A new algorithm for dimensioning resilient optical networks for shared-mesh protection against multiple link failures

Dyson Pereira Junior\textsuperscript{a,}\textsuperscript{*}, Manoel Camillo Penna\textsuperscript{b}

\textsuperscript{a} Department of Electronics, Federal Technological University of Parana, Curitiba, Brazil
\textsuperscript{b} Post-Graduate Program in Computer Science, Pontifical Catholic University of Parana, Curitiba, Brazil

\textbf{A B S T R A C T}

Failures of fiber links can result in major loss of data in high-speed optical communication networks. Survivability is of critical importance, making high levels of availability essential, given the increased level of infrastructure vulnerability to natural disasters, massive power failures, and malicious attacks. A typical approach to the design of resilient optical networks is through protection schemes that predetermine and reserve protection resources based on single and double link-failure scenarios. In this paper we propose a planning heuristic for WDM networks that computes the resource capacity required to transport the traffic demand and protect the optical connections while meeting availability requirements in scenarios of multiple link failures. The method is based on two algorithms, one for path-selection and the other for computing connection unavailability. The numerical results show that the method allows network topology to be exploited to significantly reduce connection unavailability.

\& 2014 The Authors. Published by Elsevier B.V. This is an open access article under the CC BY-NC-ND license (http://creativecommons.org/licenses/by-nc-nd/3.0/).

1. Introduction

The development of optical networks is being directed toward the provision of high transmission capacity at costs that allow these networks to accommodate the large increase in bandwidth requirements of new applications. This development has been driven by two main factors: the advent of WDM (Wavelength-Division Multiplexing) and significant advances in optical component technology [1]. Advances in optical component technology have made it possible to design all-optical WDM transport networks in which all the control functions, including intelligent switching and routing, are handled only in the optical domain [2,3].

To establish a light-path, its route must be found and a wavelength assigned to each link traversed by the path. Reconfigurable optical multiplexers have been used in WDM networks, providing flexible configuration by allowing WDM channels to be added and dropped without manual configuration. More recently, two major limitations have been addressed by the introduction of colorless and directionless optical add/drop multiplexers, where add/drop ports are not wavelength specific and the outbound direction of added signals is no longer limited [4,5]. The failure of a fiber link results in interruption of all light-paths using the link [6]. The use of recovery mechanisms in the optical layer has several advantages [7]: this layer can efficiently multiplex protection resources (such as wavelength and fiber) between the various higher-layer applications; and resilience in the optical layer provides protection for higher-layer protocols that may not have internal protection.

The ability to continue operating in the event of failures is known as resilience [8,9]. A classification of resilience in transparent optical networks and the main techniques used to achieve it can be found in [10–16]. In particular, there is...
In general, resilience can be classified according to the survival strategies and topologies used. One criterion that influences protection strategies relates to the components that will be included in the strategy. Nodes and links can be included, but for the former, resilience strategies usually involve only local structures and are therefore simpler (with multiple redundancies). For this reason most research studies consider strategies that only involve link resilience. Another criterion is the range of resilience mechanisms, which may involve paths or links (or segments). The former, which is known as end-to-end protection, tends to use the least resources, while the latter, which is also known as local protection, tends to provide faster recovery times. A further criterion influencing resilience strategies is whether to allow sharing of protection resources or not. In dedicated protection, protection resources are not shareable whereas in shared protection they are. Our study does not consider the protection of nodes, for reasons already mentioned, but is based on the use of a shared end-to-end strategy.

The timing of actions related to resilience also produces different strategies. Three key moments can be considered: the time when the service path is routed, the time when the protection path is routed and the time when the protection path is activated. The first of these leads to two possible strategies, i.e., routing the service path on demand (online) or a priori (offline). The second strategy, which requires prior knowledge of the demands, is best suited to planning algorithms and will be used in this study. The time when the protection path is routed also leads to two distinct strategies: routing the protection path along with the service path or only when a failure occurs. In the literature the first of these strategies is referred to as protection and the second as restoration [8]. Finally, the time when the protection path is activated also leads to two distinct strategies: a strategy in which the protection path is activated when it is routed or one in which it is activated when the main path fails.

The sharing topology used also leads to different resilience strategies. Two such strategies can be found in the literature [18–23]. In the first, a set of service paths with the same origin and destination are protected by a set of protection paths. In the second, service paths with different origins and destinations may share one or more links. The second strategy is known as shared-mesh and tends to consume fewer protection resources at the expense of greater complexity.

A final criterion that greatly influences resilience strategies is whether the availability of connections should be considered or not. Because availability is one of the most important indicators in service level agreements, availability-aware resilience strategies aim to achieve predefined levels of availability. Such strategies in turn are distinguished from each other by the number of simultaneous failures they are able to cater for. Because of the complexity involved, most studies only take into account one or two simultaneous failures.

In this study we investigate a resilience strategy to determine network capacity requirements in terms of wavelength allocation and light-path routing in a shared-mesh protection scheme with full wavelength conversion. We propose a planning heuristic that assumes that the demands are known a priori and takes into account connection availability requirements while catering for multiple simultaneous failures.

### 1.1. Contribution

We present a new heuristic for planning WDM networks called multiple shared backups (MSBs). Given the physical topology and traffic demand between any two nodes, the proposed algorithm simultaneously configures connections (routing of service and protection paths) and allocates and minimizes resources (wavelengths) for each demand.

In the model considered, the WDM network is defined by the physical topology, which consists of links and nodes organized in a graph, and the logical topology, which is defined by the set of configured light-paths. A link is a cable with multiple fibers, and in each (bidirectional) link some fibers are used in one direction and others (not necessarily the same number) in the opposite direction. The geographic location of the nodes and the lengths of the physical links are known. Each fiber carries a given number of wavelengths, and we assume for simplicity that all wavelengths in the network transmit data at the same bit rate. The physical capacity of a link is equivalent to the number of wavelengths that it supports.

While the physical topology is known, the capacity of each link is a variable of the problem. Nodes are optical cross-connects (OXCs), considered simultaneously as sources and destinations of traffic. A connection is defined as a unit demand between a pair of source and destination nodes. A unit demand corresponds to the capacity of a wavelength (by definition). Connections are protected services, implemented by a set of light-paths, of which one is the service path and the others are protection paths. More than one connection is needed when the required bandwidth between a pair of nodes exceeds the capacity of a wavelength.

The objective of planning is to enable all necessary connections to meet the demand with the required level of availability. The solution is found when all connections are set up, i.e., when all the service light-paths are routed and their wavelengths assigned. The capacity of the physical links is sized to accommodate the light-paths of all the connections. To achieve the availability requirements the network planning method (MSB) includes an end-to-end shared-mesh resilience scheme that takes into account the requirements described above for a scenario with multiple simultaneous failures.

### 2. Related work

The approaches discussed in [18–24], as well as the majority of studies discussed in the literature, make the simplifying assumption that a maximum of two simultaneous link failures occur. However, some studies suggest
resilience schemes that consider the possibility of multiple failures. This type of failure is dealt with in the following studies using shared end-to-end resilience schemes in dynamic routing scenarios (online).

In a study of multiple link failure recovery by Cheng et al. [25], one service path and $F$ protection paths (link disjoint) are chosen on the arrival of a new connection request in order to protect the network against multiple link failures. The paths are selected from a list of routes pre-calculated so as to minimize the additional bandwidth. When routing protection paths, every link cost is updated according to the required bandwidth. The algorithm goes through the list of pre-calculated paths by choosing the one needing the least additional capacity.

Cheng et al. [26] propose two heuristics for calculating routes and allocating bandwidth for a protection scheme in which the service and protection paths are determined from the $k$ shortest paths. In the first heuristic, the paths are calculated so as to consume as little as possible of the total bandwidth, while in the second the goal is to satisfy the path-length constraints. The algorithm is triggered by the arrival of a new connection request with a known bandwidth requirement. It starts looking for the shortest paths (disjoint or partially disjoint, i.e., two or more paths that share at least one link) between the source and the destination of the connection from a table of pre-computed routes. The additional bandwidth requirement for the service-path links is equal to the bandwidth computed routes. The additional bandwidth for service paths is calculated by simulating all possible failures and increasing the capacity reserved for protection in the corresponding links.

Wason and Kaler [27] describe two algorithms, one for routing and another for allocating wavelengths. In the routing algorithm, all possible paths are calculated for a given pair of (source and destination) nodes and arranged in order (side by side in a list) according to their length, from shortest to longest. Link failures are identified and checked on all paths. If a failure does not affect the path, then the path is selected; if it does, then the path is discarded and the next path in order of preference is checked. This process is repeated until a path that is resilient in terms of the maximum number of faults is established.

Guo et al. [28] propose a recovery mechanism with escalation for establishing routing and protection. The service path is calculated based on the shortest path and network load balancing. To avoid unbalanced use of links, the link cost is updated according to the criterion that if there are no free wavelengths on the link, the cost is set to a high value. Otherwise, the current cost is reduced by a proportion equal to the ratio of the number of wavelengths used in the fiber to the maximum number of wavelengths in the fiber.

As noted earlier, studies that take into account the availability of connections also use the simplifying assumption of no more than two simultaneous link failures. However, these studies consider online protection, i.e., search requests for service path arrive one at a time to the source node. The routing decision is made without knowledge of future requests. Diaz et al. [17] try to achieve a balance between strategies that do not incorporate a priori link risk information. Given a request for a connection between two nodes (online protection), the algorithm first prunes all non-feasible links in order to improve blocking performance removing all links with insufficient capacity. Next, from the list of $k$-shortest paths, service paths are computed between the source nodes and the destination. In particular, a load balancing approach is applied by defining dynamic link weights as follows, i.e., more congested links have higher costs. After the service paths are computed, each is processed in isolation to compute a link-disjoint protection path, by running Dijkstra’s shortest path algorithm. This step yields $k$ service/protection path pairs, from which the final pair is selected based on the lowest risk. Namely, this is done by computing a path failure probability for each service path. Using the above, the path pair with the minimum joint path failure probability is chosen, i.e., computed as a sum of service and protection path failure probabilities (products) across all failure events.

3. Problem statement

There is much interest in handling multiple failures arising from large-scale stressors such as natural disasters, massive power outages, and malicious (WDM) attacks [17]. On the other hand, a resource sharing enables a significant reduction in availability for protected connections to $F_{\text{max}}$ simultaneous failures.

In our work, traffic demand will be met by a set of light-path with the same origin and the destination. Each connection has a service light-path and a set of $F_{\text{max}}$ protection lightpaths, where $F_{\text{max}}$ corresponds to the number of simultaneous failures to be considered. Thus, a connection is implemented by $F_{\text{max}}+1$ light-paths arranged in an activation sequence, where the first is the service light-path and the others are protection lightpaths. When a link failure affecting the active light-path occurs, the source node is responsible for switching to the next light-path in the activation sequence. The allocation of routes achieved by the PS algorithm allows sharing of resources only in a network with full wavelength conversion. The network planning method proposed here is called multiple shared backups (MSBs).

The MSB method involves two algorithms, one for selecting the light-paths (Path Selection – PS) and another for calculating the unavailability of connections (Computing Connection Unavailability – CCU). From a set of $k$ shortest paths, the first algorithm selects, for each connection, the supporting light-paths and the order of activation. The second algorithm uses the method proposed by Mello et al. [19], extended to cater for multiple failures, to calculate the availability of each connection. The results of both algorithms are used by a planning procedure to meet the demand between all pairs of nodes, thus ensuring a pre-defined level of availability while using the fewest resources.

The decisions of the path-selection algorithm take into account the failure states of the network. In order to minimize the resources needed, the path-selection algorithm attempts to maintain the load balancing on the
network. To achieve this, it assumes that when a link failure occurs, the traffic stopped in each light-path is switched to the next protection light-path. If this is not possible, the traffic connection is interrupted until a link that can make the connection operational again is recovered. For a light-path to be successfully switched, there must be sufficient protection resources on the links traversed by the protection light-path. The unavailability of a connection is given by the sum of the probabilities of failure sequences for which there is an unsuccessful recovery attempt.

The load-balancing procedure uses sets of failure states from zero to \( F_{\text{max}} \). The first set contains only the state with no failure, the second contains the states with one failure, the third contains the sequences of states with two failures and so on. When applied to the first set, the load-balancing procedure defines the service light-path; when applied to the second, it defines the first protection light-path; and so on. Load balancing is achieved by evaluating all possible combinations of link failures involving a set of alternative light-paths and choosing the one that reduces the difference between the load on the most loaded link and the load on the least loaded one. As analysis of all possible combinations leads to a combinatorial explosion, our method uses a heuristic to address this issue. The load-balancing procedure is first applied to light-paths in the set of the shortest paths for connection \( c \). When evaluating other sets, each connection will have an active light-path and a set of candidate protection light-paths. In this case, balance is achieved by evaluating all possible combinations of protection light-paths for each state. When an active light-path is interrupted in more than one state, a protection path for each state would be ideal. This would lead to optimal load balancing in each state. To achieve a better load balancing most of the time with just a path protection, the procedure uses only the state with the highest probability.

The algorithm to compute connection unavailability assumes that the failure rates and recovery rates of the links are known. As shown by Zhang et al. [11], the time to failure and time to repair are exponentially distributed and there is no dependency between link failures. The probability of each sequence of failures is calculated by extending the method proposed by Mello et al. [19] for multiple link failures. Each failure sequence defines a state. The states of a same subset are represented by failure sequences with differences only in the last link. The algorithm performs the calculation by covering all states in a subset at a time. A new state is considered by appending the faulty link to the failure sequence of the last visited state. This allows information computed in previous states to be saved and reused (for example, the current active light-paths and the remaining reserved capacity in all links). This information in turn allows reserved capacity to be assigned to the next protection light-path. Hence, the availability of resources after any failure can be analyzed and the protection resources that allow the network to return to one of the previous states can be recovered. The details of the algorithm will be provided in the next section.

Connections compete for protection resources (used by best-effort traffic) on an equal basis. Some recovery attempts may not succeed, making the corresponding connection unavailable. This situation is modeled by a rejection factor, which is computed for each connection for every state in which it is unable to recover. The probability that a connection will become unavailable in a particular state is given by the product of the probability of the state occurring and the rejection factor in that state. The unavailability of a connection is obtained by adding up the probability of its becoming unavailable in each state.

The aim of the MSB method is to plan the network by selecting the minimum necessary number of light-paths for each connection so that downtime does not exceed a preset value. The result is the minimum number of light-paths capable of meeting the requirements. MSB has two stages. In the first stage, the PS algorithm is executed using the assumption that the links have unlimited capacity, i.e., for every state there are enough available resources to switch to a protection light-path without resource sharing (at this stage, some connections are overprotected, i.e., with more protection light-paths than is necessary). The first stage is interactive and at the end of each cycle MSB checks for over-protected connections (by using the CCU algorithm) reducing the number of protection light-paths by one (reducing by one the number of wavelengths on each link traversed by the deleted light-path). At the end of the first stage, the unavailability of these connections will be higher than required (under-protected). A new adjustment is then made (second stage), in which the number of protection light-paths for under-protected connections is increased by one. The PS algorithm and the CCU algorithm are run one last time with the final configuration (without increasing the capacity of the links). Resource sharing is allowed and the connection configuration and link capacities produced in the last cycle of the first stage are used.

4. Formalization and detailing of the algorithms

In this section we summarize the new heuristic for planning WDM networks called multiple shared backups (MSBs). In Section 4.1, we demonstrate the operation of the PS algorithm, the procedure of load balancing, an application example on a small network and the complexity of the algorithm. In Section 4.2, we demonstrate the operation of the CCU algorithm, the use of Markov chain to calculate the unavailability and the complexity of the algorithm. In Section 4.3, we show the network planning method (MSB) and runtime for different network sizes.

4.1. Path-selection algorithm

The input to the path-selection (PS) algorithm is a list of candidate light-paths for each connection organized in subsets, the first containing disjoint candidate light-paths, and the others containing partially disjoint candidate light-paths. Let \( C \) be the set of all connections necessary to meet all demands. \( L_c^k \) is a list that contains the set of \( k \) shortest paths for connection \( c \) (with candidate disjoint paths and partially disjoint paths). From each list \( L_c^k \), the goal is to create a list \( L_c^{k'} \) which contains the set of candidate light-paths to protect the connection \( c \), arranged
in subsets of light-paths. Let $L_p = \{L_p^c | c \in C\}$. $L_p$ is the input to the PS algorithm. Subset $L_p^{c_1}(1)$ contains the shortest light-paths that are link disjoint while the other subsets contain light-paths that are partially link disjoint in relation to the light-paths included in $L_p^{c_1}(1)$. To build $L_p^{c_1}(1), L_p^{c_1}$ is ordered from the shortest to the longest light-path and the smallest light-path in $L_p^{c_1}$ that has at least one disjoint light-path ($c_{11}$) is identified, $L_p^{c_1}(1)$ is formed by including all $c_{11}$ and all the light-paths in $L_p^{c_1}$ that are disjoint in relation to $c_{11}$ and mutually disjoint. After this procedure, $L_p^{c_1}(1) = \{c_{11}, c_{12}, ..., c_{1n}\}$ will contain $n_1 = |L_p^{c_1}(1)|$ disjoint light-paths. All light-paths in $L_p^{c_1}(1)$ will be part of the final solution, and one will be the service light-path.

If $n_1 \leq F_{\text{max}}, L_p^{c_2}(2)$ is built and this contains the shortest light-paths that are able to protect against failures in the links in the light-paths in $L_p^{c_1}(1)$. To build $L_p^{c_2}(2)$, we take the sets of links for each light-path in $L_p^{c_1}(1)$. Let $E_1, E_2, ..., E_n$ be the sets of links for $c_{11}, c_{12}, ..., c_{1n}$, respectively. The link failures that interrupt all the $n_1$ light-paths in $L_p^{c_1}(1)$ are given by the Cartesian product $E_1 \times E_2 \times ... \times E_n$. For each tuple in $E_1 \times E_2 \times ... \times E_n$, a partially disjoint light-path that does not use the links in the tuple is found ($n_2 = 1$). This is always possible because there are no disconnected nodes. These light-paths form $L_p^{c_2}(2)$.

The PS algorithm selects only one of the paths belonging to $L_p^{c_2}(2)$, and this will protect the connection against the $n_1$th failure. If $n_1 + n_2 > F_{\text{max}}, \text{ construction has ended}; otherwise the same procedure must be used to build $L_p^{c_3}(3)$ and so on until the number of subsets included in the list $L_p$ is $|L_p| = (F_{\text{max}} - \min|F_{\text{max}}, |L_p^{c_1}(1)| - 1| + 1$.

The PS algorithm handles each number of simultaneous failures sequentially (from 0 to $F_{\text{max}}$). The result is a list of light-paths for each connection organized according to the activation sequence. The service light-path is selected by the following procedure: if $L_p^{c_1}(1)$ has one light-path of unit length (i.e., path between adjacent nodes, with a single link), it is chosen. If not, the service light-path is chosen from $L_p^{c_1}(1)$ by the load balancing procedure (see below). The first protection light-path is selected from the disjoint light-paths still in $L_p^{c_1}(1)$, the second protection light-path is selected from the disjoint light-paths still in $L_p^{c_1}(1)$ (if any) or from $L_p^{c_2}(2)$ and so on. At each step the load balancing procedure determines the light-path to be used.

The nodal degree (i.e., number of neighboring nodes to a node) limits the number of paths generated by any k-link-disjoint shortest path-based algorithm. The PS algorithm selects disjoint and partially disjoint paths to protect the connections against $F_{\text{max}}$ simultaneous failures, regardless of the degree of nodes connected. For this, it is necessary to have available a large number of paths to achieve better load balancing in all states of the network. This allows for greater sharing of resources.

4.1.1. Load-balance algorithm

The load-balancing algorithm suggests the best path for each connection to protect against a combination of failures. The load of a network link is incremented by one when an additional wavelength is used exclusively by a new light-path. The input to the algorithm is the set of dropped connections in the failure state, and for each, a list of candidate paths. The overuse of some remaining links is avoided with the best choice of a candidate path to protect each broken connection. The algorithm performs the following steps:

1. Randomly select a candidate path for each connection.
2. Recalculate the load on the links based on the choice of candidate paths and their current load, the interrupted connections (or in case of no fault, connections between non-adjacent nodes).
3. Assign a weight to every link as follows: the weight of the link with the lowest load is 1, and for the others the weight is incremented by 1, from the lowest to the highest load.
4. Calculate the mean deviation of the loads of links.
5. Assign a weight to every candidate path equal to the sum of the weights of the links that it traverses.
6. For all broken connections, subtract the weight of the candidate path with the lowest weight from the weight of the currently selected candidate path.
7. For the connection that produces the greatest difference in step 6, replace the currently selected path with the candidate path that has the lowest weight.
8. Recalculate the load on the links based on the new choice.
9. Repeat 1–8 until no further reduction in the value of the mean deviation is observed.

When the algorithm has completed these steps, each broken connection will have a selected (suggested) path for that particular combination of failures. The lowest mean deviation ensures better load balancing among the links of the network in a particular state. If a connection can be interrupted in more than one state, the protection path chosen is the one suggested by the most probable state. Therefore, load balancing is not ideal, but approximate.

Table 1 shows an example of selecting the service paths and protection paths for the connection $c$ between nodes 2 and 4 of the network shown in Fig. 1, where $F_{\text{max}} = 4$. The overdone value for $F_{\text{max}}$ was chosen in order to show that the PS algorithm provides additional protection, even when using partially disjoint paths. Decision on path selection depends on the probability of failure of the links, then let us assume that the rates of failure ($\lambda$) and recovery ($\mu$) of links are known. For the subset $I_p^{(2,4)}(1)$ are chosen all disjoint shortest paths (each path is represented by a sequence of links) between the nodes of the connection, i.e., $I_p^{(2,4)}(1) = \{(2, 3), (8, 4), (1, 6, 9)\}$.

The subset $I_p^{(2,4)}(2)$ is formed by partially disjoint paths relative to paths in $I_p^{(2,4)}(1)$. This subset contains only the shortest paths with the same quantity of links. In the example, the selected paths are (1, 7, 3) and (8, 5, 9), both with 3 links. Therefore, $I_p^{(2,4)}(2) = \{(1, 7, 3), (8, 5, 9)\}$. Similarly, each partially disjoint path within the subset $I_p^{(2,4)}(3) = \{(1, 6, 5, 4), (2, 7, 6, 9)\}$ has the same minimum quantity of links, $I_p^{(2,4)}(4) = \{(2, 3), (8, 4), (1, 6, 9)\}, \{(1, 7, 3), (8, 5, 9)\}, \{(1, 6 , 5, 4), (2, 7, 6, 9)\}$ is the set of paths obtained at the end of the procedure. To simplify the presentation of the table, the subsets of candidate paths obtained will be named as follows: $I_p^{(2,4)} = \{P_{11}, P_{12}, P_{13}\}, \{P_{21}, P_{22}\}, \{P_{31}, P_{32}\}$.
algorithm is executed once to select the service path to each connection.

Load balancing is achieved with no network failure. Let us assume that \( p_{12} \) is the path providing the best possible load balancing, so it is selected. The column 0 of the table shows that the first selected path for the connection was \( p_{12} \), one that provides the best possible load balancing. Load balancing considers all disjoint paths in \( L^c_p(1) \) of each connection. The selected service path for each connection is removed from corresponding subset \( L^c_p \).

The column 1 of the table shows the selection of the first protection path for the connection \((2,4)\), obtained by evaluating the network with a link failure, using the

paths in \( L^c_p \). For each fault state \( i \) (corresponds to the failure of the link \( i \)), there will be a different set of connections affected. Each state is evaluated according to the procedure of load balancing, to seek the best solution for all connections with protection and affected by the failure considered. There will be a protection path suggested for each connection. The connection that is analyzed in the example has a service path \( p_{12} \) that will be discontinued if there is failure in one of your links \((8 \text{ or } 4)\). In case of failure of the link 8, the load balancing algorithm had the path \( p_{11} \) as the best option for protection, while a failure occurs on link 4, the path \( p_{13} \) is the best option.

To protect against the first failure, the connection will utilize \( p_{11} \) as the protection path, because the probability of failure of link 8 is greater than the probability of failure of link 4. At this stage, the probability value does not matter, but the ratio of probabilities. However, the metric used is the ratio between the failure rate and the recovery rate \((P_0 = \log(\lambda / \mu))\). The logarithm is used to speed the calculation in the case of considering more than one link simultaneously, because it allows the result to be obtained by a sum, rather than the product of the ratios, just as [29].

The column 2 of the table shows the selection of the second protection path for connection \( c \). The path is obtained by evaluating the network with two link failures,
and there is only one path to choose within $L_\mathcal{P}^i(1)\setminus\{p_{12},p_{11}\} = \{p_{13}\}$. For each fault state $(i,j)$ that corresponds to the simultaneous failure of the links $i$ and $j$ (in any order), there will be a different set of affected connections. Again, each state is evaluated according to the procedure of load balancing, considering all the affected connections, suggesting a protection path for each. In connection seen in the example, the paths $p_{12}$ and $p_{11}$ supporting the connection, and both would be affected when there is at least one of the combinations of link failure shown in the table $\{(8,2),(8,3),(4,2)\}$ or $\{(4,3)\}$. In the example, for each combination of two failures, the load balancing algorithm pointed the path $p_{13}$ as the best option (only one), and was then chosen as the protection path of a second link failure for the connection. As explained previously, $P_{(8,2)} = -\log(\lambda_8/\mu_8) + \log(\lambda_2/\mu_2)$.

The column 3 of the table shows the selection of the third protection path for the connection $(2,4)$, obtained by evaluating the network with three failed links, using an analogous procedure. It is observed that the options are the paths in $L_\mathcal{P}^i(2) = \{p_{21},p_{22}\}$, since the set $L_\mathcal{P}^i(1)$ is empty. It was also observed that a fault state is called $(i,j,k)$, and not all failure combinations can ensure that the existence of a protection path with the minimum use of resources, in this case, is represented by “#”.

The column 4 shows the selection of the fourth protection path for the connection $(2,4)$, obtained by evaluating the network with four failed links. It is observed that the options are the paths in $L_\mathcal{P}^i(3) = \{p_{31},p_{32}\}$, because the paths in the subset $L_\mathcal{P}^i(2)$ were not provided to protect the fourth failure. After the last run of the algorithm, each connection has a list of protection paths organized in order of activation ($L_\mathcal{P}^i$).

In the example, the connection $c$ has a list $L_\mathcal{P}^c(2,4) = \{\{8,4\},\{2,3\},\{1,6,9\},\{8,5,9\},\{2,7,6,9\}\}$, where the first is the service path $(p_{12})$ and others are protection paths. It is important to note that the PS algorithm can select a smaller amount of protection paths for a connection. In fact, the amount of protection paths that it selects for a connection is a parameter that will be used by the network planning algorithm (Section 4.3).

We consider a network with $N$ vertices and $E$ links. The complexity of finding all-pairs disjoint shortest paths using Dijkstra’s algorithm [1] is $O(N^3)$. The total number of demands $T$ is the length of the list $L_p$. Let $S_{max}$ be the maximum number of disjoint shortest paths between any pair of nodes and $h_{max}$ be the number of hops in the route of the longest light-path. In the PS algorithm, there is initial load balancing after the assignment of one of the disjoint paths within the set $L_\mathcal{P}^i$ for each connection. In each interaction, the load balancing is improved through the selection of connection that will have a new path selected. In each connection $O(h_{max}S_{max})$ different values of link load should be examined. Therefore, $O(Th_{max}S_{max})$ comparisons are needed (to compare the load on links between disjoint paths in each desired connection). To obtain the best possible balance are required $T/2$ interactions from a worst initial load balancing, resulting in a time complexity of at most $O(T^2h_{max}S_{max}/2)$. In each state of the network there will be $O(E^{max}T^2h_{max}S_{max}/2)$ comparisons. In total, the overall complexity of the PS algorithm is $O(S_{max}N^3 + E^{max}T^2h_{max}S_{max}/2)$.

4.2. Algorithm for computing connection unavailability

The input to the algorithm for computing connection unavailability (CCU) is the set of $F_{max} + 1$ paths for every connection produced by the PS algorithm ($L_\mathcal{P}^i$) and the list of network links with their capacity and failure and repair rates. The algorithm is based on an analytical model derived from a continuous-time Markov chain, which uses the modeling assumptions for availability analysis discussed at the end of Section 4.1. The following notation is used to represent sequences of link failures: $(i)$ represents the state where all links are operational; $(i,j,k)$ represents the state where links $i$, $j$, $k$ have failed in this order; and $\pi_{(i,j,k)}$ represents the steady-state probability of state $(i,j,k)$. The Markov chain for $F_{max}=3$ is shown in Fig. 2.

The stationary probability of a sequence of failures can be obtained from the probability of the previous state (the one before the last link failure in the sequence), as shown in the below equations:

$$\pi_{(i)} = \frac{\lambda_i}{\mu_i} \pi_{(i)}$$

Fig. 2. Example of the Markov chain for $F_{max}=3$. 
Fig. 3. Order in which the states of the Markov chain are traversed to calculate unavailability.

\[ \pi_{(i,j)} = \frac{\lambda_j}{\mu_i + \lambda_j} \pi_{(i)} \]  
(2)

\[ \pi_{(i,j,k)} = \frac{\lambda_k}{\mu_i + \mu_j + \mu_k} \pi_{(i,j)} \]  
(3)

\[ \pi_{(i,j)} \left[ 1 + L \sum_{i=1}^{L} \frac{\lambda_i}{\mu_i} + L \sum_{i=1}^{L} \sum_{j=1}^{L} \frac{\lambda_i \lambda_j}{\mu_i \mu_j + \mu_j} + L \sum_{i=1}^{L} \sum_{j=1}^{L} \sum_{k=1}^{L} \frac{\lambda_i \lambda_j \lambda_k}{\mu_i \mu_j \mu_k + \mu_i \mu_j + \mu_k} \right] = 1 \]  
(4)

where \( L \) is the number of links, \( \lambda \) is the failure rate (number of failures per unit time) and \( \mu \) is the recovery rate (number of repairs per unit time).

To calculate the unavailability of connections, the CCU algorithm performs a depth-first traversal of the Markov chain, as illustrated in Fig. 3. We introduce a notation where the identification of the currently visited state is broken down into two parts \((s^{-1}, s)\), where \( s^{-1} \) represents the sequence of link failures that occurred before the failure of link \( s \). The next state to be visited is always the concatenation of the current link failure \( s \) with the preceding failure sequence \((s^{-1})\). This approach allows the active paths for each connection and the resources that were reserved during the previous state and are still available in the visited state to be stored.

Each stage of the calculation is started by visiting a new state, which includes an additional link failure. The current failed link is used by a set of connections that will be interrupted and will seek resources for the next protection path in the activation sequence. The amount of remaining protection resources in a given state is given by the difference between the network capacity and the capacity of the links used in the previous state and may not be sufficient to meet all the connection needs.

We assume that the probability of obtaining a resource on a shared link is the same for all connections and is given by the ratio of the number of resource units available (wavelengths) to provide protection in the shared link to the number of connections competing for these resources. A connection competing for resources in more than one link must obtain the resources for all of them. The probability of connection \( c \) obtaining all the necessary resources to enable it to recover in state \( s \) is called the acceptance factor \( (AF_c) \) and is the product of the probabilities of the protection path getting a shared resource for each of its links. This product is an approximation and underestimates the real probability because it does not consider some possible combinations of interrupted connections. Although there is therefore an error in the calculation, this has a negligible effect on the final result because such scenarios mainly occur when there are large numbers of simultaneous failures, and the probability of this happening is low. The rejection factor \( (RF_c) \) is the complement of \( AF_c \). The unavailability of connection \( c \) in state \((i,j,k)\) can be estimated by

\[ U_{(i,j,k)} = \sum_{s} \pi_s RF_c \]  
(5)

When visiting state \((s^{-1}, s)\) the algorithm identifies all the connections that were affected by the failure of link \( s \). It calculates the rejection factor of the protection path of the interrupted connections \( RF_c \) and estimates their unavailability according to the following equation:

\[ U_{(s^{-1}, s)} = \pi_{(s^{-1}, s)} RF_c \]  
(6)

For connections already broken in \( s^{-1} \) (i.e., in the absence of a protection path), \( RF_c = 1 \). For connections not interrupted in \( s^{-1} \) but broken in \( s \), \( RF_c \) is calculated taking into account the resources available for the protection path as follows: (i) if at least one of the links used by the protection path of connection \( c \) has no resources, \( RF_c = 1 \); (ii) if there are sufficient resources for all connections that share any link with connection \( c \), \( RF_c = 0 \); and (iii) if there is at least one link of the protection path with non-null capacity but with insufficient resources for all the connections that share the link with connection \( c \), \( 0 < RF_c < 1 \). In this case there is contention for protection resources, and some connections will be broken because of a lack of resources.

Due to the sharing of resources, each state of the Markov chain has different amounts of available resources. The order of occurrence of faults determines each state and has influence in obtaining necessary resource for each connection. Each connection affected by the same failure should try to get network resource after activating the first uninterrupted protection path. The distribution of available resources is different in each state. Therefore, to run the CCU algorithm, each state of the Markov chain must be traversed. In state \((s^{-1}, s)\), the unavailability \( U_{(s^{-1}, s)} \) is obtained by calculating probability \( \pi_{(s^{-1}, s)} \) and rejection factor \( RF_c \) (Eq. (6)). Since each sequence failure represents a state of the Markov chain and for each state the procedure takes into account all network connections, the computational complexity of the CCU algorithm is given by \( O(|L_\lambda| E^{\text{max}}) \), where \( E \) is the number of links.
4.3. The network planning method

The network planning method (MSB) is a two-stage procedure based on the PS and CCU algorithms. For the network to be dimensioned with dedicated capacity, there are always considered to be sufficient protection resources for all connections that share any link, i.e. $RF_c^s = 0$, for all $s$ and $c$. The first stage is an iterative procedure. At the end of each iteration, MSB checks for connections that are over-protected (unavailability below the required value) and reduces the number of lightpaths protecting these connections by one. Using the list of all connections and corresponding light-paths, the first stage sizes the network with dedicated capacity so that connections with less than $F_{max} + 1$ light-paths have unavailability above the required value. At the end of the first stage, connections from which light-paths were removed will be under-protected (unavailability above the required value). In the second stage, one light-path is added to under-protected connections.

The MSB method separates the connections into two groups. Group 1 contains the under-protected connections, for which the number of protection paths was modified during the procedure, whereas group 2 contains the over-protected connections, for which the protection paths were maintained throughout the procedure. In general, connections in group 1 have shorter service and protection paths than connections in the second group. As previously stated, the first stage produces a network configuration with dedicated protection. It is important to note that when the traffic demand between a pair of nodes requires multiple connections, the unavailability of each connection is inversely proportional to the number of connections needed, as the aim is to ensure that the amount of data lost is independent of the traffic demand. For this reason, these connections tend to be in group 2.

At the end of first stage, every connection in group 1 has insufficient alternative paths to achieve the required availability, while all connections in group 2 have their original protection paths. The protection capacity to be installed on the network is calculated using this configuration. Note that this capacity implies that the availability of connections in group 1 is below the required value, so that additional protection paths are needed. These additional paths share protection resources with connections in group 2. The paths to be added are determined by the load-balancing procedure when the PS algorithm is performed for the last time with constrained resources. This will increase the unavailability of connections in group 2, but this will have a limited impact for three reasons: (i) although there are many connections in group 1, their protection paths are short; (ii) load balancing avoids overuse of the links; and (iii) there is capacity available on the links for some of the protection paths for the connections in group 2. The steps carried out by the MSB are as follows:

1. Set the number of paths to $F_{max} + 1$ (service and protection). Put all connections in group 2.

2. Update the number of paths for each connection and run the PS algorithm. The result is a list with a defined amount of paths for each connection.
3. Run the CCU algorithm assuming unlimited capacity on every link (dedicated protection). The result is the unavailability for each connection (h/year).
4. If there are connections with unavailability below the desired value, subtract one from the number of protection light-paths. If the connection is in group 2, it should be moved to group 1. Go to step 2.
5. Add one to the number of protection light-paths for the connections in group 1.
6. Run the path selection and CCU algorithms with the same link configuration.

Table 2 shows the runtime for the PS and CC algorithms. For networks randomly created, we define the number of nodes in the graph and probability of connection to each node. Then, each of the edges is replaced by a bidirectional link. This resulting topology is also checked to ensure that it is two-edge connected. For all networks there is a demand of a wavelength between each pair of nodes. The observed times are in seconds in a Intel Core i7-3517U CPU 2.40 GHz. For the CCU algorithm there are two different runtimes: (i) the algorithm runs with unlimited resources on each link for up to $F_{max}$ times (first stage); (ii) the algorithm runs with shared resources by last time (second stage). In the last column the time is not the greatest, because unlike the others, the Pan-European BT is a planar network. The network has lower average number of disjoint paths per node pair and the paths are longer.

5. Results

As noted earlier, works that address network protection for multiple link failure are dynamic algorithms (online). Our work performs static protection (offline) because it uses pre-planned protection capability and set of connections are known in advance. Therefore, to compare with the MSB method, a method of protection using a single dedicated backup path for each connection will be used (single dedicated backup – SDB).

5.1. Comparison with an exhaustive search

An experiment was conducted to evaluate the results after each of the two stages of the MSB method and to
compare the resulting unavailability values with those produced by an exhaustive search, on the small size network, the goal being to demonstrate the accuracy of MSB. The example network is shown in Fig. 4. There is a demand of one wavelength between each pair of network nodes (full-mesh), forming a group of 20 unidirectional connections. The network is to be protected against at most 3 simultaneous link failures, i.e., each connection is supported by four light-paths \((F_{\text{max}} + 1 = 4)\). The maximum required unavailability for all connections is 1 h/year. Recovery rates and failures are shown in Table 3.

Fig. 5 shows the unavailability in increasing order for all the connections in h/year. When all the connections have the maximum number of protection paths \((F_{\text{max}} = 3)\), some are over-protected. For illustration, the figure shows the unavailability after the first stage of MSB, where it can be seen that two light-paths have been removed from the first 12 under-protected connections. At this point, MSB determines the capacity of the links without sharing resources. In the second stage, it adds one light-path for every under-protected connection without increasing the capacity of the links, thus forcing capacity to be shared and bringing the availability close to the required values. Unavailability of some of the first 12 connections is above the desired value. This unavailability is already high at the end of the first stage. The amount of shared resources on the network is insufficient to meet this need, since the network is very small. A new iteration of the first stage with the same number of paths can add dedicated resources for the connection. Adding a path with purpose of sharing resources can reduce the unavailability of the connection to the desired value in the second stage.

The figure also shows the unavailability obtained using an exhaustive search procedure. The exhaustive search method aims to evaluate each of the possible permutations in the queue formed by interrupted connections competing for insufficient resources to maintain availability. Then, for each new state of the network is performed a new exhaustive search (all permutations of interrupted connections in the queue) to find the rejection factor \((RF_c)\) for each interrupted connection.

Fig. 6 shows the differences between the unavailability calculated by the MSB method and the corresponding figure for the exhaustive search. MSB introduces an error that is directly proportional to the number of shared links. This error, which is more apparent in connections located at the right end of the figure, whose paths are longer (more than one shared link), occurs because MSB does not consider all the possible connectivities that can be achieved through shared links (connections 13–20). The unavailability is always greater than the actual value, i.e., the algorithm is conservative, and is only computed exactly for connections that depend on a single shared link, here the 12 first connections in the figure. The largest difference in the example was 0.95 h/year for connection 17, an error of 31% compared with the unavailability obtained by the exhaustive search. The reason for this error is the use of longer protection paths for this connection (greater number of shared links).

Fig. 7 compares MSB with the dedicated method using a single protection path for each connection (single dedicated backup – SDB). The unavailability values produced by the SDB method are all higher than those produced by MSB. Assuming up to 3 simultaneous link failures, the

---

**Table 3**

Recovery rates and failures used in example network.

<table>
<thead>
<tr>
<th>Link</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
</tr>
</thead>
<tbody>
<tr>
<td>(i) ((10^{-4}) h(^{-1}))</td>
<td>2.48</td>
<td>3.66</td>
<td>5.71</td>
<td>2.88</td>
<td>2.98</td>
<td>2.04</td>
<td>2.82</td>
</tr>
<tr>
<td>(\mu) ((10^{-2}) h(^{-1}))</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
<td>5</td>
</tr>
</tbody>
</table>

---
average unavailability using SDB is 5.29 h/year with a standard deviation of 1.52, while the corresponding figure for MSB is 2.06 h/year, with a standard deviation of 1.14. SDB requires 1.08 times the service capacity, while for MSB this figure is 1.58.

If each failure sequence is a state of the Markov chain and \( |c_j| \) is the number of interrupted connections in each state of the network, then the computational complexity of exhaustive search algorithm would be \( O(E^{F_{max}} |c_j|) \), where \( E \) is the amount of links. In fact, the example network in Fig. 4 with the traffic matrix full-mesh, exhaustive search needs 157 min to get the final solution in an Intel Core i7-3517U CPU 2.40 GHz. With a link and a node added to this network, the runtime for this algorithm came to be several days.

5.2. Analysis with a real-size network

A second experiment was carried out to demonstrate the practical scalability of MSB. The network chosen was the Pan-European BT network, as this is of a suitable size to demonstrate how the method can be used in practice (the value \( F_{max} = 3 \) is used). Fig. 8 shows the network, which consists of 28 nodes in major cities in Europe connected by 41 links in a mesh topology. The average length of each fiber link is 625 km, and the minimum and maximum lengths are 111 km and 1500 km [30].

As stated by Mello et al. [19], the values assigned to the network parameters, such as \( \lambda = 1/\text{MTTF} = 200 \text{ FIT/km} \), where FIT = 1 failure in \( 10^9 \) hours (FIT – Failure in Time) and \( 1/\mu = \text{MTTR} = 20 \text{ h} \), in intercontinental networks, 95% of the possible network failures do not happen in more than two links simultaneously (mean time to failure (MTTF) and mean time to repair (MTTR)). The MSB method applied in such a configuration would cause a large reduction in unavailability of connections (with values of a few seconds per year) and the effects that can be demonstrated by applying the method would not have the desired emphasis. To keep the unavailability of connections in the order of h/year, and enable the minimum certification of the effects of resource sharing, were used: \( \lambda = 1/\text{MTTF} = 800 \text{ FIT/km and MTTR} = 20 \text{ h} \). The maximum required unavailability in selected connections is 4 h/year.

The traffic matrix for the Pan-European BT network was set up with the values obtained from the growth estimate suggested by De Maesschalck et al. [30] for voice and data. All wavelengths have the same capacity (20 Gbit/s). The number of connections between any two nodes was set to 8 for the highest traffic demand (160 Gbit/s), to 1 for the lowest and to proportional integer values for the others (Table 4). The capacity of one connection is the same as the capacity of one wavelength. The traffic demand for the 756 node pairs (28 \( \times \) 27 unidirectional connections) is shown in Table 3.

Fig. 9 shows the distribution of wavelengths per link; the links are arranged in ascending order of load in the service light-paths. From the number of wavelengths/fiber, the minimum number of fibers in each link can be obtained.

The wavelengths required to support all the light-paths for SDB and MSB are shown in Table 5. In comparison with the SDB method, the table shows the reduction of the average unavailability of connections by 43% (from 5.97 to
3.38 h/year) using the method MSB with additional resources of 27% (from 9206 to 11 688 wavelengths). The reduction in standard deviation is due to resource sharing.

Fig. 10 shows the unavailability calculated by SDB and by MSB. The results obtained are shown in Table 5. The reduction in average downtime is a direct consequence of increased protection, while the reduction in the standard deviation shows that there is a more homogeneous distribution of network resources, indicating that the MSB method is working correctly (i.e., it has achieved balanced use of resources to share network resources more efficiently). To facilitate explanation of the effect of sharing protection resources, the connections are arranged in increasing unavailability and separated according to the two groups formed when MSB is run (see Section 4.3). As explained, the connections in group 1 (numbers 1–956) are under-protected, while those in group 2 (numbers 957–1632) are over-protected.

In the MSB method, the unavailability of every connection is limited to the required unavailability divided by the number of connections needed to meet the corresponding traffic demand between any two nodes. However, it can be seen in Fig. 11 that there are connections belonging to group 2 that have lower unavailability than required. This occurs for traffic demands that require more than one connection. In such cases the required unavailability of the individual connections is set to a fraction of the total unavailability for all the connections used to meet the traffic demand. These connections can be observed in the left-most positions of group 2 (connections 957–1258).

The unavailability calculated by MSB for connections in the right-most positions of group 1 is higher than the unavailability calculated by SDB. Hence, for best performance such connections should be configured using SDB. This discrepancy arises because the location of these connections on the network renders sharing unfeasible. Finally, the connections in group 2 with higher unavailability than required have relatively long paths in the network. These connections have greater unavailability because they share protection resources with connections in group 1.

5.3. Validation of the proposed method by simulation

The Pan-European BT network scenario in the second experiment was simulated to validate the results obtained by MSB. Here also, the wavelength continuity constraint is overcome, i.e., every node of a route is able to convert an input wavelength to a different output wavelength. The steps in the simulation were as follows:

1. For each link, generate a random sequence of failures. The time to failure has an exponential distribution with a rate parameter equal to the failure rate of the link.
2. For each failure, generate the time to repair. The time to repair has an exponential distribution with a rate parameter equal to the repair rate of the link.
3. Identify dropped connections for all link failures. All possible failure combinations are taken into account, including those with more than $F_{\text{max}}$ links failed simultaneously.
4. For the set of dropped connections, generate a random order for treatment (in this random order) and try to activate the first uninterrupted protection path.

<table>
<thead>
<tr>
<th>Lightpaths</th>
<th>1</th>
<th>2</th>
<th>3</th>
<th>4</th>
<th>5</th>
<th>6</th>
<th>7</th>
<th>8</th>
</tr>
</thead>
<tbody>
<tr>
<td>Number of node pairs</td>
<td>337</td>
<td>196</td>
<td>107</td>
<td>54</td>
<td>26</td>
<td>20</td>
<td>12</td>
<td>4</td>
</tr>
</tbody>
</table>

Fig. 10. MSB vs. SDB (Pan-European reference network).

Fig. 11. Simulation vs. MSB (Pan-European reference network).

<table>
<thead>
<tr>
<th>Wavelengths</th>
<th>Service</th>
<th>Protection (%)</th>
<th>h/year</th>
<th>$\bar{T}$</th>
<th>$\sigma$</th>
<th>$U_{\text{max}}$</th>
</tr>
</thead>
<tbody>
<tr>
<td>5103</td>
<td>4103 (80)</td>
<td>SDB</td>
<td>5.97</td>
<td>5.16</td>
<td>26.10</td>
<td></td>
</tr>
<tr>
<td>6585 (129)</td>
<td>Simulation</td>
<td></td>
<td>3.59</td>
<td>4.09</td>
<td>24.07</td>
<td></td>
</tr>
<tr>
<td></td>
<td>MSB</td>
<td></td>
<td>3.38</td>
<td>3.70</td>
<td>22.26</td>
<td></td>
</tr>
</tbody>
</table>
5. Identify the connections for which the protection path (if any) could not be activated because of a lack of resources. In this situation the connection is considered to have failed.
6. Record the failure times for the connections.
7. Compute the unavailability of connection \( c \) as 
   \[
   U_c = \sum_k T_{c_k} / T
   \]
   where \( T \) is the simulation time and \( T_{c_k} \) is the duration of the \( k \)th failure of \( c \).

We used the method of active recovery with a reverting strategy defined by Zhang et al. [24]. In this scheme the traffic is switched to the protection path when the service path fails and switched back to the service path when the failure is repaired, thus releasing the resources used by the protection path. If there are multiple connections waiting for the released resources, the connections are recovered in the order in which the failed links are recovered. Protection features are released with the recovery of active path be it service or protection.

Fig. 11 shows the unavailability values obtained by simulation and by MSB.

The simulation was implemented with Mathematica 7. 180,000 steps were performed by the algorithm, where the step corresponds to the period of 1 h in the activation of the network, resulting in a total simulation period of over 20 years. It can be seen that both methods yielded similar results. The results obtained are shown in Table 5.

Fig. 12 shows the differences between the two sets of results. The values produced by MSB are used as a reference (zero in the figure). The difference is positive (blue) when the simulated value is greater than calculated. The difference is negative (red) when the simulated value is less than that calculated. For connections with similar unavailabilities, the greater the balance between the red and blue values, the closer the simulated and calculated values. It can be seen from the figure that there are regions where there is a greater balance (on the left of groups 1 and 2) and other regions where positive (blue) values predominate (on the right of groups 1 and 2). The average deviation was 0.513 and the mean deviation 0.583. The predominance of positive values is because failure combinations with more than \( F_{max} \) simultaneous link failures occurred during the simulation, despite the value of the calculated unavailability contain an additional error caused by overvalued rejection factor (\( RF_c \)).

This is less apparent for connections with lower downtime (the first 800 connections in group 1 and the first 300 in group 2) because they are less vulnerable. These connections have unavailabilities below the required value (4 h/year). The connections to the right of group 1 are situated in locations that are topologically unsuitable for sharing resources, and those to the right of group 2 have availabilities that are directly proportional to \( F_{max} \). The largest absolute deviation (5.672 h/year) occurred for connection number 1605, for which the simulated value was 20.23 h/year, a variation of 28%.

The differences in percentage are shown in Fig. 13. The greatest difference (64.4%) occurred in connections 335–346 and corresponded to a calculated value of 0.602 h/year and a simulated value of 1.691 h/year. This difference is because during the simulation, the occurrence of combinations with more than \( F_{max} \) simultaneous failures (not considered by CCU) increases the unavailability of all connections. The effects are easily observable in connection with low unavailability and calculated more precisely. Therefore, the graphic shows that despite this the availability requirements continue to be met.

These findings indicate that the required unavailability is not exceeded for the connections analyzed. It can therefore be concluded that MSB enables a required maximum allowable unavailability to be defined for connections in regions of the network where the topology permits sharing of resources.

The simulation result shows a natural behavior of the network, since the metric used in determining availability is the mean time between failures (MTBFs), which is the sum of MTTF and MTTR. It should be noted that there are an infinite number of sets of combinations of MTTF and MTTR that can produce the same availability. The calculated result is very similar to the simulation. In the simulation, an additional error in the unavailability is caused by the occurrence of combinations of failures of order higher than \( F_{max} \), especially for connections on the right side of group 2. The additional error is larger in the calculation of the unavailability of these connections because they use longer paths. Note that despite the imprecision, the required unavailability in selected
connections has not been exceeded (4 h/year). It can therefore be concluded that MSB enables a required maximum allowable unavailability to be defined for connections in regions of the network where the topology allows resource sharing.

6. Conclusion

In this paper we have presented a new traffic-planning method that routes and allocates resources to connections protected against multiple link failures. Protection is achieved by means of a sorted list of protection paths while at the same time ensuring optimal load balancing in the network. The method leads to a degree of protection resource sharing permitted by the network topology. It increases the resiliency of the network by prioritizing load balancing and minimizes the total amount of protection resources by increasing the degree to which they are shared.

The most vulnerable connections, those with longer paths (group 2), can also be planned using the proposed method if a maximum number of simultaneous link failures is established. This variable determines the number of protection paths, each with resources that will be shared with connections supported by more reliable shorter paths (group 1). The method takes advantage of any network topology. The method uses not only the disjoint paths, as in [17, 25–30], but also partially disjoint paths, which provide additional protection for the connection. One connection can remain available even if all its disjoint paths are broken. The method also establishes a limit for data loss for multiple-connection demands and identifies connections with limited ability to share protection resources, leaving it to the network planner to choose between multipath protection ($F_{\text{max}} + 1$) or protection with a single dedicated path.

The numerical results show that the method uses sharable resources contained in the network topology, thereby significantly reducing the vulnerability of connection. Furthermore, the proposed method shows that resource sharing can be achieved by organizing connections into two groups: less vulnerable connections that need protection features (group 1) and more vulnerable ones with abundant resources (group 2). The method identifies the connections that require special handling due to their topologically unfavorable conditions.

The following enhancements could improve the performance of the method both functionally and in terms of the results it provides and will be considered in future studies: improving the accuracy of the calculation of the rejection factor for dropped connections, using a larger value for the maximum number of simultaneous link failures and calculating unavailability based on failure combinations involving more link failures, thus improving the accuracy of the results; and skipping intermediate steps, avoiding unnecessary calculations and identifying connections in topologically unfavorable positions so that these can be addressed with special procedures, thus accelerating the planning procedure.

References

[26] X. Cheng, X. Shao, Y. Wang, Y.-K. Yeo, Differentiated resilient protection against multiple-link failures in survivable optical


