SDN Controller Placement with Availability Upgrade under Delay and Geodiversity Constraints

Dorabella Santos
INESC Coimbra, DEEC,
Rua Sílvio Lima, Polo II, 3030-290 Coimbra, Portugal
dorabella.santos@gmail.com

Teresa Gomes
University of Coimbra,
Department of Electrical and Computer Engineering,
Rua Sílvio Lima, Polo II, 3030-290 Coimbra, Portugal
INESC Coimbra, DEEC,
Rua Sílvio Lima, Polo II, 3030-290 Coimbra, Portugal
teresa@deec.uc.pt

David Tipper
University of Pittsburgh
Department of Informatics and Networked Systems
Pittsburgh, PA, USA, 15260
dtipper@pitt.edu

***Abstract*—An inherent problem in Software-Defined Networking (SDN) is the Controller Placement Problem, which addresses how many controllers to deploy in the network, and where to place them. Several variants of this problem have been addressed and researched to find the placements that adapt best to different contexts. In this paper, we address a more complex variant of this problem, to satisfy QoS requirements and to offer robustness against disaster-based failures. We address the joint optimization problem of controller placement and finding a tree subgraph which can be upgraded to have enhanced availability, in order to satisfy delay and availability constraints. Additionally, we consider geodiversity constraints as a way to enhance robustness to disaster-based failures.**

***Index Terms*—Software-Defined Networking, geodiversity, availability, controller placement problem, integer linear programming**

I. Introduction

Resiliency and availability are gaining great importance in the context of Software-Defined Networking (SDN). Due to the decoupling of the data and control planes, SDN offers network operators simplified network management and rapid innovation. Therefore, SDN networks are becoming very attractive not only for datacenters [1] and campus networks, but there is also significant effort to deploy SDN in transport networks [2].

Hence, SDN networks must provide high end-to-end availability and resiliency not only against single link/node failures, but also against disaster-based failures. Since the risk of natural disasters is increasing, as well as man-made attacks to communication networks, large scale failures have become a concern [3]. Geodiverse routing can be used to mitigate the impact of these failures, however at the expense of longer paths, making it difficult to achieve the required availability levels.

In this work, we address a joint optimization problem which involves two major subproblems. The first major subproblem is the Controller Placement Problem (CPP), introduced in [4] and shown to be NP-hard. This problem tackles the question of how many controllers and where to place them in the network, to offer the desired control plane performance given the context in study. In this paper, we consider constraints for QoS performance in the SDN networks translated into delay and availability requirements. To obtain acceptable performance we consider intercontroller maximum delay constraints and also maximum delay constraints between the switches and the controllers that manage them [5], [6], [7], [8], [9].

Single link and node failures are the most common failures, and these can be mitigated by considering path redundancy and even controller redundancy [10]. In this paper, we consider controller redundancy to offer protection against single link and node failures, consequently increasing the end-to-end availability in the network. However, path and controller redundancy may not be sufficient to guarantee the level of availability required for critical services [11]. We therefore, consider the spine concept proposed in [12], where a high-availability subgraph is chosen, together with other protection mechanisms, to guarantee the required availabilities in the network. Improving the availability of a subset of links of the network (a spanning tree) was shown [11] to be a cost effective approach to attain desired target flow availabilities without modifying the network topology. In [11] the scenario for link availability upgrade was using strategies to increase the mean time between failures of each link (e.g., burying an aerial cable) and/or decreasing the mean time to repair (e.g., better maintenance).

In this way, the second major subproblem in our optimization problem is the selection of such a subgraph and the selection of which links should be upgraded and how much. We consider the availability link upgrade strategy used in [11], and consider that the required target availability constraints are guaranteed along the connections between switches and controllers. Hence, we impose the subgraph to be a tree, since we aim to minimize the cost of upgrading the links belonging to the subgraph. The joint optimization CPP and spine design problem was presented in [13], considering path redundancy for the control paths – primary and backup paths between the switches and their controllers.

Finally, we also consider geodiversity routing to mitigate the impact of disaster-based failures, in this work. Path geodiversity has been proposed as a strategy to enhance robustness against such failures [14], [15]. In this work, we consider the path geodiversity strategy proposed in [16], [17], where each pair of paths are geographically separated by a given minimum distance. This results in a pair of longer paths, which is conflicting with the delay and availability constraints – we assume the delay is proportional to the shortest path distance [4], [6], and that the availability is given by a distance dependent expression [11].

In summary, we address the joint controller placement and availability link upgrade optimization problem for SDN networks, aiming to minimize the upgrade cost, while delay, geodiversity and availability requirements are guaranteed. We assume in-band control. The novelty of this work in comparison with [13] is including controller redundancy for protection against single link and controller failures, and geodiversity for protection against disaster-based failures. In the computational results, we compare the path redundancy, controller redundancy and geodiverse solutions in terms of the upgrade cost. In this work, the spine is initially considered to be a spanning tree as in [13], but then a pruning process is performed to eliminate unnecessary links in the final tree.

This paper is organized as follows. In Section II, we discuss the related work and point out the differences with our work. In Section III, the CPP is presented considering controller redundancy and delay constraints. In Section IV, we present the availability link upgrade model and the constraints for the target availability guarantees. In Section V, we present the D-geodiversity concept adopted and the constraints that guarantee the D-geodiversity for each pair of primary and backup paths. In Section VI, we define the joint optimization problem. In Section VII, we present numerical results for sample networks illustrating the upgrade cost versus the number of controllers trade-off for path redundancy, controller redundancy and geodiversity. Finally, in Section VIII, we present our conclusions.

II. Related work

In [11], the spine concept was explored for generic backbone networks. They consider path redundancy between end nodes, and show that improving the availability on the primary path has a more significant impact on the overall availability, than improving the availability on the backup paths or on both paths simultaneously. Furthermore, they present the incremental link availability model, that we also adopt. They consider three different cost functions for upgrading the link availability and we adopt the third function which is more realistic. An ILP model is proposed where the spine is imposed to be a spanning tree which we also consider, although we then prune the spanning tree to remove unnecessary links from it. The ILP model also accommodates improvement to the availability of backup paths, but then in the computational results only improvement to the availability of the primary paths was considered in accordance to the impact study.

In [18], the spine is adopted and the availability enhancements are considered only for the primary paths. Two objectives are studied: the maximization of the sum of the primary paths availabilities, and the maximization of the minimization (max-min) of the primary paths availabilities. An ILP model is presented and a heuristic approach is also proposed.

In [19], the effect of the spine is also studied in addition to several protection schemes. Independent random failures and regional failures modelled using Shared Risk Link Group (SRLG) are considered.

None of these works, [11], [18] and [19], consider SDN networks. In [20], the authors address the CPP in SDN networks to maximize reliability of the control paths, i.e., of the paths connecting each controller to the set of switches it manages. They propose a greedy heuristic and a simulated annealing meta-heuristic as approaches to solve their problem. We also address the CPP, but we aim to guarantee a target end-to-end availability.

In [21], the authors consider the CPP with controller redundancy, where each switch connects to a primary controller and to one or more backup controllers. They propose an ILP model guaranteeing that the availability of the control paths is at least a given target value. They also propose a heuristic method for their problem, and show that each node is required to connect to 2 or 3 controllers to achieve the target availability.

In [22], the resilient capacitated CPP was addressed considering multiple controller failures. The authors also consider switch-controller and intercontroller delays. The capacity of the controllers are taken into account to meet the traffic load of the switches. However, the authors do not consider availability target constraints.

In [23], the capacitated CPP was also addressed, but without resiliency features. The authors consider heterogeneous controllers with different capacities. The problem is formulated as an optimization problem in two phases: the first phase is to select the minimum cost set of controllers to be deloyed in the network; in the second phase with the selected set of controllers, they place the controllers in order to minimize three objectives: the maximum imbalance between unused controller processing capacity and the maximum switch-controller and intercontroller delays. They present the ILP models and propose a greedy heuristic for the first phase, and an anytime Pareto local search for the second phase.

In [24], the CPP is addressed as a multiobjective optimization problem and the authors propose a heuristic for large networks. While they do consider delay constraints, they do not consider target availability constraints.

In this paper, we consider that each switch connects to a primary and a backup controller, and so it is not always possible to achieve the required end-to-end availability. Therefore, we adopt the spine approach and jointly optimize the CPP and spine selection and upgrade.

We presented the joint optimization of the CPP and spine design problem in [13]. We considered path redundancy (protection against single link failures) and the high-availability subgraph was imposed to be a spanning tree Also the incremental availability link upgrade model used in [11] was considered. Now, we further extend this work to include controller redundancy and geodiversity to enhance robustness. We also perform a pruning process to the spanning tree, in order to remove unnecessary links.

None of these works consider geodiversity to offer resiliency to large scale failures. In [25], a routing protocol was proposed to provide multiple geographically diverse paths to end nodes, in optical networks. This protocol was extended in [26], to consider path delay and traffic skew constraints.

In [27], the problem of availability link upgrade with geodiversity constraints has been addressed in the context of generic optical networks. The authors consider the availability link upgrade given by a parallel link, which is an alternative path installed with the same availability as the original link.

In [28], anycast routing was considered where a given number of anycast nodes were chosen. This arises as a CPP variant for SDN networks. A pair of D-geodiverse paths was guaranteed for each source node to two distinct anycast nodes. The geodiversity concept adopted was that of [17], which is also the one we adopt in this work. However, delay and availability guarantees were not considered.

III. Controller Placement Problem

The CPP is the problem of how many controllers to place in the network and where to place them. Due to control plane performance issues, we assume maximum values for the delays between the switches and their primary controllers and between the controllers themselves can be specified. More precisely, the delays between each switch and its primary controller (SC delays) cannot exceed D_{sc}, while the delays between any two controllers (CC delays) cannot exceed D_{cc}. Since the communication between the switches and controllers are more frequent than between the controllers themselves, we have that $D_{sc} < D_{cc}$. In the computational results, we consider that the maximum delay values D_{sc}, D_{cc} are given as percentages of the graph diameter D_g for consistency across various topologies, as in existing works such as [6], [22], [24], where the D_g is defined as the longest shortest path between any two nodes in the graph.

We assume that a given number C of controllers have to be installed in the network. Consider that the SDN data plane can be represented by a undirected graph $G = (N, E)$, where N is the set of nodes and E is the set of edges or links. Each link is represented by its end nodes $\{i, j\}$ and has associated a given delay d_{ij}. The delay between any two nodes is given as the sum of the delays of the links belonging to the shortest path connecting them. We define the following decision variables:

- y_i binary variable that is 1 when a controller is placed in node $i \in N$, and 0 otherwise

Then, the sets of constraints of the CPP relative to the delay constraints are given by:

$$\sum_{i \in N} y_i = C \tag{1}$$

$$\sum_{\substack{j \in N: \\ d_{ij} \le D_{sc}}} y_j \ge 1 \qquad i \in N \tag{2}$$

$$y_i + y_j \le 1 \qquad i \in N, j \in N : d_{ij} > D_{cc} \tag{3}$$

$$y_i \in \{0, 1\} \qquad i \in N \tag{4}$$

Constraint (1) guarantees the placement of C controllers. Constraints (2) guarantee that for each switch, there is at least one controller distanced at most D_{sc} from it. Constraints (3) guarantee that no two controllers can be placed further than D_{cc} from each other. Constraints (4) are the variable domain constraints. In practice, the minimum possible C can be determined by solving the ILP model:

$$\text{Minimize } C \quad \text{s.t.} \quad (1) - (4) \tag{5}$$

To ensure controller redundancy, we must guarantee anycast routing for any switch to a pair of controller nodes [28]. We guarantee that the pair of paths must be node-disjoint without ensuring geodiversity.

Consider the set of arcs A, where each arc (i, j) represents the directed link from i to j. Each link is then represented by the pair of arcs (i, j) and (j, i). For each node $i \in N$, consider set $V(i)$ as the set of adjacent nodes to i. Consider the following additional decision variables and parameters:

- t_i^s binary parameter that is 1 if $i = s$, and 0 otherwise
- a_i^s binary variable that is 1 if the primary controller of switch s is placed in node i, and 0 otherwise
- b_i^s binary variable that is 1 if the backup controller of switch s is placed in node i, and 0 otherwise
- x_{ij}^s binary variable that is 1 if the arc $(i, j) \in A$ belongs to the primary path connecting switch s to its primary controller, and 0 otherwise
- u_{ij}^s binary variable that is 1 if the arc $(i, j) \in A$ belongs to the backup path connecting switch s to its backup controller, and 0 otherwise

The sets of constraints that guarantee node-disjointness routing with controller redundancy are given by:

$$\sum_{j \in V(i)} \left(x_{ij}^s - x_{ji}^s\right) = t_i^s - a_i^s \quad s \in N, i \in N \tag{6}$$

$$\sum_{j \in V(i)} \left(u_{ij}^s - u_{ji}^s\right) = t_i^s - b_i^s \quad s \in N, i \in N \tag{7}$$

$$\sum_{j \in V(i)} \left(x_{ji}^s + u_{ji}^s\right) \le 1 \qquad s \in N, i \in N \backslash \{s\} \tag{8}$$

$$\sum_{\{i,j\} \in E} d_{ij} \left(x_{ij}^s + x_{ji}^s\right) \le D_{sc} \quad s \in N \tag{9}$$

$$a_i^s + b_i^s \le y_i \qquad s \in N, i \in N \backslash \{s\} \tag{10}$$

$$a_s^s + b_s^s = 2y_s \qquad s \in N \tag{11}$$

$$\sum_{i \in N} \left(a_i^s + b_i^s\right) = 2 \qquad s \in N \tag{12}$$

$$a_i^s,\, b_i^s \in \{0,1\} \qquad s \in N,\, i \in N \tag{13}$$

$$x_{ij}^s,\, u_{ij}^s \in \{0,1\} \qquad s \in N,\, (i,j) \in A \tag{14}$$

Constraints (6) are the flow conservation constraints for the primary path of node s to its primary controller, located at node i such that $a_i^s = 1$. Likewise, constraints (7) are the flow conservation constraints for the backup path of node s to its backup controller, located at node i such that $b_i^s = 1$. Since the controller placement is not known *a priori*, the variables a_i^s and b_i^s account for the primary and backup controller placement.

Constraints (8) guarantee the node-disjointness of the pair of paths, and constraints (9) guarantee that the delays of the primary paths do not exceed D_{sc}. Constraints (10) guarantee that any primary or backup controller must be placed in a controller node. Constraints (11) guarantee that if node s is a controller node, then the switch in that node is controlled by the controller deployed there. Otherwise, constraints (12) guarantee that the primary and backup controllers of switch s must be distinct. Constraints (13) and (14) are variable domain constraints.

IV. Spine Design Problem

Controller redundancy is capable of increasing switch-controller (SC) availability significantly. However, considering only one backup controller cannot always achieve the SC availabilities required for critical services [21].

Therefore, we assume that a set of links can be selected to have enhanced availability at a given cost, which can be achieved in practice by reducing the average time to repair or reducing the time between failures (eg. by installing more robust equipment on these links or burying these links or prioritizing the repair of these links). These set of links form the spine, which we impose must be a tree.

A. Availability Link Upgrade

We assume each link $\{i,j\} \in E$ has a default availability which depends on its length, as given by [29] (pages 185-186):

$$\alpha_{ij}^0 = 1 - \frac{MTTR}{MTBF_{ij}} \tag{15}$$

where $MTTR = 24h$ denotes the mean time to repair, and $MTBF_{ij} = CaCu \cdot 365 \cdot 24/\ell_{ij}$ denotes the mean time between failures of link $\{i,j\}$, with $CaCu = 450$ km being the cable cut rate and ℓ_{ij} being the link length.

We consider the incremental availability link upgrade adopted in [11], where in each increment the link unavailability is decreased by a factor of $0 < \varepsilon < 1$. Denoting the default unavailability of link $\{i,j\}$ by μ_{ij}^0, which is given by $\mu_{ij}^0 = 1 - a_{ij}^0$ we have that $\mu_{ij}^k = (1-\varepsilon)\mu_{ij}^{k-1}$ with $k = 1,...,\kappa$, where κ is the number of upgrade levels considered. Then, we have that $\alpha_{ij}^k = \alpha_{ij}^{k-1} + \varepsilon - \varepsilon\alpha_{ij}^{k-1}$, $k = 1,...,\kappa$.

The cost of upgrading the availability of link $\{i,j\}$ to level k is given by:

$$c_{ij}^k = -\ell_{ij} \cdot \ln\left(\frac{1-\alpha_{ij}^k}{1-\alpha_{ij}^0}\right), \quad k = 1,...,\kappa \tag{16}$$

In this way, the decision variables x_{ij}^s and u_{ij}^s, introduced in Section III, are now extended to accommodate another index:

- x_{ij}^{sk} binary variable that is 1 if the arc $(i,j) \in A$ belongs to the primary path connecting switch s to its primary controller, when link $\{i,j\}$ is upgraded to level $k = 1,...,\kappa$ or not upgraded ($k=0$), and 0 otherwise
- u_{ij}^{sk} binary variable that is 1 if the arc $(i,j) \in A$ belongs to the backup path connecting switch s to its backup controller, when link $\{i,j\}$ is upgraded to level $k = 1,...,\kappa$ or not upgraded ($k=0$), and 0 otherwise

Therefore, constraints (6)-(9) and (14) now become:

$$\sum_{j\in V(i)} \sum_{k=0}^{\kappa} \left(x_{ij}^{sk} - x_{ji}^{sk}\right) = t_i^s - a_i^s \quad s \in N,\, i \in N \tag{6'}$$

$$\sum_{j\in V(i)} \sum_{k=0}^{\kappa} \left(u_{ij}^{sk} - u_{ji}^{sk}\right) = t_i^s - b_i^s \quad s \in N,\, i \in N \tag{7'}$$

$$\sum_{j\in V(i)} \sum_{k=0}^{\kappa} \left(x_{ji}^{sk} + u_{ji}^{sk}\right) \le 1 \qquad s \in N,\, i \in N\backslash\{s\} \tag{8'}$$

$$\sum_{j\in V(i)} \sum_{k=0}^{\kappa} d_{ij} x_{ij}^{sk} \le D_{sc} \qquad s \in N \tag{9'}$$

$$x_{ij}^{sk},\, u_{ij}^{sk} \in \{0,1\} \qquad s \in N,\, (i,j) \in A,\, k = 0,...,\kappa \tag{14'}$$

B. Control Path Availability Targets

The end-to-end availability of each control path, given by the pair of primary and backup paths, is required to be at least a target value $\lambda \lesssim 1$. The availability of a pair of node-disjoint paths is given by $\mathcal{A}_{ee} = 1 - (1-\mathcal{A}_p)(1-\mathcal{A}_b)$, where $\mathcal{A}_p$ and $\mathcal{A}_b$ denotes the availabilities of the primary and backup paths, respectively.

The availability of a path is found by the product of the availabilities of the individual components in the path (i.e., the links and nodes in the path). Here we assume the node availabilities to be one, simplifying the path availability to the product of the link availabilities. This approach has been adopted in the wide area network literature [30], with the argument that node equipment availability is typically higher than link availability to begin with and node availability can be further increased through redundancy and hardening methods to where the unavailability is orders of magnitude higher than link unavailability.

Since the end-to-end availability $\mathcal{A}_{ee}$ is difficult to linearize, we consider target availabilities for the primary and backup paths themselves. In this way, we assume that each primary path must have an availability of at least λ_p, while each backup path must have an availability of at least λ_b, where $1 - (1-\lambda_p)(1-\lambda_b) \ge \lambda$ [11]. With λ_p and λ_b, we ensure that the required control path availability is achieved. Since the primary paths tend to be shorter than the backup paths, especially because we impose a maximum value D_{sc} for the delay of the primary paths, we assume that $\lambda_p > \lambda_b$.

To achieve the required availability of λ, a set of links is selected that can be upgraded to level $k = 1,...,\kappa$ as needed, which forms the high-availability subgraph or spine. Although neither the primary nor backup paths are forced on the spine, they will use the necessary links of the spine to achieve their target availabilities, while satisfying the geodiversity separation between them.

The linearized expression of the availability of the primary path of switch s, assuming κ levels of upgrade, is given by:

$$\log(\mathcal{A}_p^s) = \sum_{\{i,j\}\in E} \sum_{k=0}^{\kappa} \left(x_{ij}^{sk} + x_{ji}^{sk}\right) \log\left(\alpha_{ij}^k\right) \quad (17)$$

We briefly schematize the proof here. The detailed proof is given in [13]. The availability of the primary paths of switch s is given by:

$$\begin{aligned} \mathcal{A}_p^s &= \prod_{\substack{(i,j)\in A:\\ x_{ij}^{s0}=1}} \alpha_{ij}^0 \prod_{\substack{(i,j)\in A:\\ x_{ij}^{s1}=1}} \alpha_{ij}^1 \cdots \prod_{\substack{(i,j)\in A:\\ x_{ij}^{s\kappa}=1}} \alpha_{ij}^\kappa \\ &= \prod_{(i,j)\in A} \prod_{k=0}^{\kappa} \left[1 - x_{ij}^{sk}\left(1-\alpha_{ij}^{sk}\right)\right] \end{aligned} \quad (18)$$

By applying logarithms, it is possible to linearize the expression as:

$$\begin{aligned} \log(\mathcal{A}_p^s) &= \sum_{(i,j)\in A} \sum_{k=0}^{\kappa} \log\left[1 - x_{ij}^{sk}(1-\alpha_{ij}^{sk})\right] \\ &= \sum_{(i,j)\in A} \sum_{k=0}^{\kappa} x_{ij}^{sk} \log\left(\alpha_{ij}^k\right) \end{aligned} \quad (19)$$

Consider the following additional decision variables:

- z_{ij}^k binary variable that is 1 if link $\{i,j\}\in E$ is upgraded to level $k=1,...,\kappa$, and 0 otherwise

The the set of constraints that account for the link upgrade are given by:

$$x_{ij}^{s0} + x_{ji}^{s0} \le 1 - z_{ij}^k \quad s\in N,\ \{i,j\}\in E,\ k=1,...,\kappa \quad (20)$$
$$x_{ij}^{sk} + x_{ji}^{sk} \le z_{ij}^k \quad s\in N,\ \{i,j\}\in E,\ k=1,...,\kappa \quad (21)$$
$$u_{ij}^{s0} + u_{ji}^{s0} \le 1 - z_{ij}^k \quad s\in N,\ \{i,j\}\in E,\ k=1,...,\kappa \quad (22)$$
$$u_{ij}^{sk} + u_{ji}^{sk} \le z_{ij}^k \quad s\in N,\ \{i,j\}\in E,\ k=1,...,\kappa \quad (23)$$
$$\sum_{k=1}^{\kappa} z_{ij}^k \le 1 \quad \{i,j\}\in E \quad (24)$$
$$z_{ij}^k \in \{0,1\} \quad \{i,j\}\in E,\ k=1,...,\kappa \quad (25)$$

Constraints (20) account for the links of the primary paths that are not upgraded, while constraints (21) account for the links that are upgraded to level $k=1,...,\kappa$. Likewise, constraints (22) account for the links of the backup paths that are not upgraded, while constraints (23) account for the links that are upgraded to level $k=1,...,\kappa$. Constraints (24) guarantee that each link is upgraded to one and only one level $k>1$, or not upgraded at all ($k=0$).

The set of constraints that guarantees the target availabilities for the primary and backup paths are given by:

$$\sum_{\{i,j\}\in E} \sum_{k=0}^{\kappa} \left(x_{ij}^{sk} + x_{ji}^{sk}\right) \log\left(\alpha_{ij}^k\right) \ge \log(\lambda_p) \quad s\in N \quad (26)$$
$$\sum_{\{i,j\}\in E} \sum_{k=0}^{\kappa} \left(u_{ij}^{sk} + u_{ji}^{sk}\right) \log\left(\alpha_{ij}^k\right) \ge \log(\lambda_b) \quad s\in N \quad (27)$$

Finally, we impose that the set of upgraded links must sit on a spanning tree subgraph. Consider the following additional parameter and decision variables:

- γ arbitrary node chosen and referred as the root node to model the spanning tree for the spine
- θ_{ij} binary variable that is 1 if arc $(i,j)\in A$ belongs to the spanning tree, and 0 otherwise
- β_{ij}^s binary variable that is 1 if arc $(i,j)\in A$ belongs to the routing path from node s to root node γ on the spanning tree, and 0 otherwise

The set of constraints that guarantee that the upgraded links belong to a spanning tree subgraph are given by:

$$\sum_{j\in V(i)} \left(\beta_{ij}^s - \beta_{ji}^s\right) = t_i^s \quad s\in N\backslash\{\gamma\},\ i\in N\backslash\{\gamma\} \quad (28)$$
$$\theta_{ij} \ge \beta_{ij}^s \quad s\in N\backslash\{\gamma\},\ (i,j)\in A \quad (29)$$
$$\sum_{j\in V(i)} \theta_{ij} = 1 - t_i^\gamma \quad i\in N \quad (30)$$
$$z_{ij}^k \le \theta_{ij} + \theta_{ji} \quad \{i,j\}\in E,\ k=1,...,\kappa \quad (31)$$
$$\theta_{ij} \in \{0,1\} \quad (i,j)\in A \quad (32)$$
$$\beta_{ij}^s \in \{0,1\} \quad s\in N\backslash\{\gamma\},\ (i,j)\in A \quad (33)$$

Constraints (28) guarantee that there is a routing path from node s to root node γ. Constraints (29) account for the links belonging to the spanning tree, by making use of the links in the routing paths to root node γ. Constraints (30) guarantee that the routing paths form a directed arborescence towards the root node γ. Constraints (31) guarantee that the upgraded links belong to the spanning tree.

When the spanning tree is finally selected, we prune the unnecessary links. The pruning process consists in identifying any non-upgraded link incident to the leaves, which are the terminal nodes of the tree. Once identified, these links are removed and we continue the pruning process, until we obtain a tree such that all the links connecting to the leaves are upgraded.

V. D-GEODIVERSITY

Besides the delay and availability constraints, we also consider geodiversity constraints. For each switch, the pair of primary and backup paths connecting it to its primary and backup controllers, respectively, must be D-geodiverse for a given value $D>0$. In this way, we guarantee that a regional failure with a coverage diameter of at most D, affecting one of the paths, does not affect the other path.

We adopt the D-geodiverse concept used in [17], which extends the concept in [16] to include geodiversity for the links incident to the source and destination nodes. Hence, for each switch s the pair of node-disjoint primary and backup paths, must be geographically distanced from each other at least D. To ensure this separation, each link of the primary path that is not incident to s must be distanced at least D from any link of the backup path, and vice-versa. We define the distance between two links $e_1, e_2 \in E$ as the minimum geographical distance between any point of e_1 and any point of e_2, and denote it by $\delta(e_1, e_2)$; in other words,

$$\delta(e_1, e_2) = \inf_{\substack{t_1 \in e_1 \\ t_2 \in e_2}} \delta(t_1, t_2) \tag{34}$$

For the pair of links incident to s, the distance between them is zero, since they share the node s. So, we can ignore these pairs of links as do many authors, since having a failure that affects s cannot be protected anyway. However, we prefer the approach in [17], and consider that for this particular case, the primary path link incident to s must be distanced at least D from the opposite node of the backup path link, and vice-versa. We define the distance between a node $n \in N$ and a link $e \in E$ as the minimum geographical distance between the node and any point of the link:

$$\delta(n, e) = \inf_{t \in e} \delta(n, t) \tag{35}$$

Denoting the primary and backup path links incident to s as $\{s, i\}$ and $\{s, j\}$, respectively, and denoting their modified distance as $\delta'(\{s, i\}, \{s, j\})$:

$$\delta'(\{s, i\}, \{s, j\}) = \min\{\delta(i, \{s, j\}), \delta(j, \{s, i\})\} \tag{36}$$

The D-geodiversity concept is described in detail in [17] for one pair of paths, and in [28] for anycast routing. We briefly illustrate the special cases for the links incident to s in Fig. 1.

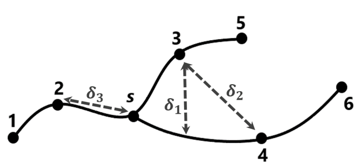

Fig. 1. Geographical distance between links incident to node s

Assume that the backup path for switch s is given by the links $\{s, 4\}$ and $\{4, 6\}$ having the backup controller placed in node 6. If the primary path is given by links $\{s, 3\}$ and $\{3, 5\}$ having the primary controller placed in node 5, then the modified distance between links $\{s, 3\}$ and $\{s, 4\}$ is given by $\delta'(\{s, 3\}, \{s, 4\}) = \min\{\delta_1, \delta_2\} = \delta_1$.

If the primary path is given by links $\{s, 2\}$ and $\{1, 2\}$ having the primary controller placed in node 1, then the modified distance between links $\{s, 2\}$ and $\{s, 4\}$ is given by the geographical distance between nodes s and 2: $\delta'(\{s, 2\}, \{s, 4\}) = \delta_3$. In this case, as pointed out in [28], if a failure affects both the primary and backup path, then it affects node s, and so it is impossible to protect both paths. Therefore, it is not worthwhile considering such cases.

We define set P_s as the set of incompatible link pairs in relation to switch node s. Consider set P_{s1} as the set of incompatible link pairs not sharing node s, i.e., the set of link pairs $\{i_1, j_1\}, \{i_2, j_2\}$ such that $i_1, j_1 \neq s \vee i_2, j_2 \neq s$ and $\delta(\{i_1, j_1\}, \{i_2, j_2\}) < D$. As in [28], we ignore the link pairs with common node different from s, since these are automatically removed by the node-disjoint constraints. Consider set P_{s2} as the set of link pairs sharing a common node different to s. Consider set P_{s3} as the set of incompatible link pairs incident to s, i.e., the set of link pairs $\{s, i\}, \{s, j\}$ such that $\delta'(\{s, i\}, \{s, j\}) < D$. Consider set P_{s4} as the set of link pairs incident to s that cannot be protected, i.e., the set link pairs $\{s, i\}, \{s, j\}$ such that $\delta'(\{s, i\}, \{s, j\}) = \min\{\ell_{si}, \ell_{sj}\}$. Finally, the set P_s of incompatible link pairs of interest is given by $P_s = (P_{s1} \cup P_{s3}) \backslash (P_{s2} \cup P_{s4})$.

The set of geodiverse constraints, guaranteeing that the primary and backup paths of switch s are D-geodiverse, is given by:

$$\sum_{k=0}^{\kappa} x_{i_1 j_1}^{sk} + x_{j_1 i_1}^{sk} + u_{i_2 j_2}^{sk} + u_{j_2 i_2}^{sk} \leq 1 \tag{37}$$

$$\sum_{k=0}^{\kappa} u_{i_1 j_1}^{sk} + u_{j_1 i_1}^{sk} + x_{i_2 j_2}^{sk} + x_{j_2 i_2}^{sk} \leq 1$$

$$s \in N, \; (\{i_1, j_1\}, \{i_2, j_2\}) \in P_s \tag{38}$$

By removing set P_{s2}, the number of constraints (37) and (38) are reduced significantly, improving computational efficiency when solving the ILP model.

Recall that these constraints, together with (10)-(12), ensure anycast routing of each switch (except if it is a controller node) to two distinct controller nodes with geodiversity.

VI. Formulation of the Optimization Problem

We can finally define our optimization problem, which consists of selecting the controller placements and selecting the spanning tree such that its links can be upgraded to κ levels of improved availability. The problem aims at minimizing the upgrade cost, while guaranteeing that the SC and CC delays be within delay maximum requirements, that required target path availabilities are achieved, and that the D-geodiversity between each pair of primary and backup paths is guaranteed. The ILP model is given by:

$$\text{Minimize} \sum_{k=1}^{\kappa} c_{ij}^k z_{ij}^k \tag{39}$$

s.t.

$(1) - (4)$, $(6') - (9')$, $(10) - (13)$, $(14')$,

$(20) - (25)$, $(26) - (27)$, $(28) - (33)$, $(37) - (38)$

Recall that (1)-(4) are the constraints relative to the controller placement delays. Constraints (6′)-(9′) and (14′) are relative to the node-disjoint pair of primary and backup paths, with extended variables to account for the level of upgrade of the respective arcs. Constraints (10)-(13) account for the primary and backup controller placement of each switch, by anycast routing to two distinct controller nodes (except if the switch is also a controller node). Constraints (20)-(25) account for the upgraded links, by relating the extended path variables x_{ij}^{sk} and u_{ij}^{sk}, with the upgraded link variables z_{ij}^{k}, where each link is either not upgraded ($k = 0$) or upgraded to one and only one level $k = 1, ..., \kappa$. Constraints (26)-(27) guarantee the target path availabilities. Constraints (28)-(33) are relative to the spanning tree selection which harbours the upgraded links. Finally, constraints (37)-(38) guarantee that each pair of primary and backup paths are D-geodiverse.

The problem of minimizing the cost of upgrading the spine can be reduced to a problem of minimizing the cost of a pair of link-disjoint paths, such that their lengths are bounded (by the path availability target values), which is known to be NP-complete [11], [31]. Moreover, the CPP is NP-hard and, therefore, our optimization problem is NP-complete.

Our problem is bi-objective since we want to minimize the upgrade cost, but also minimize the number of controllers C to avoid intercontroller communication overhead (which can result in synchronization issues between the controllers). This trade-off can be studied, thanks to the discrete nature of C. We can determine the minimum possible C, only under the delay constraints, given by the ILP model (1)-(5) presented in Section III. With this minimum value C, by solving the ILP model of our optimization problem we determine the minimum upgrade cost for the minimum number of controllers. By incrementing C iteratively, we can solve the ILP model again and obtain the Pareto front for the two objectives. Eventually, C will be too large for the CC delays to satisfy D_{cc}, rendering the problem infeasible.

VII. Computational Results

For the computational results, we used the spain topology from [32] and five networks from SNDlib [33]. Their characteristics are summarized in Table I, which shows the number of nodes, number of links, average node degree and the graph diameter D_g for each network.

Network	#nodes	#links	avg deg	D_g [km]
abilene	12	15	2.5	4706
polska	12	18	3.00	811
spain	14	22	3.14	1034
nobel_germany	17	26	3.06	790
janos_us	26	42	3.23	4690
nobel_eu	28	41	2.93	3365

TABLE I
Characteristics of the networks

Recall that the maximum delay values D_{sc} and D_{cc} are given as percentages of D_g. We chose D_{sc} to be 30% ($0.3D_g$) or the tightest possible value above 30% for the given topology. Similarly, we chose D_{cc} to be 60% ($0.6D_g$) or the tightest possible value above 60% for the given topology. To obtain a second set of values, we increased each maximum value by 5%. The different pairs of maximum values D_{sc}, D_{cc} are shown in Table II, for each network.

We considered the geodiversity values to be multiples of 20 km up to 300 km, or until the problem became infeasible for a given $D_{\max}$ in this range. The second last column of Table II shows the $D_{\max}$ used for each topology. The last column (D_{geo}) will be explained in Section VII-A. For each network, we computed the link lengths as the geographical distance between their end nodes over the Earth's surface [17].

Network	D_{sc}	D_{cc}	$D_{\max}$ [km]	D_{geo} [km]
abilene	40%	75%	220	220
	45%	80%	300	220
polska	35%	70%	140	100
	40%	75%	140	100
spain	35%	70%	160	100
	40%	75%	220	160
nobel_germany	35%	65%	100	20
	40%	70%	100	20
janos_us	30%	60%	160	160
	35%	65%	160	140
nobel_eu	30%	60%	260	140
	35%	65%	300	–

TABLE II
Maximum delay values D_{sc} and D_{cc}, and geodiversity maximum $D_{\max}$ and induced D_{geo} values

We implemented our code in C/C++, where we use the CPLEX 12.9 Callable libraries to solve the ILP model. The model was solved for the minimum number of controllers C, and then C was iteratively incremented and the ILP was solved again, until eventually the problem became infeasible. All instances were run on an Intel 8-core i7 PC with 64 GB of RAM, running at 3.6 GHz.

Note that $D_{sc} = 30\%$ and $D_{cc} = 60\%$ is only possible for the larger networks, janos_us and nobel_us (larger in terms of number of nodes and links). For the other networks, we needed to start with a larger value of $D_{sc} = 35\%$, while for the smallest network we needed $D_{sc} = 40\%$. For D_{cc}, it was possible to use 65% for nobel_germany, but for the other networks we needed larger values. So as the network grows in number of nodes and links, the percentages used in D_{sc} and D_{cc} w.r.t. the graph diameter D_g can be tighter.

Also note that for nobel_germany which has the smallest diameter, a geodiversity of over 100 km is not feasible, for the considered delays. Polska allows a geodiversity of up to 140 km, while janos_us allows up to 160 km for the considered delays. For abilene, which has a graph diameter comparable to janos_us, we see that for the tighter delay values, we can go up to 220 km, while relaxing the delays by 5%, we can go up to 300 km. Similar observations can be made for spain and nobel_eu. Note that although intuitively it makes sense, that the geodiversity maximum $D_{\max}$ is dependent on the graph diameter, we can see from abilene and janos_us that this is not sufficient. As we will see in Section VII-B, the induced geodiversity depends on the relevant nearest link pairs not sharing a controller node in the network. In the text that follows the units of D (km) will often be omitted for simplicity.

The target path availabilities were chosen to be $\lambda_p = 0.999$ for each primary path, and $\lambda_b = 0.99$ for each backup path, in all networks. In this way, it was possible achieve an end-to-end availability of at least $\lambda = 0.99999$ (referred to as 'five-nines' availability).

A. Upgrade Cost versus Number of Controllers for Path Redundancy, Controller Redundancy and Geodiversity

In [13], we addressed the joint CPP and spine design problem with path redundancy. We reported computational results for polska and nobel_germany and use them here for comparison purposes. Note that when we ignore the geodiversity constraints (37)-(38) in our ILP model, our problem reduces to that with controller redundancy, where each switch connects to a primary controller and a backup controller via a pair of node-disjoint paths that meet delay requirements.

We have used an availability improvement factor of $\varepsilon = 0.5$ and $\kappa = 4$ levels of availability link upgrade. We note that by solving our optimization problem for controller redundancy alone, the solution already provides a certain degree of geodiversity. This is shown in the last column D_{geo} of Table II. These values are found as the maximum values of D (in steps of 20 km) for which the D-geodiverse solutions are exactly the same as the ones for controller redundancy alone. Therefore, in Figs. 2-5 the D-geodiverse solutions shown are for $D_{\text{geo}} < D \leq D_{\max}$.

In these figures, we show the upgrade cost versus the number of controllers for path redundancy (for polska and nobel_germany only), controller redundancy and D-geodiversity. We denote path redundancy as PR, controller redundancy as CR (without geodiversity constraints), and D-geodiversity as the value of the corresponding $D > D_{\text{geo}}$ (with controller redundancy).

We begin by showing the figures for polska and nobel_germany, since for these networks we have path redundancy to compare with [13]. In Fig. 2, we show the results for polska network. Note that the orange line (solid line with circles) is for path redundancy, and is the most costly strategy. The black line (dotted line with circles) is for controller redundancy (without geodiversity constraints). From Table II, we know that this solution induces a geodiversity of 100 km, meaning that the curve for CR is the same for geodiversity of $D = 20, 40, 60, 80$ and 100 km. Moreover, it is possible to observe that the CR solution induces a geodiversity of 120 km for $C \geq 4$, since the curves are identical (the grey dashed curve coincides with the dotted black line for $C \geq 4$.)

Note that for $D_{sc} = 35\%$ and $D_{cc} = 70\%$, the minimum number of controllers $C_{\min}$ is 3 for CR and up to $D = 100$ km, while for $D = 120$ km we have $C_{\min} = 4$. Also note that the maximum number of controllers $C_{\max}$ that can be deployed is 8 (top chart). In turn for $D_{sc} = 40\%$ and $D_{cc} = 75\%$, we note that by relaxing the maximum delay values we now have $C_{\min} = 3$ also for $D = 120$ km, and $C_{\max} = 9$ (bottom chart). This is expected since relaxing D_{sc} and D_{cc} means that the controllers can be more spread from each other, making them closer to the switches. This leads to needing less controllers, but at the same time being able to deploy more controllers if desired. In fact, D_{sc} imposes $C_{\min}$ to guarantee that there is a controller close enough to each node, while D_{cc} imposes $C_{\max}$ from which having more controllers cannot satisfy D_{cc}.

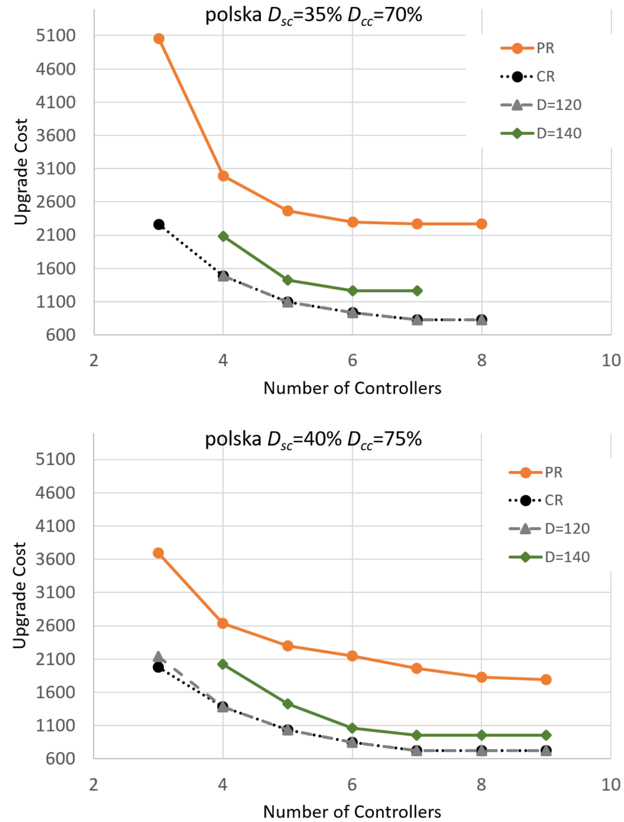

Fig. 2. Upgrade cost versus number of controllers for polska network

We can see that the cost of having 3 controllers for $D = 120$ km is slightly higher than for $D \leq 100$. This is due to the repositioning of the controllers in order to guarantee the geodiversity requirement of 120 km, leading to a solution that requires a more costly upgrading of the spanning tree. In both charts, we note that $D = 140$ km given by the green curve (solid curve with diamonds) shows that $C_{\min} = 4$ and that the cost is higher than the previous options, but still significantly lower than PR. Furthermore, note that the costs tend to be lower in the bottom chart, as expected due to the more relaxed maximum delay values.

Also note that having more than 6 controllers does not improve the upgrade cost significantly, and may even not improve the cost at all as seen in the bottom chart of Fig. 2. Finally, from Table II, we can see that $D_{\max} = 140$, meaning that our problem is infeasible for $D = 160$ km.

The results for nobel_germany are shown in Fig. 3. Recall from Table II, that CR induces a geodiversity of 20 km. We will see in Section VII-B, that this induced geodiversity is imposed by a pair of links in the network. Note that in both charts CR with only 2 controllers has a higher upgrade cost than PR, but this is then inverted for $C \geq 3$. Intuitively, we expect path redundancy to be more costly, since the backup path has to be node-disjoint but still connect to the same

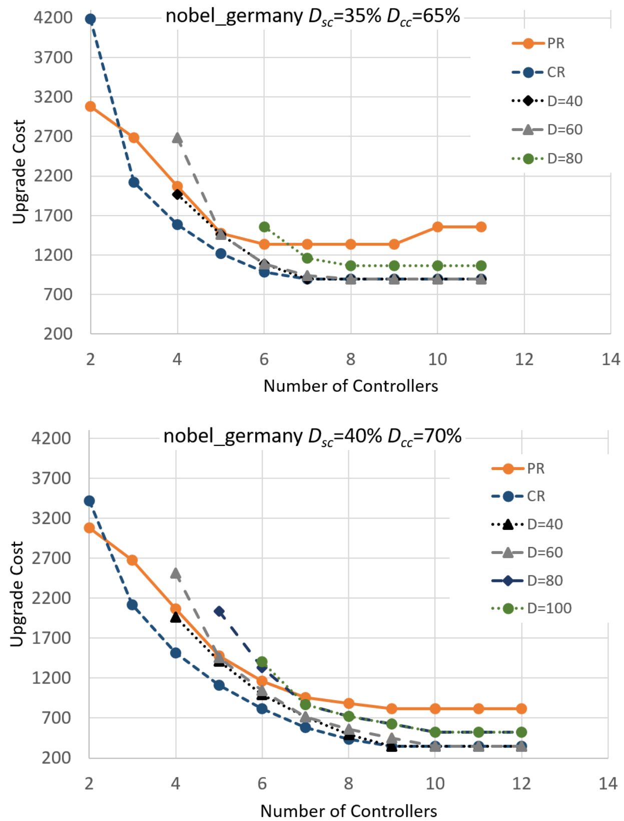

Fig. 3. Upgrade cost versus number of controllers for nobel_germany network

controller. This can lead to very long backup paths. Having the freedom to connect to a second controller, can lead to shorter backup paths while still being node-disjoint to the primary path. However, when we only have 2 controllers in the network, due to the D_{sc} requirement between switches and their primary controllers, the 2 controllers may be placed the furthest possible from each other, without exceeding D_{cc}. Since we require controller redundancy, each switch is forced to connect to the only 2 controllers in the network, making the backup paths quite long.

Another interesting observation, already reported in [13], is how the PR curve increases for $C = 10$ in the top chart, a tendency not observed for the other curves. This indicates that having 10 controllers is overcrowding, forcing controllers to be repositioned due to the D_{cc} requirement, making the solution more costly. Although an increase in cost is not observed for the other curves, we can see in the top chart that having more than 7 controllers does not improve the cost significantly or at all, while the same is observed for $C \geq 8$ in the bottom chart. For this network, this is quite clear that the costs for the more relaxed maximum delay values (bottom chart) are much lower. We can see as expected that as D increases there is a tendency for the upgrade cost to increase (for the same number of controllers). It is interesting to observe that geodiversity can be less costly than PR: note that geodiversity even for 100 km is less costly than PR for $C \geq 7$ (in both charts).

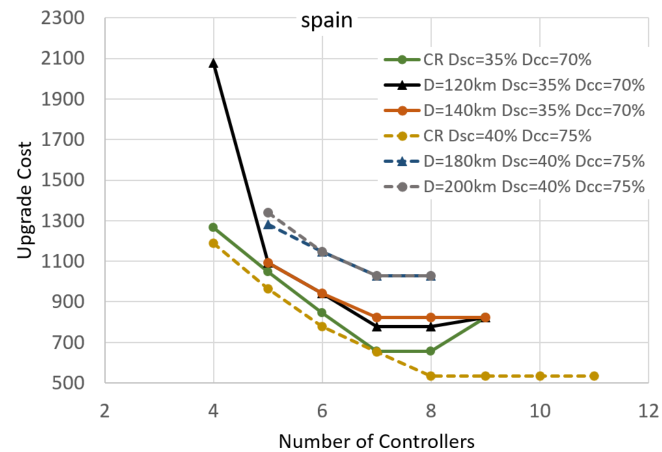

Fig. 4. Upgrade cost versus number of controllers for spain network

The results for spain are shown in Fig. 4. The solid lines are for $D_{sc} = 35\%$ and $D_{cc} = 70\%$, while the dashed lines are for the more relaxed values. For the former set of maximum delays, CR induces a geodiversity of 100 km, while for the more relaxed delay maximums, the controllers are repisitioned to guarantee a slightly cheaper solution, and now guaranteeing a geodiversity of 160 km. Unfortunately, due to space limitations we will not go into detail on this.

We also note that for the tighter delay maximums, it is impossible to guarantee a geodiversity of 180 km or more, while the second set of maximum delays allows us to go up to 220 km, although at a much higher cost. We mention that for $D_{sc} = 35\%$ and $D_{cc} = 70\%$ the curve for 160 km coincides with 140 km, and for the more relaxed maximum delays, the curve for 220 km coincides with 200 km (therefore, omitted in the figure).

Now, we analyze the US networks, janos_us and abilene. Due to space limitations, we only show the results for janos_us in Fig. 5, since abilene behaves similarly.

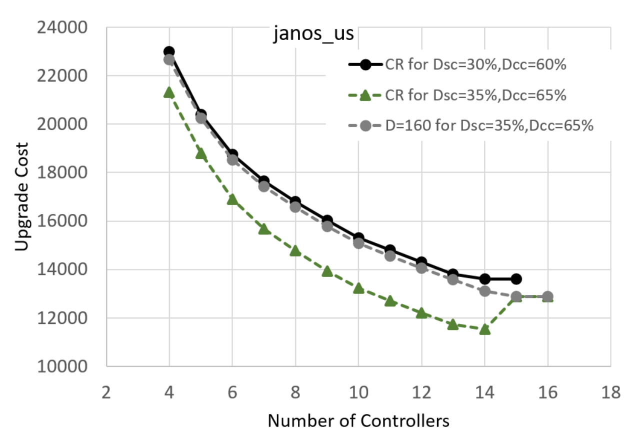

Fig. 5. Upgrade cost versus number of controllers for janos_us network

The solid line refers to $D_{sc} = 30\%$ and $D_{cc} = 60\%$, while the dashed lines refer to the more relaxed delay set. From

Table II, we know that for $D_{sc} = 30\%$ and $D_{cc} = 60\%$, CR induces a geodiversity of 160 km and that we cannot guarantee a geodiversity of 180 km or more – the problem becomes infeasible. For the second set of delay maximums, we note that CR can retrieve a much cheaper solution, but now this solutions induces a geodiversity of only 140 km. Needing to guarantee $D = 160$ km, increases the cost close to that of CR for $D_{sc} = 30\%$ and $D_{cc} = 60\%$ (but still slightly lower).

Also note that since the graph diameter of janos_us is much larger than for the previous networks, we can see an increase in an order of magnitude in terms of cost. This is also true for abilene, for which the costs are around the same values as janos_us (which is not surprising, given they represent networks across the US). In abilene, the CR solution induces a geodiversity of 220 km, which also happens to be $D_{\max}$ for the first set of delay maximums. When relaxing the delay maximums, we can go up to 300 km (maximum D we considered). The costs for $D = 240$ km are much higher than for $D = 220$ km, while the costs for $D = 260$ km coincide with 240 km.

Finally, we mention that for nobel_eu, for $D_{sc} = 30\%$ and $D_{cc} = 60\%$, the CR curve goes from a cost of little less than 16000 with $C_{\min} = 4$ to a little over 8000 with $C_{max} = 20$. This solution induces a geodiversity of 140 km. Increasing D does not incur in a significant cost increase. For the more relaxed delay maximums, the models reached the time limit of 6 hours without ending, except for $D = 300$ km, which took over a little more than 2 hours as can be seen in Table III.

Network	D_{sc}	D_{cc}	avg [s]	min [s]	max [s]
abilene	40%	75%	0.89	0.86	0.94
	45%	80%	1.62	0.98	2.19
polska	35%	70%	2.65	0.92	3.18
	40%	75%	8.85	5.55	9.68
spain	35%	70%	1.94	1.73	2.46
	40%	75%	2.31	1.30	2.78
nobel_germany	35%	65%	230.43	20.50	422.60
	40%	70%	879.29	129.79	1889.26
janos_us	30%	60%	564.53	559.52	568.28
	35%	65%	2231.18	1911.92	3503.05
nobel_eu	30%	60%	1764.81	1278.73	2267.96
	35%	65%	–	7574.139	–

TABLE III
AVERAGE, MINIMUM AND MAXIMUM RUNTIMES IN SECONDS, FOR SOLVING THE ILP MODELS

We also consider the solution times for each instance, which is the sum of the solution times for solving the ILP model for the minimum C (see (5)), and then incrementally increasing C until the problem becomes infeasible. In Table III, we show the average, minimum and maximum solution times. In general, the problems tend to be quicker to solve as D increases, and more difficult to solve as the network increases in number of nodes and links. Clearly, the ILP is not efficient for larger networks, as observed with nobel_eu.

B. Controller Placement and Tree Subgraph

In this subsection, we analyze the solution of the optimization problem, in terms of controller placement and tree subgraph, for some of the above instances.

In Fig. 6, the controller placement and tree subgraph are shown for nobel_germany with $D_{sc} = 35\%$ and $D_{cc} = 65\%$. The controller nodes are shown as big grey circles, while the other nodes are represented as small black nodes. The dashed grey lines represent the links not belonging to the tree. The links of the tree are in black, where the thinnest links are not upgraded and the thickest links are upgraded to level $\kappa = 4$. The graphs are shown for $C = 6$ controllers and for $D = 20$ (top), $D = 60$ (middle) and $D = 100$ (bottom).

Note that as D increases, the controllers tend to be placed on the nodes of the smallest links, since it is not possible to guarantee a pair of D-geodiverse paths for such nodes, when D is large enough – a similar observation is made in [28].

The tree subgraph forms the spine whose links can have upgraded availability. To achieve the target path availabilities of $\lambda_p = 0.999$ and $\lambda_b = 0.99$, we can see that for $D = 20$ which is already guaranteed for controller redundancy alone, 4 links are upgraded to level $k = 1$, while 3 shorter links are upgraded to level $k = 2$ and one short link is upgraded to level $k = 3$. The controllers are spread out in the network. The distance between the pair of links marked by the red arrow, imposes a geodiversity of exactly 29 km – recall that the pairs of links belonging to set P_{s4} are not included in the geodiversity constraints. We translated this as 20 km in Table II, due to the 20 km steps we were considering.

For $D = 60$, 2 controllers are repositioned. Note that the same link is upgraded to level $k = 3$ and its neighbor to level $k = 2$, and 4 of the previous links remain upgraded to level $k = 1$. In the previous case, two additional short links were upgraded to level $k = 2$, while now three additional longer links are upgraded to level $k = 1$. Although the cost function given by (16) leads to an exponential increase in cost when upgrading to a higher level, the function also increases with the link lengths. Thus, the longer link lengths although upgraded to a lower level, result in a more costly solution than that for $D = 20$. Moreover, now the geodiversity is imposed to be 74, given by the link pairs shown by the red arrow.

Therefore, for $D = 100$, we can see that 2 controllers are repositioned again. In this case, it is interesting to see that no link is upgraded for levels higher than $k = 2$; there are 5 links upgraded to level $k = 1$ and 4 links upgraded to level $k = 2$. Note that two fairly long links are upgraded to level $k = 2$, making this solution more costly, than the previous ones. The geodiversity now imposed is 101.

In Fig. 7, the solutions for janos_us are shown with $D_{sc} = 35\%$ and $D_{cc} = 65\%$. The graphs are shown for $C = 10$ controllers with $D = 80$ (top) and $D = 160$ (bottom). Doubling the value D from 80 to 160 km, we can see that some of the controllers were repositioned to nodes of the shorter links (as observed in Fig. 6). Note that for $D = 80$: 1 short link is upgraded to $k = 4$, on the upper right side of the network; 7 links are upgraded to level $k = 3$, spread throughout the network; 6 relatively short links are upgraded to level $k = 2$, all except one located from center to the right side of the network; and two very short links are upgraded to level $k = 1$, on the upper right side of the network.

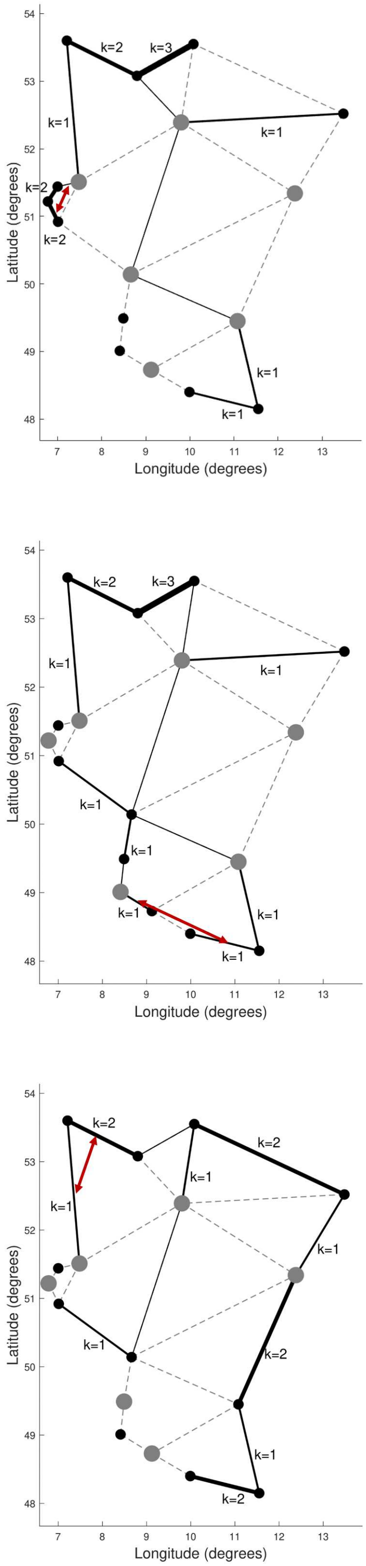

Fig. 6. Controller placement and tree subgraph for nobel_germany network with $D = 20$ (top), $D = 60$ (middle) and $D = 100$ (bottom)

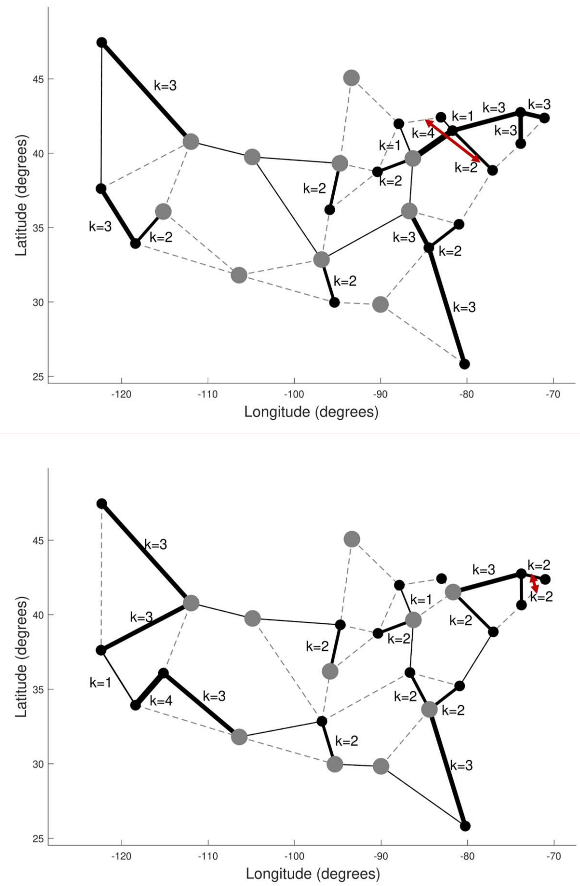

Fig. 7. Controller placement and tree subgraph for janos_us network with $D = 80$ (top) and $D = 160$ (bottom)

In comparison, for $D = 160$ we see that: 1 short link is upgraded to level $k = 4$, now on the lower left side of the network; 5 links are upgrade to level $k = 3$, instead of 7 links but with much longer total lengths and spread throughout; 8 relatively short links are upgraded to $k = 2$ all located from center to the right side of the network; and 2 links are upgraded to level $k = 1$, one to the far lower left side of the network and one on the upper right side. Note that the geodiversity is imposed by the link pairs at the top of the network, shown by the red arrows. For $D = 80$ km, the imposed geodiversity is 149 km, while for $D = 160$ km the imposed geodiversity is 178 km.

Note that the networks in Fig. 6 correspond to the curves in the top chart of Fig. 3, for 6 controllers and the corresponding D values, which show that the cost increases with D. In turn, the networks in Fig. 7 correspond to the dashed curves in Fig. 5, for 10 controllers and the corresponding D values, which show that the cost for $D = 80$ is much lower than for $D = 160$.

C. Comparing Variations in ε and κ

Besides the results presented above, we also obtained results for a upgrade factor of $\varepsilon = 0.4$, while maintaining the $\kappa = 4$

levels of upgrade, for polska, nobel_germany and janos_us. In turn, by maintaining $\varepsilon = 0.5$ we obtained results for $\kappa = 2$. For janos_us, we had to use $\kappa = 3$, since many instances were infeasible for only 2 levels of upgrade.

We start by comparing the results for $\varepsilon = 0.5$ and $\varepsilon = 0.4$. For smaller values of ε, more links may need to be upgraded to achieve the target path availability levels of $\lambda_p = 0.999$ and $\lambda_b = 0.99$. This may lead to an increase in the upgrade cost. On the other hand, higher values of ε may lead to unnecessarily high levels of availability, which may be leveraged better by lowering ε, consequently reducing the upgrade cost. We did some preliminary tests by reducing ε even more, but many instances became infeasible (especially for polska and janos_us).

In Fig. 8, we show the results for polska network with $D_{sc} = 35\%$, $D_{cc} = 70\%$ and $\kappa = 4$. The solid lines are for $\varepsilon = 0.5$, while the dashed lines are for $\varepsilon = 0.4$. Note that for $D \leq 120$, $\varepsilon = 0.4$ is able to achieve the target availabilities, decreasing the overall upgrade cost. The difference in cost is greater as C increases. However, for 140-geodiversity, the opposite occurs: the lower value of ε leads to an increase in the cost, since many more links need to be upgraded due to the much longer paths imposed by the required geodiversity, in order to achieve the target availabilities, when compared to $\varepsilon = 0.5$. The chart for $D_{sc} = 40\%$ and $D_{cc} = 75\%$ is omitted, since the observations are similar.

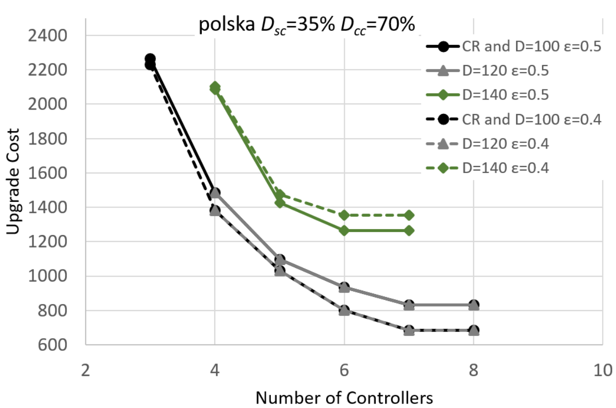

Fig. 8. Upgrade cost versus number of controllers for polska ($\kappa = 4$)

In Fig. 9, we show the results for nobel_germany with $D_{sc} = 40\%$, $D_{cc} = 70\%$ and $\kappa = 4$. For better readability, we only show the results for $D = 20$, 60 and 100 km. It is possible to see that $\varepsilon = 0.4$ (dashed lines) always achieves a cheaper solution, except for $C = 2$ and $D = 20$. The chart for $D_{sc} = 35\%$ and $D_{cc} = 65\%$ is omitted, since the observations are similar but the differences between $\varepsilon = 0.5$ and $\varepsilon = 0.4$ are less significant.

In Fig. 10, we show the results for janos_us, for both pairs of D_{sc}, D_{cc} values. In the top chart, for $D_{sc} = 30\%$ and $D_{cc} = 60\%$, recall that controller redundancy alone already guarantees a geodiversity of 160 km. Furthermore, we can see that the differences between varying ε from 0.5 to 0.4 are

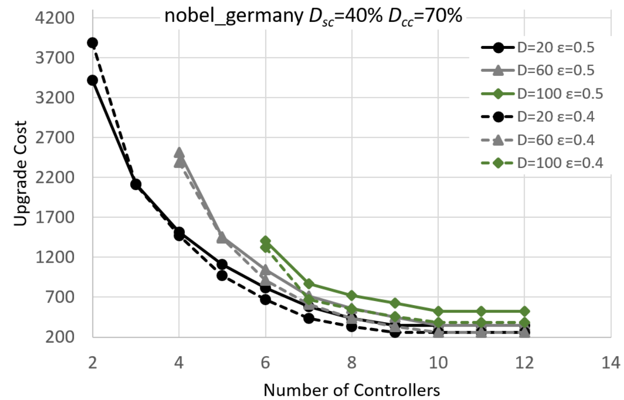

Fig. 9. Upgrade cost versus number of controllers for nobel_germany ($\kappa = 4$)

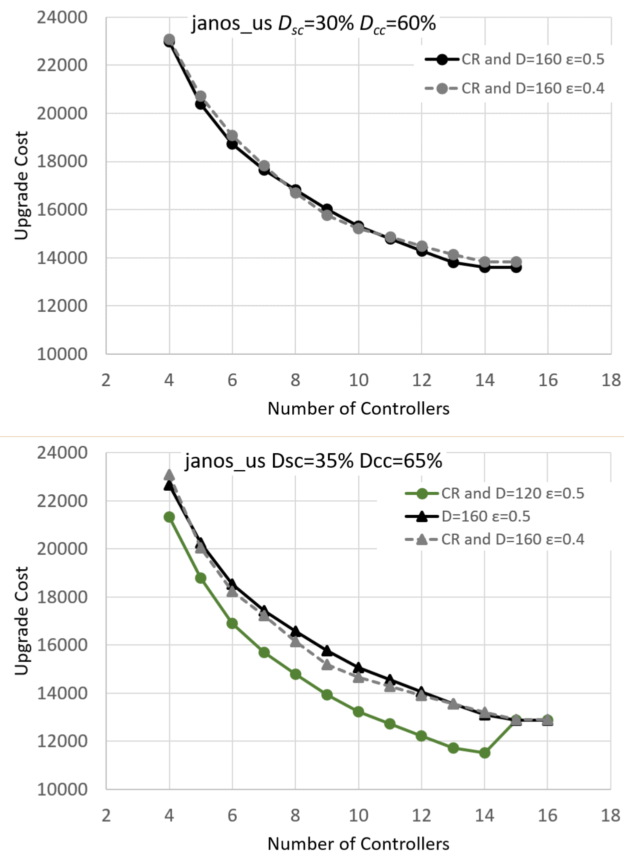

Fig. 10. Upgrade cost versus number of controllers for janos_us ($\kappa = 4$)

negligible. In the bottom chart for more relaxed delays, we recall that the controller redundancy guarantees a geodiversity of 120 km (which is the solid line with circle markers). The solid line with triangle markers are for $D = 160$ km. Note that for $\varepsilon = 0.4$, the dashed line is for controller redundancy which coincides with the geodiversity of 160 km. This means that for $\varepsilon = 0.4$, the solution guarantees geodiversity for 160 km. Also note that this curve (dashed line) is very close that of 160-geodiversity for $\varepsilon = 0.5$.

We now compare the results for $\kappa = 4$ versus $\kappa = 2$ for polska and nobel_germany and $\kappa = 3$ for janos_us. In Fig. 11, we show the results for polska network with $D_{sc} = 35\%$, $D_{cc} = 70\%$ and $\varepsilon = 0.5$. Note that the solid grey line with triangle markers is for $D = 120$ and is the same for $\kappa = 2$ and $\kappa = 4$. In other words, for $D = 120$ km the optimal solution only needs 2 levels of upgrade even though more may be available. Note that the dashed line which is for $D = 100$ with $\kappa = 2$, has a much higher cost than that with $\kappa = 4$, since many more links need to be upgraded to achieve the target availabilities. This impact attenuates as C increases. We omitted the curve for $D = 140$ km, since the problem is infeasible for $\kappa = 2$.

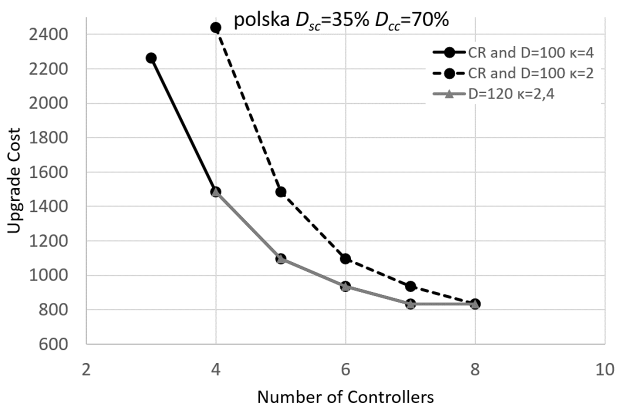

Fig. 11. Upgrade cost versus number of controllers for polska ($\varepsilon = 0.5$)

In Fig.12, we can see that the impact of $\kappa = 2$ versus $\kappa = 4$ is mostly negligible, since the optimal solution for most instances of this network only need up to 2 levels of link upgrade to achieve the target availabilities.

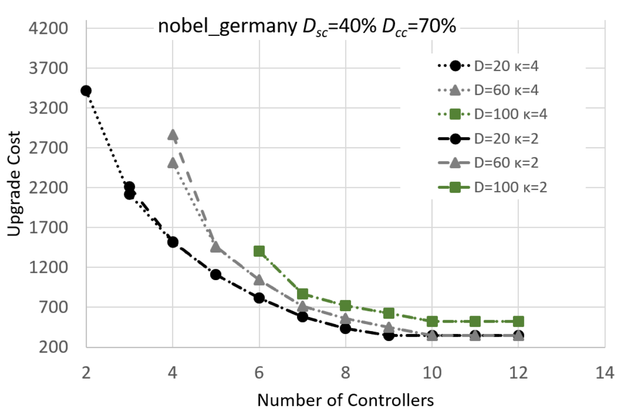

Fig. 12. Upgrade cost versus number of controllers for nobel_germany ($\varepsilon = 0.5$)

In Fig. 13, we show the results for janos_us, for both pairs of D_{sc}, D_{cc} values. In the top chart, for $D_{sc} = 30\%$ and $D_{cc} = 60\%$, we can see that varying κ from 4 to 3 has a negligible impact on the upgrade cost, although it is always slightly higher for $\kappa = 3$. Recall that the results for controller redundancy alone and for geodiversity up to 160 km are the same. In the bottom chart, for $D_{sc} = 35\%$ and $D_{cc} = 65\%$, as observed in Fig. 10 for $\varepsilon = 0.4$, the controller redundancy alone imposes a geodiversity of 160 km, while for $\varepsilon = 0.5$ controller redundancy alone imposes a geodiversity of 120 km. The curve for $\kappa = 3$ is very similar to that of $D = 160$ for $\kappa = 4$, although again always slightly more costly.

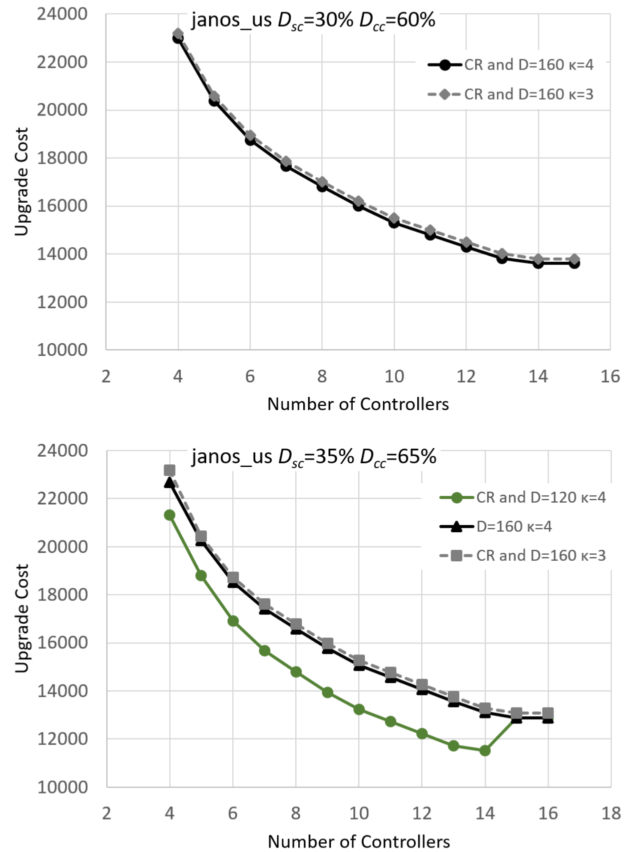

Fig. 13. Upgrade cost versus number of controllers for janos_us ($\varepsilon = 0.5$)

VIII. Conclusions

In this paper, we have addressed the joint optimization of controller placement and spine design, subject to SC and CC maximum delay constraints, target path availabilities for the primary and backup paths constraints, constraints that ensure the spine is a tree subgraph, and finally geodiversity constraints. We have discussed each group of constraints and formulated an ILP model. We presented computational results, comparing path redundancy, controller redundancy and geodiversity. We have observed that path redundancy strategy is generally more costly than the controller redundancy strategy, and also computationally more expensive to solve. We have also observed that controller redundancy alone guarantees in many cases a reasonable degree of geodiversity between the pair of primary and backup paths.

As the value of D for geodiversity increases, we have observed that the controllers tend to migrate to nodes with shorter links. This is because the maximum geodiversity possible in the topology is given by the relevant nearest link pairs, not sharing a controller node, in the network. Migrating controllers to the nodes incident to these links, allows to guarantee larger geodiversity. Necessarily, this is a consequence of the adopted geodiversity definition and of the set P_s used in the geodiversity constraints. We have also observed that, as expected, the upgrade cost increases as D increases, since the paths tend to be longer. In turn, the upgrade cost tends to decrease as the number of controllers are increased and also when the maximum delay values D_{sc}, D_{cc} are more relaxed. Nevertheless, having a large number of controllers has little or no effect on the upgrade cost, and can even increase the cost. This is reassuring, since from the perspective of the control plane, it is desirable to have a small number of controllers to avoid intercontroller communication overhead.

Moreover, the case studies that we analyzed for the controller placement and tree subgraph, show that it is not always necessary to upgrade the links to the top levels to achieve the desired geodiversity and availability. The results for a smaller ε and a smaller κ range from a significant impact to negligible impact, depending on the topology.

So, in summary we conclude that: it may be sufficient to have only controller redundancy since this alone may guarantee geodiversity to some extent; having a few levels of upgrade may be sufficient without jeopardizing the cost too much, if the networks are not too small; it is generally cheaper to have controller redundancy instead of path redundancy alone; having more than the minimum number of controllers impacts the cost significantly, but having much more is negligible or even detrimental in some cases; the fact that we obtain a set of upgraded links shows that without the spine subgraph, the target availabilities could never be achieved for controller redundancy (considering only one backup controller).

Since the ILP model does not scale well for larger networks, we plan to improve the ILP model by finding a more efficient formulation (improving the linear relaxation). However, for large networks, we need to develop an efficient heuristic. Since the problem is quite complex and because of the additional geodiverse feature, the most promising approach may be matheuristics, i.e., combining heuristic methods and mathematical programming techniques, in order to achieve good quality suboptimal solutions. Switch and controller availabilities will also be taken into account in the model.

ACKNOWLEDGMENTS

This work is funded by ERDF Funds through the Centre's Regional Operational Program and by National Funds through the FCT - Fundação para a Ciência e a Tecnologia, I.P. under the project CENTRO-01-0145-FEDER-029312. This work was also partially supported by FCT under project grant UIDB/00308/2020.

REFERENCES

[1] S. Jain *et al.*, "B4: Experience with a globally-deployed software defined WAN," *ACM SIGCOMM Computer Communication Review*, vol. 43, no. 4, pp. 3–14, 2013.

[2] R. Alvizu, G. Maier, N. Kukreja, A. Pattavina, R. Morro, A. Capello, and C. Cavazzoni, "Comprehensive survey on T-SDN: Software-defined networking for transport networks," *IEEE Communications Surveys & Tutorials*, vol. 19, no. 4, pp. 2232–2283, 2017.

[3] J. Rak, D. Hutchison, J. Tapolcai, R. Bruzgiene, M. Tornatore, C. Mas-Machuca, M. Furdek, and P. Smith, *Fundamentals of Communication Networks Resilience to Disasters and Massive Disruptions*. Computer Communications and Networks, Springer, 2020, pp. 1–43.

[4] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *ACM HotSDN*, New York, USA, 2012, pp. 7–12.

[5] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," in *ITC*, Shanghai, China, 2013, pp. 1–9.

[6] N. Perrot and T. Reynaud, "Optimal placement of controllers in a resilient SDN architecture," in *DRCN*, Paris, France, 2016, pp. 145–151.

[7] T. Das, V. Sridharan, and M. Gurusamy, "A survey on controller placement in SDN," *IEEE Communications Surveys & Tutorials*, vol. 22, pp. 472–503, First Quarter 2020.

[8] G. Wang, Y. Zhao, J. Huang, and W. Wang, "The controller placement in software defined networking: A survey," *IEEE Network*, pp. 21–27, Sept./Oct. 2017.

[9] ——, "The controller placement in software defined networking: A survey," *IEEE Network*, pp. 21–27, Sept./Oct. 2017.

[10] P. Vizarreta, C. M. Machuca, and W. Kellerer, "Controller placement strategies for a resilient SDN control plane," in *RNDM*, Halmstad, Sweden, 2016, pp. 253–259.

[11] A. Alashaikh, D. Tipper, and T. Gomes, "Embedded network design to support availability differentiation," *Annals of Telecommunications*, vol. 74, no. 9-10, pp. 605–623, 2019.

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

[13] D. Santos, T. Gomes, and D. Tipper, "Software-defined network design driven by availability requirements," in *DRCN*, Milan, Italy, 2020, pp. 1–7.

[14] Y. Cheng, J. Li, and J. P. G. Sterbenz, "Path geo-diversification: Design and analysis," in *ICUMT*, Almaty, Kazakhstan, 2013, pp. 46–53.

[15] J. P. Rohrer, A. Jabbar, and J. P. G. Sterbenz, "Path diversification for future internet end-to-end resilience and survivability," *Telecommunication Systems*, vol. 56, no. 1, pp. 49–67, 2014.

[16] S. Trajanovski, F. A. Kuipers, A. Ilić, J. Crowcroft, and P. V. Mieghem, "Finding critical regions and region-disjoint paths in a network," *IEEE/ACM Transactions on Networking*, vol. 23, no. 3, pp. 908–921, June 2015.

[17] A. de Sousa, D. Santos, and P. Monteiro, "Determination of the minimum cost pair of D-geodiverse paths," in *DRCN*, Munich, Germany, 2017.

[18] R. Girão-Silva, L. Martins, T. Gomes, A. Alashaikh, and D. Tipper, "Improving network availability - a design perspective," in *ICICT*, London, United Kingdom, 2018, pp. 799–815.

[19] L. Pašić, A. Pašić, F. Mogyorósi, and A. Pašić, "FRADIR meets availability," in *DRCN*, MIlan, Italy, 2020, pp. 1–6.

[20] Y. Hu, W. Wendon, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *IFIP/IM*, Ghent, Belgium, 2013, pp. 672–675.

[21] F. J. Ros and P. M. Ruiz, "Five nines of southbound reliability in software-defined networks," in *ACM HotSDN*, New York, USA, 2014, pp. 31–36.

[22] M. Tanha, D. Sajjadi, R. Ruby, , and J. Pan, "Capacity-aware and delay-guaranteed resilient controller placement for software-defined WANs," *IEEE Transactions on Network and Service Management*, vol. 15, no. 3, pp. 991–1005, 2018.

[23] M. Khorramizadeh and V. Ahmadi, "Capacity and load-aware software-defined network controller placement in heterogeneous environments," *Computer Communications*, vol. 129, pp. 226–247, 2018.

[24] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic approaches to the controller placement problem in large scale SDN networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 1, pp. 4–17, March 2015.

[25] Y. Cheng, M. T. Gardner, J. Li, R. May, D. Medhi, and J. P. Sterbenz, "Analysing geopath diversity and improving routing performance in optical networks," *Computer Networks*, vol. 82, pp. 50–67, 2015.

[26] Y. Cheng, D. Medhi, and J. P. G. Sterbenz, "Geodiverse routing with path delay and skew requirement under area-based challenges," *Networks*, vol. 66, no. 4, pp. 335–346, 2015.

[27] A. de Sousa, T. Gomes, R. Girão-Silva, and L. Martins, "Minimization of the network availability upgrade cost with geodiverse routing for disaster resilience," *Optical Switching and Networking*, vol. 31, pp. 127–143, 2019.

[28] A. de Sousa and D. Santos, "The minimum cost D-geodiverse anycast routing with optimal selection of anycast nodes," in *DRCN 2019*, Coimbra, Portugal, 2019.

[29] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network Recovery – Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Elsevier, 2004.

[30] M. Tarifeño-Gajardo, A. Beghelli, and E. Moreno, "Availability-driven optimal design of shared path protection in WDM networks," *Networks*, vol. 68, no. 3, pp. 224–237, 2016.

[31] A. Beshir and F. Kuipers, "Variants of the min-sum link-disjoint paths problem," in *IEEE SCVT'09*, Louvain-la-Neuve, Belgium, November 2009.

[32] R. Martinez, R. Casellas, R. Vilalta, and R. Munoz, "GMPLS/PCE-controlled multi-flow optical transponders in elastic optical networks [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 11, pp. B71–B80, 2015.

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

Dorabella Santos is a Research Assistant at the Institute for Systems Engineering and Computers at Coimbra (INESC Coimbra) since December 2018. She was a post-doc at Instituto de Telecomunicações, Aveiro, from 2008 until 2018. She received her Msc in Mathematics in 2003 by the University of Aveiro (Portugal), and her PhD degree in Electrotechnical Engineering in 2007 by the University of Aveiro (Portugal). Her research interests include optimization problems for telecommunication networks involving network design, traffic engineering, and more recently, software-defined networking.

Teresa Gomes is an Assistant Professor at the University of Coimbra, at the Department of Electrical and Computer Engineering, since 1998, and with the permanent position since 2003. In 2013, from April until July, she was a Visiting Researcher at the School of Information Sciences of the University of Pittsburgh (USA). She is also a researcher at the Institute for Systems Engineering and Computers at Coimbra (INESC Coimbra). She received her PhD diploma in Electrical Engineering (Telecommunications and Electronics) at the University of Coimbra in 1998. She is the author/co-author of over 90 technical publications in international journals, conference proceedings, and book chapters, including one European patent. She has been responsible for several national projects, mainly with industry. She has also served as a TPC member of numerous international conferences and was Co-Chair of DRCN 2019. Teresa Gomes is an Associate Editor of Springer's Journal of Network and Systems Management. Her main present interests are routing, protection and reliability analysis models and algorithms for communication networks.

David Tipper is Professor in the Department of Informatics and Networked Systems at the University of Pittsburgh. He is a graduate of the University of Arizona (Ph.D. E.E., M.S. S.I.E.)and Virginia Tech (B.S. E.E.). At Pitt he served as Director of the Graduate Telecommunications and Networking Program from 2007 – 2016. His current research interests are wired and wireless network design, performance analysis, network reliability and critical infrastructure resilience. His teaching and research has been supported by grants from various government and corporate sources such as NSF, ARO, NIST, NSA, IBM, Bechtel Bettis and AT&T. He is a Senior member of the Institute of Electrical and Electronics Engineers (IEEE).