This is an Accepted Manuscript of an article published by Elsevier in Computer Networks on August 2015, available online:https://doi.org/10.1016/j.comnet.2015.02.020.

The Spine Concept for Improving Network Availability

Abdulaziz Alashaikh

Graduate Telecommunications and Networking Program
University of Pittsburgh
Pittsburgh, PA 15260
Email: azizoozi@gmail.com

Teresa Gomes

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

David Tipper

Graduate Telecommunications and Networking Program
University of Pittsburgh
Pittsburgh, PA 15260
Email: tipper@tele.pitt.edu

Abstract

Telecommunications networks need to guarantee that all node pairs involved in critical service communications are highly available. Here we adopt a novel approach to the problem of how to provide high levels of availability in an efficient manner. The basic idea is 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 end-to-end availability. We first explore the spine concept through simple topologies illustrating the potential benefits of the approach in improving the overall network availability and the capability to support quality of resilience classes. Then, we study how the structural properties of a network topology can be used to determine heuristics to select a suitable spine and compare this with the case where all network components have the same availability. This is followed by a numerical based study comparing the heuristics with all possible spanning tree based spines for sample topologies. Our results demonstrate how to best design a physical network to support protection methods in achieving high levels of availability efficiently.

Keywords: availability; quality of resilience classes; network design

1. Introduction

Communication networks are one of the critical national infrastructures upon which society depends [1], thus it is imperative that they are highly available and resilient to failure. In particular networks need to guarantee that all node pairs involved in critical service communications (e.g., financial transactions, emergency calls, smart grid communications, etc.) have a high end-to-end availability. The traditional approach to improving availability in systems is to add parallel redundancy [2], which in the context of typical optical backbone networks this would imply adding additional links and possibly nodes to the network topology to support additional parallel routes. However, from a service provider's perspective communication networking is increasingly becoming a commodity type of business with severe cost constraints limiting improvements to network availability. Thus adding links to nationwide or continent wide backbone networks simply to improve availability is difficult to economically justify. Furthermore, only a small number of users and services need very high levels of availability and these users/services produce only a small fraction of the

total network traffic. Unfortunately, the small amount of high availability traffic drives the network design giving rise to a free rider scenario where the majority of customers get a higher level of availability than they need or are willing to pay for. Hence, from a service provider's perspective, there is a need to support classes of quality of resilience (QoR) in a fashion similar to quality of service classes. The basic concept is to categorize traffic into classes and provide different levels of availability and fault protection for each class. The goal of providing QoR classes is to just meet availability requirements without over-engineering. the network for the lowest classes of traffic. Providing quality of resilience classes has been mentioned in the current literature in both a qualitative and quantitative fashion (see the survey paper [3]). Most of the current work focuses on providing QoR classes by using different restoration mechanisms per traffic type in a particular network layer (e.g., WDM, MPLS, etc.). For example, providing gold, silver and bronze service classes by giving the gold traffic dedicated reserved bandwidth backup paths, silver class shared backup path restoration and the bronze class no protection relying on rerouting after failure. Other approaches in the literature address the problem from upper layers by having an overlay network maximize path diversity and dual homing or preconfiguring logical rings or protection cycles (p-cycles) between nodes for gold class traffic. These approaches suffer from the crosslayer mapping issues discussed in the literature 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) [4]. Essentially, the current approach is to take the physical network availability as a given and deploy redundancy and restoration techniques at various layers to provide QoR classes with different fault recovery capabilities and availabilities.

We believe that high availability must begin at the physical layer and work it's way up the various layers. Note that typical service provider optical backbone networks are at least two-way connected supporting some number of disjoint/partially disjoint paths between node pairs at the physical layer. Here we assume the network topology is fixed and the cost of adding links/nodes is prohibitive. We propose an innovative technique of embedding a higher availability sub-structure into the network at the physical layer to improve the overall network availability without substantial modifications to the topology. We term the high availability sub-structure portion of the network the *spine* [5]. 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 would 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. One can think of the spine approach as assuming a restoration method (path restoration) or set of restoration methods (i.e., no protection, shared backup path, dedicated backup path etc.) are to be used, then determining how should availabilities be assigned to the physical network components to best support the availability requirements.

The higher availability of the spine, in comparison to the non-spine part of the network, can be accomplished using a variety of techniques. For example, on the spine more expensive equipment can be utilized that is arranged and configured to provide high availability (e.g. hot standby line card, redundant fans, etc.) with redundant equipment deployed locally in parallel as needed (e.g., hot standby fiber in physically diverse duct, etc.). Also, the equipment along the spine can be situated to increase the mean time to failure (MTTF) using a number of techniques such as longer back up power supplies, better heating/cooling, stronger outside cabinets, underground cabling instead of above ground, etc. In a similar vein, methods can be employed to reduce the mean time to repair (MTTR) along the spine. For instance, one can follow best practices and training procedures as determined by several government and trade organizations (e.g., NRIC, FCC, TIA) and standards bodies (e.g., ITU) [6]. The operator can pre-position spare parts, equipment, software and test equipment along the spine. Similarly, the network operations center (NOC) can more closely monitor the spine portion of the network. Additionally, the operator can assign the most experienced staff to the operations, administration and management (OAM) of the spine portion of the network. Many of the methods above are employed in other critical infrastructures (e.g., the power grid) and industries and studies show that the average MTTR can be reduced by 5 - 25% resulting in a significant improvement in the availability. Of course exactly which combination of techniques (hardware, equipment siting, workforce training, etc.)

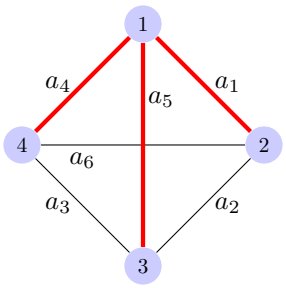


Figure 1: Full mesh network (thicker red lines denote *Spine*)

are adopted to improve the reliability of the spine will depend on the cost versus benefit structure of the network owner. Even using techniques to improve the MTTF and MTTR of links and nodes that comprise the spine, we assume additional protection, either end-to-end, segment or local [7] is needed to achieve the desired level of end-end availability for the most stringent QoR class.

In this paper, we explore the spine concept and its potential advantages. We will assume that high availability communication services between all node pairs in the network is required and thus the spine will be a spanning tree. We show that if intelligently deployed, the spine approach can be leveraged to provide higher overall average end-to-end availability or lower downtime per year efficiently. The remainder of the paper is organized as follows. Section 2 presents the spine concept through analysis of a simple network, which shows that the spine can be advantageous from the point of view of the average end-to-end availability and support a wider spread in the availabilities. In Section 3 we study how the structural properties of the network topology can be used to determine heuristics to select a suitable spine and compare this with the case where all network components have the same availability. This is followed by a numerical based study comparing the heuristics with all possible spanning tree based spines in Section 4. Section 5 studies the sensitivity of the heuristic spine selection methods, the effects of heterogenous availabilities and other practical issues including cost. Our conclusions and future work are given in Section 6.

2. The Spine Concept

The spine concept is to take a physical network topology graph $\mathcal{G} = (\mathcal{N}, \mathcal{L})$ which consists of a set of $\mathcal{N}$ nodes and a set $\mathcal{L}$ of links (undirected edges), then embed a substructure $\mathcal{G}_s = (\mathcal{N}_s, \mathcal{L}_s)$ with higher availability in a fashion so as to efficiently improve the overall availability or some availability based metric. In the general case the spine structure could take any subgraph form as dictated by availability requirements and cost. Here we assume that high availability communication service is needed between all $|\mathcal{N}| \times (|\mathcal{N}| - 1)$ source-destination pairs and the spine takes the form of a spanning tree. We illustrate the potential of the spine concept via a simple example. Consider a full-mesh four node network as shown in Figure 1. For simplicity, we restrict our study to differentiated availability of links only (i.e., nodes are assumed to not fail). To improve the end-to-end availability, we assume the network has the ability to employ disjoint working and backup path protection for each source-destination pair if desired. We ignore the option of multiple backup paths (both two and three hop). Thus each of the 12 source-destination pairs has a single hop direct working path (WP) and a disjoint two hop backup path (BP). Let $A_{\mathcal{S}}$ denote the average over all source-destination pairs of the end-to-end availability of a flow between a source-destination pair. $A_{\mathcal{S}}$ can be found using standard parallel and series availability calculations [2]. First we study the homogeneous case, $a_l = a$, for all $l \in \mathcal{L}$. The average system availability $A_{\mathcal{S}}$ is simply the parallel combination of the one hop working path and a two hop backup path which is given by:

$$A_{\mathcal{S}}(a) = 1 - (1 - a)(1 - a^2) = -a^3 + a^2 + a \tag{1}$$

Now, lets consider the non-homogeneous edge availability case corresponding the spine concept. We define a spanning tree as the spine consisting of edges 1, 5 and 4 as shown by the thicker red lines in

Table 1: Effect of varying ϵ on $A_{\mathcal{S}}$ and Downtime

Case	$A_{\mathcal{S}}$	Downtime (hours/year)
$a = .9$, $\epsilon = 0$	.981	166.44
$a = .9$, $\epsilon = 0.09$	.99712	25.23756
$a = .9$ $\epsilon = 0.099$	.999701	2.61749

Table 2: Flow Availability Values with No Protection

Case		1Hop WP	2Hop WP	3Hop WP
$a = .9$, $\epsilon = 0$	All $s-d$ pairs the same	0.9	0.81	0.729
$a = .9$, $\epsilon = 0.09$	6 $s-d$ pairs with 1Hop WP ON spine	0.99	0.8019	0.64595
–	6 $s-d$ pairs with 1Hop WP OFF spine	0.81	0.9801 or 0.6561	0.79388

Table 3: Flow Availability Values with Protection

Case		1Hop ∥ 2Hop	1Hop ∥ 3Hop	2Hop ∥ 2Hop
$a = .9$, $\epsilon = 0$	All $s-d$ pairs the same	0.981	0.9729	0.9639
$a = .9$, $\epsilon = 0.09$	6 $s-d$ pairs with 1Hop WP ON spine	0.998	0.99645	0.960756
–	6 $s-d$ pairs with 1Hop WP OFF spine	0.996219 or 0.934659	0.96084	0.993156

Figure 1. Further we assume the availability of edges on the *spine* (a_1, a_4, a_5) are equal with value a_S and the availability of the edges off the spine (a_2, a_3, a_6) are equal with value a_O. Six of the node *s-d* pairs have a single hop WP on the spine and a two hop BP with one hop on the *spine*, so the corresponding availability is $1-(1-a_S)(1-a_Sa_O)$. The other six node pairs have a WP with two hops on the spine and a single hop BP off the spine, and the corresponding availability is $1-(1-a_S^2)(1-a_O)$. So, the average end-to-end availability, as a function of a_S and a_O is: $A_{\mathcal{S}}(a_S, a_0) = \frac{1}{12}(6(1-(1-a_S)(1-a_Sa_O))+6(1-(1-a_S^2)(1-a_O)))$. If we assume that $a_S = a+\epsilon$ and $a_O = a-\epsilon$, then $A_{\mathcal{S}}$ can be shown to be

$$A_S(a,\epsilon) = -a^3 + (1-\epsilon)a^2 + (1+\epsilon)a + a\epsilon^2 + \epsilon^3 \tag{2}$$

Note, that since $a_S = a+\epsilon$ and $a_O = a-\epsilon$, then the average link availability and the sum of the link availabilities network wide are the same for the spine based network and the homogeneous case (i.e., $\sum a_i = 6a$). We define δ as the difference in $A_{\mathcal{S}}$ between the the spine and homogeneous scenarios, then $\delta = A_S(a,\epsilon) - A_{\mathcal{S}}(a)$, which can be shown to be $\delta = \epsilon^3 + a\epsilon^2 + a\epsilon(1-a)$, and $\delta > 0$ if $\epsilon > 0, a > 0$. Hence using edges with different availabilities results in larger average availability than using an homogeneous edge availability. Thus the spine has the potential to improve the average end-to-end availability.

In Table 1, we show numerical results of the effects of varying ϵ on $A_{\mathcal{S}}$ and the downtime per year for the four node full mesh network. From the table one can clearly see that embedding a spine with differential availability of the links has the potential to improve $A_{\mathcal{S}}$. We also note that in the spine the different *s-d* node pairs do not always get the same level of availability. For example, when $\epsilon = 0.09$ the group of six *s-d* pairs with a single hop WP on the spine have end-to-end availability of 0.998, while the second group of *s-d* pairs with a two hop WP on the spine have end-to-end availability of 0.9962. Observe that both groups have an end-to-end availability greater than the uniform end-to-end availability provided by the $\epsilon = 0$ homogeneous case. An important point is that the choice of the spanning tree spine is not unique in maximizing $A_{\mathcal{S}}$ as selecting edges 1, 2, and 6 results in the same $A_{\mathcal{S}}$. However, the choice of the spine is not arbitrary as selecting edges 1, 5, and 6 for the spine results in a lower $A_{\mathcal{S}}$.

In addition to improving $A_{\mathcal{S}}$, the spine also can provide a wider range of availability options for deploying QoR classes. For example, consider the scenario where one class has no protection (i.e., only a WP) and the other class has protection with a WP and disjoint BP. For the network of Figure 1 with no protection, Table 2 lists the availabilities that a service provider can achieve by routing the unprotected flows on routes of different length. For the spine based network we group the $s-d$ pairs into two groups based on whether they have a direct one hop WP route ON the spine {(1,2), (1,3), (1,4), (2,1), (3,1), (4,1)} or a one hop

WP route OFF the spine {(2,3), (2,4), (3,4), (3,2), (4,2),(4,3)}. Notice that in the case of the spine based network (i.e., bottom two rows) routes can contain links that are ON the spine, OFF the spine or a mix of ON and OFF the spine links. Hence there is a greater range of availabilities in the spine case (i.e., max - min = 0.344 for spine vs. 0.171 for no spine case). Table 3 lists the availability values that can be selected from with routing in the case of a WP with a disjoint BP. As in the no protection case, the spine-based network provides a wider range in the availabilities (i.e., range = .0633 for spine vs. 0.0171 no spine case). From a service provider perspective, one can compose classes of resilience in the spine based network by combining routing and protection/no protection to provide top of the line customers a higher quality of resilience than in the non-spine based network without giving the best effort lower paying customers a corresponding increase in resilience.

Now we consider a slightly different scenario of, given the topology, what is the effect of improving the availability of the components that make up the spine while leaving the rest of the network untouched. Specifically, we assume that $a_S = a + \Delta$ and $a_O = a$. Again considering the four node network in Figure 1 with the spine consisting of edges 1, 5 and 4 as shown by the thicker red lines, then A_S can be shown to be

$$A_S(a, \Delta) = \frac{1}{2}[-2a^3 + 2a^2 + 2a - (4a^2 - 3a - 1)\Delta - (2a - 1)\Delta^2]. \quad (3)$$

Figure 2 shows the average downtime in minutes per year for different a and Δ combinations. Each line corresponds to one Δ value, starting from zero (top line) and ending with 0.09999 in 100 steps, while varying the value of a. Thus each line shows how the average downtime decreases with increasing link availability a for a specific value of Δ. The inset figure in the top right corner is a magnification of the far right of the orginal figure. From Figure 2, one can see that to achieve a specific downtime it is better to increase Δ than a. Hence, from a system downtime point of view it is more effective to increase the availability of the spine components then to increase the availability of all the components in the network.

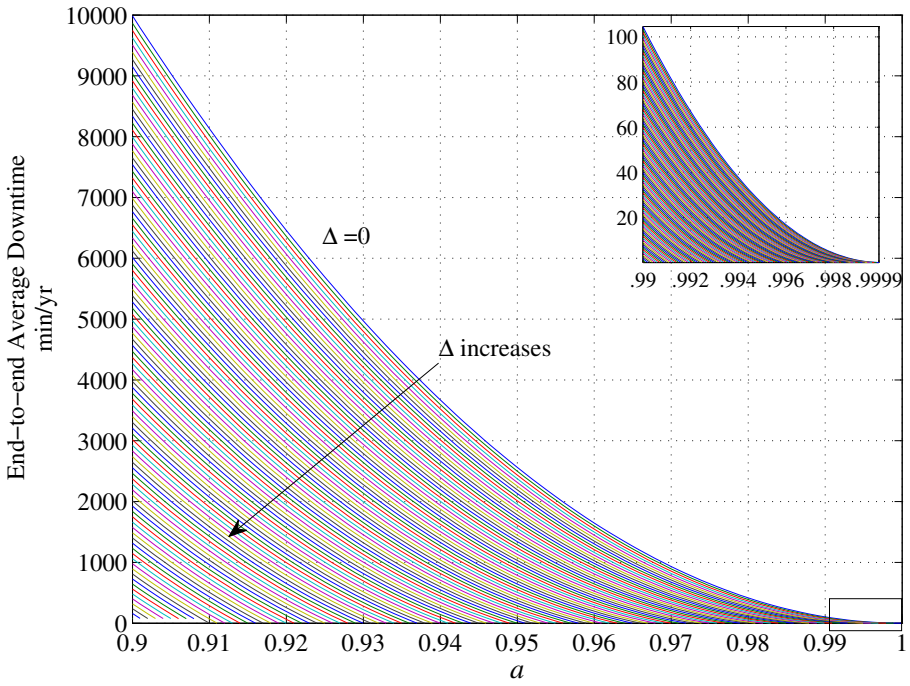
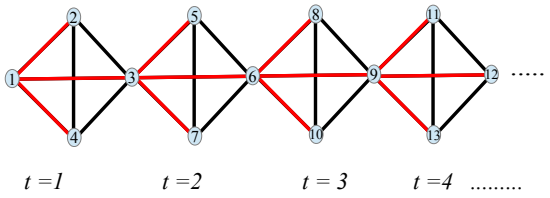
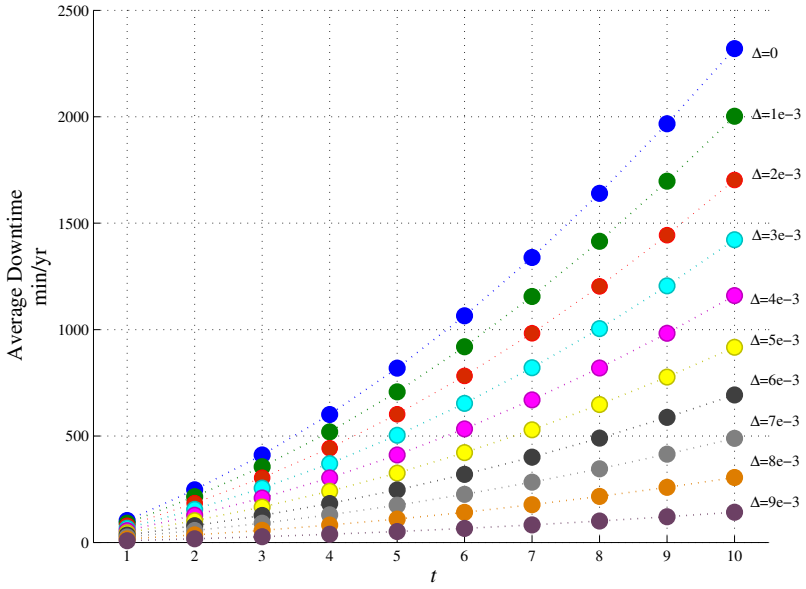


Figure 2: Average Downtime corresponding to A_S versus a and Δ.

We expect the spine to be a more beneficial approach for large networks, where the longer paths between



(a) Extension version of Figure 1 network with t stages.



(b) Average Downtime for $t = 1 : 10$, with $a = 0.99$, $\Delta = 0 : 9e-3$ with 1e-3 step size.

Figure 3: The effect of differential links availabilities on different sizes networks

node pairs decreases the end-to-end flow availability significantly. Consider an extended version of the four node network of Figure 1, if we repeat the same structure with the same spine layout, we can produce a chain-like network as shown in Figure 3a with t stages (i.e., t repetitions). The overall average availability A_S considering each WP on the spine and the BP as the corresponding min-hop edge-disjoint path can be derived as:

$$A_S = \frac{6}{n(n-1)}[t(a_S + a_S a_O - a_S^2 a_O + a_S^2 + a_O - a_S^2 a_O) + \sum_{r=2}^{t}(t-r+1) \\ [a_S^r + a_S^r a_O^r - a_S^{2r} a_O^r + 2(a_S^{r+1} + a_S^{r-1} a_O^r - a_S^{2r} a_O^r)]] \quad (4)$$

where n is the number of nodes and t is the number of repetitions of the original network structure (i.e. *stages*). As above we assume that $a_S = a + \Delta$ and $a_O = a$. In Figure 3b, we show the average downtime (in min/yr) for different t stage networks with a fixed $a = 0.99$ and Δ. A set of values are generated by varying Δ in steps of 0.001 over the range of 0 to 0.009. The top set of points in the figure shows the downtime for

a homogenous case (with $\Delta = 0$). It can be seen that introducing differential link availability reduces the average downtime even for the larger networks as shown by the lower set of points. Note that for a specific Δ, the absolute change in average downtime is greater the larger the network.

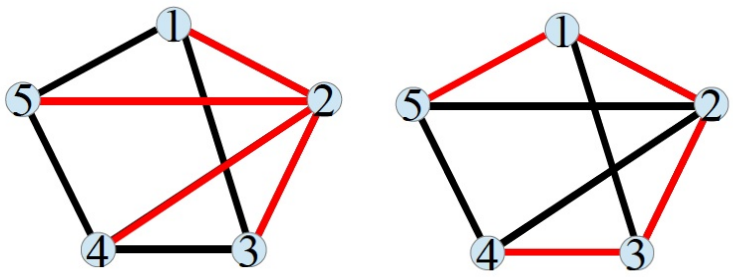
Lastly we note that the choice of a subgraph selected as the spine impacts the overall availability $A_{\mathcal{S}}$. Consider the simple 5-node network shown in Figure 4 with two different spine layouts. The spine in the leftmost network is a star whereas the spine in the rightmost network is ring-like. As above, we assume that $a_S = a + \Delta$ and $a_O = a$ and one can show that for the leftmost star-like spine network

$$A_S(a, \Delta) = \frac{1}{5}[-a^4 - 4a^3 + 6a^2 + 4a - \Delta^4 + 4\Delta^3 + (2a^2 + 4a + 2)\Delta^2 - (4a^2 - 4a)\Delta]. \tag{5}$$

Similarly for the rightmost ring-like spine network it can be shown that

$$A_S(a, \Delta) = \frac{1}{10}[a^4 - 5a^3 + 9a^2 + 8a - (3a - 1)\Delta^4 - (12a^2 - 4a - 2)\Delta^3 - (18a^3 - 6a^2 + a - 3)\Delta^2 \\ -(12a^4 - 4a^3 + 8a^2 - 12a - 4)\Delta]. \tag{6}$$

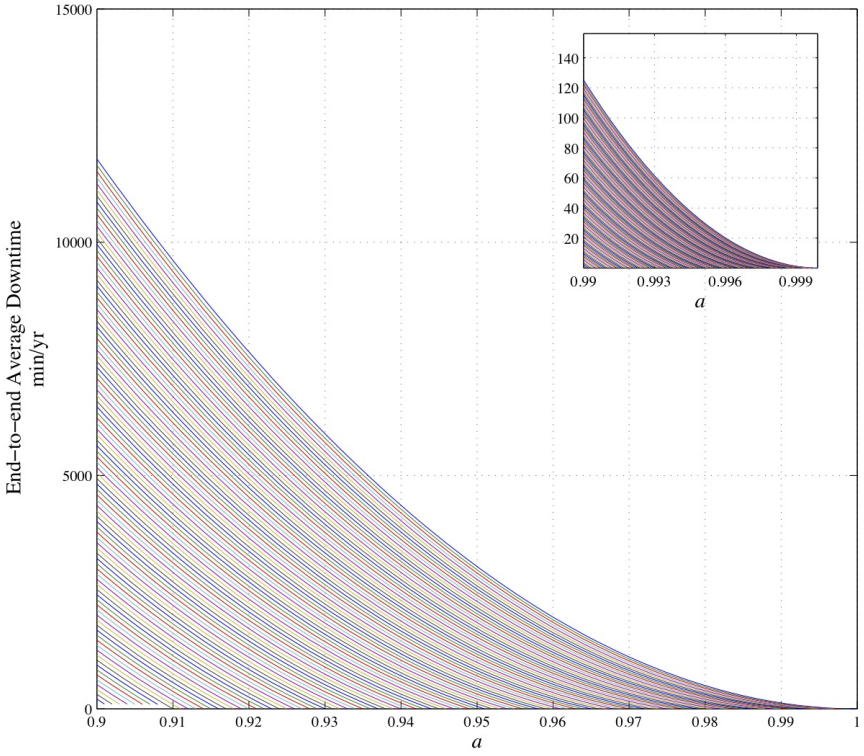
Figure 5 (a) and (b) show plots of the downtime per year for different a and Δ for the star-like and ring-like spines respectively. Each line in the downtime plots corresponds to one Δ value, starting from zero (top line) and ending with 0.09999 in 100 steps, while varying the value of a. The spine in Figure 5a has much lower downtimes than the one in Figure 5b for a given a and Δ. Hence, the star-like spine is more efficient to reach a target downtime level and would be preferred to the ring-like spine.



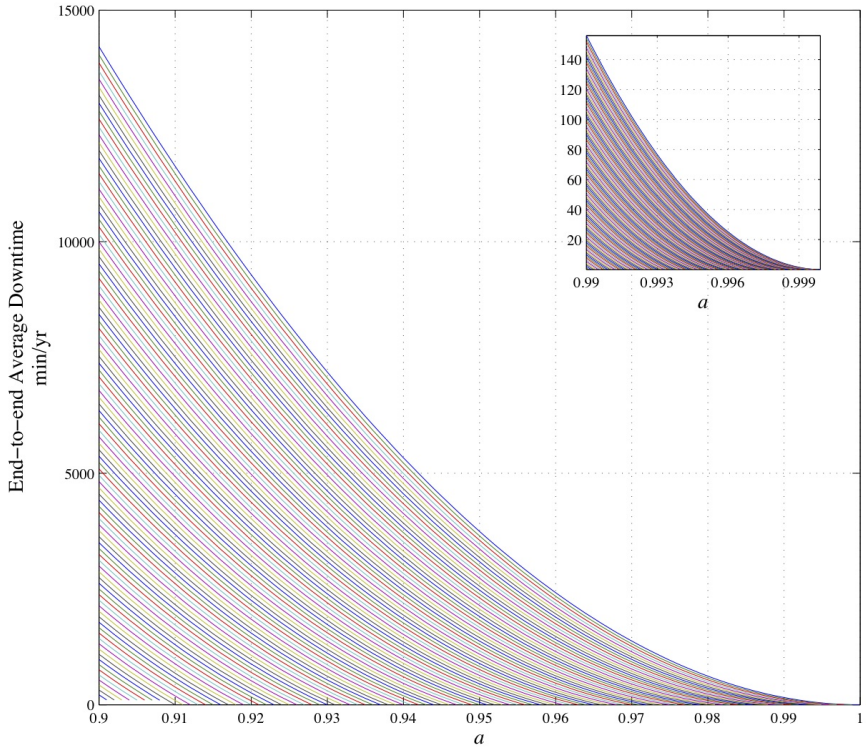
(a) Star-like spine (b) Ring-like spine

Figure 4: An example of a 5 nodes network with two spine designs

In general, the choice of the spine is not unique and its selection not only impacts the average end-to-end availability $A_{\mathcal{S}}$, but also the variability of the availability among $s - d$ pairs and the range of availabilities that can be selected by routing. For a realistic network topology, there will be many possible candidates to select the spine from and many factors come in to play in selecting the spine. For example, the length of the spine diameter ($d_{\mathcal{S}}$) and the value of Δ are related. For any given flow, we may require the working path availability A_f^{WP} to be larger than the flow backup path availability A_f^{BP}. Thus $A_f^{WP} \geqslant A_f^{BP}$. Consequently, ${a_{\mathcal{S}}}^{hc^{WP}} \geqslant {a_O}^{hc^{BP}}$, where hc^{WP} and hc^{BP} are the hop count for working and backup paths respectively. This relation should hold for all flows, and the worst case can be found as flow with longest WP and shortest BP. The longest WP is obviously the diameter of the spine ($di_{\mathcal{S}}$), and the shortest can be one hop — this is a conservative approximation. Hence, if ${a_{\mathcal{S}}}^{di_{\mathcal{S}}} \geqslant a_O$, then $di_{\mathcal{S}} \leqslant \frac{\ln a_O}{\ln a_{\mathcal{S}}}$. This also constrains the minimum Δ value that can be applied to a given spine with a specific diameter, where $\Delta \geqslant (a^{1/di_{\mathcal{S}}} - a)$ must hold for the spine meet the constraint. Hence, small values of Δ are worthwhile only for short $di_{\mathcal{S}}$. In the following section, we study how the properties of the network topology can be used to select a good spine.



(a) Star-like spine



(b) Ring-like spine

Figure 5: Average Downtime corresponding to $A_{\mathcal{S}}$ versus a and Δ for the two 5 nodes examples

3. Exploring *Spine* Selection

Here, we consider how to select a good spine using minimum cost spanning trees, where the cost of using a link (or edge) was defined to take into account the edge betweenness centrality and the edge degree. The objective was to define the spine so that it would most likely include the edges that are important from the structural point of view of the network topology. However a spine is only considered admissible if an edge-disjoint WP and BP path can be calculated for each end-to-end $s-d$ node pair. Before presenting how we generate and evaluate candidate spines, we detail our notation and provide some definitions.

3.1. Notation

Sets:

$\mathcal{N}$ set of physical nodes in the graph.
$\mathcal{L}$ set of physical links in the graph (undirected edges).
$\mathcal{G}$ network graph: $\mathcal{G} = (\mathcal{N}, \mathcal{L})$.
$\mathcal{F}$ set of end-to-end flows
$\mathcal{S}$ set of links in the *Spine*.
$\mathcal{G}_{\mathcal{S}}$ network subgraph defined by the spine, $\mathcal{G}_{\mathcal{S}} = (\mathcal{N}, \mathcal{S})$.

Indexes:

n node index.
l link (edge) index ($l \in \mathcal{L}$).
f a bidirectional symmetric flow ($f \in \mathcal{F}$).
s, d end nodes of a flow ($s, d \in \mathcal{N}$).

Paths:

WP_f Working Path for flow f.
BP_f Backup Path for flow f.

Availability:

a_l availability of link l.
A_f^{WP} Working Path availability for flow f:

$$A_f^{WP} = \prod_{l \in WP} a_l \tag{7}$$

A_f^{BP} Backup Path Availability for flow f (similar to equation (7)).
A_f availability of flow f. Assuming WP_f and BP_f are edge-disjoint, $A_f = 1 - (1 - A_f^{WP})(1 - A_f^{BP})$.
$A_{\mathcal{S}}^{WP}$ average value of A_f^{WP} when WP on the *Spine*.
$A_{\mathcal{S}}^{BP}$ average value of A_f^{BP} when WP on the *Spine*.
$A_{\mathcal{S}}$ average value of A_f when the WP on the *Spine*.
σ_S^{WP} standard deviation around $A_{\mathcal{S}}^{WP}$.
σ_S standard deviation around $A_{\mathcal{S}}$.

Performance and Structural measures:

eb_l The edge l betweenness centrality which is determined from:

$$eb_l = \frac{2}{|\mathcal{N}|(|\mathcal{N}|-1)} \sum_{s,d \in \mathcal{N}} \frac{\sigma(s,d|l)}{\sigma(s,d)} \tag{8}$$

where $\sigma(s,d)$ is the number of shortest paths between nodes s and d and $\sigma(s,d|l)$ is the number of those paths that use edge l.

$eb_{\mathcal{S}}$ ($eb_{\mathcal{G}}$) is the average value of eb_l in $\mathcal{G}_S$ ($\mathcal{G}$), that is considering only the edges in $\mathcal{S}$ ($\mathcal{L}$).
$h_{\mathcal{S}}$ ($h_{\mathcal{G}}$) is the average shortest paths in $\mathcal{G}_{\mathcal{S}}(\mathcal{G})$.
ed_l is the degree of edge l, defined as the sum of the degree of the edge's end nodes.
$ed_{\mathcal{S}}$ ($ed_{\mathcal{G}}$) is the average of ed_l over all edges in $\mathcal{G}_{\mathcal{S}}(\mathcal{G})$.

$di_{\mathcal{S}}$ ($di_{\mathcal{S}}$) is the spine diameter, that is the length (hops) of the longest shortest path in $\mathcal{G}_{\mathcal{S}}(\mathcal{G})$.
c_l cost of using edge l.
$PL_{\mathcal{S}}$ ($PL_{\mathcal{G}}$) is the total number of links (hops) that all flows in $\mathcal{F}$ traverse as WPs and BPs in $\mathcal{G}_S$ ($\mathcal{G}$)
$\triangle PL\%$ $\frac{PL_{\mathcal{S}}-PL_{\mathcal{G}}}{PL_{\mathcal{G}}} \times 100$

3.2. Generating Candidate Spines

To generate candidate spines, we used Kruskal's minimum spanning tree (MST) algorithm with the cost of the edges c_l^i defined as a weighted combination related to the edge betweenness centrality and the edge degree. The costs of the edges c_l^i, $i \in \{A, B, C, D\}$ were determined as follows:

- Case A: for a given $\alpha > 0$, the larger the edge degree and the larger the edge betweenness centrality, the smaller the cost of edge l:

$$c_l^A = (1-\alpha)\frac{(\min_l ed_l)}{ed_l} + \alpha\frac{(\min_l eb_l)}{eb_l} \tag{9}$$

- Case B: for a given $\alpha > 0$, the larger the edge degree and the smaller the edge betweenness centrality, the smaller the cost of edge l:

$$c_l^B = (1-\alpha)\frac{(\min_l ed_l)}{ed_l} + \alpha\frac{eb_l}{(\max_l eb_l)} \tag{10}$$

- Case C: for a given $\alpha > 0$, the smaller the edge degree and the larger the edge betweenness centrality, the smaller the cost of edge l:

$$c_l^C = (1-\alpha)\frac{ed_l}{(\max_l ed_l)} + \alpha\frac{(\min_l eb_l)}{eb_l} \tag{11}$$

- Case D: for a given $\alpha > 0$, the smaller edge degree and the smaller edge betweenness centrality, the smaller the cost of edge l:

$$c_l^D = (1-\alpha)\frac{ed_l}{(\max_l ed_l)} + \alpha\frac{eb_l}{(\max_l eb_l)} \tag{12}$$

In all cases (i.e., A - D) the weight α was varied from zero to one in increments of 0.1. algorithm [8] was used for generating a MST for each value of α. If the resulting MST was equal to one previously obtained, it was dropped. Also, if the obtained MST (*Spine*) did not allow for all WPs in the spine to be protected by an edge disjoint BP, the MST was dropped. In this case, the set (X) of all the common edges between a WP and its BP for all *s-d* pairs was collected. Then, sequentially, each combination (1 to $|X|$) of the common edges was temporarily removed from the graph and Kruskal's algorithm was again used, until either an admissible MST was obtained or the network became disconnected.

As in Section 2, we assume all links on the spine have the same availability $a_l = a_S \ \forall l \in \mathcal{S}$ and all links off the spine have the same availability $a_l = a_O \ \forall l \in \mathcal{L} - \mathcal{S}$. The WPs were routed entirely on the spine while each BP, edge-disjoint with the corresponding WP, was calculated with high edge cost on the spine (i.e., to avoid routing the BP on the spine)[1]. Specifically, prior to determining each BP, the cost of the edges of the protected WP was defined equal to a sufficiently large number, the cost of the rest of the edges in the spine was increased and the remaining edges had their cost changed to one. This way the BP is maximally edge-disjoint with the corresponding WP, while avoiding the edges in the spine (if possible). The common edges were used to generate new candidate MSTs as described above.

The set of candidate spines were evaluated considering the metrics: $A_{\mathcal{S}}$, $A_{\mathcal{S}}^{WP}$, $h_{\mathcal{S}}$, $di_{\mathcal{S}}$, $\min_f A_f$, $PL_{\mathcal{S}}$ and $a'_l \ \forall l \in \mathcal{L}$ the uniform edge availability required to achieve the same $A_{\mathcal{S}}$ as the spine based solution.

[1]Note that this is not meant to be a routing problem. However, this way we make sure that the spine is capable of providing disjoint path pair for each node pair. Furthermore, the routing problem might consider multiple protection levels, e.g., dedicated, shared, or no protection, and can be designed to route flows in a static or dynamic fashion

Table 4: Test Network Topology Data

Network	$\|\mathcal{N}\|$	$\|\mathcal{L}\|$	$eb_{\mathcal{G}}$	$ed_{\mathcal{G}}$	$h_{\mathcal{G}}$	$di_{\mathcal{G}}$	$PL_{\mathcal{G}}$
Polska	12	18	0.1187	6.3333	2.1364	4	356
NSF	14	19	0.1180	5.7895	2.2418	4	559
EPAN16	16	23	0.1149	6.0870	2.6417	6	806
Italia	32	69	0.0425	9.4493	2.9315	6	3378

3.3. Numerical Results

Here we present sample results for four network topologies often adopted in the literature, Polska, NSF [9] , EPAN16 [10], and Italia [11]. In Table 4, data on the topologies of the four test networks is given. In the results presented here, we use $a_O = 0.99$ and $a_S = 0.999$ and a step size of 0.0001 in determining a'_l. Boldface is used in the table of numerical results to make the corresponding maximum (minimum) values in some columns more visible, depending on which was considered relevant for that column.

Table 5: Numerical Results for Heuristics

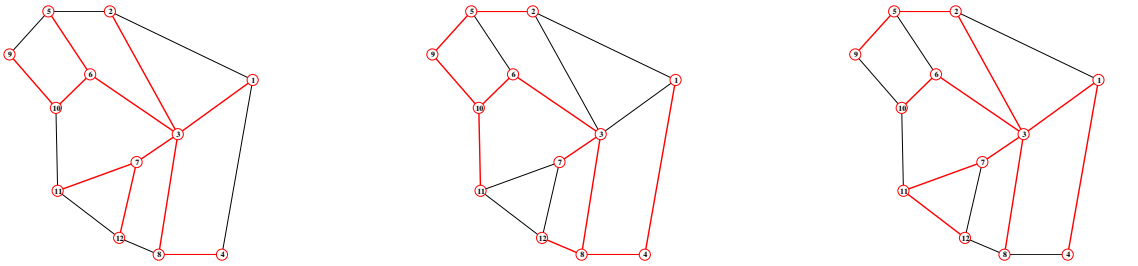
Network	eq. for c_l^i	α	$A_{\mathcal{S}}^{WP}$	$A_{\mathcal{S}}$	$\min_f A_f$	$eb_{\mathcal{S}}$	$ed_{\mathcal{S}}$	$h_{\mathcal{S}}$	$di_{\mathcal{S}}$	$PL_{\mathcal{S}}$	a'_l
Polska	(10)	0.1	**0.99734**	0.9999294	0.9997033	**0.2424**	**5.2727**	**2.6667**	**5**	**391**	0.9969
	(11)	0.6	0.99660	**0.9999480**	**0.9998543**	0.3099	4.3636	3.4091	8	425	**0.9973**
	(11)	0.8	0.99714	0.9999398	0.9997890	0.2603	4.9091	2.8636	6	395	0.9971
NSF	(9)	0.8	**0.99685**	0.9999382	0.9997986	**0.2426**	4.7692	**3.1538**	**6**	**593**	0.9973
	(11)	0.8	0.99643	**0.9999405**	0.9997527	0.2747	4.6154	3.5714	7	634	**0.9974**
	(10)	0.1	0.99662	0.9999347	**0.9998518**	0.2604	4.6154	3.3846	7	616	0.9973
	(9)	0.8	0.99672	0.9999391	0.9997859	0.2527	4.6154	3.2857	7	603	0.9973
EPAN16	(9)	0.10	**0.99670**	0.9999149	0.9997116	**0.2200**	**5.0667**	**3.3000**	**7**	**838**	0.9973
	(11)	0.5	0.99642	**0.9999218**	**0.9997412**	0.2389	4.6667	3.5833	8	890	**0.9974**
	(10)	0.30	0.99642	0.9999157	0.9997116	0.2389	4.8000	3.5833	8	861	0.9973
Italia	(9)	0.10	**0.99623**	0.9998015	0.9992670	**0.1217**	**6.9677**	**3.7742**	**8**	4367	0.9960
	(11) or (12)	0	0.99348	**0.9998115**	**0.9993595**	0.2110	4.5806	6.5423	15	4837	**0.9961**
	(9)	0.50	0.99616	0.9998041	0.9992670	0.1241	**6.9677**	3.8468	**8**	**4341**	0.9960

Table 5 shows numerical results for the four networks studied where the spines were found using the cost functions above. First we consider the results for the Polska network. In Table 5 the first row corresponds to the MSTs with largest $A_{\mathcal{S}}^{WP}$, and the second row corresponds to the MST with the largest $A_{\mathcal{S}}$, using equations (10)-(11). The corresponding spines are shown in Figures 6 and 6b, respectively. It can be seen that the largest $A_{\mathcal{S}}$ corresponds to a MST that presents a large diameter (twice the network diameter $di_{\mathcal{G}}$), while the MST with the largest $A_{\mathcal{S}}^{WP}$ has a diameter of only 5. Also note that the row with maximum $A_{\mathcal{S}}^{WP}$ has the smallest $PL_{\mathcal{S}}$. In the third row we present a compromise solution, a MST that has diameter 6 – the corresponding spine is shown in Figure 6c.

Considering the spines with the largest $A_{\mathcal{S}}$ (second row in Table 5) it is worth noting that the corresponding value of a'_l (0.9973) is larger than the average value of the edges availability (a_l) considering the spine 0.9955. This confirms the results in Section 2. In the second row of Table 5 one can also find the largest obtained value for $\min_f A_f$ (considering 1+1 protection).

The NSF network results in Table 5 list in the fourth row the values corresponding to the spine with largest $A_{\mathcal{S}}^{WP}$, in the fifth row values corresponding to the spine with largest $A_{\mathcal{S}}$, in the sixth row the values corresponding to the spine with the largest value for $\min_f A_f$ and finally in the seventh row, a compromise solution. As in the case of the Polska network a larger A_S can be obtained at the cost of a larger spine diameter. In this case the spine that results in the maximal value for $\min_f A_f$ does not coincide with the spine with the largest A_S, and it also has a lower A_S than the corresponding value in the fourth row (row of maximum $A_{\mathcal{S}}^{WP}$). The compromise solution has $A_{\mathcal{S}}^{WP}$, A_S and $\min_f A_f$ in the interval defined by the corresponding values in rows four and five of Table 5.

The results obtained for the EPAN16 network (shown in line eight to ten in Table 5) are similar to those obtained for the Polska and NSF networks. The proposed compromise solution has $A_{\mathcal{S}}$ equal to 0.9999157,



(a) The MST with the largest $A_{\mathcal{S}}^{WP}$ $di_{\mathcal{S}} = 5$ – obtained using equation (10)

(b) The MST with the largest $A_{\mathcal{S}}$ $di_{\mathcal{S}} = 8$ – obtained using equation (11)

(c) A compromise solution $di_{\mathcal{S}} = 6$ – obtained using equation (11)

Figure 6: The MSTs obtained by heuristics for Polska network – red/thicker lines represent the *Spine*

slightly larger than the corresponding value in row eight of Table 5, while presenting the same $\min_f A_f$ as shown in row eight.

In the case of the Italia network, the largest value for $A_{\mathcal{S}}^{WP}$ was obtained considering the costs given by equation (9) with α equal to 0.1, and is presented in row eleven of Table 5. A compromise solution can be found in the last line of the table. It achieves the $\min_f A_f$ value shown in line eleven and has a larger $A_{\mathcal{S}}$ than the corresponding value found in that same row. The spine resulting in the largest $A_{\mathcal{S}}$ (and $\min_f A_f$) was obtained twice (see row twelve), because when α is zero, the cost given by equations (11) and (12) is equal to $ed_l / \max_l ed_l$. It can also be observed that the required uniform edge availability (a'_l), to achieve the value of $A_{\mathcal{S}}$ in Table 5, is 0.9961, while using a spine this can be achieved with an average of 0.99404. Note that in the case of Italia network the minimal value obtained for $PL_{\mathcal{S}}$ does not correspond to the spine with maximal $A_{\mathcal{S}}^{WP}$ (as was the case of the previous networks).

Overall, from the numerical results, it was observed that for each type of cost (9)-(12) the MST with the largest average WP availability $A_{\mathcal{S}}^{WP}$ often also corresponds to the MST with the smallest average shortest path $h_{\mathcal{S}}$, the smallest $PL_{\mathcal{S}}$, the smallest diameter $di_{\mathcal{S}}$ and smallest average edge betweenness centrality $eb_{\mathcal{S}}$, and with the largest average edge degree $ed_{\mathcal{S}}$. However the MST that corresponds to the largest $A_{\mathcal{S}}^{WP}$ rarely coincides with the MST with the largest value for $A_{\mathcal{S}}$. Nevertheless the MST that maximizes $A_{\mathcal{S}}$ (for each type of cost) tends to present, a small $h_{\mathcal{S}}$, $di_{\mathcal{S}}$, $eb_{\mathcal{S}}$ (although these are usually larger than the corresponding values for the MST that maximizes $A_{\mathcal{S}}^{WP}$), and a large $ed_{\mathcal{S}}$ (although usually smaller than the MST that maximizes $A_{\mathcal{S}}^{WP}$). Also, the results from the tested networks seem to indicate that maximizing $A_{\mathcal{S}}^{WP}$ does not maximize $\min_f A_f$.

4. Considering All Spanning Trees in Determining the Spine

In this section, we study the metrics used to evaluate spine solutions over the space of all spanning trees in order to see their behavior and gain insight into spine selection.

4.1. Generating All Spanning Trees

The number of spanning trees (ST) in a connected graph $\mathcal{G}$ can be quite large even for small $|\mathcal{N}|$ and $|\mathcal{L}|$. The exact number of STs in a graph can be related to the Laplacian spectrum of the graph [12] as follows. Let A denote the $|\mathcal{N}| \times |\mathcal{N}|$ adjacency matrix of a graph, where $a_{ij} = 1$ if and only if there is a link between node i and node j, otherwise $a_{ij} = 0$. The degree matrix D is a $|\mathcal{N}| \times |\mathcal{N}|$ matrix with the node degree placed along the diagonal (i.e., d_{ii} = number of adjacent nodes of i) and zero every where else. The Laplacian matrix L of a graph is defined as $A - D$ and the eigenvalues $\lambda_i, i = 1, 2, ...|\mathcal{N}|$ of L form the Laplacian spectrum. It has been shown in the algebraic graph theory literature [12] that the number of

spanning trees in a graph can be determined from the Laplacian spectrum by:

$$\textit{No. of Spanning Trees in } \mathcal{G} = \frac{1}{n} \prod_{i,i>1} \lambda_i \tag{13}$$

Table 6 shows the number of spanning trees for the networks studied here. One can clearly see, that for even modest size networks such as EPAN16, the number of spanning trees is quite large.

Table 6: Number of spanning trees for sample networks

$\mathcal{G}$	Network	$\lvert\mathcal{N}\rvert$	$\lvert\mathcal{L}\rvert$	$\frac{\lvert\mathcal{L}\rvert}{\lvert\mathcal{N}\rvert}$	No. of ST	*Valid* STs
1	Polska	12	18	1.50	5161	1862
2	NSF	14	19	1.36	5862	1466
3	EPAN16	16	23	1.44	43.7E+03	7535
4	Italia	32	69	2.16	53.3E+14	NA

Table 7: Results considering all spanning trees

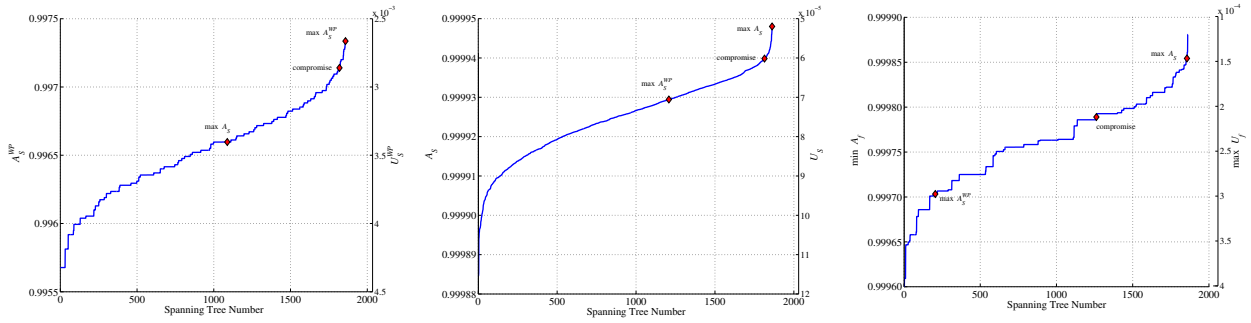
Network	$A_{\mathcal{S}}^{WP}$	$A_{\mathcal{S}}$	$\min_f A_f$	$eb_{\mathcal{S}}$	$ed_{\mathcal{S}}$	$h_{\mathcal{S}}$	$di_{\mathcal{S}}$	$PL_{\mathcal{S}}$	$\triangle PL\%$
Polska	**0.99734**	0.9999322	0.9997554	**0.2424**	5.2727	**2.6667**	**5**	385	8.2
	0.99660	**0.9999480**	0.9998543	0.3099	4.3636	3.4091	8	425	19.4
	0.99732	0.9999440	0.9997967	0.2438	**5.4545**	2.6818	**5**	**379**	6.0
	0.99654	0.9999442	**0.9998809**	0.3154	4.1818	3.4697	8	411	15.5
NSF	**0.99687**	0.9999390	0.9997642	**0.2409**	4.6154	**3.1319**	**6**	**583**	4.3
	0.99665	**0.9999424**	0.9998518	0.2578	4.6154	3.3516	7	604	8.1
	0.99637	0.9999415	**0.9998736**	0.2798	4.4615	3.6374	9	639	13.0
EPAN16	**0.99670**	0.9999149	0.9997116	0.2200	5.0667	3.3000	**7**	838	4.0
	0.99649	**0.9999254**	0.9997724	0.2344	4.6667	3.5167	8	869	7.8
	0.99552	0.9999163	**0.9998128**	0.2994	4.1333	4.4917	11	936	16.1

In order to generate all STs, we use Prim's algorithm to determine a ST implementing a binary code of size $(2^{|\mathcal{L}|})$ with $|\mathcal{L}|$ digits each corresponding to a specific link on the graph, with value 1 if the link is on the spine and 0 if not. Then, we run a counter from $(2^{|\mathcal{N}|} - 1)$ to $(2^{|\mathcal{L}|} - 1)$ to enumerate all possible combinations of links on a spine. Each generated combination of links that constructs a valid ST is saved to be further tested. A valid ST is verified by checking that the sum of all columns on the adjacency matrix of the spine is greater or equal to one and the number of links is $|\mathcal{N}| - 1$. Once all spanning trees are created, for each ST we route the WPs for all $s-d$ pairs on the spine while BPs are routed such they avoid the spine if possible but constrained to be fully disjoint from the corresponding flow's WP.

4.2. Numerical Results

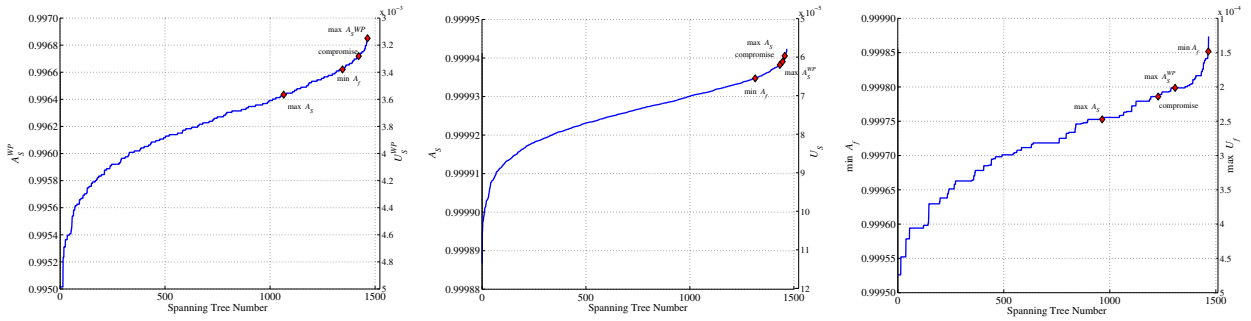
We studied the Polska, NSF, and EPAN16 networks by generating all STs and routing all *s-d* flows with disjoint protection. The Italia network was not considered due to the computational complexity. As in the previous sections all links on the spine have the same availability a_S and all links off the spine have the same availability a_O. Here we use $a_S = 0.999$ and $a_O = 0.99$. We evaluate the results using the same performance and structural measures as in the previous section. Table 7 shows the results for the networks considered. For each network in the table, the first row corresponds to the ST with the largest $A_{\mathcal{S}}^{WP}$, the second row the ST corresponding to the largest $A_{\mathcal{S}}$ and the subsequent rows can be either the ST with largest $\min_f A_f$ or a compromise solution between the results of the first two rows, typically with a lower $di_{\mathcal{S}}$ then the maximum $A_{\mathcal{S}}$ case.

We find the observations from Section 3 regarding the coincidence between $A_{\mathcal{S}}$ and relatively to average, low $eb_{\mathcal{S}}$, $h_{\mathcal{S}}$, $di_{\mathcal{S}}$, and high $ed_{\mathcal{S}}$ to hold. Similarly, the correspondence of $A_{\mathcal{S}}^{WP}$ on the spine with minimum $di_{\mathcal{S}}$ and $h_{\mathcal{S}}$. Figures 7-9 show, for each network, $A_{\mathcal{S}}^{WP}$, $A_{\mathcal{S}}$, and min-A_f for all STs examined with the



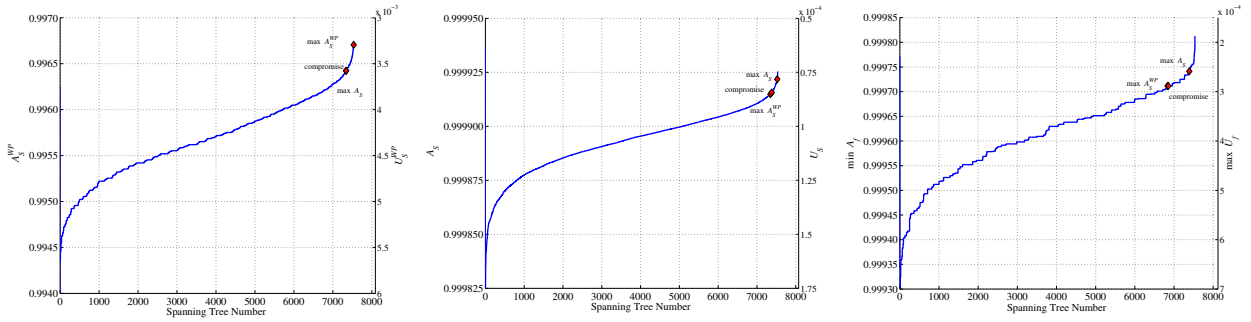
(a) (b) (c)

Figure 7: $A_{\mathcal{S}}^{WP}$, $A_{\mathcal{S}}$ & min-A_f calculated over all spanning trees for Polska Network



(a) (b) (c)

Figure 8: $A_{\mathcal{S}}^{WP}$, $A_{\mathcal{S}}$ & min-A_f calculated over all spanning trees for NSF Network



(a) (b) (c)

Figure 9: $A_{\mathcal{S}}^{WP}$, $A_{\mathcal{S}}$ & min-A_f calculated over all spanning trees for EPAN16 Network

results sorted from largest to smallest. On each plot the right side scale is the corresponding unavailability. Also, we mapped (shown as dots) the STs obtained from the heuristics on the plots. Note that the figures have fairly consistent behavior in terms of the shapes of the curves across the three networks. For $A_{\mathcal{S}}^{WP}$ there appears to be a relatively small set (in comparison to the number of STs) of STs with the largest values. Where as for $A_{\mathcal{S}}$ there is a larger percentage of STs with reasonably high values. But these values corresponds to different min-A_f. In Polska network, the difference between the largest and smallest min-A_f values is 2.89e-4, compared to 6.35e-5 for $A_{\mathcal{S}}$, which indicates that A_S values range is narrower. It might

be more appropriate to consider the *minimum* flow availability(A_f) instead of $A_{\mathcal{S}}$, to ensure that each flow can be given a route with highest possible end-to-end availability.

Comparing with the heuristics of Section 3, the largest value obtained from the heuristics for $A_{\mathcal{S}}^{WP}$ in the Polska network coincides with the maximum found over all STs in Table 7 − there are multiple STs of the same $A_{\mathcal{S}}^{WP}$ value. For the EPAN16 network, the heuristics in Section III calculated a spine with $A_{\mathcal{S}}^{WP}$ equal to 0.99670 which also coincides with the maximum found over all STs in Table 7. Regarding the case of the NSF network, the maximum value obtained by the heuristics for $A_{\mathcal{S}}^{WP}$ (0.99685) was also close to the maximum 0.99687 in Table 7.

In terms of $A_{\mathcal{S}}$, for the Polska network the maximum value found from the heuristics (0.9999480) is actually the maximum over all STs in Table 7. Similarly, the heuristic in Section 3 managed to generate a spine for the NSF network, such that the resulting A_S has the first 5 digits correct (according to Tables 5 and 7). In the case of the EPAN16 network, the first 5 digits match too. Overall these numerical results seem to indicate that the heuristics from Section III work reasonably well in identifying a viable spine for a network topology.

Regarding path lengths, Table 7 (last two columns) shows the total number of links/hops utilized by the network $PL_{\mathcal{S}}$ when the spine is considered. This is compared to the total number hops used when the link-disjoint path pair is calculated using shortest path pair (i.e., *min-min*). It is clear form the results that the spine incurs more resources. Despite this, the increase in path lengths $\triangle PL\%$ can be as low as 6%, 4.3%, and 4.0% for Polska, NSF, and EPAN16 networks, respectively.

In the next section, we study the sensitivity of the results to the link availability values a_l, the relaxation of the assumption of homogeneous spine and off-spine link availabilities and discuss the effects of monetary cost.

5. Further Analysis

5.1. Sensitivity

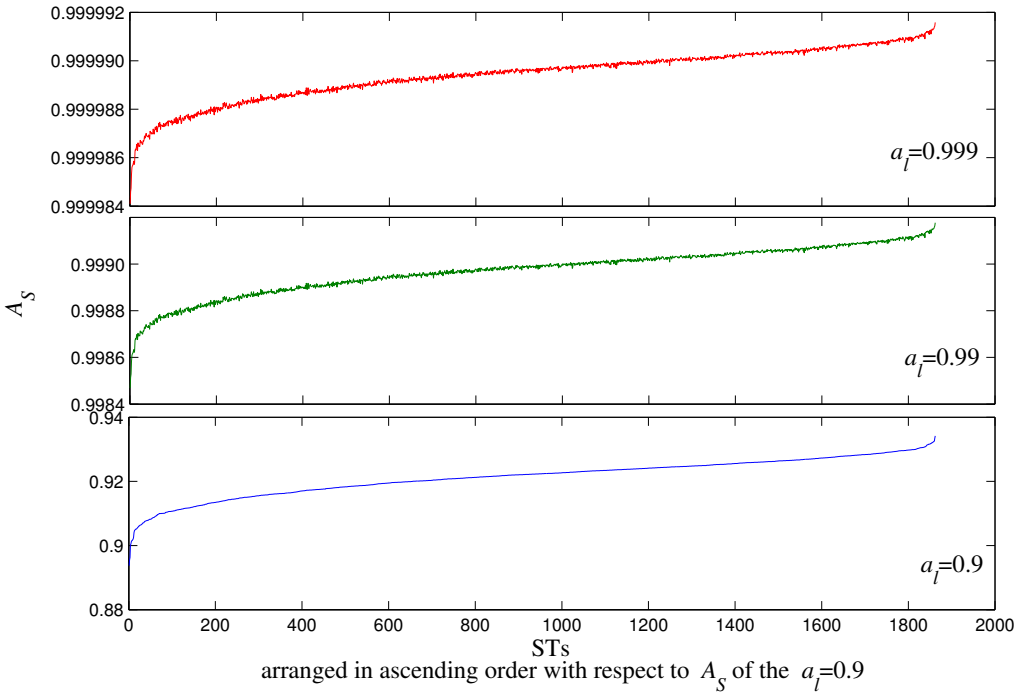
In our previous numerical analysis, we considered $a = 0.99$ and $\Delta = 0.009$ (i.e., $a_O = 0.99$, $a_{\mathcal{S}} = 0.999$). Here we examine how the results would change if we considered different values of a and Δ. Figure 10 (a) and (b) shows the availability values ($A_{\mathcal{S}}$ and minimum A_f) for all STs in the Polska network using 3 different values of $a = \{0.9, 0.99, 0.999\}$. By visual inspection, we can see that the general pattern for the ordered STs remain unchanged regardless of the metric $A_{\mathcal{S}}$ or minimum A_f. Only slight changes in terms of the ranking of the individual STs from best to worst occur in both measures.

For varying Δ, Figure 11 shows the results for $A_{\mathcal{S}}$ and minimum A_f using 5 different Δ values. When Δ changes, we see a consistent pattern for the values. However, more variation is noticed. This means that the ranking of the STs (from best to worst) with respect to a specific measure would slightly change as Δ changes and an ST might exchange its rank with another ST within its close range. In addition, for each value of Δ there can be different best/worst STs. Figure 12 visualizes this behavior, especially at large Δ values. For example, in Figure 12a at $\Delta = 6e-3$ the upper three STs (corresponds to worst $A_{\mathcal{S}}$) exchange their positions after the merging point as the variation of the values diminishes. This is also true for the NSF network as shown in Figure 12b.

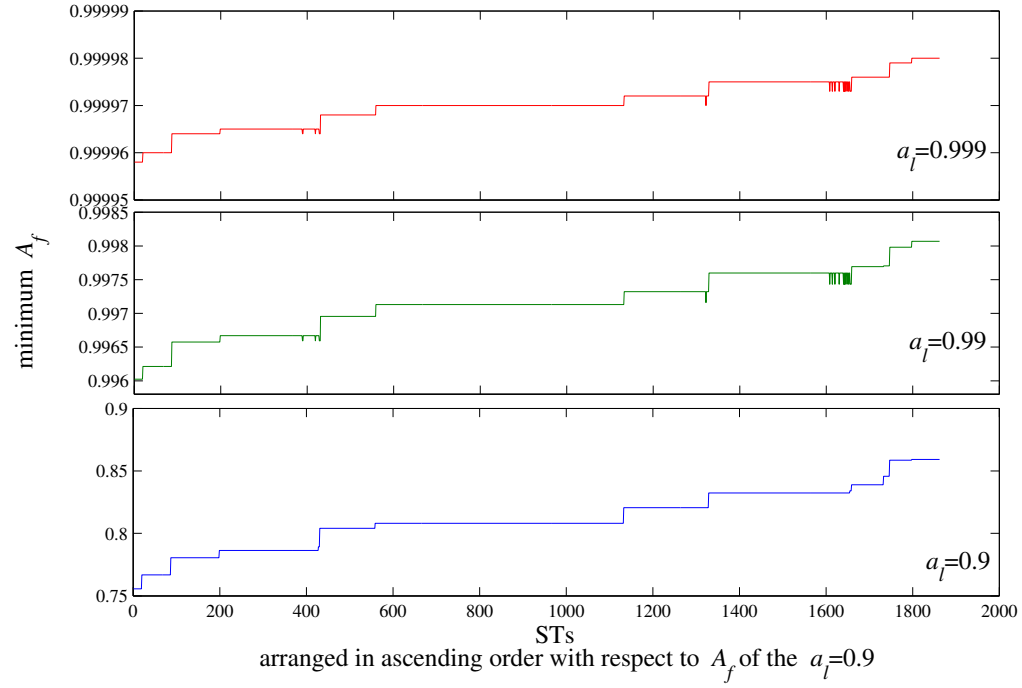
5.2. Heterogenous Scenario

So far, we considered different values for a and Δ but we also assume that all links on the spine have the same availability $a_l = a + \Delta \ \forall l \in \mathcal{S}$ and all links off the spine have the same availability $a_l = a \ \forall l \in \mathcal{L} - \mathcal{S}$. However, typically, links on a network would have different availabilities. In this part, we relax our assumption of considering homogenous link availability. We consider a distance-based link availability found in [7]. The link availability is calculated as $a_l = a_c \times a_t$ where a_t is the product of cable-ends equipments (i.e., OXC, ADM etc...), and a_c is the fiber cable availability that can be calculated from:

$$a_c = 1 - \frac{MTTR}{MTBF} \tag{14}$$

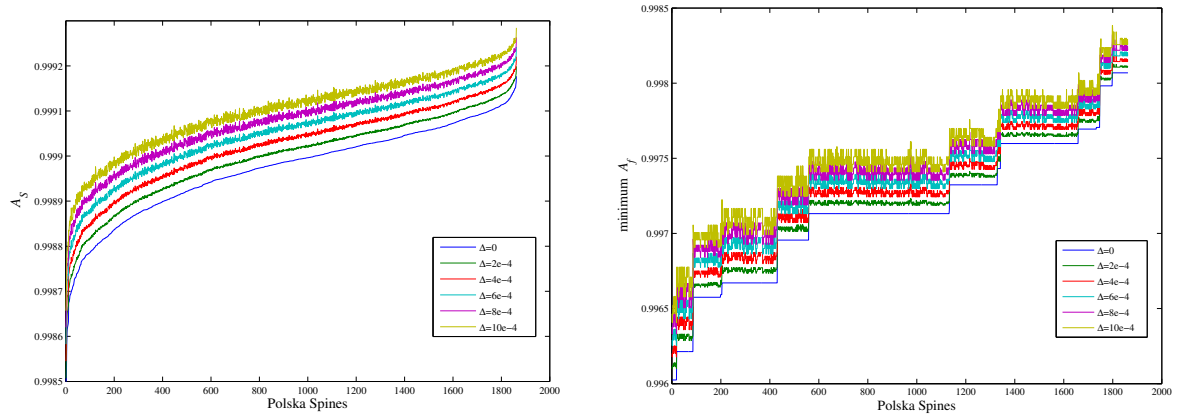


(a) $A_{\mathcal{S}}$ over all STs for different a



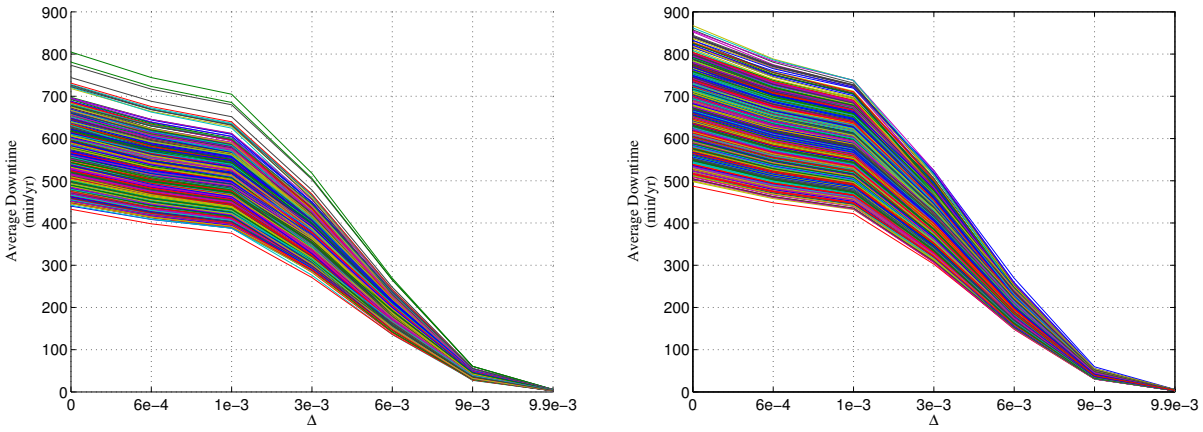
(b) Min-A_f over all STs for different a

Figure 10: The effect of varying a in Polska network.



(a) (b)

Figure 11: The effect of varying Δ in Polska network.



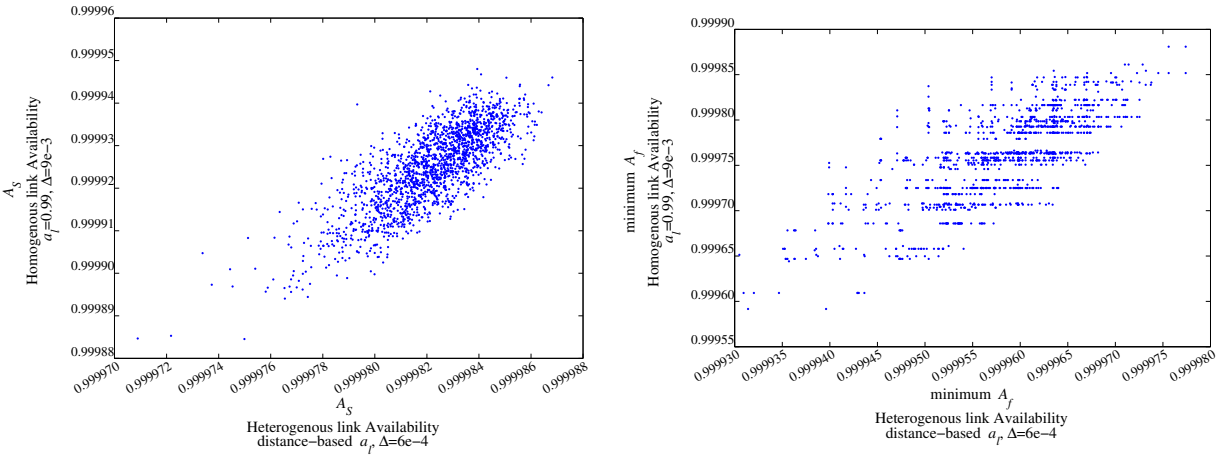
(a) Polska Network (b) NSF Network

Figure 12: Average Downtime corresponding to $A_{\mathcal{S}}$ for all STs at different Δ values.

$$MTBF_{hrs} = \frac{CC \times 365 \times 24}{\text{cable length}_{km}} \tag{15}$$

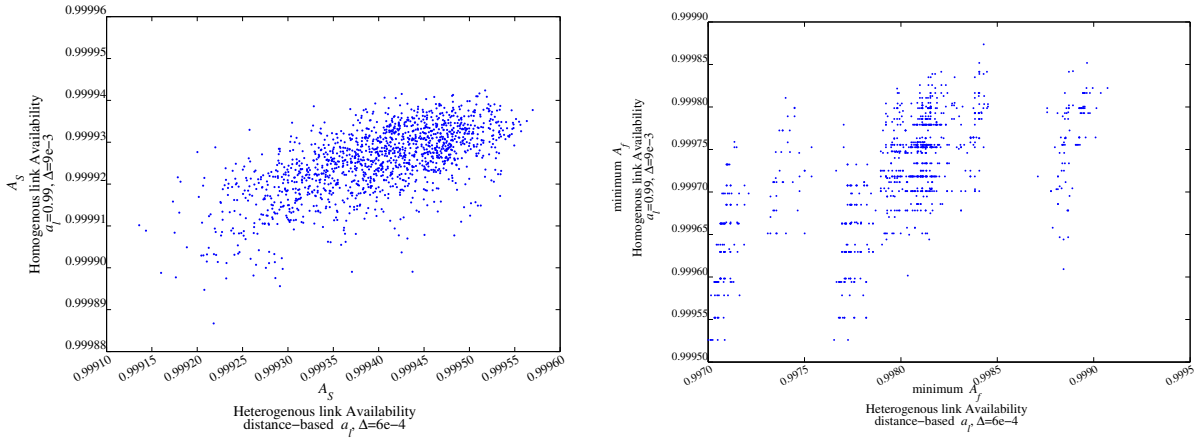
where $MTBF$ and $MTTR$ are the mean time between failures and mean time to repair in hours, respectively. CC is the cable cut metric in km.

Recall that we want to find a spine with high availability measures, and these measures vary on the different spines based on their graphical structure as we showed in the previous section. In here, we involve heterogenous link availability which complicates the problem furthermore. Now, we want to examine to what extent the added input changes the results. To inspect this, we study the Polska and NSF networks, and we calculate distance-based link availability by setting $MTTR = 24$, $CC = 450$, and $a_t = 0.9995$, and using the actual distance of the cables. Then, for all STs we calculate $A_{\mathcal{S}}$ and min-A_f with different Δ values. We compare these results to the results obtained from a homogenous case. As an example, Figures



(a) (b)

Figure 13: Scatterplot of Polska STs measures with homogenous versus heterogenous link availability.



(a) (b)

Figure 14: Scatterplot of NSF STs measures with homogenous versus heterogenous link availability.

13 and 14 show a scatterplot for the A_S and min-A_f values of the STs for Polska and NSF networks. The x-axis and y-axis depict a ST availability value in heterogenous and homogenous case, respectively. For A_S, we can see some variation for STs values around the linear correlation line which indicates an STs change in ranking.

5.3. Monetary Cost and Implementation Issues

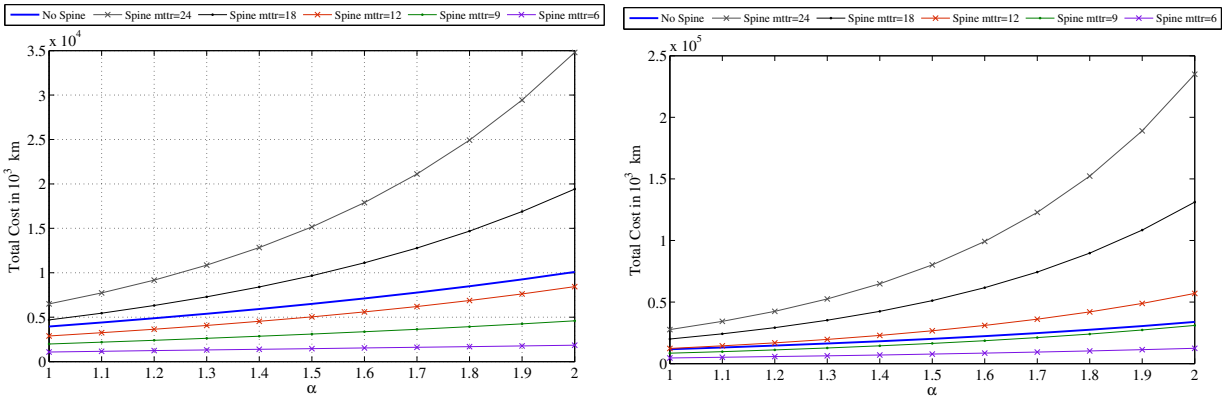
The discussion and analysis thus far illustrates the potential of the spine concept in improving A_S and reducing the average downtime per year. In reality there are several factors that will determine the usefulness and practicality of the spine approach. The paramount factor is the financial cost versus benefit tradeoff of the spine approach versus non-spine based methods of improving the availability. Note, that the cost of the spine design is the cost of improving the availability of only the spine components. On the other hand, the cost of the non-spine design is the cost of improving all the components in the network to meet the same average flow availability A_S achieved by the spine design. Hence, the spine is a monetary cost

effective option if and only if the cost of the spine design is lower than the cost of the improved non-spine design. This will depend on the financial cost structure of improving the availability for the network under consideration and the desired levels of availability.

In the networking literature the cost of improving availability has not been widely discussed, the majority of papers focus on technical techniques to improve or quantify the availability of components or systems. Financial cost is usually given in a qualitative fashion (e.g., low, medium, high) or in a few cases as a numerical value for a specific technology and application scenario [13]. Determining a precise generally applicable formula on cost of availability is difficult as the cost is dependent on a number of technical and non-technical issues and is typically scenario and organization dependent. In general one can note that the availability of information and communications technology can be improved up to certain point then there are diminishing returns with increased cost and perfect availability is not attainable (i.e., downtime = 0) [14]. A few attempts to provide mathematical models relating cost and availability have appeared in the literature. Grover and Sack [6] proposed to model the reduction of the mean-time-to-repair (MTTR) and the associated cost in terms of % of budget for improving availability as having an inverse relationship of the form $Cost = (MTTR_o/MTTR)^{1/\alpha}$ where $MTTR_o$ is the baseline mean-time-to-repair and α is a parameter. Recently [15], models the cost of increasing the mean-time-between-failure (MTBF) as polynomial function of MTBF, namely $Cost = MTBF^{\alpha} + K$ where α is a parameter and K is a constant fixed cost. Note, that these two works each focus on only one side of the techniques to improve availability. In practice, one typically adopts a two-pronged approach to increase availability by investment in organization improvement to reduce MTTR and technical improvements to directly or indirectly (e.g., backup electrical power) increase MTBF [14]. In [16], the author takes a different viewpoint and relates the cost of improving availability to potential financial loss L due to downtime. The cost function $Netcost = 1–f(A_o, c))L + c$ is proposed where A_o is the baseline availability, c is the cost of investment in improving the availability as a percentage of L and $f(A_o, c)$ is a nonlinear function relating the improvement in availability as a function of investment. Different forms of $f(A_o, c)$ are proposed such as $f_1(A_o, c) = 1–(1 - A_o)e^{-\alpha c}$ and $f_2(A_o, c) = 1–(1 - A_o)/(1 - \alpha c)$.

Another practical deployment issue is that in terms of equipment improvement one does not get continuous changes in the availability but discrete changes in the MTBF by direct component modifications (e.g., spare mirrored line card) or indirect modifications (e.g., backup power supply of 8 hours). However, adjustment of MTTR can occur in a more fine grained fashion. To illustrate this with an example, consider the candidate spines in Table 5 in the second and eleventh rows which correspond to the Polska and Italia networks, respectively. We compare the total cost of the spine and non-spine designs, where the total cost of a design is the sum of the costs of improving each link. Assume that the availability of all links on the network initially is 0.99, with MTTR = 24 *hrs*, which corresponds to MTBF = 2400. If we were to increase the MTTR in one link (i.e., to reach $a_{\mathcal{S}}$ or a'_l), the improvement is subject to the cost function $Cost = MTBF^{\alpha}$ from [15], where the MTBF value is for 1 km. For α, we use different values range from 1 to 2 with step size of 0.1, and also we use the geographic distances for links in both networks to calculate the MTBF. In addition, the links on the spine can be given MTTR values from 24 down to 6 hrs, whereas MTTR for the off-spine links in the spine design and all the links in the non-spine design remains unchanged. Figure 15 shows the results for the two networks and the two design options. It is clear from the figure that the spine design can be more cost effective for the right combination of both MTBF and MTTR even at large value of α. In practice one will typically not be able to tune the availability in a continuous fashion as in the analysis of previous sections, but there will be discrete options around which the availabilty can be tuned somewhat as in Figure 15 .

Lastly we observe that throughout this work we have focused on uncapacitated networks. Since the spine can lead to non-shortest path routes it may require more capacity in contrast to a non-spine based design. However, as discussed in the introduction the spine is primarily proposed to satisfy the requirements of high availability traffic and enable the use of QoR classes. Note that only a small fraction of the total flows are expected in the highest availability class, thus one would expect the potential capacity increases to be minimal. In the event the percentage of high availability traffic increases significantly then the capacity of the spine may need to be increased accordingly.



(a) spine in the second row of table 5

(b) spine in the eleventh row of table 5

Figure 15: Comparison between total cost of the spine design and non-spine design.

6. Conclusion

In this paper we presented the novel concept of embedding a subgraph structure with higher availability (termed the spine) in a network together with protection mechanisms to improve the overall end-to-end availability. The spine based approach was shown to have the potential to improve the network availability in a more efficient fashion compared to improving the availability of all network components in a homogeneous fashion. Heuristic spine selection methods based on structural properties of the network topology were given and the results appear promising compared to optimal spine values determined by a brute force search. Obviously, much additional work needs to be done to fully flesh out the spine concept, including detailed design algorithms that incorporate a range of node and link availabilities, the discrete nature of availabilty components and realistic monetary cost functions. The final goal is to find a way to embed a spine to achieve a maximum average end-to-end availability, and given such a spine, to identify the minimum increase in availability of the links, to achieve a given level of end-to-end availability for all node pairs, considering protection and monetary costs. In general the spine is hoped to provide larger differences in the range of availability values to quality of resilience classes resulting in less over engineering of the network to meet the most stringent availability requirements.

7. Acknowledgment

The work of Teresa Gomes has been supported in part by the Portuguese Foundation for Science and Technology under project grant PEst-OE/ EEI/UI308/2014 and by ICIS Project CENTRO-07-0224-FEDER-002003.

References

[1] T. G. Lewis, Critical Infrastructure Protection in Homeland Security: Defending a Networked Nation, Wiley-Interscience, 2006.

[2] M. Rausand, A. Hoyland, System Reliability Theory, Wiley, 2003.

[3] P. Cholda, A. Mykkeltveit, B. E. Helvik, O. J. Wittner, A. Jajszczyk, A survey of resilience differentiation frameworks in communication networks, IEEE Communications Surveys & Tutorials 9 (4) (2007) 32–55.

[4] P. Pacharintanakul, D. Tipper, Crosslayer survivable mapping in Overlay-IP-WDM networks, in: Design of Reliable Communication Networks, 2009. DRCN 2009. 7th International Workshop on, Washington, D.C., 2009, pp. 168–174.

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

[6] W. Grover, A. Sack, High availability survivable networks: When is reducing mttr better thank adding protection capacity, in: IEEE 6th International Workshop on the Design of Reliable Communication Networks (DRCN), La Rochelle, France, 2007.
[7] J.-P. Vasseur, M. Pickavet, P. Demeester, Network Recovery – Protection and Restoration of Optical, SONET-SDH, IP, and MPLS, Elsevier, 2004.
[8] J. B. Kruskal, On the shortest spanning subtree of a graph and the traveling salesman problem, Proceedings of the American Mathematical Society, 7 (1) (1956) 48–50.
[9] S. Orlowski, M. Pióro, A. Tomaszewski, R. Wessäly, SNDlib 1.0–Survivable Network Design library, Networks 55 (3) (2010) 276–286. doi:10.1002/net.20371.
URL http://www3.interscience.wiley.com/journal/122653325/abstract
[10] S. Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, B. M. R. Inkret, J. Derkacz, Pan-european optical transport networks: An availability-based comparison, Photonic Network Communications 5 (3) (2003) 203–225.
[11] M. Tornatore, G. Maier, A. Pattavina, Availability design of optical transport networks, IEEE Journal on Selected Areas in Communications 23 (8) (2005) 1520–1532.
[12] C. Godsil, G. Royle, Algebraic Graph Theory, Springer, 2001.
[13] J. Chen, L. Wosinska, C. Machuca, M. Jaeger, Cost vs. reliability performance study of fiber access network architectures, IEEE Communications Magazine (2010) 56–65.
[14] F. Piedad, M. Hawkins, High Availability: Design, Techniques, and Processes, Prentice Hall, 2001.
[15] S. Herker, W. Kiess, X. An, and A. Kirstädter, On the Trade-off between Cost and Availability of Virtual Networks, in: Proc., IFIP Networking Conference, Trondheim, Norway, 2014.
[16] U. Franke, Optimal it service availability: Shorter outages, or fewer?, IEEE Transactions on Network and Service Management 1 (9) (2012) 22–33.