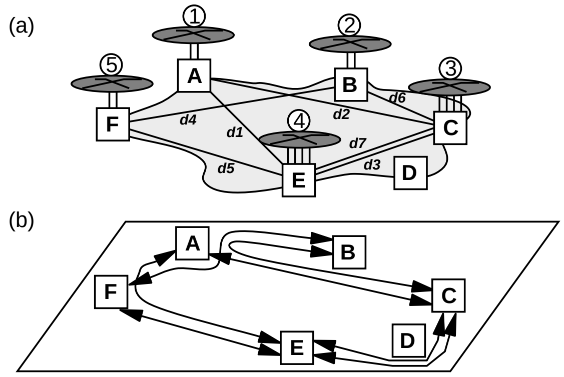

Figure 3: (a) The higher level network view and (b) a network mapping example for the connectivity problem. The physical network is shown in Figure 1.

level network is still connected, so the traffic should be able to reach all the high level nodes. But there are five higher level link failures, and the higher level network has not enough spare capacity to reroute all the traffic. By having $(l6)$ (instead of $(l4, l1)$) as the route on the optical network for higher level link $d4$ and $(l4)$ (instead of $(l1, l6)$) as the route on the optical network for higher level link $d10$ we would avoid any propagation failure for all single link failures in the physical topology, as there is at most three higher level links broken by any failure. The algorithm we propose addresses this problem.

Connectivity Problem Consider for the second example the higher level network and network mapping shown in Figure 3. The links that the higher level network would like to set up are shown in (a), and a possible network mapping on the physical network of Figure 1 in (b).

We assume here that there is one single protected group with protection level $= +\infty$. This means that there is no constraint on the number of broken higher level links as long as it is still possible to reach all the nodes spanned by a protected group using its remaining higher level links. An IP network is an example.

One can see that if physical link $l4$ fails, the protected group is broken in two points only ($d4$ and $d6$ were using link $l4$). But high level node 2, attached to node B, is unreachable at the higher level due to this single link failure even though this higher level node is reachable at the physical level. By having $(l6)$ (instead of $(l4, l1)$) as the route on the optical network for higher level link $d4$ we would avoid any propagation failure for all single link or node failure in

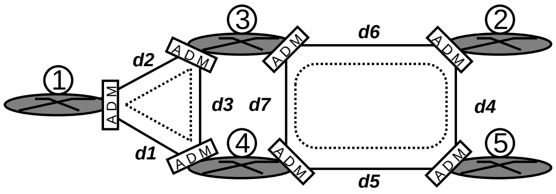

Figure 4: The topology in the case of a sample SONET network, and the two Self-Healing Rings, $(d1, d2, d3)$ and $(d4, d5, d7, d6)$. There are always two disjoint paths between any two Add-Drop Multiplexers (ADMs) on the same ring, the working one and the protection one. For example, between 1 and 3, $(d2)$ and $(d1, d3)$

the physical topology. The problem here is to find a network mapping that leaves all the protected groups connected whatever the single failure is. The algorithm we develop in section 4 solves this problem.

A problem with several groups Consider for the third example a small SONET network, the topology of which is given in Figure 4. One protection possibility for SONET are Self-Healing Rings (SHRs) [10], which rely on a ring topology for protection. This protection method is precomputed, as those rings are set when the SONET topology is designed.

Let us assume that our example of SONET network has two Self-Healing Rings, $(d1, d2, d3)$ and $(d4, d5, d7, d6)$, as shown in Figure 4.

Figure 3 can be used to show how our SONET example is mapped onto the WDM infrastructure proposed in Figure 1. In (a) we see the topology of Figure 4 in an alternate form. The SONET Add-Drop Multiplexers (ADMs) are linked by bidirectional higher level links that need to be set up through the optical network (ADM 2 is linked to ADMs 3 and 5 with higher level links $d6$ and $d4$). In (b) a possible network mapping is represented.

With the protection strategy proposed in Figure 4 and the network mapping shown in Figure 3 one can see that if physical link $l4$ fails, the self-healing ring $(d4, d5, d7, d6)$ cannot be fully protected, as it is broken in two points ($d4$ and $d6$ were using link $l4$). So ADM 2, attached to node B, is unreachable at the SONET level due to a single link failure even though this ADM is reachable at the physical level. By having $(l6)$ (instead of $(l4, l1)$) as the route on the optical network for higher level link $d4$ we avoid any propagation failure for all single link or node failure in the physical topology.

Those three examples show that if the network mapping is done without taking into account the protection strategy deployed by the higher level network, the higher level network could fail due to single failures, even if it was implementing multiple paths or rerouting protection strategies.

The aim of our solution, called *Protection Interoperability*, is to avoid the consequences of single failures in the physical topology. Our solution determines a mapping algorithm knowing the protection strategy of the higher level network. Of course it is still necessary for the higher level network to plan spare capacity as part of its connectivity in order to make service restoration possible [14, 11, 19].

With our first example our algorithm would have as inputs the list of protected groups given in Figure 3 and the physical topology of the optical network given in Figure 1. Based on this information it would issue a network mapping where protected groups remain connected after any single-link failure. One such network mapping is the one shown in Figure 3 with one modification, $(l6)$ as the route of $d4$ instead of $(l4, l1)$.

The problem of finding a network mapping protecting the higher level links from hardware failure consequences was first studied in [7] for a simplified version of the problem and the proposed algorithm solved only the connectivity problem. A new version of this algorithm including the possibility to have capacity constraints on the optical links is presented in [22]. However, both algorithms assume the higher level network is a single protected group.

3 The Protection Interoperability Problem

We have given in the previous section an overview of a WDM network and of the consequences of failure propagation. Here we formally define the problem and the constraints, allowing us to detect if a proposed network mapping is an acceptable solution with respect to failure propagation.

3.1 Formal Definition

We develop in this section a general model and formulate the protection interoperability problem as an optimization problem.

(1) Types We start by defining six types of elements which are used in the formulation. We also give some notation convention used to simplify the notation in this paper. This convention is *not* implemented in the code of our algorithm, it is just used in the English text.

(i) A **Node** represents an optical cross-connect node of the WDM network. In the examples in this paper, the identifiers for objects of type"Node" are a single capital letter.

(ii) A **Port** represents an access point to the optical network for the higher level network. Each port is always associated to a node; in the examples in this paper, the identifiers for objects of type "Port" are the concatenation of a letter (the corresponding node identifier) and an integer (the port number).

(iii)A **Demand** is a couple of two ports. It represents a bidirectional higher level link between two ports.

(iv) A **Link** is a triple (node, node, integer). It represents a bidirectional optical physical link between two nodes. The integer is the maximum number of WDM channels that can be set up on it simultaneously. Note that the transmission can be both full-duplex on a single fiber or half-duplex on a pair of fibers. The integer can be accessed by the function maxcapa(Link l).

(v) A **Clear-Channel** is a list (= ordered set) of nodes. It represents a route in the physical network between two nodes. Clear-channels are also called lightpaths.

(vi) A **p-Group** is a couple made of a set of demands and an integer. It represents demands belonging to a common protected group of the higher level network. The integer, called "protection level" of the p-group, is the maximum number of demands that can fail without causing failure propagation. The underlying assumption is that, for a p-group with protection level k, the higher-level network is able, by means of its own intra-level restoration mechanisms, to continue operation as long as k or fewer higher-level-links are broken, and that the group remains fully connected. The protection level k can be accessed by means of function protectlevel(p-Group p)

(2) Variables Used to model our Problem We model our generic problem with the following variables, which are set of objects, each object being of one of the types defined above.

(a) N is a set of nodes. It represents the cross-connect nodes of the WDM network. For Figure 1, $N = \{A, B, C, D, E, F\}$.

(b) L is a set of links. It represents the topology of the WDM network. We have the following constraints on L: there should be only one physical link between two nodes (if many fibers are present, they are considered as one physical link having as capacity the sum of their respective capacities); no self-loops are allowed; the physical network should be at least two-connected. For Figure 1, $L = \{(A, F, 4), (E, F, 3), (D, E, 3), (A, B, 3), (A, C, 3), (B, F, 2), (C, D, 3)\}$.

(c) D is a list of demands. It represents the higher level links that the higher level networks require to set up using the optical infrastructure. There is only one constraint on D: self-loops are not allowed. Note that there is no constraint on the number of ports (transmitter, receiver) per node. This is a constraint for the higher level topology mapping algorithm [23, 24], and we assume that the given set of demands D fulfills these constraints. For Figure 3, $D = ((A1, E2), (A2, C2), (C4, E4), (B1, F1), (E1, F2), (B2, C1), (C3, E3))$.

(d) P is a set of p-groups. It represents the protection strategy of the higher level network. P should fulfill the following constraints. Firstly, all the p-groups of P should be at least two-connected, as no protection path can be found in a p-group where a demand that disconnected the network was removed. Secondly, all demands in a p-group should be members of D. Note however that p-groups need not to be disjoint. In the case of adaptive re-routing in the higher level networks, the same demand could belong to more than one p-group. Also note that different higher level networks (for example SONET, ATM, ...) can coexist on the top of the same optical infrastructure, and will result in different protected groups. For Figure 4, $P = \{(\{d1, d2, d3\}, 1), (\{d4, d5, d6, d7\}, 1)\}$.

(e) M is a list of clear-channels. It represents the network mapping, namely the mapping onto the physical topology L of all the demands present in the logical topology D. The mapping establishes a 1:1 correspondence between clear-channels and the demands in D. The network mapping found by the mapping algorithm should fulfill the following constraints: the capacity constraint of the physical links should be respected; clear-channels should have no cycles; each clear-channel of M should be associated to one and only one demand of D and should be connected to the same nodes; each demand of D should have a clear-channel associated to it. For Figure 3 $M = ((A, F, E), (A, C), (C, D, E), (F, A, B), (E, F), (B, A, C), (C, D, E))$.

3.2 The problem

We now define the requirements that the higher level poses in order to avoid failure propagation.

(1) The capacity constraints of the links should be respected. If this is not the case, then not all demands can be routed.

(2) Each p-group should remain connected in presence of any single link or node failure.

(3) The number of demands belonging to a same p-group affected by a single link or node failure does not exceed the protection level of the p-group.

We now give a formal definition of those three constraints.

(1) Capacity Constraint We define a function $\mathcal{F}_{Cap}(Z, y)$, where Z is a set of clear-channels and y a physical link as follows. $\mathcal{F}_{Cap}(Z, y)$ gives, for the set of clear-channels Z, the number of demands that uses link y. To solve the protection interoperability problem, we have to find, given L and P, a network mapping M such that

$$\forall l \in L \quad \mathcal{F}_{Cap}(M, l) \ \leq \ \text{maxcapa}(l) \qquad (1)$$

where l is a physical link of L and maxcapa(l) the maximum number of channels on l. This ensures that the capacity constraint of the physical network is respected by the mapping proposed by the solution.

(2) Connectivity Constraints We define two functions $\mathcal{F}_{ConL}$ and $\mathcal{F}_{ConN}$. $\mathcal{F}_{ConL}(Z, p, y)$ gives, for the set of clear-channels Z, and a p-group p, the subset of the set of demands of the p-group which do not use link y in their associated clear-channel Z. $\mathcal{F}_{ConN}(Z, p, y)$ is similar, but here y is a node instead of a link. To solve the protection interoperability problem, we have to find, given L and P, a network mapping M such that the two graphs with edges

$$\mathcal{F}_{ConL}(M, p, l) \text{ with } p \in P, l \in L \qquad (2)$$

$$\mathcal{F}_{ConN}(M, p, n) \text{ with } p \in P, n \in N \qquad (3)$$

are connected. This ensures that for any single link or node failure in the physical topology there is still a path between any involved higher level nodes for all the p-groups of P.

(3) Bottleneck Constraints We define two functions $\mathcal{F}_{BotL}$ and $\mathcal{F}_{BotN}$. $\mathcal{F}_{BotL}(Z, p, y)$ gives, for the set of clear-channels Z, and a p-group p, the number of demands of the p-group which were using link y in their associated clear-channel in Z (except the one having a port in node y, in case of node failure). $\mathcal{F}_{BotN}(Z, p, y)$ is similar, but here y is a node instead of a link. To solve the protection interoperability problem, we have to find, given L and P, a network mapping M which satisfies:

$$\mathcal{F}_{BotL}(M, p, l) \leq \text{protectlevel}(p) \qquad (4)$$

$$\mathcal{F}_{BotN}(M, p, n) \leq \text{protectlevel}(p) \qquad (5)$$

for all $p \in P$, $l \in L$ and $n \in N$. This ensures that the number of broken demands is always small enough to allow the higher level network to restore the traffic, whichever resource (link or node) or p-group is considered.

The problem thus becomes: *For given sets N, L, D, P, find, if it exists, a network mapping M which satisfies Eqs 1 to 5.*

The search for a solution may require the evaluation of all the functions $\mathcal{F}$ for all the possible values of M if a full solution space search is performed. This entails calculating, for each demand of D, all the possible elementary routes between the two nodes it is connected to in the physical network.

Our simulations have shown that the number of solutions for which $\mathcal{F}_{Cap}$, $\mathcal{F}_{ConL}$, $\mathcal{F}_{ConN}$, $\mathcal{F}_{BotL}$ and $\mathcal{F}_{BotN}$ need to be computed was very large, even for small problems. Moreover, this number rapidly grows with the size of the problem. This is why we have not tried to find the optimum solution directly, but developed a heuristic which tries to minimize the number of times the criteria of Eq. 1 to Eq. 5 are not met, as explained in section 4.

The problem is NP-Complete. It is NP, as one can check if a given network mapping fulfills all the constraints in polynomial time. It is also NP-Complete, as at least three of its sub-problems are NP-Complete. It is easy to show that the Capacity Constraint (Eq. 1) and the two Bottleneck Constraints (Eq. 4 and 5) problems are each equivalent to the *multicommodity flow problem with integer link flows* [25] and this problem has been shown to be NP-Complete for undirected graphs in [26].

3.3 Related Problems

A number of related problems can be solved as subsets of our main problem.

Subset 1 If the aim is to find a mapping which protects completely the higher level networks from the consequences of failure propagation at any cost, the capacity constraint may be set aside if it is possible to rent more optical channels from the provider. In that case, the constraints are Eq 2 to 5.

Subset 2 If protected groups have been built to protect the higher level networks from multiple demands blocking in case of link failure only (by having large amount of spare capacity and a high connectivity), it may not be necessary to check the constraints involving $\mathcal{F}_{ConL}$, $\mathcal{F}_{ConN}$ and $\mathcal{F}_{BotN}$. Indeed, by satisfying Eq 4 only, the number of blocked demands in a p-group due to a single-link failure should always be smaller than the connectivity. The Bottleneck problem presented in section 2 (Figure 2) is of this kind.

Subset 3 If we are interested only in protection against link failures, then constraints in Eq 3 and Eq 5 should be dropped.

Slight variations of our problems, which can be solved with minor modifications of our algorithms, are as follows.

(1) Consider the case where the higher level network does not give any information on protectlevel(p). The problem is now to try and minimize the number of demands of the same group using each resource. The bottleneck constraints would be in this case:

$$\forall p \in P, l \in L \mathcal{F}_{BotL}(M,p,l) \quad \text{is minimum} \tag{6}$$

$$\forall p \in P, n \in N \mathcal{F}_{BotN}(M,p,n) \quad \text{is minimum} \tag{7}$$

This alternative is not presented here. It could be implemented by slightly modifying the algorithm described in section 4.

(2)In this paper we focus on the search for one solution which satisfies all constraints. In the case where a solution exists, a natural problem is to find one which minimizes the total usage of optical links. This can be solved by a slight modification of our algorithm, as explained in Section 4.1.

4 The PIW Algorithm

We develop in this section an algorithm to perform protection interoperability. The results of our algorithm on various test networks are discussed in section 5. These show that failure propagation should be taken into account when performing the network mapping and the strong enhancement in protection achieved by the Protection Interoperability for WDM (PIW) algorithm compared to the one achieved by simple mapping strategies, such as a Shortest-Path Routing algorithm.

4.1 Outline of the Algorithm

The PIW algorithm tries to find a solution to the network mapping problem which respects the capacity constraints, defined in Section 3. It does this in the following way. We transform the search for a solution to the set of constraints into an optimization problem. The function to optimize is designed such that we greatly penalize states where capacity constraints are violated.

The reason for this transformation comes from the way search heuristics operate. A strict enforcement of the capacity constraints may confine the algorithm to a small "island" in the possible solution space, without any possibility of leaving it by relaxing the capacity constraints. It also makes difficult the computation of the initial solution or the use of the algorithm if the capacity constraints cannot be respected by any solution.

The algorithm tries to find the minimum of a function $\mathcal{F}'$. This function, which is a modification of Eqs 1 to 5, measures the number of times that the constraints are not met. To solve the protection interoperability problem, we have to find a network mapping M which minimizes:

$$\begin{aligned}\mathcal{F}'(M,P,L) = \sum_{l\in L}\sum_{p\in P}(a{\mathcal{F}'}^2_{Cap}(M,l) \\ +b\mathcal{F}'_{ConL}(M,p,l) + c\mathcal{F}'_{ConN}(M,p,n) \\ +d{\mathcal{F}'}^2_{BotL}(M,p,l) + e{\mathcal{F}'}^2_{BotN}(M,p,n))\end{aligned} \tag{8}$$

where

(1) $\mathcal{F}'_{Cap}(Z,y) = \max[0, \text{clear-channels using } y - \text{maximal capacity of } y]$ gives, for a network mapping Z and a resource (link or node) y, the number of times this resource is overused.

(2)$\mathcal{F}'_{ConL}(Z,p,y)$ gives, for a network mapping Z, a protected group p and a physical link y, the sum over each group of the number of demands in p with two end-points that do not belong to the same subgraph (and therefore cannot be protected by the higher level network) if physical link y is broken. Note that when $\mathcal{F}'_{ConL} = 0$, the conditions of Eq. 2 are met.

(3) $\mathcal{F}'_{ConN}(Z,p,y)$ is similar to $\mathcal{F}'_{ConL}$, but in case of node failure.

(4) $\mathcal{F}'_{BotL}(Z,p,y) = \max[0, \text{clear-channels using } y - \text{protectlevel}(p))]$ gives, for a network mapping Z, a protected group p and a physical link y, the number of demands exceeding the spare capacity of protected group p if this link is broken.

(5)$\mathcal{F}'_{BotN}(Z,p,y)$ is similar to $\mathcal{F}'_{BotL}$, but in case of node failure.

(6) a, b, c, d, e are arbitrary parameters chosen such that $a >> b$, $c >> d$ and $c >> e$. For the cases shown in Section 5, values such as $a = c = 10$, $b = d = e = 1$ worked well.

We have mentioned in Section 3.3 some subsets of the problems. These subsets correspond to dropping some of the constraints. With our algorithm, this is translated into zero values for the corresponding coefficients. The parameters should thus be chosen as follows for these special cases. For subset 1, let $a = 0$. For subset 2, let $b = c = e = 0$. For subset 3, let $c = e = 0$.

The exponent 2 for $\mathcal{F}'_{Cap}$, $\mathcal{F}'_{BotL}$ and $\mathcal{F}'_{BotN}$ in Eq. 8 is used to lower the variance of the number of demands blocked by each resource ($\sum_{i=1-3} x_i^2$ is smaller for (2,2,2) than for (1,3,2)).

Note that for $\mathcal{F}'_{ConN}$ and $\mathcal{F}'_{BotN}$ (node failure protection), the demands that have a port in the broken node are not counted. They are not counted because there is no possibility to reach higher level nodes connected to the broken node anyway, as all the physical links connected to it are out of order.

We have also mentioned in Section 3.3 a variant of the problem which would consist in finding a solution which satisfies all constraints and minimizes the total usage of optical links. This problem would be solved by adding to $\mathcal{F}'$ the term $+f\mathcal{F}'_{Usage}(M)$, where $\mathcal{F}'_{Usage}(M)$ is the sum of wavelengths per link that are allocated to a demand in the network mapping M, over all links. The parameter f is an arbitrary parameter much smaller than a, b, c, d and e. In the rest of the paper we focus on the main problem and do not consider this variant any further.

The PIW algorithm uses the Taboo Search heuristic to find a network mapping M fulfilling the criteria defined in section 3.2 by iteratively minimizing Eq. 8. Taboo Search can be defined as a general heuristic in which a local search procedure is applied at each step of the general iterative process. It can be superposed on other heuristics to prevent those being trapped in a local optimum. For interested readers, a recent paper [27] gives a detailed description of the method.

The PIW algorithm starts by initializing the network mapping M to an arbitrary mapping M_0 (given by randomly placing clear-channels on the physical network for all of the demands). Then, at each iteration, it explores the solution space near the actual M by slightly modifying the route of a clear-channel to have a new network mapping. It chooses thanks to taboo search the best network mapping not yet visited and stores it in M. After a given number of iterations without finding a network mapping M better than the best one found until now, the algorithm stops.

Two features added to "plain" taboo search enlarge the visited solution space and reduce the number of iterations required to reach an optimum solution [28]. Firstly, the size of the taboo list changes randomly from time to time in order to alternate between intensive search and diverse search. Intensive search occurs when the size of the taboo list is small: a recent move can be accepted again and the search focuses on a small region. In contrast diverse search is when the size of the taboo list is large: a recent move cannot be accepted again for a long time, which forces the algorithm to select new or rarely chosen moves, thus visiting new regions. Secondly, the "aspiration criteria" allows acceptance of a move even if it is already present in the taboo list, provided that this move leads to a state whose value of $\mathcal{F}'$ is smaller than the best solution found previously. This guarantees that the move, even if normally forbidden, leads to a state never visited before.

The PIW algorithm structure is as follows. The solution is M_{best}

```
M = M_0;    /*initialization of network mapping,*/
M_best = M_0; /*M_0 arbitrary*/
nbit = 0;
while (nbit < nbitmax) {
   for (all d of D) {
      M_d = M with clear-channel of d modified};
   M = the M_d which minimizes F' and not yet visited;
   nbit = nbit + 1;
   if (F'(M) < F'(M_best)) {
      M_best = M; nbit = 0};
}; /*while*/
```

4.2 Tunable Parameters of the Algorithm

The PIW algorithm uses the taboo search parameters as defined below. The values for the six parameters described in this section have been found by empirical testing. The results with various values for them are not shown here, as the main aim of this paper is the description of Protection Interoperability and a way to achieve it, and not the fine tuning of parameters.

Move A move m consists of changing the clear-channel of a demand d without using the links present in the movelist mld.

Movelist A movelist mld is associated with each demand d. It consists of the list of all the links that have already been forbidden by a previous move, plus the new forbidden link, associated with this move. If no route for d can be found on the physical topology due to this constraint (there is a cutset [29] between the two nodes that d connects), the oldest included link of mld is removed. The reason for forbidding the use of a list of links instead of just a link is to enlarge the visited solution space by avoiding oscillation between two routes. The link to be added to the list is chosen randomly between all the links that are part of the clear-channel associated to this demand, with a higher probability to choose a physical link l for which $\mathcal{F}'_{Cap}(M,l)$, $\mathcal{F}'_{ConL}(M,p,l)$ or $\mathcal{F}'_{BotL}(M,p,l)$ were not equal to zero (if there was one) or a physical link l connected to a node n for which $\mathcal{F}'_{ConN}(M,p,n)$ or $\mathcal{F}'_{BotN}(M,p,n)$ were not equal to zero. The randomness in choosing this link renders cycling less probable and higher probability of choosing a link causing $\mathcal{F}'_{Cap}, \mathcal{F}'_{ConL}, \mathcal{F}'_{ConN}, \mathcal{F}'_{BotL}$ or $\mathcal{F}'_{BotN} > 0$ directs the search faster towards good solutions.

Taboo List The taboo list is the list of all the demands that have been chosen as best move for the last *t_size* iterations.

Size of Taboo List The size of the taboo list, *t_size*, varies between $\lfloor 0.9 \cdot mts \rfloor$ and $\lceil 1.1 \cdot mts + 4 \rceil$[2] [28], where *mts* is defined as a function of the number of demands in the logical topology ($|D|$). A new value for *t_size* is uniformly randomly chosen between the two bounds after every $2 \cdot \lceil 1.1 \cdot mts + 4 \rceil$ iterations. *mts* is set to $|D|/4$.

Stopping Criteria The stopping criteria *maxitnb* has been defined as the maximum number of iterations allowed with no decrease of $\mathcal{F}'$ (Eq. 8) over the best found solution until now. *maxitnb* is needed because we do not know the theoretical minimum of $\mathcal{F}'$. *maxitnb* is set to $3 \cdot |D|$.

4.3 Complexity of the Algorithm

The complexity of an iteration of the PIW algorithm can be expressed in function of the number of physical links ($|L|$), of demands ($|D|$), of p-groups in P ($|P|$) and of nodes (N). An iteration requires the computation of a new clear-channel in the physical topology for all the demands, which leads to the complexity $|D| \times \mathrm{O(Dijkstra)}$. The computation of $\mathcal{F}'$ for all the moves of an iteration requires, for each p-group of P, each link of L and for all the demands using it, to find an alternate path, which adds a complexity of $|P| \times |L| \times |D| \times \mathrm{O(Dijkstra)}$.

Generally, the upper bound for fully connected graphs of Dijkstra's algorithm is given by N^2 [30], so $\mathrm{Time(PIW)} = |D| \times N^2 + |P| \times |D| \times |L| \times N^2$. If $|D|$ and $|L|$ are expressed as $\alpha \times N$ and $\beta \times N$ with α and β being the average connectivity per vertex, the complexity can be expressed as $\mathrm{O}(\alpha \times N^3 + |P| \times \alpha\beta \times N^4)$ $= \mathrm{O}(N^4)$, or $\mathrm{O}(N^5)$ if $|P|$ also depends on the number of nodes.

5 Numerical Results

Two simple mapping algorithms, SPR-P and SPR-CC, are compared to PIW. By studying the results of the three algorithms on various topologies, we show that protection interoperability is an important issue that should be addressed when finding clear-channels for the demands in a WDM network, and that the PIW algorithm solves the problem. The three algorithms are:

PIW The Protection Interoperability for WDM algorithm, as described in section 4.

SPR-CC The Shortest-Path Routing – Capacity Constraint algorithm. The use of SPR to perform the network mapping means that each demand is placed on the physical topology using the shortest route [23], without any optimization towards resilience against failure protection. Capacity constraints are, when possible, respected by SPR-CC. It always chooses the shortest clear-channel which minimizes the number of times the capacity constraints are violated. SPR-CC maps all the demands on the physical network one after the other, the next one being randomly chosen among those not yet placed.

SPR-P The Shortest-Path Routing – Plain algorithm. The Shortest Path on the physical topology is chosen for each demand. The Capacity Constraints are not taken into account in this algorithm, namely the algorithm acts as if infinite capacity was associated with each of the links.

SPR-P, SPR-CC and PIW have been implemented in C++ on a Sparc 20 workstation and applied to two kind of configurations to be compared.

For all the tests, the cost associated with the use of a link is set to 1 for all the links, in which case the

[2] small problems need proportionally a longer taboo list than large ones. This is why there is a "+4" in the upper bound value.

length of a path corresponds to the number of links it uses. The maximal number of wavelength channels per optical link is set to the same value Λ_{max} for all the existing links. The values of Λ_{max} for the various examples have been chosen in such a way that there exists at least one network mapping which respects the capacity constraints. It should be noted that the PIW algorithm may also be used for problems where different values are associated with each optical link for the cost (like delays, distance between nodes) or the capacity (various types of fibers, or nodes with different number of transceivers).

The tests have been performed to demonstrate the protection achieved by the three algorithms against link or node failure. In the case of node failure, none of the fibers having an end-point in the broken node is then usable. Consequently all the demands that had a port or which were routed through that node are broken. Note that demands which had a port in the broken node are not counted when computing $\mathcal{F}'_{BotN}$ as the higher level nodes they were starting from have no more access to the optical infrastructure. Their traffic cannot be rerouted anyway.

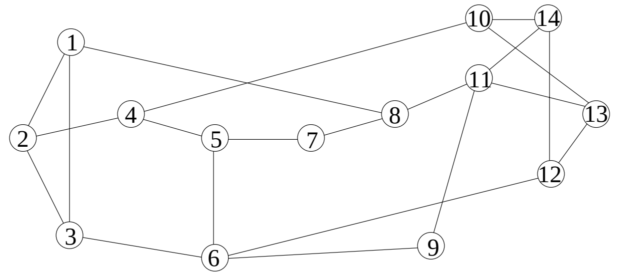

Figure 5: The 14-nodes 21-links NFSNET physical network.

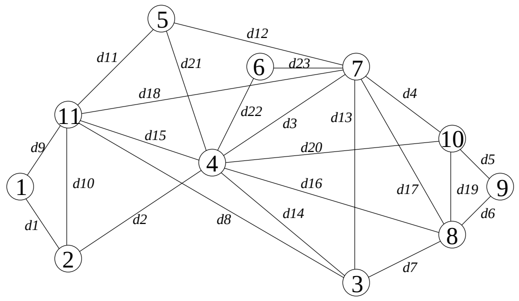

Figure 6: The SONET logical topology containing 23 demands.

5.1 SONET over WDM

We consider in this section the case where a SONET network is set up over a WDM infrastructure. One protection possibility for SONET are Self-Healing Rings (SHRs) [10], which rely on a ring topology for protection. This protection method is precomputed, as those rings are set when the SONET topology is designed.

The example below shows what happens to the SHRs protection strategy if the network mapping is done without taking into account the ring structure of the protection. It also show the protection improvement achieved by the PIW algorithm:

(1) The physical topology used for this example is the NFSNET network, as shown in Figure 5. The maximum number of channels is the same for all the fibers and has been set to $\Lambda_{max} = 4$.

(2) The SONET logical topology is shown in Figure 6. This logical topology is taken from a paper by W. Grover [6]. Note that optical cross-connect nodes 12, 13, 14 in Figure 5 are transit nodes and do not have any demand connected to them, but may be used by clear-channels.

(3) The protection strategy consists of 5 protected groups, each one being a ring (from [6]). Note that, as each protected group is a ring, there could be at most one demand of a p-group blocked by any failure and no other one in case of failure of a node having demands starting in it. If there is more than one demand blocked, the p-group would be disconnected. In such a case (ring) the connectivity constraint is equivalent to the bottleneck one, with protectlevel$(p) = 1$ for all the p-groups (cf. Eq. 6 and Eq. 7). So we have $P = \{((d1, d2, d3, d4, d5, d6, d7, d8, d9), 1), ((d2, d10, d11, d12, d13, d14), 1), ((d15, d16, d17, d18), 1), ((d16, d19, d20), 1), ((d12, d21, d22, d23), 1)\}$.

For the computation we have used only the capacity and bottleneck constraints, by having $a = 10$, $b = 0$, $c = 0$, $d = 1$ and $e = 1$ for the PIW algorithm. The number of demands, p-groups, demands per p-group, the protection level and the value of the links' capacities (Λ_{max}) are summarized in Table 2. The results for this configuration are discussed in section 5.3.

It should be noted that all SONET Ring-based protection strategies (ULSR, UPSR, BLSR using [9] terminology) can be easily mapped into protected groups. Each pair of unidirectional SONET links between two ADMs constitute a demand, and all the demands of the ring belong to one protected group with protection level one. Figure 7 shows as example of how it can be done for a SONET 4-fiber Bidirectional Line Switched Ring (BLSR). We have in (a) a 4-fiber BLSR with four rings, and in (b) how those

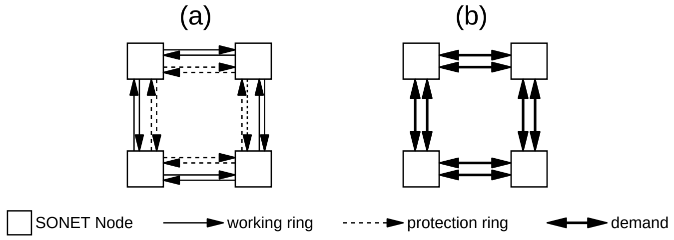

Figure 7: (a) A 4-fiber Bidirectional Line Switched Ring (BLSR) with four parallel unidirectional rings and (b) The mapping of those rings into bidirectional demands at the logical level.

are mapped to demands. In normal conditions, only two SONET rings (one ring of demands) are used to transmit the traffic.

The protection strategy of the 4-fiber BLSR is the most resilient of SONET protection schemes. It is achieved by having the pair of protection rings and the pair of working ones set on physically disjoint paths. When mapping such rings onto a WDM infrastructure, care should be taken to ensure disjoint paths for each demand of a ring, but also between the two pairs of rings. It can be achieved by having all the demands of the two rings in one p-group with protection level 1 (protectlevel$(p) = 1$). Having a protection level equal to 1 allow the routing of only one demand of this p-group per fiber. As the demands of the two rings are in the same p-group, this constraint ensures that the two pairs of SONET rings are using physically disjoint paths.

5.2 Arbitrary Network

We consider in this section a general case where an arbitrary logical topology has to be mapped onto a randomly generated physical network:

(1) The physical topology used for this example is the network shown in Figure 8. This network has been created by randomly generating links to obtain a two-connected topology of average degree[3] three.

(2) The logical topology has been defined by randomly generating demands to obtain a (at least) two-connected network of average degree three.

(3) The protection strategy (set of p-groups) for each logical topology has been defined by building five p-groups, each one being at least two-connected and containing $\frac{1}{4}$ of the demands. The choice is done randomly, and so a demand may belong to none of the

[3]The *nodal degree* of a node is the number of edges having one end-point in it. The nodal degree of a network is the mean of its nodes' nodal degree

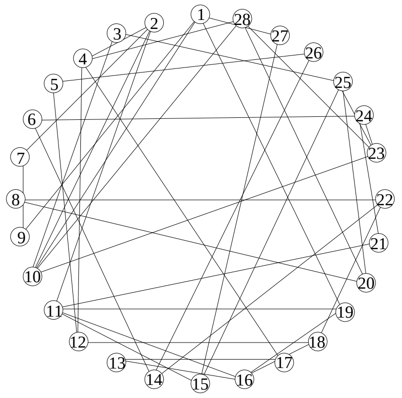

Figure 8: The 28-nodes 42-links Random physical network.

p-groups, or to more than one. The protection level of all the p-groups has been arbitrarily set to three.

This example has complex protected groups (overlapping, degree larger than two). In this case the five constraints have been included in the optimization algorithm, as we want to find network mappings that respect the capacity constraints, leave all the protected groups connected after any single failure and minimize the number of demands of a same p-group using the same physical link. This is done by having $a = 100$, $b = 10$, $c = 10$, $d = 1$ and $e = 1$ in the PIW algorithm.

The number of demands, p-groups, demands per p-group, the protection level and the value of the links' capacities (Λ_{max}) are summarized in Table 2. The results for this configuration are discussed in the next section.

5.3 Discussion

The values of $\mathcal{F}'_{Cap}$, $\mathcal{F}'_{ConL}$, $\mathcal{F}'_{ConN}$, $\mathcal{F}'_{BotL}$ and $\mathcal{F}'_{BotN}$ (cf. Eq. 8) for the network mapping proposed by the three algorithms are shown for the SONET over NFSNET example in Figure 9 and for the arbitrary logical topology over random network in Figure 10. $\mathcal{F}'_{Cap}$ represents the number of time the capacity constraints are not respected, $\mathcal{F}'_{ConL}$ and $\mathcal{F}'_{ConN}$ the number of times the connectivity constraints are not respected in case of link and node failure, respectively. $\mathcal{F}'_{BotL}$ and $\mathcal{F}'_{BotN}$ do the same for the bottleneck constraints.

The numbers shown in Figures 9 and 10 are average values of ten simulation runs for SPR-CC and PIW.

Those two figures show that SPR-P and SPR-CC do not in any way fulfill the various constraints and that PIW performs better for all the constraints in the two examples. The PIW algorithm finds network mappings fulfilling the capacity constraint, which was

	$\|D\|$	$\|P\|$	$\|D/p\|$	protectlevel(p)	Λ_{max}	a	b	c	d	e
SONET	23	5	5	1	4	10	0	0	1	1
Random	42	5	11	3	7	100	10	10	1	1

Table 1: Number of demands in the logical topology ($|D|$), the number of p-groups ($|P|$), average number of demands in each p-group ($|D/p|$), protection level (protectlevel(p)) and maximum number of demands allowed per physical link (Λ_{max}) for the two examples. The table also shows the values of the parameters which select what should be minimized by the PIW algorithm when solving Eq. 8.

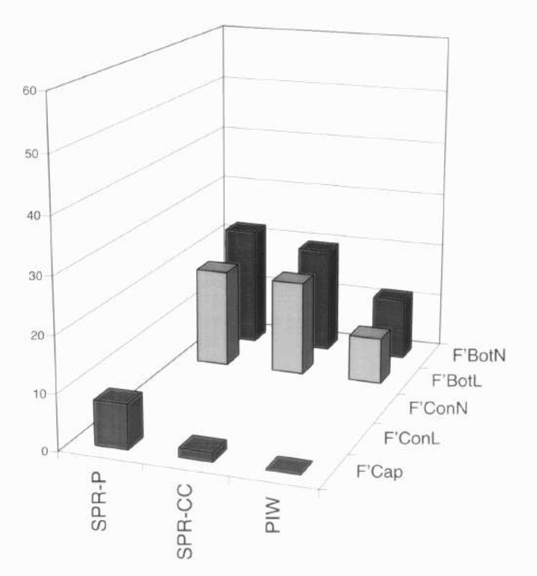

Figure 9: The values for $\mathcal{F}'_{Cap}$, $\mathcal{F}'_{BotL}$ and $\mathcal{F}'_{BotN}$ (Eq. 8) found by the three algorithms for the SONET over WDM example. Note that $\mathcal{F}'_{ConL}$ and $\mathcal{F}'_{ConN}$ have no values displayed, as they were not used in the computation ($b = c = 0$). One can see that the PIW algorithm always finds network mappings fulfilling the Capacity Constraint ($\mathcal{F}'_{Cap} = 0$), which is not the case for SPR-P and SPR-CC. Moreover, the network mappings of SPR-P and SPR-CC have values for $\mathcal{F}'_{BotL}$ and $\mathcal{F}'_{BotN}$ almost twice as large as those of network mappings found by PIW. That means that SPR-CC and SPR-P produce network mappings where spare capacity could be overflowed twice as often as in those produced by PIW.

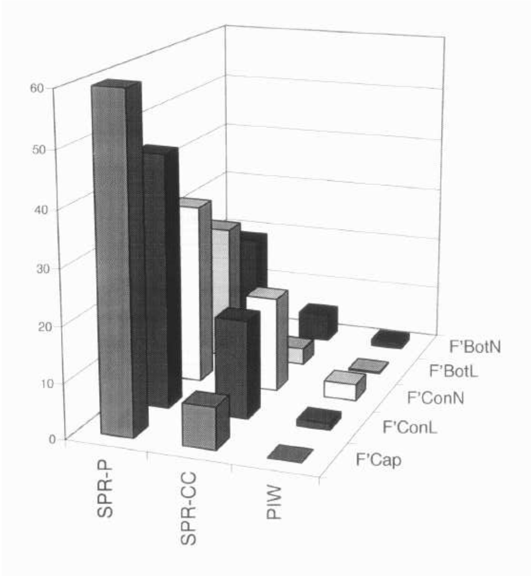

Figure 10: The values for $\mathcal{F}'_{Cap}$, $\mathcal{F}'_{ConL}$, $\mathcal{F}'_{ConN}$, $\mathcal{F}'_{BotL}$ and $\mathcal{F}'_{BotN}$ (Eq. 8) found by the three algorithms for the arbitrary logical topology over random physical network. One can see that the PIW algorithm always finds network mappings fulfilling the Capacity Constraint ($\mathcal{F}'_{Cap} = 0$), which is not the case for SPR-P and SPR-CC. The PIW algorithm resolves also the connectivity and bottleneck constraints for both link and node failures reasonably well, as we have average values very close to zero. SPR-CC and SPR-P produce network mappings where disconnections in the protected groups may happen quite often and where spare capacity could be overflowed ($\mathcal{F}'_{ConL}$, $\mathcal{F}'_{ConN}$, $\mathcal{F}'_{BotL}$ and $\mathcal{F}'_{BotN}$ are large).

selected as the major constraint. It should be noted that, even if the network mappings found by PIW have to fulfill the capacity constraint first, they are more resilient against connectivity and bottleneck problems than those found by SPR-P and SPR-CC in the two examples. Also note that in the SONET over WDM example a network mapping fulfilling the bottleneck constraints does not exist and that PIW finds the optimum solution.

Figures 9 and 10 show that the constraints are respected by PIW whereas SPR-P and SPR-CC perform poorly. However, it may be difficult to understand the implications of high values of the various $\mathcal{F}'$s in these figures on failure protection.

To have a clearer idea we computed various measurements for the network mappings found by the three algorithms and summarized them in Tables **??**-4. These three tables show the consequences of a bad network mapping in terms of probability of failure propagation unrestorable by the higher level network and of average number of blocked demands for such an unrecoverable failure in case of single failure in the optical network.

Table **??** corresponds to the SONET over WDM example and summarizes information about the capacity and bottleneck constraints. It shows in the first and second columns that the capacity constraint is respected for all the links by the PIW network mappings, whereas it is not the case for the two other algorithms (9.5% of the links are overused by SPR-P for example). In case of link failure (columns three and four) one can see that PIW reduces the percentage of links causing an unrecoverable error in the higher level network and the significance of such an event (1.1 unrestorable demand on average versus 1.5 for SPR-CC for example). In case of node failure (columns five and six) SPR-P seems to perform better as its probability of unrecoverable failure propagation is smaller. However, the average number of non restorable demands is twice as small for PIW, and the PIW network mappings are nonetheless better than the two others by almost 25%. The difference between PIW and SPR network mappings is smaller than what is achieved in the arbitrary logical topology over a random physical network, but this is because the solution found by PIW is the mathematical minimum for this problem, and that a solution with $\mathcal{F}' = 0$ is not achievable.

Table 3 shows the same measurements as Table **??**, but for the arbitrary logical topology over random network. One can see in this table that the values for PIW are a lot better than those for the network mappings found by SPR-P or SPR-CC. For example, only 1% of the links may cause a failure propagation in PIW (as opposed to 16.7% for SPR-P) and such a failure would cause, on average, only 0.3 unrecoverable demand (as opposed to 1.6 for SPR-P).

Table 4 shows the measurements for the arbitrary logical topology over random network and for the connectivity constraint. The values for the capacity constraints are the same as those of Table 3 and are repeated here to show that the improvement concerning the connectivity constraint is not achieved by relaxing the capacity constraint, but that both are achieved jointly. One can see in this table that PIW finds network mappings which are better than those found by SPR-P and SPR-CC also for the connectivity constraint. For example only 4% of the nodes may cause a failure propagation using PIW (as opposed to 23.9% for SPR-CC) and such a failure would cause on average only 1.3 unrecoverable demands (as opposed to 2.6 for SPR-CC).

The three tables show that protection interoperability is an important problem that should be explicitly addressed when performing the network mapping. Not doing so could lead to unrecoverable failures in the higher level network using the optical infrastructure even if it implements a protection strategy.

6 Conclusion

WDM optical networks allow an optimal use of the fiber capacity by offering a wide bandwidth, necessary to respond to the needs of actual applications. However we are also witnessing a rapidly increasing concern about the reliability of the communications, as the consequences of a failure in high-speed networks are proportional to the amount of information flowing through the network.

The independence between the logical and physical topologies offers a lot of possibilities to build the network mapping. We have shown the problems that failure propagation causes if the choice of clear-channels is realized without taking into account the interaction between the two levels.

In this article, we have argued that protection inside WDM networks has to be implemented jointly with the higher level networks using the optical infrastructure. This should be done in order to find a network mapping fulfilling the capacity, connectivity and bottleneck constraints that are required by the protection strategy of the higher level networks. As the resolution of this problem is NP-Complete, a heuristic, the PIW algorithm, has been developed to find a good network mapping rapidly.

The PIW algorithm includes parameters which can be used to adapt the problem to different cases of higher level protection strategies (like SONET rings, IP or ATM networks) and to various kinds of failure protection (node or link).

Numerical results have shown that by using this PIW algorithm to place the higher level demands onto the physical infrastructure, it is possible for the higher level networks to rely on their protection strategy. This is not the case if the network mapping is done using algorithms which do not try to solve the protection interoperability problem, such as SPR-P and SPR-CC for example. Should these algorithms be used, single failure in the WDM infrastructure could propagate and cause unrecoverable errors in the higher level networks.

References

[1] C. A. Brackett, "Dense Wavelength Division Multiplexing networks: Principles and Applications,"

	$\|D\|$	$\|P\|$	$\|D/p\|$	protectlevel(p)	Λ_{max}	a	b	c	d	e
SONET	23	5	5	1	4	10	0	0	1	1
Random	42	5	11	3	7	100	10	10	1	1

Table 2: Number of demands in the logical topology ($|D|$), the number of p-groups ($|P|$), average number of demands in each p-group ($|D/p|$), protection level (protectlevel(p)) and maximum number of demands allowed per physical link (Λ_{max}) for the two examples. The table also shows the values of the parameters which select what should be minimized by the PIW algorithm when solving Eq. 8.

	Capacity Constraint		Bottleneck Constraint			
			Link Failure		Node Failure	
	Over vs Total (in %)	Average Over	Bad vs Total (in %)	Average Broken	Bad vs Total (in %)	Average Broken
SPR-P	16.7	2.7	16.7	1.6	14.3	1.8
SPR-CC	6.9	1.4	6.7	1.0	12.3	1.2
PIW	0.2	0.1	1.0	0.3	3.9	1

Table 3: Arbitrary logical topology over random network case. Capacity constraint: It is respected by most of the network mappings proposed by PIW. Bottleneck constraint: The percentage of *bad* elements is very low when using the PIW algorithm and the average number of broken demands is also small compared to SPR-P and SPR-CC.

Journal of Selected Areas in Communication, vol. 8, pp. 948–963, Aug. 1990.

[2] P. E. Green, "The Future of Fiber-Optic Computer Networks," *IEEE Computer*, Sept. 1991.

[3] G. Hill *et al.*, "A transport network layer based on optical network elements," *Journal of Lightwave Technology*, vol. 11, pp. 667–679, May 1993.

[4] R. Ramaswami, "Multiwavelength Lightwave Network for Computer Communication," *IEEE Communications Magazine*, pp. 78–88, Feb. 1993.

[5] O. Gerstel, R. Ramaswami, and G. Sasaki, "Fault Tolerant Multiwavelength Optical Rings with Limited Wavelength Conversion," in *Proc. of INFOCOM*, Apr. 1997.

[6] W. D. Grover, "Case Studies of Survivable Ring, Mesh and Mesh-Arc Hybrid Networks," in *Globecom'92*, vol. 1, pp. 633–638, Dec. 1992.

[7] J. Armitage, O. Crochat, and J.-Y. Le Boudec, "Design of a survivable WDM photonic network," in *Proc. of INFOCOM*, Apr. 1997.

[8] W. et al, "Survivability in a new pan-european carriers' carrier network based on wdm and sdh technology," *IEEE Communication Magazine*, 1999.

[9] W. Goralski, *SONET: A Guide to Synchronous Optical Networks*. McGraw-Hill, 1997.

[10] T.-H. Wu, *Fiber Network Service Survivability*. Artech House, 1992.

[11] W. Grover, T. Bilodieau, and B. Venables, "Near Optimal Spare Capacity Planning in a Mesh Restorable Network," in *Globecom'91*, pp. 2007–2012, 1991.

[12] K. Ishida, "Multiple Node-Disjoint Path Protocol for Virtual Paths in ATM Networks," in *Proc. of Joint workshop on Parallel and Distributed Real-Time Systems*, pp. 91–97, 1997.

[13] L. Nederlof *et al.*, "End-to-end survivable broadband networks," *IEEE Communication Magazine*, vol. 33, pp. 63–70, Sept. 1995.

[14] B. Gavish and I. Neuman, "Routing in a Network with Unreliable Components," *IEEE Transactions on Communications*, vol. 40, pp. 1248–1257, July 1992.

[15] M. Iwashita *et al.*, "Design of Highly Reliable Optical Fiber Cable Network in Access Networks," *IEICE Transactions on Communications*, pp. 1033–1042, July 1995.

[16] B. Mukherjee *et al.*, "Spare-Capacity-Based Fault-Tolerant Routing in Wavelength-Routed Optical Networks," in *Proc. of DIMACS Workshop on Optical Networks*, 1998.

[17] Y. Rekhter, "Cisco systems' tag switching architecture overview," February 1997. RFC 2105, IETF.

[18] D. A. et al., "Requirements for traffic engineering over mpls," Septembre 1999. RFC 2702, IETF.

[19] J. Slevinsky, W. Grover, and M. MacGregor, "An algorithm for survivable network design employing multiple self-healing rings," in *Globecom*, pp. 1568–1573, 1993.

[20] "Physical and networking layer issues of COBNET," Deliverable AC069/COB/WP1/DS/R/111/b1, COBNET Consortium, Mar. 1996.

	Capacity Constraint		Connectivity Constraint			
			Link Failure		Node Failure	
	Over vs Total (in %)	Average Over	Bad vs Total (in %)	Average Broken	Bad vs Total (in %)	Average Broken
SPR-P	16.7	2.7	28.6	3.8	35.7	3.3
SPR-CC	6.9	1.4	15.6	2.7	23.9	2.6
PIW	0.2	0.1	1.0	0.8	4.0	1.2

Table 4: Arbitrary logical topology over random network case. Capacity constraint: The values of Table 3 are repeated here to show that improvement concerning the connectivity constraint is not achieved by relaxing the capacity constraint. Connectivity constraint: The percentage of *bad* elements (which would cause disconnections in the higher level network protected groups) is very low for PIW compared to SPR-P and SPR-CC. There is also a strong reduction achieved by PIW concerning the average number of demands which cannot be restored by the failure of such a *bad* element.

[21] "Special Issue on Multiwavelength Optical Technology and Networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 14, June 1996.

[22] O. Crochat and J.-Y. Le Boudec, "Design Protection for WDM Optical Networks," *IEEE JSAC Special Issue on High-Capacity Optical Transport Networks*, Oct. 1998.

[23] B. Mukherjee *et al.*, "Some principles for designing a wide-area optical network," *IEEE/ACM Transactions on Networking*, vol. 4, pp. 684–696, Oct. 1996.

[24] R. Ramaswami and K. N. Sivarajan, "Design of Logical Topologies for Wavelength-Routed Optical Networks," *IEEE Journal on Selected Areas in Communications*, vol. 14, pp. 840–851, June 1996.

[25] D. Banerjee and B. Mukherjee, "A Practical Approach for Routing and Wavelength Assignment in Large Wavelength-Routed Optical Networks," *IEEE Journal on Selected Areas in Communication*, vol. 14, pp. 903–908, June 1996.

[26] S. Even, A. Itai, and A. Shamir, "On the Complexity of Timetable and Multicommodity Flow Problems," *SIAM Journal on Computing*, vol. 5, pp. 691–703, Dec. 1976.

[27] F. Glover, E. Taillard, and D. de Werra, "A user's guide for tabu search," *Annals of Operations Research*, no. 41, pp. 3–28, 1993.

[28] E. D. Taillard, *Recherches Iteratives Dirigees Paralleles*. PhD thesis, EPFL / DMA, 1993. Thesis Nr 1153.

[29] B. Carre, *Graphs and Networks*. Oxford University Press, 1979.

[30] N. Christofides, *Graph Theory: A algorithmic approach*. Academic Press, 1975.