A fault-tolerance solution to any set of failure scenarios on Dynamic WDM Networks with wavelength continuity constraints

Nicolas Jara, Hermann Pempelfort, Gerardo Rubino, Reinaldo Vallejos

A fault-tolerance solution to any set of failure scenarios on Dynamic WDM Networks with wavelength continuity constraints.

Nicolás Jara, Hermann Pempelfort, Gerardo Rubino and Reinaldo Vallejos

***Abstract*—Survivability of internet services is a significant and crucial challenge in designing optical networks. A robust infrastructure and transmission protocols are needed to maintain communications, despite the existence of one or more failed components on the system. Here, we present a generalized approach to tolerate any set of failure scenarios, to the extent the user can still communicate with the remaining components, where a scenario is an arbitrary set of links in a non-operational state.**

To assess the survivability problem, we propose a joint solution to the issues listed next: the set of primary routes, a collection of alternate routes associated to each failure scenario, and the capacity required on the network to allow communication between all users, in spite of any considered failure scenario, while satisfying for each user a specific predefined quality of service threshold, defined in the Service Level Agreement (SLA).

Numerical results show that the proposed approach not only enjoys the advantages of low complexity and ease of implementation, but it is also able to achieve significant resource savings compared to existing methods. The savings are higher than 30% on single link failures and more than a 100% on two simultaneous link failures scenarios or in more complex failure scenarios.

***Index Terms*—Dynamic WDM Optical Networks, Blocking Probability, Wavelength Continuity, Wavelength Dimensioning, Wavelength Assignment, Routing, Fault Tolerance.**

I. Introduction

A remarkable issue to be solved in designing WDM optical networks is to ensure that the network will still be able to provide its transmission service after the failure of one or more of its links. The solution to this problem consists in providing the necessary infrastructure to rapidly re-establish communications between all source-destination pair of nodes affected by these link failures. This type of mechanism is known as "Fault Tolerance".

The frequency of link failures occurrences is significant. For instance, [1], [2] reports measures of the mean time between failures of about 367 year/km. This failure frequency explains why failures on links may significantly impact the performance of the optical networks. For example, in a 26,000 km-long network such as NSFNet [3], there is an average of one fiber cut every 5 days. Moreover, the frequency with which two simultaneous links failures occur is high enough to be considered in the design process. In fact, Schupke [2] reported that the probability of two simultaneous failures occurring in a network like NSFNet is approximately 0.0027, implying a downtime of about 24 hours per year, on average, which in addition to the high transmission rate of this kind of networks, means an unacceptable data loss.

The previous elements justify the need to provide an efficient methodology for multiple fault tolerance, which should ensure (with a certain probabilistic guarantee) successful communications among all network users, despite the occurrence of failures in some of the links, and at the lowest possible cost regarding the network infrastructure. Note that node failure may be modeled as the failure of all the links connected to the node, so the general problem can be modeled as a set of link failures.

The fault tolerance methods proposed so far generally have been devoted to finding alternative paths considering single link failure (one bidirectional link), affecting all the users with routes passing through the failed link in both directions. Then, the number of wavelengths in the network is dimensioned to tolerate this situation [2], [4]–[8]. However, as already pointed out, the probability of occurrence of two or more simultaneous failures is high enough that it is needed to consider this kind of event in the design of the network. Some studies have focused on this 2-failures scenario [9]–[12]. Also, some studies have considered more complex cases of failures, such as Disaster risk constraints and Shared-Risk-Group scenarios. Disaster risk constraints [13]–[15] considers the possible service disruptions in case of a natural disaster or a targeted attack, in which case, the failures affect various links simultaneously. The case of Shared-Risk-Group (SRG) [16], [17] considers cases where some fibers are placed physically together, even if they are connecting different optical nodes. This situation makes those fibers liable to physical cuts since they can be cut together at the same time.

The previous discussion supports the need to produce an adequate strategy concerning multiple fault tolerance scenarios, which should ensure (with a certain probabilistic guarantee) successful communications among all network

N. Jara, H. Pempelfort and R. Vallejos are with the Department of Electronics in Universidad Tecnica Federico Santa Maria (UTFSM), Valparaíso, Chile, e-mail: nicolas.jara@usm.cl, hermann.pempelfort@usm.cl, reinaldo.vallejos@usm.cl

G. Rubino is with INRIA Rennes – Bretagne Atlantique, Rennes, France e-mail: gerardo.rubino@inria.fr

Manuscript received; revised.

users, notwithstanding the existence of failures in some of the links, with the lowest cost regarding the network infrastructure.

Here we propose a new fault-tolerance scheme, which we call the "Cheapest Shared Alternate Paths" method *CSAP*. In this approach, we go one step further concerning previous works, and we take into account the case of arbitrary sets of links failures scenarios, where a failure scenario is composed by a set of links in failure state. This means that we solve the fault-tolerance problem in its more general aspects.

The method also evaluates the number of wavelengths for each link of the network, ensuring that the blocking probability of any user request, of any user, is lower than a given corresponding predefined threshold (β_c), despite the possible occurrence of those simultaneous link failures. The value of β_c is defined on the Service Level Agreement (SLA), signed by the service providers and their clients, which defines the minimum quality of service (QoS) acceptable for each user, measured here as a probabilistic guarantee. The definition of these bounds is obtained considering objective criteria, such as: different quality of service requirements [18]–[20]; and subjective decisions, such as network scalability requirements. Based on these QoS agreements, engineers must design the network fulfilling said QoS requirements. Thus, we assume that the β_c values are given and acknowledge by the users and the network service providers.

The remainder of this paper is as follows. In Section III, we summarize the state of art of fault tolerance strategies. In Section III, we present the proposed method. In section IV we compare some results obtained by the proposed algorithm with those obtained with the current best methods in a set of different scenarios. Finally, the conclusions are given in Section V.

II. State of Art

Next, we briefly describe the most common methods currently used to provide fault tolerance in optical networks with wavelength continuity constraints.

One of the most frequent ways used to address simple and double fault tolerance, called "1+1", can be found in [5], [21], [22]. In this technique, a secondary route is associated with each primary one (with the restriction that they don't share any link), and the information is transmitted simultaneously through both of them, avoiding restoration delays in case of a failure. To dimension the number of wavelengths of each link -a task usually done by simulation-, each secondary route is considered as just another network route with a load equal to the corresponding primary one. The 1+1 method is also scalable to provide tolerance to $K \geq 1$ simultaneous failures. In this case, for each user, $K+1$ supplementary disjoint routes must be found, one as the primary route and the remaining K as secondary routes. Observe that a necessary and sufficient condition that allows this scheme to work is that the graph defined by the set of nodes and links is $(K+1)$-connected.

Another fault tolerance strategy is known as "*Shared Path Protection*" (SPP) [12], [23]–[25]. In this scheme, the extra resources (wavelengths) assigned to the secondary routes can be shared by different users, and are assigned only when a fault occurs. The SPP can be executed in two different ways. The first one consists of running the algorithm off-line, which means that the routes are calculated prior to the operation of the network (off-line SPP). The second way is the on-line implementation (on-line SPP). In this last case, the primary routes are computed before the network is operating, however, it must be executed again every time that one or more simultaneous failures occur, to compute alternate paths to the affected communications. For this reason, it is said that this is a proactive and reactive approach at the same time.

In [9], [10], [26]–[28] another method of fault tolerance called "*p-cycle*" is discussed, which provides survivability through fixed secondary routes that have a cyclic form. These cyclic routes are shared between several primary routes. One problem associated with this approach is that its applicability is very dependent on the size of the network, because it may introduce an excessive additional delay for a user in protection state on large networks. Also, to perform multiple fault tolerance, it requires a large number of cycles (e.g., hundreds of cycles for the 11 nodes pan-European COST 239 network [26]), which is impractical from various points of view.

III. The Proposed Fault Tolerance Method

A. Model

The network topology is represented by a graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, where $\mathcal{N}$ is the set of network nodes or vertices and $\mathcal{L}$ is the set of unidirectional links (the arcs in $\mathcal{G}$), with respective cardinalities $|\mathcal{N}| = N$ and $|\mathcal{L}| = L$. The set of users $\mathcal{X} \subseteq \mathcal{N}^2$, with cardinality $|\mathcal{X}| = X$, is composed by all the source-destination pairs with communication between them.

We use an ON-OFF model (as in [29]–[32]) to represent the traffic between a given source-destination pair. Consider user c. During any of its ON periods, whose average length is $t_{ON\,c}$, the source transmits at a constant rate (which is the rate associated with the wavelength). During an OFF period, with average length $t_{OFF\,c}$, the source refrains from transmitting data. Observe that we address here the general case where the load can be different for each user, the so-called heterogeneous situation.

The used technology determines the constant transmission rate during the ON periods, but to simplify the presentation, it is our rate unity. Consequently, the traffic load of user c, denoted by ϱ_c, is given by:

$$\varrho_c = \frac{t_{ON\,c}}{t_{ON\,c} + t_{OFF\,c}}. \tag{1}$$

Let $\mathcal{R} = \{r_c \mid c \in \mathcal{X}\}$ be the set of routes that enable communications among the different users, where r_c is the route associated with user $c \in \mathcal{X}$. The set $\mathcal{R}$ is known as

the set of *primary* routes, since this set alone does not offer any fault tolerance to the possible failure of network links.

Let $\mathcal{W} = \{W_\ell \mid \ell \in \mathcal{L}\}$ be the set containing the number of wavelengths of each unidirectional network link, where W_ℓ, $\ell \in \mathcal{L}$, is the number of wavelengths on link ℓ. The value W_ℓ, for every $\ell \in \mathcal{L}$, must be evaluated so that the blocking probability BP_c of each user $c \in \mathcal{X}$ is less than or equal to a given pre-specified threshold β_c, and the total number of available network wavelengths is as small as possible (saving resources).

Remark that the pre-defined threshold value β_c can be different for each network user, which means that we treat the general case where there are classes of users with different quality of services (QoS).

As in several works [31], [33], [34], in this proposal the total network cost C_{net} is defined as the sum of all wavelengths of all network links, that is, $C_{net} = \sum_{\ell \in \mathcal{L}} W_\ell$. Because we are considering fault tolerance capabilities, this cost must includes all the additional wavelengths needed to provide tolerance to the desired failures scenarios.

Let Ω be the set of every possible failure scenarios, where each scenario is a subset $\mathcal{F}$, with $\mathcal{F} \subset \mathcal{L}$, a set of links in failure state. The method explained below can be applied to any possible set of failure scenarios. For example, every possible single failure scenario ($|\mathcal{F}| = 1$); every possible double link failure scenario ($|\mathcal{F}| = 2$); a node failure, where all the links connected to that node are considered non-operational; disaster risk constraints [35], [36] where all the links affected by the disaster are considered non-operational; Shared-Risk-Group (SRG) [37], where $\mathcal{F}$ is composed by every link that can be affected by the same physical cut. Note that the previous examples consider all kind of failure scenarios already treated in the literature, however the method proposed here is applicable to any set of failure scenarios, with the condition that the network remains connected after any of the failure scenarios considered, which implies that the method can provide alternative routes for all affected users.

B. Definitions and sub-procedures needed by our method

Since the graph representing the network topology and the set of users are fixed data, as well as the upper bounds β_c, $\forall c \in \mathcal{X}$ (the maximum acceptable blocking probabilities of the users), we omit them in the list of the parameters of the procedures. For simplicity, when we refer to the network capacity, we write C_{net}, because we must change the capacities of the links many times during the computational process.

Some definitions required for the explanation of the method are presented in the following list:

- $\mathcal{G}_{-\mathcal{F}} = (\mathcal{N}, \mathcal{L} \setminus \mathcal{F})$, is the partial graph of $\mathcal{G}$ (same nodes, part of the edges), containing only the non-failed links, where $\mathcal{F}$ contains the set of failed links;
- $\mathcal{X}_\mathcal{F} = \{c \in \mathcal{X} \mid \ r_c \cap \mathcal{F} \neq \phi\}$,is the set of users c affected by the failure $\mathcal{F}$;
- $\mathcal{A}_\mathcal{F} = \{r_c \in \mathcal{R} \mid r_c \cap \mathcal{F} \neq \phi\}$, is the subset of the routes in $\mathcal{R}$ disables because of the failure $\mathcal{F}$;
- $\mathcal{R}_\mathcal{F}$ is a set of routes that replace those in $\mathcal{A}_\mathcal{F}$ when all links in $\mathcal{F}$ are failed;
- $\mathcal{S}_\mathcal{F}$ is the total set of routes guaranteeing fault tolerance to the failure event "all links in $\mathcal{F}$ fail". That is, the set defined by $\mathcal{S}_\mathcal{F} = \big(\mathcal{R} \setminus \mathcal{A}_\mathcal{F}\big) \cup \mathcal{R}_\mathcal{F}$;
- $\mathcal{C}_\mathcal{F} = \{\mathcal{C}_\ell \mid \text{ for all } \mathcal{L} \setminus \mathcal{F}\}$ is the costs (to be defined later) of each link non-affected by the failure $\mathcal{F}$.

The method also needs a few sub-procedures to work. They are described next.

- $PrimaryRoutes()$. A procedure that computes a set of primary routes. The selection of the routes can be made by any available technique, e.g., Dijkstra algorithm [38].
  To represent the execution of this sub-procedure, let us symbolically write $\{\mathcal{R}, \mathcal{W}\} := PrimaryRoutes()$
- $SecondaryRoutes()$. Considering that we have a set of failure links $\mathcal{F}$, the set of costs $\mathcal{C}_\mathcal{F}$, and a set of users $\mathcal{X}_\mathcal{F}$ affected by the failure of the links in $\mathcal{F}$. We want to find a new set of routes allowing to connect each user in $\mathcal{X}_\mathcal{F}$ despite the failure scenario $\mathcal{F}$, while still satisfying the QoS required by each user.
  The search for the new routes is done as follows. We run Dijkstra's algorithms looking, for each user $c \in \mathcal{X}$, the cheapest route, where the link costs are now given by the link costs in $\mathcal{C}_\mathcal{F}$ (explained later in the algorithm). This procedure creates a new set of routes, that we denote $\mathcal{R}_\mathcal{F}$.
  Symbolically, the execution of this sub-procedure is done by calling $\mathcal{R}_\mathcal{F} := SecondaryRoutes(\mathcal{X}_\mathcal{F}, \mathcal{F}, \mathcal{C}_\mathcal{F})$.
- $Dimensioning()$. This procedure consists in finding, for each link $\ell \in \mathcal{L}$, a capacity W_ℓ such that the end-to-end blocking probability BP_c of every user $c \in \mathcal{X}$ passing through the link ℓ is less than the given threshold β_c. For different reasons, the usual dimensioning procedures consider homogeneity in the links' capacities, that is, look for the minimum capacity $\mathcal{W}$, the same on all links, such that the performance objective is reached [39]–[41]. We will then follow here the same approach, because this can facilitate further comparisons with existing methods.
  The idea is simple: we are given the operational links of the networks, the set of routes $\mathcal{R}$ (because the procedure is used for a diverse set of routes), and the set of quality of service bounds β_c. We then initialize the network capacity W by value 1 and we evaluate the blocking probabilities per user; then, we check if the blocking probability of each user is less than the one defined on the SLA. If the condition is satisfied, we stop the algorithm. If not, we increase W by 1 and we repeat the procedure.
  Let us define $\mathcal{Q} \subseteq \mathcal{X}$, as the set of users with their QoS constraint satisfied (maximum acceptable blocking probability). Then, symbolically we evaluate the dimensioning sub-procedure by writing: $\{\mathcal{W}\} =$

```
function Dimensioning(L, R, β_c)
1   Q := φ;
2   foreach link ℓ
3      W_ℓ := 1;
4   do
5      BP_c := Blocking(G, R);
6      foreach user c ∉ Q
7         if BP_c ≤ β_c
8            Q := Q ∪ {c};
9      if Q ≢ X
10        for all ℓ ∈ L
11           W_ℓ := W_ℓ + 1;
12  until Q ≡ X
13 return W
```

Figure 1. Dimensioning procedure to compute the number of wavelengths on the network.

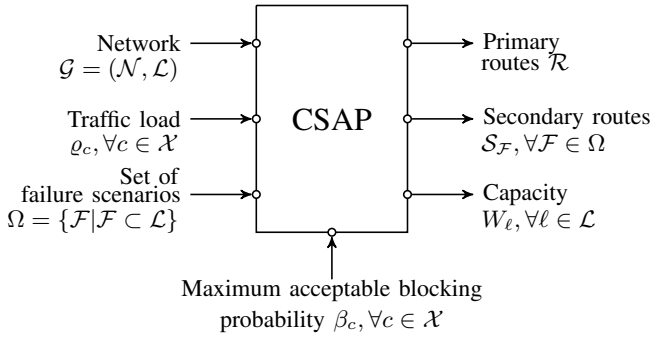

Figure 2. Diagram showing the inputs required to run the CSAP method, the condition to be guarantee, and the outputs delivered to solve the four problems jointly.

$Dimensioning(\mathcal{L}, \mathcal{R}, \beta_c)$. Figure 1 contains in pseudo-algorithmic form the procedure just described.

C. Fault Tolerance method

Figure 2 contains a diagram with the inputs required, the condition to be guaranteed, and the outputs obtained by the method execution.

The inputs are: the network topology $\mathcal{G} = (\mathcal{N}, \mathcal{L})$, which can be any network topology; each user traffic load ϱ_c, for all users in $c \in \mathcal{X}$ (notice that, the value ϱ_c of each user c can be different); and the set $\Omega = \{\mathcal{F} | \mathcal{F} \subset \mathcal{L}\}$ composed by all the link failure scenarios to be considered by the method execution.

The constraint to be satisfied by the method is to guarantee a given blocking probability β_c to each network user c, predefined on the Service Level Agreement (SLA).

The method's outputs are the set of primary routes $\mathcal{R}$, allowing to provide communication to each network user c, for all $c \in \mathcal{X}$, under the condition of no link failure; the set of alternative routes $\mathcal{S}_\mathcal{F}$, for each failure scenarios $\mathcal{F} \in \Omega$, which allow communicating in spite of the links in $\mathcal{F}$ are not operational; and the amount of wavelength W_ℓ necessary on each network link ℓ, for all $\ell \in \mathcal{L}$ (considering every possible failure scenario in Ω), fulfilling the QoS constraints to each user in spite of the failure occurrence of any scenario in Ω.

We use LIBPE method [30] to compute the users' blocking probabilities necessary to evaluate the quality of service offered to each user c. This procedure is an accurate and fast technique to evaluate the blocking probability of each user, on networks with wavelength continuity constraints. Note that fast evaluation of the QoS is significant, since solving the previously listed problems (the routing of the primary and secondary paths, with the corresponding dimensioning of each failure scenario), it is necessary to compute the blocking probability a lot of times (hundreds) considering all failure cases of the set Ω, and in each of these cases to execute the dimensioning procedure. Therefore, simulation techniques are not a possibility due to the time-consuming task involved.

Additionally, the method depends of the wavelength assignment scheme used during the network operation. These problem refers to the procedure to search for an available wavelength during network operation [33], [42]. The wavelength assignment problem has been widely covered in the literature [33], [41]–[43], and First-Fit is the most popular procedure in the literature since it performs better in terms of blocking probabilities, with low complexity. As a consequence, on our research we use this procedure to allocate the wavelengths.

In algorithmic form, the CSAP method is presented in Figure 3.

In **line 1**, by using the sub-procedure $PrimaryRoutes$, we use Dijkstra algorithm [38] to obtain an initial set of primary routes $\mathcal{R}$. However, it must be noted that the fault-tolerance mechanism presented here, is not associated to any particular routing decision, thus any routing method can be applied to obtain the primary routes.

Then in **line 2**, we include the set of every possible failure scenarios Ω, where each of these scenarios is a subset of failed network links $\mathcal{F}$. To explain how the procedure works, initially, assume that the only possible failure scenario is the simultaneous failures of all links in a specific subset $\mathcal{F}$ of $\mathcal{L}$.

In **lines 3 to 7**, we first start by finding replacement routes in case of the failure of all links in the subset of links $\mathcal{F}$. If a route r_c does not use any link of $\mathcal{F}$, it is not changed. However, for all users c whose route $r_c \in \mathcal{R}$ uses at least one link of $\mathcal{F}$ (that is, for all $c \in \mathcal{X}_\mathcal{F}$), we must find a new route that avoids the links of $\mathcal{F}$. To this end, for every link $\ell \in \mathcal{L} \setminus \mathcal{F}$, we define its cost $\mathcal{C}_\ell$ through the expression:

$$\mathcal{C}_\ell = \mathrm{e}^{\varrho_\ell - \overline{\varrho}}, \tag{2}$$

where ϱ_ℓ is the traffic load offered to the link ℓ by the users non-affected by the failed links, and $\overline{\varrho}$ is the mean traffic load on all the links ℓ, such that $\ell \in \mathcal{L} \setminus \mathcal{F}$ (the non affected links). Said cost function ($\mathcal{C}_\ell$) stands for one of many ways to represent how unbalance the traffic load is on the network,

```
function CSAP()
// --- input: the graph (the network), the users,
//     the bounds on the blocking probabilities,
//     and the set Ω of links failure scenarios,
//     where at most one of the events 'all links in F
//     fail simultaneously' occurs, all seen as global variables
// --- output: the primary routes, the secondary routes
//     and the wavelengths per link

// first compute the primary routes
1   R := PrimaryRoutes();

// calculates secondary paths in all failure scenarios.
2   foreach F in Ω
3       for all links ℓ ∈ L \ F
4           ϱ_ℓ := Σ_{c∈X\X_F∧ℓ∈r_c} ϱ_c; // non-affected routes
5       ϱ̄ := (Σ_{ℓ∈L\F} ϱ_ℓ) / L;
6       for all links ℓ ∈ L \ F
7           C_ℓ := e^{ϱ_ℓ − ϱ̄};
8       R_F := SecondaryRoutes(X_F, F, C_F); // compute alternative routes
9       S_F := (R \ A_F) ∪ R_F;
10      W_F := Dimensioning(L \ F, S_F);

// Decide the final wavelength dimensioning
11  for all links ℓ ∈ L
12      W_ℓ := Max(W_{ℓ,1}, ..., W_{ℓ,|Ω|})
13  return (R, S, W)
```

Figure 3. Algorithm for solving the fault tolerance problem, providing alternative routes to any failure scenario in Ω.

therefore seeking to balance the network traffic load, since balancing the network load may achieve remarkable savings by using network resources as even as possible [44], [45] Then, with these $\mathcal{C}_\ell$ values as weights, we run the Dijkstra's algorithm to find the cheapest route for each user $c \in \mathcal{X}_\mathcal{F}$. The set of all these routes are denoted by $\mathcal{R}_\mathcal{F}$. Symbolically, we execute the call: $\mathcal{R}_\mathcal{F} := SecondaryRoutes(\mathcal{X}_\mathcal{F}, \mathcal{F}, \mathcal{C}_\mathcal{F})$.

After that, **line 9** defines the set of routes $\mathcal{S}_\mathcal{F}$:

$$\mathcal{S}_\mathcal{F} = \left(\mathcal{R} \setminus \mathcal{A}_\mathcal{F}\right) \cup \mathcal{R}_\mathcal{F}.$$

In words, $\mathcal{S}_\mathcal{F}$ is the set of routes to be used when all links in $\mathcal{F}$ are failed. Under this condition, we must dimension the links again, because we must always respect the QoS constraints. For this purpose, we restrict the analysis to the graph $\mathcal{G}^{-\mathcal{F}}$, that is, we remove the links in $\mathcal{F}$ from $\mathcal{L}$. Then in **line 10**, we run a dimensioning phase. In pseudo-algorithmic form, we execute the function call $\mathcal{W}_\mathcal{F} := Dimensioning(\mathcal{L} \setminus \mathcal{F}, \mathcal{S}_\mathcal{F}, \{\beta_c, \forall c \in \mathcal{X} \setminus \mathcal{X}_\mathcal{F}\})$.

Repeating the steps explained above for each different failure scenario (**line 2 to 9**), we obtain a set of secondaries routes for each failure scenario $\mathcal{F}$ of Ω, and the corresponding links dimensioning for each failure scenario.

To finish, in **lines 11 to 12**, we compare each $W_{\mathcal{F},\ell}$, the number of wavelengths of link ℓ under each failure scenario $\mathcal{F}$, for all $\ell \in \mathcal{L}$, and the procedure determines the capacity of the link ℓ as the maximum between them. Formally, we add a procedure $Max()$, that performs this task. We symbolically write $W_\ell := Max(\mathcal{W}_{\ell,1}, ..., \mathcal{W}_{\ell,|\Omega|})$, where $\mathcal{W}_{\ell,\mathcal{F}}$ is the link ℓ capacity when failure scenario $\mathcal{F}$ takes place. Together each final link capacity W_ℓ, $\ell \in \mathcal{L}$ conform the final dimensioning set $\mathcal{W}$.

IV. Numerical Results

To quantify the quality of the CSAP method, the proposed solution should be compared against the optimal solution. However, it is known that the Routing and Wavelength Dimensioning (RWD) problem is an NP-complete problem [46]. In fact, those who solved this problem optimally only have been able to achieve it to very small networks (with less than 10 nodes) [47], [48]. Consequently, for real network topologies (dozens to hundreds of nodes), the fault-tolerance problem cannot be optimally solved, since it solves the RWD problem multiple times. Given this situation, our best alternative was to compare the CSAP method with those methods considered as the most competitive at this moment.

In order to make a comparison, the most important metrics on the survivability problem are the capacity of the network, and the delay in the restoration procedure in case of the

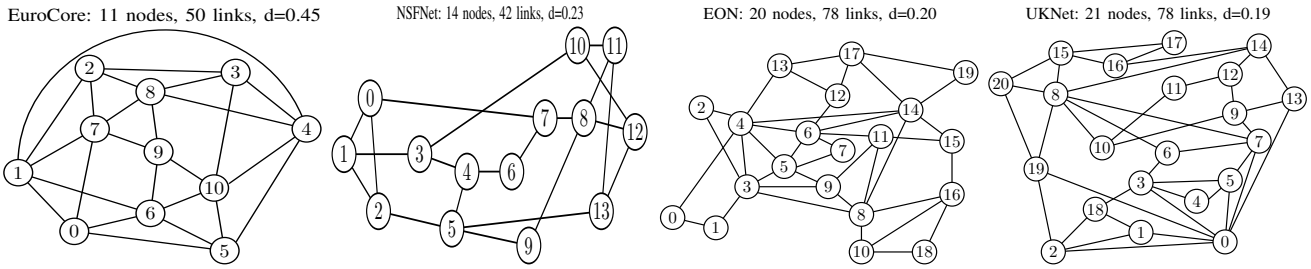

Figure 4. Some of the mesh networks evaluated. The number of links refers to the amount of bi-directional arcs. For instance, the picture shows the EON network topology with 39 edges, which corresponds to 78 arcs. The parameter d is a measure of density: if the graph has a arcs (twice the number of edges) and n nodes, $d = a/\Big(n(n-1)\Big)$.

occurrence of failures. Next, we analyze which are the most suitable methods to be compare with.

As mentioned in the introduction, there are several types of fault tolerance algorithms proposed so far, such as Shared Path Protection, p-Cycle, and 1+1. Hereafter, we discuss the pertinence in comparing CSAP with each of these different types of algorithms.

Shared Path Protection (SPP) Method. As discussed in this paper introduction, this strategy provides tolerance to multiple network links failure. There are two methods for implementing this algorithm (on-line and off-line). Both methods require between 40 to 80% of additional wavelengths (compared to the case without fault tolerance) to provide single link fault tolerance capability [23]. Another aspect that must be considered is that the SPP off-line method has the additional weakness that the percentage of restoration obtained (percentage of users that remain connected in case of link failure) is deficient (80% to 90% [23]), which means that it does not provide complete fault-tolerance to the network. Therefore, it is not a possible competitor to the method proposed in this work, which ensures that the blocking probability pre-established by the network designer is satisfied. On the other hand, the implementation of the SPP-online method requires to run on demand a route search algorithm (whenever one or more links fail) to find an alternative route to each affected user. Evidently, this on-line strategy causes a slow re-routing, which added to the fact that many of the applications that use computer networks require swift on-line responses in case of failures [49], which implies that this type of method does not represent a practical fault-tolerant mechanism for many practical applications. Due to the facts just commented, the SPP method was not considered for comparison.

The p-cycle Method. As discussed early, to provide tolerance to multiple failures, this method requires a large number of cycles (which implies a high cost when defining secondary routes), so it is not scalable for multiple faults. Given the fact that in this paper, we consider the multiple fault-tolerant cases, it is unreasonable to compare our method with the p-cycle one.

Method 1+1. This method provides tolerance to multiple failures, using as many disjoint routes as simultaneous link failures considered. It solves the problem of primary and secondary routes before the network dimensioning (off-line) sub-task. Then, the number of wavelengths is computed, having as a constraint to provide enough resources to all routes, and providing sufficient information to re-route each user in case of failure. Consequently, 1+1 is a suitable fault-tolerance method to compare with our algorithm.

In summary, the most appropriate methods for comparison is the 1+1 for the fault-tolerance mechanism. Additionally, reviewing current methods of Routing, we notice that the algorithms most commonly referenced today, and considered the best so far, use the shortest path, together with a First Fit wavelength assignment scheme. This is SP-FF (Shortest Path with First-Fit allocation scheme) [31], [33], [41], [43], [50]. Therefore, the routing strategy used on the 1+1 fault-tolerance method is the SPFF. Both methods together are denoted SPFF1+1 in the text.

To assess the blocking probabilities in both SPFF1+1 and CSAP strategies we use the mathematical method called LIBPE [30], and the final results were validated by simulation.

As previously discussed, the Wavelength Dimensioning method most commonly used nowadays is the homogeneous dimensioning, that is, all links have the same amount of wavelengths. Consequently, in this work, we consider a homogeneous dimensioning strategy on both fault-tolerance mechanisms.

To evaluate the performance of the methods under different scenarios, the algorithms were executed for different real network topologies, having different sizes and different degrees of connection d, where d is the average number of neighbors of a node in the network. Some of the selected topologies and their respective parameters N, L and d are shown in Figure 4.

The total network capacity C_{net} is one of the metric chosen to compare the algorithms, which is given by the total number of wavelengths necessary, to satisfy the users QoS constraints, including the primary and secondary routes needed on each different failure scenario $\mathcal{F} \in \Omega$. Thus, in Figure 5 we show the total cost C_{net} obtained by the CSAP and SPFF1+1 methods for the case of a single link failure, as a function of the traffic load, for different network topologies, and a maximum acceptable blocking user of 10^{-3}. In Figure 6 we show the C_{net} value for

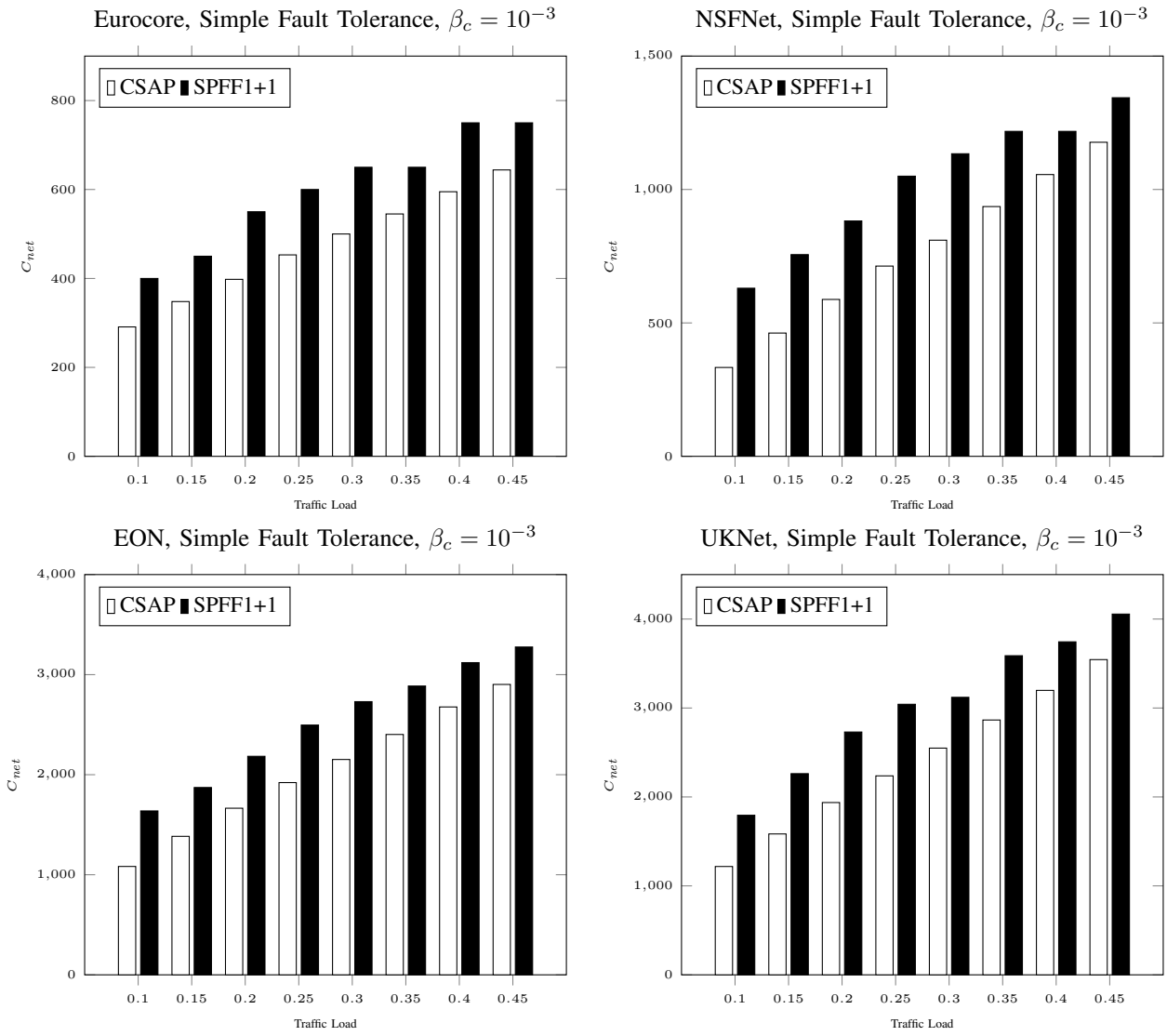

Figure 5. Total number of wavelengths C_{net} obtained with our method (CSAP) and SPFF1+1 on Eurocore, NSFNet, EON and UKNet real mesh network topologies, for different connection traffic loads, with a blocking probability threshold β_c equal to 10^{-3} in the single fault tolerance case.

the same methods, but double-link failures (i.e., any pair of simultaneous link failure possible). We show only single and double link failure scenarios. This is because to appropriately represent another kind of failures scenarios, such as SRG or Disaster Risk constraint may be hard to replicate and may achieve a disconnected network. However, as stated before, the algorithm developed can quickly evaluate any fault tolerance scenario.

Note that, for all the scenarios evaluated in our experiments for the case of single link failure, the SPFF1+1 method requires in general 30% more wavelengths (for $\varrho = 0.3$, which is a representative network load [49]) than the cost of the method proposed herein. Moreover, in the case of tolerance to two simultaneous failures of links (Figure 6), the CSAP method also significantly outperforms the SPFF1+1 technique. In this case, the SPFF1+1 method requires in the order of 160% more wavelengths (always for $\varrho = 0.3$ [49]) than our proposal.

Remark that for each scenario analyzed herein, both compared methods achieve to connect the same users with the same QoS requirements (maximum acceptable blocking probability), but our proposal requires significantly fewer resources than SPFF1+1 to do so.

To provide a more in-depth discussion of the results obtained by CSAP, next we present an analysis about the memory size the methods need, and the time required to access the memory during the network operation (Sub-Section IV-B and IV-C respectively).

A. Complexity Analysis

The total computational complexity of the CSAP method depends on the wavelength dimensioning algorithm used to compute the network links capacities, which in turn depends on the blocking probability evaluation techniques used. This

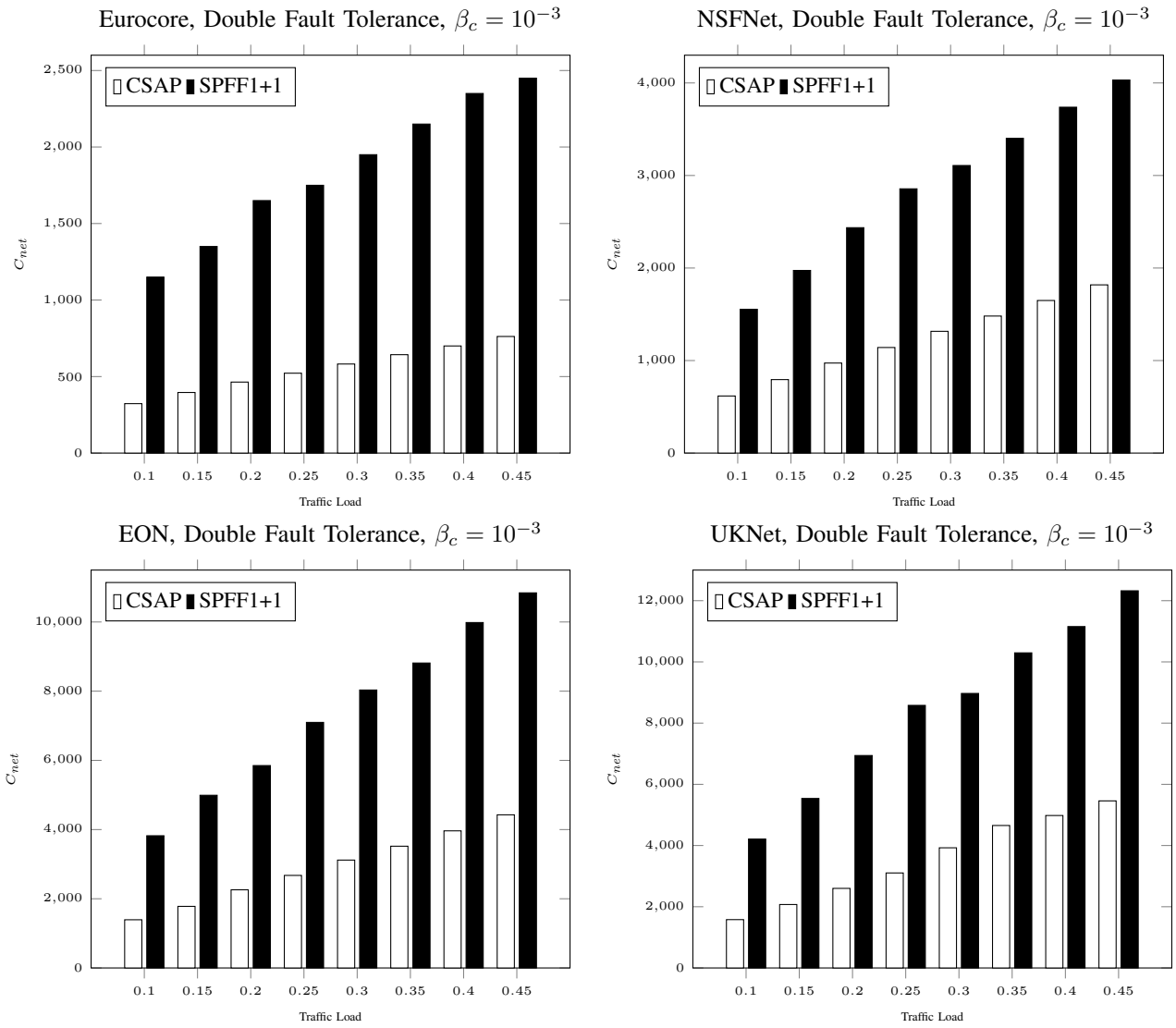

Figure 6. Total number of wavelengths C_{net} obtained with our method (CSAP) and SPFF1+1 on Eurocore, NSFNet, EON and UKNet real mesh network topologies, for different connection traffic loads, with a blocking probability threshold β_c equal to 10^{-3} in the simultaneous double fault tolerance case.

dependency is important since the dimensioning procedure is executed as many times as failure scenarios considered in the set Ω. Even more, the dimensioning algorithm executes several times the blocking probability procedure to calculate the network capacity. Therefore, the computational complexity of the proposed strategy is calculated in three stages: the Blocking probability evaluation, the wavelength dimensioning solution, and the survivability solution.

a) Blocking Probability: As mentioned in this work, we used the method LIBPE [30] to compute the blocking probability of each network user for a given network capacity W. LIBPE procedure is an iterative procedure, thus let the value I be the number of iterations that the method executes to converge in a solution, and $\overline{r}$ be the mean length of all the users path. The iterative solution executes sequentially:

- an update of the t_{OFF} values of all the network users $c \in \mathcal{X}$ (complexity $\mathcal{O}(X)$);
- per network link $\ell \in \mathcal{L}$ ($\mathcal{O}(L)$), the method solves the links blocking probability. It solves a Markov chain covering all the users passing through the given link, thus the complexity is given by the mean number of user per link $\mathcal{O}(\frac{X \cdot \overline{r}}{L})$. Then, the stage complexity is given by $\mathcal{O}(X \cdot \overline{r})$;
- and finally the method evaluates the blocking probability for all the users $c \in \mathcal{X}$, with complexity $\mathcal{O}(X \cdot \overline{r})$.

In a nutshell, the blocking probability then iterates I times executing the 3 sequential stages on all the wavelengths W, thus with a complexity $\mathcal{O}(I \cdot W \cdot (X + X \cdot \overline{r} + X \cdot \overline{r}))$. This leads to a computational complexity of

$$\mathcal{O}(I \cdot W \cdot X \cdot \overline{r}). \tag{3}$$

b) Wavelength Dimensioning: the computational complexity of the algorithm displayed in Algorithm 1 is as follows. From line 2 to 3 the complexity is $\mathcal{O}(L)$. Then, the iterative section starts from line 4 to 12, solving the blocking probability (complexity $\mathcal{O}(I \cdot W \cdot X \cdot \overline{r})$ IV-A0a); then checking for each user if they succeeded their blocking threshold $\mathcal{O}(X)$; and finally updating the links capacity ($\mathcal{O}(L)$). The iterative section is executed until the wavelength dimensioning is computed, thus W times. Consequently, the computational complexity together is ($\mathcal{O}(W(X + I \cdot W \cdot X \cdot \overline{r} + L)$, leading to

$$\mathcal{O}(W^2 \cdot I \cdot X \cdot \overline{r}). \tag{4}$$

For instance, in an Eurocore topology, the number of nodes is $N = 11$, $L = 100$ unidirectional links, $X = 110$ users, and the mean length of users' routes is $\overline{r} = 2$ using Dijkstra algorithm to compute the users' path. Now, for a mean traffic load $\varrho = 0.3$ scenario, the wavelength dimensioning obtain is $C_{net} = 400$, thus $W = 4$, and the number of iterations performed were $I = 5$. This leads us to 17600 instructions. In a common PC nowadays, the instructions are measured in MIPS (millions of instructions per second), leading to $1.76 \cdot 10^{-2}$ seconds. In LIBPE's article [30], the dimensioning procedure execution time was measured, obtaining for the exact same example $3.20 \cdot 10^{-2}$ seconds. Consequently, the computation complexity is well obtained.

c) CSAP method: the computational complexity of this method (shown in Algorithm 3) is then presented.

- In line 1, the primary routing problem is solved using Floyd-Warshall's (or Dijkstra's) algorithm, known to have an $O(N^3)$ computational complexity;
- from line 2 to 9 the secondary paths are computed. Be Z the number of failure scenarios in Ω, and F the maximum number of simultaneous links in failure state on a given scenario in Ω. We iterate for all the failure scenarios considered Z (line 2), computing the cost to all the operational links from line 3 to 6, with $\mathcal{O}(L \cdot \frac{X \cdot \overline{r}}{L}$ in line 3 to 4, $\mathcal{O}(L)$ in line 5, and $\mathcal{O}(L)$ in line 6 to 7. Later, the secondary routes are computed, executing Dijkstra ($\mathcal{O}(N^2)$ for each user affected by the failure scenario ($O(F \cdot \frac{X \cdot \overline{r}}{L})$). Finally, in line 9 the dimensioning is executed with the previously calculated complexity $\mathcal{O}(W^2 \cdot I \cdot X \cdot \overline{r})$.
- The last stage (line 10 to 11) computes the final wavelength dimensioning, comparing the dimensioning obtained on all the Z scenarios in Ω, then the complexity is $\mathcal{O}(L \cdot Z)$.

Summarizing, the complexity is given then by the sum of the 3 stages. This is:

$$\begin{aligned} &\mathcal{O}(N^3) + \\ &\mathcal{O}\left(Z\left[L \cdot \frac{X \cdot \overline{r}}{L} + L + L + N^2 \cdot F \cdot \frac{X \cdot \overline{r}}{L}\right]\right) + \\ &\mathcal{O}(Z \cdot W^2 \cdot I \cdot X \cdot \overline{r}) + \\ &+\mathcal{O}(L \cdot Z). \end{aligned} \tag{5}$$

Consequently, the final computation complexity of the complete CSAP procedure is as follows:

$$\mathcal{O}\left(Z \cdot F \cdot \frac{X \cdot \overline{r}}{L} \cdot N^2 + Z \cdot W^2 \cdot I \cdot X\right). \tag{6}$$

B. Memory size

Other aspects that influence the network performance are the storage size used by the routing tables, and the delay imposed by the routing procedure when each user attempts to transmit over a path.

The routing tables storage size depends on how many routes are computed to each user by the implemented procedure. If the 1+1 method provides fault tolerance to a single link failure, it computes only one secondary path for each user. Likewise, to offer fault tolerance to simultaneous double link failure, the 1+1 technique provides two secondary routes per user. Therefore, the number of entries stored on the routing tables are two and three times the number of users in $\mathcal{X}$, to provide single and double fault tolerance, respectively (with centralized management).

On our method, the number of paths computed changes based on the different failure scenarios and the network topology (size and node degree) evaluated. This situation occurs because, on each failure case, our method searches a new route to each user affected by the failed links on that scenario. In the executed experiments, our method required a similar number of alternate paths to provide single and fault tolerance than 1+1. For example, on the Eurocore network topology, to provide single and double fault tolerance, our method computed the same number of alternate paths than 1+1. Moreover, on a bigger network such as Arpanet, our methods required, on average, three and four paths per user to provide single and double fault tolerance, respectively.

C. Routing delay

During network operation, there is a delay incurred by the routing procedure, due to the time required to find the corresponding path and to transmit by it successfully, or to be finally blocked. We define this delay as $\tau(A)$, where A is the algorithm considered (SPFF1+1 or CSAP). Since both methods compared in this work use fixed predefined routes, the delay is mainly composed by the time needed to access the routing table and the corresponding transmission. Note that one access can be considered as a constant T, then, $\tau(A)$ measures how many times it is required to access the routing tables to have successful communication, or to be blocked, using the routing scheme obtained by the method A.

Note that both methods store the alternate paths in routing tables, but the technique to route each user on every communication request differs. The 1+1 fault tolerance scheme sends the information on each alternate path every time the user attempts to transmit; thus the access to routing tables requires to read two routes per user on single fault tolerance and three routes per user on simultaneous double fault tolerance. On the other hand, our method has only one route per link failure case; thus, it requires to read only one entry on the routing table on each attempt of transmission.

In a nutshell, the $\tau(A)$ value per method is:

- $\tau(SPFF1+1) = 2 \cdot T$, considering tolerance to single link failure.
- $\tau(SPFF1+1) = 3 \cdot T$, considering tolerance to simultaneous double link failure.
- $\tau(CSAP) = T$, for any link failure scenario.

showing the advantage of the CSAP method respect to the routing delay.

V. Conclusions

A novel method was proposed to solve the fault-tolerance problem for any possible set of scenarios, where each scenario is defined by a specific set of link failures.

The method differs considerably from those published so far, obtaining better results in terms of the necessary number of wavelengths and delay. Additionally, the dimensioning method does not make any distinction between primary and alternative routes, with the constraint that it only evaluates scenarios that may happen during the network operation (for each user, it considers either a primary or a secondary route, not both simultaneously). Consequently, the method allows sharing the resources between all the secondary routes, while guaranteeing a maximum blocking probability to each network user.

The proposed fault tolerance technique is scalable to any set of simultaneous link failures, as long as the network topology allows re-connection via the links that remain operational. This scheme is executed before the network operation, requiring a few seconds to solve the task. This fast execution also allows to quickly solve any link failure scenario during network operation if needed (for example, traffic load variations). Additionally, the network operation based on our approach is fast and straightforward, since the routes (both primary and secondary) are stored in routing tables and consulted only on demand.

Acknowledgments

This work received financial support from FONDEF ID14I20129, CONICYT and STICAMSUD 19STIC-01. These projects are then gratefully acknowledged.

References

[1] M. To and P. Neusy, "Unavailability analysis of long-haul networks," *Selected Areas in Communications, IEEE Journal on*, vol. 12, no. 1, pp. 100–109, 1994.

[2] D. A. Schupke, A. Autenrieth, and T. Fischer, "Survivability of Multiple Fiber Duct Failures," in *Third International Workshop on the Design of Reliable Communication Networks (DRCN*, pp. 7–10, 2001.

[3] R. Ramaswami and K. N. Sivarajan, "Design of logical topologies for wavelength-routed all-optical networks," in *INFOCOM '95. Fourteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Bringing Information to People. Proceedings. IEEE*, pp. 1316–1325 vol.3, 1995.

[4] S. S. Ahuja, S. Ramasubramanian, and M. M. Krunz, "Single-link failure detection in all-optical networks using monitoring cycles and paths," *IEEE/ACM Transactions on Networking*, 2009.

[5] H. Singh, J. Prakash, D. Arora, and A. Wason, "Fault Tolerant Congestion Based Algorithms in OBS Network," *International Journal of Engineering (IJE)*, vol. 5, no. 5, 2011.

[6] M. F. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee, "Design of disaster-resilient optical datacenter networks," *Journal of Lightwave Technology*, 2012.

[7] F. S. H. Souza, D. L. Guidoni, and G. R. Mateus, "A column generation-based heuristic for the GRWA with protection and QoS in WDM optical networks," in *Computers and Communications (ISCC), 2013 IEEE Symposium on*, pp. 922–927, 2013.

[8] C. Y. Chu, K. Xi, M. Luo, and H. J. Chao, "Congestion-aware single link failure recovery in hybrid SDN networks," in *Proceedings - IEEE INFOCOM*, 2015.

[9] D. S. Mukherjee, C. Assi, and A. Agarwal, "Alternate Strategies for Dual Failure Restoration Using p-Cycles," in *Communications, 2006. ICC '06. IEEE International Conference on*, vol. 6, pp. 2477–2482, 2006.

[10] R. Yadav, R. Yadav, and H. Singh, "Intercycle switching (ICS)-based dynamic reconfiguration of p-cycle for dual-failure survivability of WDM networks," *Photonic Network Communications*, vol. 24, no. 2, pp. 160–165, 2012.

[11] D. S. Yadav, S. Rana, and S. Prakash, "A mixed connection recovery strategy for surviving dual link failure in {WDM} networks," *Optical Fiber Technology*, vol. 19, no. 2, pp. 154–161, 2013.

[12] M. Jinno, T. Takagi, and Y. Uemura, "Enhanced survivability of translucent elastic optical network employing shared protection with fallback," in *2017 Optical Fiber Communications Conference and Exhibition (OFC)*, pp. 1–3, March 2017.

[13] D. Serre and C. Heinzlef, "Assessing and mapping urban resilience to floods with respect to cascading effects through critical infrastructure networks," *International Journal of Disaster Risk Reduction*, 2018.

[14] F. Dikbiyik, M. Tornatore, and B. Mukherjee, "Minimizing the risk from disaster failures in optical backbone networks," *Journal of Lightwave Technology*, 2014.

[15] S. Ferdousi, F. Dikbiyik, M. F. Habib, M. Tornatore, and B. Mukherjee, "Disaster-aware dynamic content placement in optical cloud networks," in *Conference on Optical Fiber Communication, Technical Digest Series*, 2014.

[16] X. Shao, Y. Bai, X. Cheng, Y.-K. Yeo, L. Zhou, and L. H. Ngoh, "Best Effort SRLG Failure Protection for Optical WDM Networks," *Journal of Optical Communications and Networking*, 2011.

[17] P. Babarczi, J. Tapolcai, P. H. Ho, and M. Medard, "Optimal dedicated protection approach to shared risk link group failures using network coding," in *IEEE International Conference on Communications*, 2012.

[18] W. Liao and C. H. Loi, "Providing service differentiation for optical-burst-switched networks," *Journal of Lightwave Technology*, vol. 22, no. 7, pp. 1651–1660, 2004.

[19] D. H. Hailu, G. G. Lema, E. A. Yekun, and S. H. Kebede, "Unified study of Quality of Service (QoS) in OPS/OBS networks," *Optical Fiber Technology*, vol. 36, pp. 394–402, jul 2017.

[20] S. Mohd Sam, S. Mohd Daud, K. Kamardin, and N. Maarop, "Study of Qos Performance in Optical Burst Switched Networks (OBS)," *Indian Journal of Science and Technology*, vol. 9, dec 2016.

[21] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *Lightwave Technology, Journal of*, vol. 21, no. 4, pp. 870–883, 2003.

[22] M. Wang, S. Li, E. W. M. Wong, and M. Zukerman, "Performance Analysis of Circuit Switched Multi-Service Multi-Rate Networks With Alternative Routing," *Journal of Lightwave Technology*, vol. 32, no. 2, pp. 179–200, 2014.

[23] D. Schupke and R. Prinz, "Capacity Efficiency and Restorability of Path Protection and Rerouting in WDM Networks Subject to Dual

Failures," *Photonic Network Communications*, vol. 8, no. 2, pp. 191–207, 2004.

[24] A. Wason and R. S. Kaler, "Fault-tolerant routing and wavelength assignment algorithm for multiple link failures in wavelength-routed all-optical {WDM} networks," *Optik - International Journal for Light and Electron Optics*, vol. 122, no. 2, pp. 110–113, 2011.

[25] D. Pereira and M. Camillo Penna, "A new algorithm for dimensioning resilient optical networks for shared-mesh protection against multiple link failures," *Optical Switching and Networking*, 2014.

[26] D. A. Schupke, "Multiple failure survivability in WDM networks with p-cycles," in *Circuits and Systems, 2003. ISCAS '03. Proceedings of the 2003 International Symposium on*, vol. 3, pp. III–866–III–869 vol.3, may 2003.

[27] L. Tang, M. Cai, B. Li, and R. Wu, "A novel multi-link fault-tolerant algorithm for survivability in multi-domain optical networks," *Photonic Network Communications*, vol. 24, no. 2, pp. 77–85, 2012.

[28] F. Ji, X. Chen, W. Lu, J. J. Rodrigues, and Z. Zhu, "Dynamic p-cycle configuration in spectrum-sliced elastic optical networks," in *GLOBECOM - IEEE Global Telecommunications Conference*, 2013.

[29] N. Jara, H. Pempelfort, G. Rubino, and R. Vallejos, "How much the wavelength dimensioning methods and a tightened qos provision impact on the dynamic wdm optical networks capacity?," *Optical Switching and Networking*, vol. 35, p. 100540, 2020.

[30] N. Jara, R. Vallejos, and G. Rubino, "Blocking Evaluation and Wavelength Dimensioning of Dynamic WDM Networks Without Wavelength Conversion," *Journal of Optical Communications and Networking*, vol. 9, no. 8, p. 625, 2017.

[31] A. Zapata-Beghelli and P. Bayvel, "Dynamic Versus Static Wavelength-Routed Optical Networks," *Lightwave Technology, Journal of*, vol. 26, pp. 3403–3415, oct 2008.

[32] M. Zukerman, E. W. M. Wong, Z. Rosberg, G. M. Lee, and H. L. Vu, "On teletraffic applications to {OBS}," *Communications Letters, IEEE*, vol. 8, pp. 116–118, feb 2004.

[33] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical Networks: A Practical Perspective, 3rd Edition*. San Francisco, CA, USA: Morgan Kaufmann Publishers Inc., 3rd ed., 2009.

[34] N. Jara, R. Vallejos, and G. Rubino, "A method for joint routing, wavelength dimensioning and fault tolerance for any set of simultaneous failures on dynamic WDM optical networks," *Optical Fiber Technology*, vol. 38, pp. 30–40, 2017.

[35] F. Dikbiyik, A. S. Reaz, M. De Leenheer, and B. Mukherjee, "Minimizing the disaster risk in optical telecom networks," in *Optical Fiber Communication Conference*, pp. OTh4B—-2, Optical Society of America, 2012.

[36] S. Ferdousi, F. Dikhiyik, M. F. Habib, and B. Mukherjee, "Disaster-aware data-center and content placement in cloud networks," in *2013 IEEE International Conference on Advanced Networks and Telecommunications Systems (ANTS)*, pp. 1–3, IEEE, 2013.

[37] H. Zang, C. Ou, and B. Mukherjee, "Path-protection routing and wavelength assignment in WDM mesh networks under shared-risk-group constraints," in *Asia-Pacific Optical and Wireless Communications Conference and Exhibit*, pp. 49–60, International Society for Optics and Photonics, 2001.

[38] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.

[39] X. Zhang, S.-i. K. S.-i. Kim, and S. Lumetta, "Dimensioning WDM Networks for Dynamic Routing of Evolving Traffic," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 2, no. 9, pp. 730–744, 2010.

[40] L. Tan, Q. Yang, J. Ma, and S. Jiang, "Wavelength Dimensioning of Optical Transport Networks Over Nongeosychronous Satellite Constellations," *Journal of Optical Communications and Networking*, vol. 2, no. 4, pp. 166–174, 2010.

[41] R. T. Koganti and D. Sidhu, "Analysis of routing and wavelength assignment in large WDM networks," in *Procedia Computer Science*, vol. 34, pp. 71–78, 2014.

[42] B. Mukherjee, *Optical WDM networks*, vol. 26. Springer Science & Business Media, 2006.

[43] B. C. Chatterjee, N. Sarma, and P. Pratim Sahu, "Review and Performance Analysis on Routing and Wavelength Assignment Approaches for Optical Networks.," *IETE Technical Review*, vol. 30, no. 1, pp. 12–23, 2013.

[44] A. A. Neghabi, N. J. Navimipour, M. Hosseinzadeh, and A. Rezaee, "Load Balancing Mechanisms in the Software Defined Networks: A Systematic and Comprehensive Review of the Literature," 2018.

[45] R. Vallejos and N. Jara, "Join routing and dimensioning heuristic for dynamic WDM optical mesh networks with wavelength conversion," *Optical Fiber Technology*, vol. 20, no. 3, 2014.

[46] V. López and L. Velasco, eds., *Elastic Optical Networks*. Optical Networks, Cham: Springer International Publishing, 2016.

[47] C. Meza, N. Jara, V. M. Albornoz, and R. Vallejos, "Routing and spectrum assignment for elastic, static, and without conversion optical networks with ring topology," in *2016 35th International Conference of the Chilean Computer Science Society (SCCC)*, pp. 1–8, oct 2016.

[48] R. Vallejos, A. Zapata-Beghelli, V. Albornoz, and M. Tarifeño, "Joint routing and dimensioning of optical burst switching networks," *Photonic Network Communications*, vol. 17, no. 3, pp. 266–276, 2009.

[49] A. A. M. Saleh and J. M. Simmons, "Technology and architecture to enable the explosive growth of the internet," *Communications Magazine, IEEE*, vol. 49, no. 1, pp. 126–132, 2011.

[50] N. Charbonneau and V. M. Vokkarane, "A survey of advance reservation routing and wavelength assignment in wavelength-routed WDM networks," *IEEE Communications Surveys and Tutorials*, vol. 14, no. 4, pp. 1037–1064, 2012.