Designing A High Availability Subnetwork to Support Availability Differentiation

Abdulaziz S. Alashaikh, David Tipper
Graduate Telecommunications and Networking Program
School of Computing and Information
University of Pittsburgh, Pittsburgh, PA 15260
Email: azizoozi@gmail.com, tipper@tele.pitt.edu

Teresa Gomes
Department of Electrical and Computer Engineering
University of Coimbra / INESC Coimbra
3030-290 Coimbra, Portugal
Email: teresa@deec.uc.pt

***Abstract*—In a recent series of papers [1–3] we studied the problem of how to provide both high levels of availability and service differentiation to traffic flows in a cost efficient manner. The basic idea developed was to embed at the physical layer a high availability set of links and nodes (termed the *spine*) in the network topology to support protection and routing in providing differentiated classes of resilience with varying levels of end-to-end availability. In this paper, we present an optimization model formulation of the spine design problem considering link availability and the cost of upgrading link availability. Numerical results show the advantages of modifying the availability of a subset of the network topology to provide QoR classes.**

I. Introduction

The predominant approach to supporting multiple level of resilience is based on assigning different protection and restoration schemes at a certain technology layer to flows based on their Quality of Resilience (*QoR*) classes [4]. However, such a scenario-based approach might result in a resilience level below what is needed or inefficient utilization of network resources. Availability-based routing was proposed to provision services with explicit availability constraints in the routing algorithm, combined with protection schemes (*e.g.,* alternate path routing with dedicated protection [5], and shared protection [6]). Another line of research tried to achieve differentiation through time in order to minimize both resource usage and the risk of not meeting availability targets [7, 8]. The main drawback of this approach is the need to maintain failure status for all flows and make real time routing decisions.

In general, there are some limitations of the existing approaches. First, the range and the spacing between availability classes for existing approaches are somewhat narrow. Both need to be extended to cover a wider range of classes. Second, high availability for mission-critical services (*i.e.,* four to six 9's) might not be achieved using basic protection schemes such as 1+1 [9]. For example, in [10], the 1+1 recovery scheme of the gold class is insufficient to support extremely high availability levels (*e.g.,* five or six 9's). Possible options to improve the availability include using a higher configuration of dedicated protection (*e.g.,* two disjoint backup paths for each working path) or reserving adequate sharable spare capacity to restore traffic from multiple simultaneous failures (*e.g.,* dual failure shared backup path protection [11]). Both approaches, however, lead to inefficient utilization of network resources and are also constrained by the diversity of the network which implies that in some cases adding new links and possibly nodes to the network topology to support additional parallel routes is required. However, expanding a nationwide backbone network simply to improve availability is difficult to justify economically.

An alternative way to achieve high availability is by improving the availability of network components. The authors of [12, 13] try to optimize network availability by improving the availability of a subset of physical links via shielding. In addition, Botton et al. [14] study a network design problem with a subset of edges that for a given cost can be upgraded to be more reliable. They show that having a set of more reliable edges as a substitute to having edge-disjoint path-pairs can improve overall resource efficiency. These approaches, however, do not support resilience differentiation. The third limitation of existing approaches is related to the application of these approaches to layered networks. With just a few exceptions, most of the existing approaches suffer from crosslayer mapping issues [15], as without full knowledge of the physical layer and the mappings between layers no hard guarantees on availability can be provided (*i.e.,* due to fault propagation).

Our approach to provide high and differentiated levels of availability stems from the Brinbaum's importance measure. According to this measure, improving the component with the higher availability in parallel configuration yields the best overall availability [16]. To illustrate this, assume we have a flow f routed over a working path (WP) and a disjoint backup path (BP) with availability $A_f^{WP} = 0.99$ and $A_f^{BP} = 0.90$, respectively. The end-to-end availability A_{e-e} of flow f is based on its A_f^{WP} and A_f^{BP}, and is calculated as a parallel configuration in which $A_{e-e} = 1 - (1 - A^{WP})(1 - A^{BP})$. Assume we want to strengthen one or both of the links by adding some availability units Δa, with an option of ($A_f^{WP}+\Delta a$), ($A_f^{BP}+\Delta a$), or ($A_f^{WP}+\Delta a/2$ and $A_f^{BP}+\Delta a/2$) to add these units. Figure 1 plots the overall end-to-end availability of flow f for the three options. It shows that improving A_f^{WP} only, achieves better end-to-end availability A_{e-e} than any other option. From this, it is clear that having relatively highly available components forming a working path in parallel combination with a lower availability backup path can achieve better overall availability than homogenous

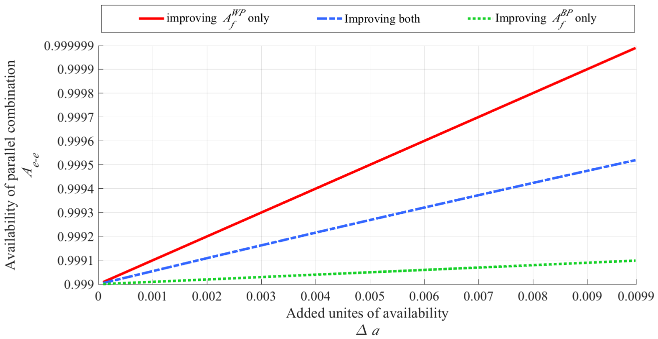


Fig. 1: Improving availability in a parallel configuration

availability components. Adopting this concept, a network provider can allocate investment towards improving network reliability in an economic-efficient way via implementing a network spine.

Definition. *The spine* is a substructure with comparatively higher availability embedded into a network at the physical layer to improve the overall network availability without substantial modifications to the topology.

Our approach requires designing a network with heterogenous link availabilities such that a substructure of the network has relatively larger availability values. The high availability substructure portion of the network is termed the spine. The spine would connect those nodes with traffic needing a high level of availability and provide a basis for differentiated classes of resilience. For example, the highest quality of resilience class traffic could be routed on the spine or use the spine as a backup path. The nodes, link interfaces and links on the network spine would have higher availability than the equipment that is not part of the spine. This provides *levels of availability differentiation at the physical level* which can be leveraged with restoration techniques, logical virtual network topology routing, cross layer mapping and other methods to further differentiate resilience classes and provide an extended range of availability guarantees.

In our previous work [1–3], we designed the spine as a subnetwork that takes the layout of a minimum spanning tree embedded at the physical layer and use this subnetwork for routing connections of the higher availability *QoR* classes. We provided heuristics based on graph structural properties to find candidate spines and study the design performance. We showed numerically that the spine approach actually widens the availability ranges of network flows, and achieves high availability values that cannot be achieved with standard protection configurations. In [2, 3], we assumed all links on the spine have the same availability (a_S) and similarly all links off the spine have the same availability (a_O) with ($a_O < a_S$). Our sensitivity analysis in [3] shows that modifying either value; the improvement step of availability Δ, or considering heterogenous link availabilities, results in a change in the ranking of the best spines with respect to the availability metrics considered. In follow on work we showed that using spine-aware crosslayer routing, the spine concept provides levels of availability differentiation in multilayer networks with upper [17] or lower layer restoration [18]. In this paper, we revisit our proposal of the spine concept and formulate the spine design problem as a mixed integer linear programming (MILP) problem that considers the cost of upgrading link availability. For simplicity we consider the uncapacitated network case and note that the our previous work [17, 18] showed only modest increases in the capacity are needed with the spine approach with the amount depending on the percentage of highest QoR class traffic.

The remainder of the paper is organized as follows. In Section II, we provide some background on the spine and describe our model. In Section III, we present the spine design optimization problem. In Section IV, we conduct a numerical study to evaluate our model and show a sample of the results. We conclude our paper in Section V.

II. The Spine Model

Here we adopt an optimization model approach to determining the spine that involves taking into consideration factors such as the possibility to improve the links and a target level of availability. The problem can be stated as follows. Given a physical network $\mathcal{G}_P = (\mathcal{V}_P, \mathcal{E}_P)$ with a set of nodes $\mathcal{V}_P$ and a set of physical links $\mathcal{E}_P$, a set $\mathcal{K}$ of improvement options for each link with associated cost c_{ij}^k; $(i,j) \in \mathcal{E}, k \in \mathcal{K}$ and a set of end-to-end connections/flows $(s,t) \in \mathcal{F}$ that need high availability, one seeks to determine the subnetwork, that forms the spine $\mathcal{G}_S = (\mathcal{V}_S, \mathcal{E}_S)$, where $\mathcal{G}_S \subset \mathcal{G}_P$, that minimizes the total cost while achieving an availability target.

A. Incremental Link Availability Model

For a given network, each link is assigned an initial link availability value a_{ij} based on its length, with longer links being less reliable. Specifically we use a distance-based link availability formula found in [19]. The link availability is calculated as $a_{ij} = a_{c_{ij}} \times a_{t_{ij}}$ where $a_{t_{ij}}$ is the product of cable-ends equipments (i.e., OXC, ROADM etc...), and $a_{c_{ij}}$ is the fiber cable availability that can be calculated from:

$$a_{c_{ij}} = 1 - \frac{MTTR}{MTBF} \tag{1}$$

$$MTBF_{hrs} = \frac{CC \times 365 \times 24}{\text{cable length}_{km}} \tag{2}$$

where CC is the cable cut rate, $MTBF$ and $MTTR$ are the mean time between failures and mean time to repair in hours, respectively.

We study the scenario where each link in a network, can be purposely strengthened so that its MTBF is increased, for example by altering the cable implementation method (*e.g.,* burying an aerial cable) [20, 21] or adding physical protection [12, 13] or the MTTR is reduced by focussed maintenance and repair efforts [20, 22]. For each link, one can collect possible options to improve its availability, and each option would result in a different availability level and incurred cost. Specifically, if the link e spans node-pair (i, j) and has availability a_{ij}, using method k, the link availability can be improved to a_{ij}^k with cost c_{ij}^k, whereas using method $k+1$ that costs c_{ij}^{k+1}, availability is improved to a_{ij}^{k+1}. We assume "K" possible availability values ($a_{ij}^k, k = 1, 2, ...K$) with ($a_{ij}^1 = a_{ij}$). For each value k, the corresponding unavailability is reduced by ϵ, so that $u_{ij}^k = u_{ij}^{k-1} \cdot (1-\epsilon)$, where $u_{ij}^k = 1 - a_{ij}^k$. Reducing a link unavailability is analogous to reducing its expected downtime. Note that, in reality the k different options might not have fixed downtime differences within nor across links. Here, we choose a fixed ϵ for illustration purpose. In practice, not all links will likely have the same number of options as this depends on several factors (*e.g.,* the terrain, cable type, the associated cost, etc). However, we assume that this is the case here in order to simplify the model. The cost associated with each improvement step k is calculated by a cost function, $c_{ij}^k = f_c(a_{ij}^k, a_{ij}^1)$.

A precise formula for cost of availability is difficult to determine in practice. Instead, many researcher rely on some mathematically known models (*e.g.,* constant, linear, quadratic, etc) to relate cost to availability [23]. Also, note that their are diminishing returns in that the cost of improving a link becomes larger as the availability gets higher [22]. For example, the cost of improving an already high availability link (*e.g.,* 0.999 to 0.9999) typically costs more than improving a link with a moderate availability (*e.g.,* 0.9 to 0.9009) by the same amount. We consider the following cost functions, f_c's, to compute the cost of improving the link availability per unit of length. The cost function, f_{c1}, is a polynomial in the availability improvement $\Delta a_{ij}^k = a_{ij}^k - a_{ij}^1$.

$$f_{c1}(a_{ij}^k, a_{ij}^1) = \left(a_{ij}^k - a_{ij}^1\right)^{\alpha} \qquad , \ k > 2 \tag{3}$$

where α is a scaling parameter. This function captures the idea that the greater the improvement in availability, the larger the cost. The second cost function, f_{c2}, is a polynomial in the availability improvement $\Delta a_{ij}^k = a_{ij}^k - a_{ij}^1$ but also weighted by the unavailability of the link. Hence for equal Δa_{ij}^k, it compounds the cost for the link with higher availability. This formula is very similar to f_2 in [24]

$$f_{c2}(a_{ij}^k, a_{ij}^1) = \left(\frac{a_{ij}^k - a_{ij}^1}{1 - a_{ij}^1}\right)^{\alpha} \qquad , \ k > 2 \tag{4}$$

The third cost function, f_{c3}, is derived from f_1 in [24] and models the notion that the impact of the cost on the improved availability decreases exponentially.

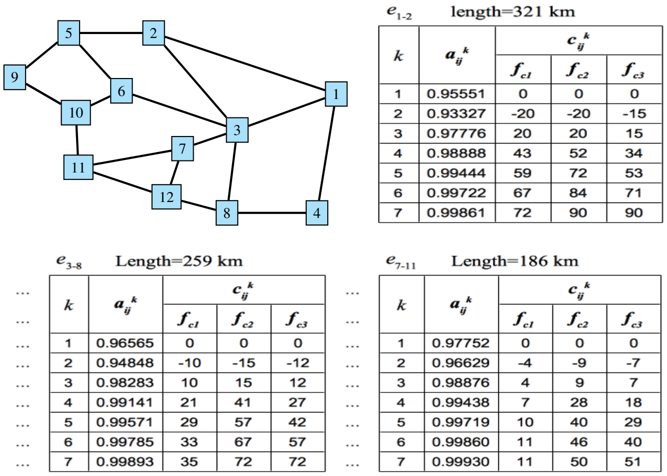
$$f_{c3}(a_{ij}^k, a_{ij}^1) = -\ln\left[\frac{1 - a_{ij}^k}{1 - a_{ij}^1}\right] \qquad , \ k > 2 \tag{5}$$

The length factor in the upgrade cost is included by the relationship,

$$c_{ij}^k = f_{cl}(a_{ij}^k, a_{ij}^1) \times d_{ij} \quad l = 1, 2, 3 \tag{6}$$

where d_{ij} is link (i, j) length. The exponent α in (3) and (4) was set to 2 to impose quadratic growth of the cost.

Figure 2 shows the Polska network topology and the availability options for three different links with $K = 7$. Each table in the figure shows the availability levels of a link and the corresponding cost for the different cost functions. In the figure, $k = 1$ denotes the initial link availability. The case of $k = 2$ models transferring maintenance capabilities between links. Therefore, we set $u_{ij}^2 = u_{ij}^1 \cdot (1+\epsilon)$ and $c_{ij}^2 = -c_{ij}^3$. Thus the expected downtime of a link with $k = 2$ (*i.e.,* off the spine link) would increase and incur negative cost which would reduce the total cost C. Subsequently, the availability of other links (*i.e.,* on the spine) can be improved by the relocation of maintenance and repair capabilities, and take advantage of transferring operational expenditure from degraded links.



e_{1-2} length=321 km

k	a_{ij}^k	c_{ij}^k		
		f_{c1}	f_{c2}	f_{c3}
1	0.95551	0	0	0
2	0.93327	-20	-20	-15
3	0.97776	20	20	15
4	0.98888	43	52	34
5	0.99444	59	72	53
6	0.99722	67	84	71
7	0.99861	72	90	90

e_{3-8} Length=259 km

k	a_{ij}^k	c_{ij}^k		
		f_{c1}	f_{c2}	f_{c3}
1	0.96565	0	0	0
2	0.94848	-10	-15	-12
3	0.98283	10	15	12
4	0.99141	21	41	27
5	0.99571	29	57	42
6	0.99785	33	67	57
7	0.99893	35	72	72

e_{7-11} Length=186 km

k	a_{ij}^k	c_{ij}^k		
		f_{c1}	f_{c2}	f_{c3}
1	0.97752	0	0	0
2	0.96629	-4	-9	-7
3	0.98876	4	9	7
4	0.99438	7	28	18
5	0.99719	10	40	29
6	0.99860	11	46	40
7	0.99930	11	50	51

Fig. 2: Polska network incremental link model.

III. The Spine Link Selection Design Problem

A. Notation

Indices:

ij represent a physical link by its two end-nodes, $i, j \in \mathcal{V}_P$, $(i, j) \in \mathcal{E}_P$.

st represent a connection/flow between two end-nodes, $s, t \in \mathcal{V}_P$, $(s, t) \in \mathcal{F}$.

Parameters:

k method of link improvement.

a_{ij}^k availability of link (i, j) after applying improvement option k, with $a_{ij}^1 = a_{ij}$, (a_{ij}: initial link availability).

$\widehat{a_{wp}}(\widehat{a_{bp}})$ flow WP (BP) availability target.

$A_{\mathcal{S}}$ ($A_{\mathcal{S}}^{WP}$) average end-to-end flow (flow WP) availability.

c_{ij}^k cost of improving link (i, j) using method k.

$H_{\mathcal{G}}$ total number of (undirected) links required by shortest (min-hop) path pairs.

Variables:

x_{ij} a binary variable indicating whether link (i, j) is selected on the spine (x_{ij}=1) or not (x_{ij}=0).

x_{ij}^{st} (y_{ij}^{st}) a binary variable denoting whether physical link (i, j) is used for routing the WP (BP) of connection (s, t).

r_{ij}^k a binary variable indicating if method k is used for link (i, j).

p_{ij}^{st} (q_{ij}^{st}) a continuous variable denoting link (i, j) unavailability given that it is on connection (s, t) WP (BP).

B. Optimization Model Formulation

The spine design problem aims at finding the best combination of links to form the spine and selecting the improvement options for all links in order to achieve a target end to end availability, while minimizing the cost. We route all flows on the spine with fully link-disjoint backup paths. This ensures that all high QoR priority traffic supported by the spine can be given 1+1 dedicated protection. Note that, this also enables 1:N shared protection, however this topic is left for future work. Furthermore, we assume that the QoR class of service needing high availability has traffic between all possible node-pairs (*e.g.,* a full mesh of demand of one unit between each node pair), hence we adopt a minimum spanning tree (MST) for the spine structure. The high QoR class flow availability is constrained to be greater or equal to target values ($\widehat{a_{wp}}$, $\widehat{a_{bp}}$). Instead of looking for the spine that maximizes the average availability, here we require that the minimum WP availability on the spine to be greater than or equal to a target availability, $\widehat{a_{wp}}$. As shown earlier, increasing the availability on the working path improves the end to end availability more effectively then improving all links at once. Finally, the objective C of the design problem aims at minimizing the total costs of embedding the spine and improving flow availabilities to reach or exceed the target value.

Given the notation above, the spine design problem can be formulated as an optimization model as follows:

$$\textbf{Minimize} \qquad C = \sum_{ij}\sum_{k} r_{ij}^{k} \times c_{ij}^{k} \tag{7}$$

s.t.

WP and BP computation:

$$\sum_{hj\in\mathcal{E}_P} x_{hj}^{st} - \sum_{ih\in\mathcal{E}_P} x_{ih}^{st} = \begin{cases} 1 & \text{if } h=s \\ -1 & \text{if } h=t \\ 0 & \text{otherwise} \end{cases} \quad ,\forall h\in\mathcal{V}_P,(s,t)\in\mathcal{F} \tag{8}$$

$$\sum_{hj\in\mathcal{E}_P} y_{hj}^{st} - \sum_{ih\in\mathcal{E}_P} y_{ih}^{st} = \begin{cases} 1 & \text{if } h=s \\ -1 & \text{if } h=t \\ 0 & \text{otherwise} \end{cases} \quad ,\forall h\in\mathcal{V}_P(s,t)\in\mathcal{F} \tag{9}$$

Loopless Routing:

$$x_{ij}^{st} + x_{ji}^{st} \le 1, \quad y_{ij}^{st} + y_{ji}^{st} \le 1 \quad ,\forall (i,j)\in\mathcal{E}_P,(s,t)\in\mathcal{F} \tag{10}$$

$$\sum_{\substack{j\in\mathcal{V}_P\\ ij\in\mathcal{E}_P}} x_{ij}^{st} + \sum_{\substack{h\in\mathcal{V}_P\\ hi\in\mathcal{E}_P}} x_{hi}^{st} \le 2 \quad ,\forall i\in\mathcal{V}_P,(s,t)\in\mathcal{F} \tag{11}$$

$$\sum_{\substack{j\in\mathcal{V}_P\\ ij\in\mathcal{E}_P}} y_{ij}^{st} + \sum_{\substack{h\in\mathcal{V}_P\\ hi\in\mathcal{E}_P}} y_{hi}^{st} \le 2 \quad ,\forall i\in\mathcal{V}_P,(s,t)\in\mathcal{F} \tag{12}$$

Disjointness constraints:

$$x_{ij}^{st} + y_{ij}^{st} \le 1 \quad x_{ji}^{st} + y_{ji}^{st} \le 1 \quad ,\forall (i,j)\in\mathcal{E}_P,(s,t)\in\mathcal{F} \tag{13}$$

Hop-count constraint:

$$H_{\mathcal{S}} = \sum_{ij\in\mathcal{E}_P}\sum_{st\in\mathcal{F}} (x_{ij}^{st} + y_{ij}^{st}) \;\le\; \delta \times H_{\mathcal{G}}, \qquad \delta \ge 1 \tag{14}$$

MST formation:

$$x_{ij} \ge x_{ij}^{st}, \quad \forall (s,t)\in\mathcal{F} \quad (15) \qquad x_{ij} = x_{ji}, \quad \forall (i,j)\in\mathcal{E}_P \quad (16)$$

$$\sum_{\substack{ij\in\mathcal{E}_P\\ i<j}} x_{ij} \le |\mathcal{V}_P| - 1 \tag{17}$$

Availability constraints:

$$\sum_{k} r_{ij}^{k} = 1, \quad \forall (i,j)\in\mathcal{E}_P \quad (18) \qquad r_{ij}^{k} = r_{ji}^{k} \quad ,\forall (i,j)\in\mathcal{E}_P \quad (19)$$

$$p_{ij}^{st} = x_{ij}^{st} \times \sum_{k} r_{ij}^{k}\,(1 - a_{ij}^{k}) \quad ,\forall (i,j)\in\mathcal{E}_P,(s,t)\in\mathcal{F} \tag{20}$$

$$q_{ij}^{st} = y_{ij}^{st} \times \sum_{k} r_{ij}^{k}\,(1 - a_{ij}^{k}) \quad ,\forall (i,j)\in\mathcal{E}_P,(s,t)\in\mathcal{F} \tag{21}$$

Flow availability targets:

$$A_{st}^{WP} = 1 - \sum_{ij\in\mathcal{E}_P} p_{ij}^{st} \ge \widehat{a_{wp}} \tag{22}$$

$$A_{st}^{BP} = 1 - \sum_{ij\in\mathcal{E}_P} q_{ij}^{st} \ge \widehat{a_{bp}} \tag{23}$$

Variables:

$$x_{ij}^{st}, y_{ij}^{st}, x_{ij}, r_{ij}^{k} \quad \text{binary} \quad (24) \qquad p_{ij}^{st}, q_{ij}^{st} \quad \in [0,1] \quad (25)$$

The core of the formulation is the flow conservation constraints (8) and (9), which find primary and backup paths for all flows. A flow conservation constraint pushes a unit of demand along a path between the two end-nodes of a given flow. Constraint sets (10)-(12) ensure loop free routing. Constraint set (13) ensures that for each flow the primary and backup paths are fully link-disjoint. The sum of these paths $H_{\mathcal{S}}$, however, is constrained by (14) which sets a maximum limit for the hop count of the sum of the link-disjoint path-pairs where δ is a scaling factor and $H_{\mathcal{G}}$ is the total length of the shortest path pairs between all node-pairs in a network.

Each link used by a primary path of any flow is considered as an on-spine link. Constraint (15) enforces this by turning the spine link selector variable for a link, x_{ij}, to 1 if the link is used in a primary path of at least one flow. Due to the route of different (s,t) flows and equation (15), both x_{ij} and x_{ji} may take the value 1, but not necessarily both every time. Hence, because the network is undirected, constraint (16) is required. Then, constraint (17) limits the number of the (undirected) links selected for the spine to $|\mathcal{V}_P| - 1$ which is the number of links for a MST. Next, for the availability constraints, constraint set (18) ensures that only one improvement method is selected for each link. Constraint (19) requires that a link has the same improvement method in both directions. Constraint sets (20) and (21) are used to relate a flow WP and BP unavailability to the unavailability of each link along the flow path. Variable p_{ij}^{st} or q_{ij}^{st} will have an unavailability value only if flow (s,t) WP or BP is routed through link (i,j). These two sets of constraints, turn the optimization problem into a integer nonlinear programming (INLP) model, because the product of two variables *i.e.,* x_{ij}^{st} with r_{ij}^{k} in (20) and y_{ij}^{st} with r_{ij}^{k} in (21). Note that, to compute a single path availability for a given flow, one can multiply the availability of the links along the path, but this results in a nonlinearity. Instead, we use the approximate version of the unavailability formula for a system connected in series, ($u^{st} \approx \sum_{ij}\ u_{ij}^{st}$). Hence, WP availability can be computed as $(1-\sum_{ij}\ p_{ij}^{st})$. BP availability is computed in the same way. Constraints (22) and (23) require that a flow WP and BP are above target availability values $\widehat{a_{wp}}$ and $\widehat{a_{bp}}$, respectively. Lastly, constraint sets (24) and (25) declare binary and continuous variables.

To remove the nonlinearity of the INLP, constraints set (20) can be replaced with constraint sets eqs. (26) to (28). The three constraints provide the same function as (20). Similarly, constraints set (21) that computes BP unavailability can be replaced with the set of eqs. (29) to (31):

$$p_{ij}^{st} \le x_{ij}^{st} \tag{26}$$

$$p_{ij}^{st} \le \sum_k r_{ij}^k\ (1-a_{ij}^k) \tag{27}$$

$$p_{ij}^{st} \ge x_{ij}^{st} + \sum_k r_{ij}^k\ (1-a_{ij}^k) - 1 \qquad \forall (i,j) \in \mathcal{E}_P, (s,t) \in \mathcal{F} \tag{28}$$

$$q_{ij}^{st} \le y_{ij}^{st} \tag{29}$$

$$q_{ij}^{st} \le \sum_k r_{ij}^k\ (1-a_{ij}^k) \tag{30}$$

$$q_{ij}^{st} \ge y_{ij}^{st} + \sum_k r_{ij}^k\ (1-a_{ij}^k) - 1 \qquad \forall (i,j) \in \mathcal{E}_P, (s,t) \in \mathcal{F} \tag{31}$$

IV. Numerical Study

We studied the Polska, Spain, and Italia14 network topologies drawn from published literature (see [25]) but due to space limitations only report results from the Polska network. We set $K = 7$, and δ in constraint (14) is set to 110%, allowing for a maximum of 10% increase in total resources over the resources required by shortest path-pairs, $H_{\mathcal{G}}$ (*i.e.,* $H_{\mathcal{S}} \le 1.1H_{\mathcal{G}}$). We solved the spine design optimization problem for the test networks using AMPL/Gurobi. Here, the availability goal for WP, $\widehat{a_{wp}}$ in Equation (22), is set to $\widehat{a_{wp}} = \{0.99, 0.995, 0.996, 0.9964\}$. At each each $\widehat{a_{wp}}$ setting, we solved the model for the three cost functions. The solution time to achieve 0% integrality gap ranged from 11 to 687 seconds. Here we ignore the $\widehat{a_{bp}}$ constraint. Note, that increasing the target value $\widehat{a_{wp}}$ increases the total cost of the design and may also result in a different spine layout.

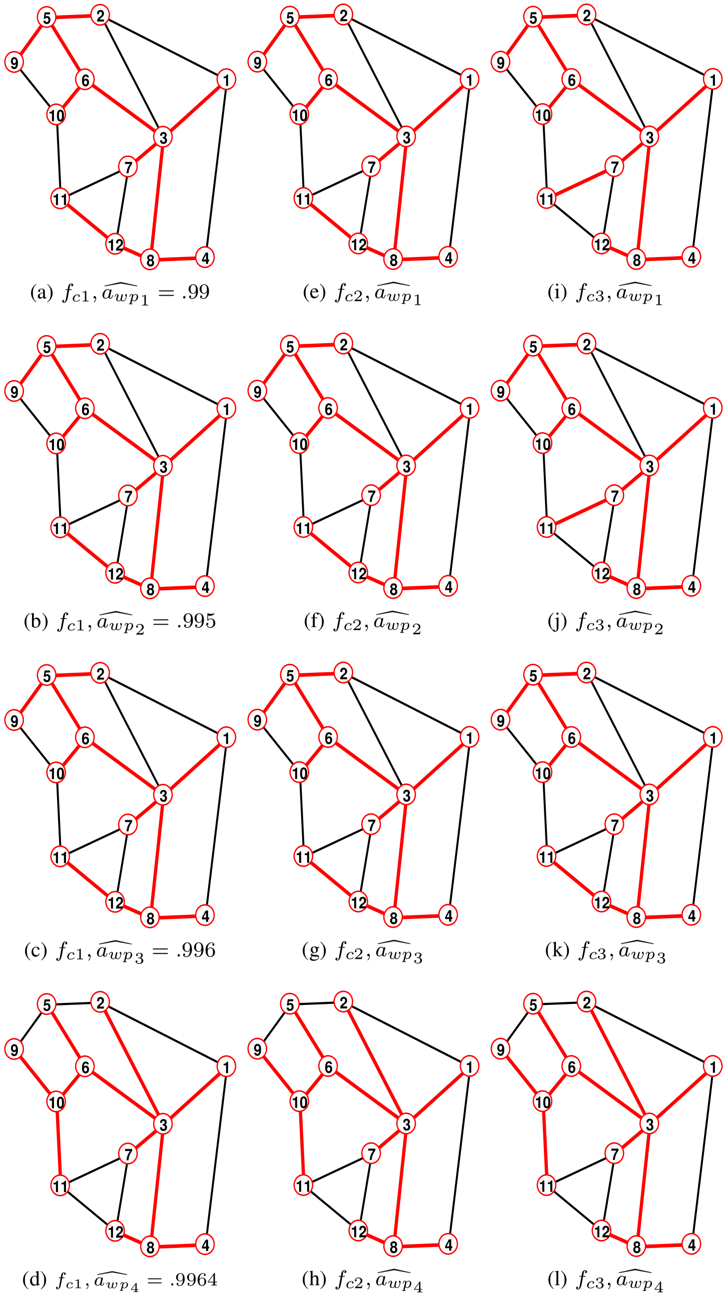
Figure 3 shows the spine layouts obtained for the Polska network, as the target $\widehat{a_{wp}}$ increases. First, the spine topology varies slightly as the target availability, $\widehat{a_{wp}}$, or the cost function changes. Though one can see that there is a persistence substructure that appears in almost all the spines, *e.g.,* the star-like substructure rooted at node 3. Table I shows the graph theoretic structural properties of the spines, namely: eb_S the average edge betweeness centrality of the spine, ed_S the average edge degree of the spine, h_S the average shortest path on the spine, and di_S the diameter of the spine. The first row shows the corresponding measures for the full physical layer graph $\mathcal{G}_P$ with no spine. Observe that the spines tend to have a comparatively small edge betweenness eb_S and average shortest path h_S and a large edge degree ed_S. Only in a few cases does the corresponding measure match the minimum (or maximum) value found by generating all MSTs for the network as given in [3]. However, these results are consistent with the findings of the heuristic algorithm (reported in [2, 3]) with respect to the spine that maximizes A_S^{WP} and indicate that the spines tend to have a star-like rather than a ring-like layout.

cost function	scenario	eb_S	ed_S	h_S	di_S
	$\mathcal{G}_P$	0.12	6.33	2.14	4
f_{c1}	$\mathcal{G}_S, \widehat{a_{wp}}_1$	0.26	4.91	2.89	6
	$\mathcal{G}_S, \widehat{a_{wp}}_2$	0.26	4.91	2.89	6
	$\mathcal{G}_S, \widehat{a_{wp}}_3$	0.26	4.91	2.89	6
	$\mathcal{G}_S, \widehat{a_{wp}}_4$	0.24	5.45	2.68	5
f_{c2}	$\mathcal{G}_S, \widehat{a_{wp}}_1$	0.26	4.91	2.89	6
	$\mathcal{G}_S, \widehat{a_{wp}}_2$	0.26	4.91	2.89	6
	$\mathcal{G}_S, \widehat{a_{wp}}_3$	0.26	4.91	2.89	6
	$\mathcal{G}_S, \widehat{a_{wp}}_4$	0.24	5.45	2.68	5
f_{c3}	$\mathcal{G}_S, \widehat{a_{wp}}_1$	0.26	4.91	2.82	5
	$\mathcal{G}_S, \widehat{a_{wp}}_2$	0.26	4.91	2.82	5
	$\mathcal{G}_S, \widehat{a_{wp}}_3$	0.26	4.91	2.89	6
	$\mathcal{G}_S, \widehat{a_{wp}}_4$	0.24	5.45	2.68	5

TABLE I: Structural properties of the spines.

Moreover, as the spine layout might be attributed to the structural importance of the links and nodes, it is also shaped by the cost associated with the links and their availability as well as the hop-count constraint.

It was observed that similar spines within and across cost functions are likely to have different availability and link types. This is illustrated by Figure 4 where downtime per year for each link versus the link length and the link improvement method/type k selected for each link is shown for the cases corresponding to Figure 3. In the figures, each circle represents a link and the number inside the circle is the improvement method/type k. The red circles represent links comprising the spine and the blue are the off spine links. For example, the spines obtained with $\widehat{a_{wp4}} = 0.9964$ for cost functions f_{c1} and f_{c2} are identical in the layout, as shown in Figures 3d and 3h, but the corresponding link improvement method k of the links are different as shown in Figures 4d and 4h. However, the spine and the selected methods are identical for f_{c2} and f_{c3}, whereas, they were completely different for $\widehat{a_{wp1}}$ as shown in Figures 3e and 3i. One can also see that, for the same cost function, different link improvement methods k can be selected as the WP availability target $\widehat{a_{wp}}$ increases. For example, the first three spines obtained for the Polska network for cost function f_{c1}, shown in Figures 3a to 3c, have different link improvement assignments as the WP availability target $\widehat{a_{wp}}$ changes. This is shown in the downtime and availability assignment Figures 4a to 4c. Initially, shorter links (*i.e.,* with higher availability and lower improvement cost) are favored as a spine link, thus exploiting existing heterogeneity. As the availability target increases, expensive links are selected to meet the more stringent requirement. For example, consider how the spine layout changes from the initial one of Figure 3a with $\widehat{a_{wp1}}$, to that of Figure 3d in order to achieve $\widehat{a_{wp4}}$. Note that link (2,3) (that is, the third longest link) is selected to be on the spine despite its high cost. Also Table I shows that the spine for $\widehat{a_{wp4}}$ has better structural measures (*e.g.,* smaller di_S) than the spine for $\widehat{a_{wp1}}$. The results for Spain and Italia14



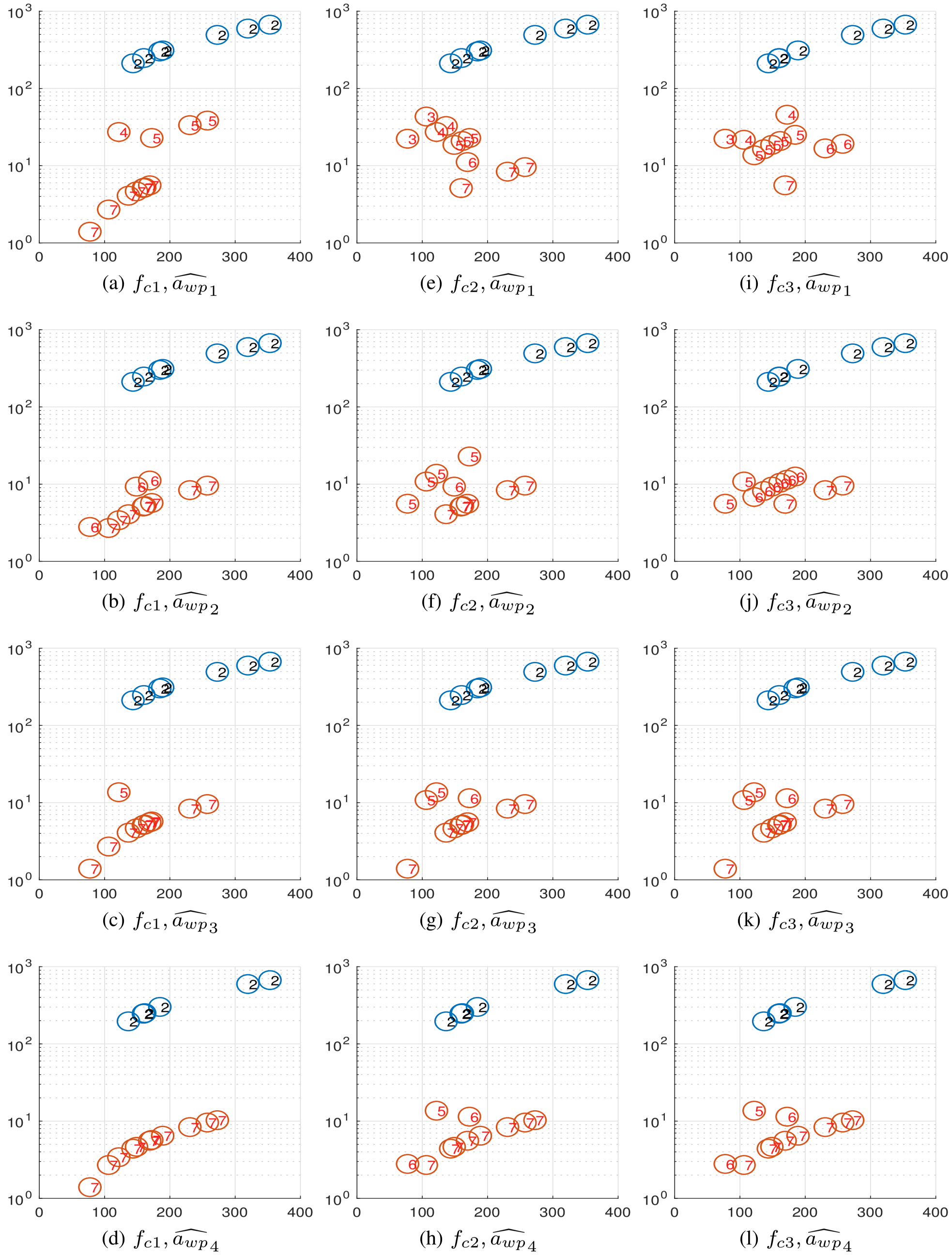
(a) $f_{c1}, \widehat{a_{wp_1}} = .99$ (e) $f_{c2}, \widehat{a_{wp_1}}$ (i) $f_{c3}, \widehat{a_{wp_1}}$

(b) $f_{c1}, \widehat{a_{wp_2}} = .995$ (f) $f_{c2}, \widehat{a_{wp_2}}$ (j) $f_{c3}, \widehat{a_{wp_2}}$

(c) $f_{c1}, \widehat{a_{wp_3}} = .996$ (g) $f_{c2}, \widehat{a_{wp_3}}$ (k) $f_{c3}, \widehat{a_{wp_3}}$

(d) $f_{c1}, \widehat{a_{wp_4}} = .9964$ (h) $f_{c2}, \widehat{a_{wp_4}}$ (l) $f_{c3}, \widehat{a_{wp_4}}$

Fig. 3: Polska network spines using the three cost functions.



(a) $f_{c1}, \widehat{a_{wp_1}}$ (e) $f_{c2}, \widehat{a_{wp_1}}$ (i) $f_{c3}, \widehat{a_{wp_1}}$

(b) $f_{c1}, \widehat{a_{wp_2}}$ (f) $f_{c2}, \widehat{a_{wp_2}}$ (j) $f_{c3}, \widehat{a_{wp_2}}$

(c) $f_{c1}, \widehat{a_{wp_3}}$ (g) $f_{c2}, \widehat{a_{wp_3}}$ (k) $f_{c3}, \widehat{a_{wp_3}}$

(d) $f_{c1}, \widehat{a_{wp_4}}$ (h) $f_{c2}, \widehat{a_{wp_4}}$ (l) $f_{c3}, \widehat{a_{wp_4}}$

Fig. 4: Polska network link downtime/year vs. length

networks exhibit similar observations. Notice that the off spine links are selected as type $k = 2$ which are less reliable, but provide additional budget resources to improve the spine links from $k = 1$ to better quality links (*i.e.,* $k > 2$).

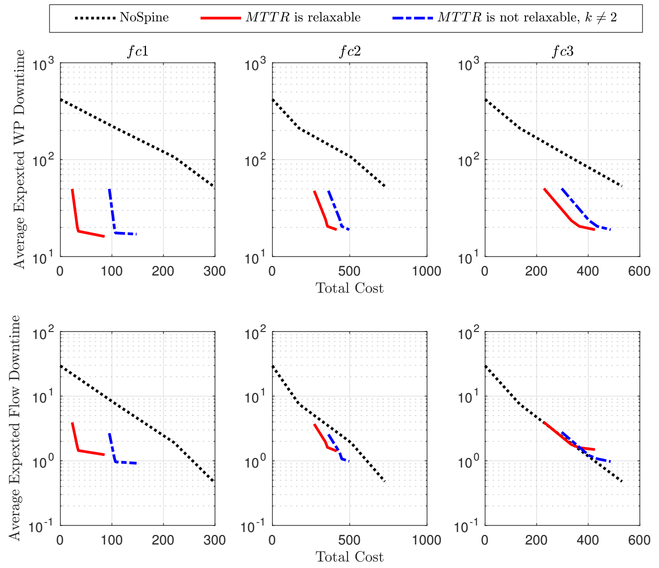


Fig. 5: Average expected WP and end-to-end flow downtime/year versus cost for the Polska network

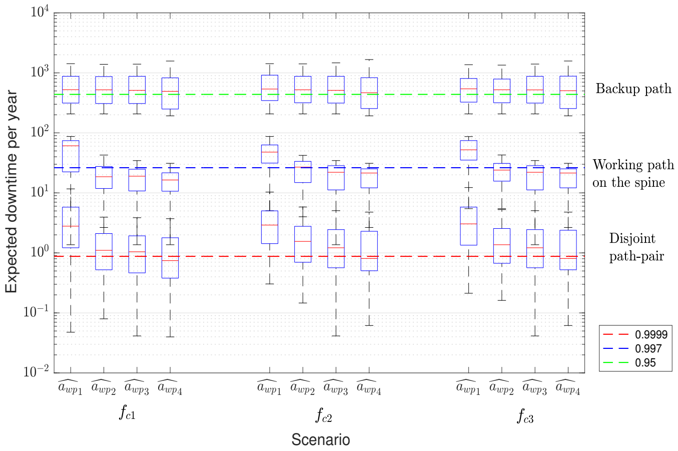


Fig. 6: The range of expected downtimes in the spines of the Polska network

In addition to the structural properties, we compare the average expected flow downtime $d_{\mathcal{S}} = (1 - A_{\mathcal{S}}) \times 8760\ hrs/year$ and average expected WP downtime $d^{WP} = (1 - A^{WP}_{\mathcal{S}}) \times 8760\ hrs/year$ for the different scenarios and cost functions. We also include the corresponding downtime of an equivalent network with no spine and considering link-disjoint path pairs. In the no spine network case, all links are improved using the same method, k, and the total cost C is computed accordingly. Figure 5 shows the average expected WP and end-to-end flow downtimes for for the Polska network considering the different cost functions and $\widehat{a_{wp}}$'s. Also the results are shown for three cases: no spine (dotted line) which is the baseline case; then the case where the MTTR is relaxable for the off spine links and the third that forbids relaxing the MTTR for the off spine links. From the figures we can make a number of observations. First, there is significant improvement in the downtime values over the no spine model when using cost function f_{c1}, slight improvement for the case of f_{c2}, and no improvement with f_{c3}. Lastly, comparing relaxable and non-relaxable MTTR cases shows a significant saving in cost when relaxing the off spine links MTTR. The results for the Spain network were similar to the Polska network. However, the Italia14 network which is more dense, achieves lower downtime in all cases across all cost functions these results are not shown due to page limits.

Recall that the spine concept aims to create different levels of availability and also meet the most stringent availability requirements. Figure 6 shows the expected downtime for each path type for the optimal spines obtained for the Polska network. The downtime results are represented for each scenario as a box plot. The upper and lower edges of each box represent the third and first quartile of the values, respectively, the middle bar (in red) represents the median, and the upper and lower bars represent the maximum and minimum downtime values across all paths, respectively. Note that, even for the spine with the lowest cost (*i.e.,* $\widehat{a_{wp_1}}$ and relaxable $MTTR$), there are three different levels of availability classes resulting from using only one protection scheme. The lower availability class can be given an unprotected path equivalent to the backup path with large expected downtime. Then, the middle class is routed on an unprotected path on the spine which achieves shorter expected downtime compared to the lower class. The higher class is routed on the spine and protected by a link-disjoint backup path, and its expected downtime is minimal. One also can see from the graph that the target availability also controls the downtime of the higher class since the WP of this class is routed on the spine, and its downtime decreases as the target availability increases. The spacing between each level of availability is mainly determined by the range of link availabilities (initial and improved) and the WP target availability.

V. Conclusions

In this paper, we revisit the spine concept of embedding a subgraph structure with higher availability in a network together with protection mechanisms aiming to improve the overall end-to-end availability. We provided an optimization based formulation for designing the spine taking into consideration that links availability are upgradeable for a given cost. The design problem aims at exploiting existing link availability heterogeneity and the upgradeability of links to achieve a target flow availability while minimizing the total cost. Our results demonstrate the spine model efficiency in terms of average flow availability and potential advantage over the shortest path model with no spine. This efficiency, however, depends primarily on network density and link improvement cost distribution.

Acknowledgment

T. Gomes work has been supported by the Portuguese Foundation for Science and Technology (FCT) under project grant UID/MULTI/00308/2013.

References

[1] D. Tipper, "Resilient network design: Challenges and future directions," *Telecommunication Systems*, vol. 56, no. 1, pp. 5–16, 2014.

[2] T. Gomes, D. Tipper, and A. Alashaikh, "A novel approach for ensuring high end-to-end availability: The spine concept," in *10th International Conference Design of Reliable Communication Networks (DRCN)*. IEEE, 2014, pp. 1–8.

[3] A. Alashaikh, T. Gomes, and D. Tipper, "The spine concept for improving network availability," *Computer Networks*, vol. 82, pp. 4–19, May 2015.

[4] H. Zhang and A. Durresi, "Differentiated multi-layer survivability in IP/WDM networks," *IEEE/IFIP Network Operations and Management Symposium, NOMS 2002*, pp. 681–694, 2002.

[5] M. Tornatore, D. Lucerna, B. Mukherjee, and A. Pattavina, "Multilayer Protection with Availability Guarantees in Optical WDM Networks," *Journal of Network and Systems Management*, vol. 20, no. 1, pp. 34–55, Sep. 2012.

[6] L. Zhou, M. Held, A. Member, and U. Sennhauser, "Connection Availability Analysis of Shared Backup Path-Protected Mesh Networks," *Journal of Lightwave Technology*, vol. 25, no. 5, pp. 1111–1119, 2007.

[7] M. Xia, M. Tornatore, S. Sevilla, L. Shi, C. U. Martel, and B. Mukherjee, "A Novel SLA Framework for Time-Differentiated Resilience in Optical Mesh Networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 3, no. 4, pp. 312–322, 2011.

[8] B. Kantarci, H. T. Mouftah, and S. Oktug, "Arranging Shareability Dynamically for the Availability-Constrained Design of Optical Transport Networks," *Proceedings - IEEE Symposium on Computers and Communications*, pp. 68–73, 2008.

[9] D. A. Garbin, J. E. Knepley, F. Park, D. South, and F. Church, "Design and Analysis of High Availability Networks," in *IEEE Conference on Technologies for Homeland Security*, 2009, pp. 1–6.

[10] S. Verbrugge, D. Colle, P. Demeester, R. Huelsermann, and M. Jaeger, "General availability model for multilayer transport networks," in *Design of Reliable Communication Networks, 2005. DRCN 2005. 5th IEEE International Workshop on*, 2005.

[11] V.Y.Liu and D. Tipper, "Spare capacity allocation using shared backup path protection for dual link failures," *Computer Communications*, vol. 36, no. 6, pp. 666–677, 2013.

[12] J. Zhang, E. Modiano, and D. Hay, "Enhancing network robustness via shielding," in *11th International Conference on the Design of Reliable Communication Networks (DRCN)*, March 2015, pp. 17–24.

[13] ——, "Enhancing Network Robustness via Shielding," *IEEE/ACM Transactions on Networking*, vol. 25, no. 4, pp. 2209–2222, August 2017.

[14] Q. Botton, B. Fortz, and L. Gouveia, "On the hop-constrained survivable network design problem with reliable edges," *Computers & Operations Research*, vol. 64, pp. 159–167, 2015.

[15] P. Pacharintanakul and D. Tipper, "Crosslayer survivable mapping in overlay-IP-WDM networks," in *Proceedings of the 2009 7th International Workshop on the Design of Reliable Communication Networks, DRCN 2009*. IEEE, 2009, pp. 168–174.

[16] M. Rausand and R. Hoyland, *System Reliability Theory: Models, Statistical Methods and Applications*, 2ed, Ed. Wiley Interscience, 2003.

[17] A. Alashaikh, D. Tipper, and T. Gomes, "Exploring the Logical Layer to Support Differentiated Resilience Classes in Multilayer Networks," *Annals of Telecommunications*, pp. 1–17, 2017.

[18] ——, "Supporting differentiated resilience classes in multilayer networks," in *12th International Conference on the Design of Reliable Communication Networks (DRCN)*. IEEE, mar 2016, pp. 31–38.

[19] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Morgan Kaufmann Publishers, Elsevier, 2004.

[20] I. Rados, T. Sunaric, and P. Turalija, "Suggestions for Availability Improvement of Optical Cables," in *Circuits and Systems for Communications, 2002. Proceedings. ICCSC '02. 1st IEEE International Conference on*. IEEE, 2002, pp. 234–239.

[21] H.-Y. Chang, I.-K. Chen, and W.-C. Chiang, "A Simple Network-Engineering Approach for Improving Connection Availability in WDM Networks," *2011 IEEE Workshops of International Conference on Advanced Information Networking and Applications*, pp. 805–807, Mar. 2011.

[22] E. Marcus and H. Stern, *Blueprints for High Availability*, 2nd ed. Wiley Publishing, 2003.

[23] T. Nagae and H. Wakabayashi, "Differences in Network Reliability Improvement by Several Importance Indices," *Transportation Research Procedia*, vol. 10, pp. 155–165, 2015.

[24] U. Franke, "Optimal IT service availability: Shorter outages, or fewer?" *IEEE Transactions on Network and Service Management*, vol. 1, no. 9, pp. 22–33, 2012.

[25] S. Orlowski, M. Pióro, A. Tomaszewski, and R. Wessäly, "SNDlib 1.0–Survivable Network Design library," *Networks*, vol. 55, no. 3, pp. 276–286, 2010.