Notions of Connectivity in Overlay Networks[⋆]

Yuval Emek[1], Pierre Fraigniaud[2], Amos Korman[3], Shay Kutten[4], and David Peleg[5]

[1] ETH Zurich, Zurich, Switzerland. yuval.emek@tik.ee.ethz.ch
[2] CNRS and University of Paris Diderot, France. pierref@liafa.jussieu.fr
[3] CNRS and University of Paris Diderot, France. amos.korman@liafa.jussieu.fr
[4] The Technion, Haifa, Israel. kutten@ie.technion.ac.il
[5] The Weizmann Institute of Science, Rehovot, Israel. david.peleg@weizmann.ac.il

Abstract. "How well connected is the network?" This is one of the most fundamental questions one would ask when facing the challenge of designing a communication network. Three major notions of connectivity have been considered in the literature, but in the context of traditional (single-layer) networks, they turn out to be equivalent. This paper introduces a model for studying the three notions of connectivity in *multi-layer* networks. Using this model, it is easy to demonstrate that in multi-layer networks the three notions may differ dramatically. Unfortunately, in contrast to the single-layer case, where the values of the three connectivity notions can be computed efficiently, it has been recently shown in the context of WDM networks (results that can be easily translated to our model) that the values of two of these notions of connectivity are hard to compute or even approximate in multi-layer networks. The current paper shed some positive light into the multi-layer connectivity topic: we show that the value of the third connectivity notion can be computed in polynomial time and develop an approximation for the construction of well connected overlay networks.

1 Introduction

1.1 Background and Motivation

The term "connectivity" in networks has more than one meaning, but these meanings are equivalent in "traditional" networks. While the graph theoretic definition of connectivity refers to the ability to reach every node from every other node (a.k.a. 1-*connectivity*), the term connectivity is often related also to the *survivability* of a network, namely, the ability to preserve 1-connectivity whenever some links fail.[6] In other words, a network G is said to be k-*connected* if it satisfies the following "connectivity property" (CP):

(CP1) G remains connected whenever up to $k-1$ links are erased from it.

[⋆] Supported by a France-Israel cooperation grant ("Mutli-Computing" project) from the France Ministry of Science and Israel Ministry of Science.

[6] The current paper does not deal with *node* connectivity.

However, there are also other meanings to connectivity. A network G is also said to be *k-connected* if it satisfies the following "connectivity property":

(CP2) There exist k pairwise *link-disjoint paths* from s to t for every two nodes s, t in G.

The equivalence of these two properties [13] enables numerous practical applications. For example, one of the applications of the existence of link-disjoint paths (Property (CP2)) is to ensure survivability (Property (CP1)). Often, a backup (link-disjoint) path is prepared in advance, and the traffic is diverted to the backup path whenever a link on the primary path fails. An example is the backup path protection mechanism in SONET networks (see, e.g., [15]).

A third property capturing connectivity is based on the amount of flow that can be shipped in the network between any source and any destination, defining the capacity of a single link to be 1. In other words, a network G is said to be *k-connected* if it satisfies the following "connectivity property":

(CP3) It is possible to ship k units of flow from s to t for every two nodes s, t in G.

This property too is equivalent to the first two [8], and is also used together with them. For example, routing some flow of information around congestion (which may be possible only if the network satisfies property (CP3) and thus can support this additional flow) uses the second property, i.e., it relies on the existence of multiple link-disjoint paths.

Current networks, however, offer multi-layered structures which yield significant complications when dealing with the notion of connectivity. In particular, the overlay/underlying network dichotomy plays a major role in modeling communication networks, and overlay networks such as peer-to-peer (P2P) networks, MPLS networks, IP/WDM networks, and SDH/SONET-WDM networks, all share the same overall structure: an *overlay* network H is implemented on top of an *underlying* network G. This implementation can be abstracted as a *routing scheme* that maps overlay connections to underlying routes. We comment that there are sometimes differences between such a mapping and the common notion of a routing scheme. Still, since the routing scheme often defines the mapping, we shall term this mapping the *routing scheme*.

Often, the underlying network itself is implemented on top of yet another network, thus introducing a multi-layer hierarchy. Typically, the lower level underlying network is "closer" to the physical wires, whereas the higher level network is a traffic network in which edges capture various kinds of connections, depending on the context. For the sake of simplicity, we focus on a pair of consecutive layers G and H. This is sufficient to capture a large class of practical scenarios.

The current paper deals with what happens to the different connectivity properties once we turn to the context of overlay networks. As discussed later on, connectivity has been studied previously in the "overlay network" world under the "survivability" interpretation (CP1), and it has been observed that, in this context, the connectivity parameter changes, i.e., the connectivity of the overlay network may be different from that of the underlying network. Lee et al. [12] demonstrated the significance of this difference by showing that the survivability property is computationally hard and even hard to approximate in the multi-layer case. Since the three aforementioned connectivity parameters may differ in multi-layer networks (see Sect. 1.2), they also showed a similar result for the disjoint paths connectivity property.

Interestingly, the motivation of Lee et al. for addressing the disjoint paths connectivity property was the issue of flow. One of the contributions of the current paper is to directly address this issue, showing that in contrast to the previous two notions of connectivity, the maximum flow supported by an overlay network can actually be computed in polynomial time.

In the specific context of survivability, there has been other papers that have shown that the issue of connectivity in an overlay network is different from that of connectivity in underlying networking. Consider, for example, a situation where several overlay edges (representing connections) of H pass over the same physical link. Then all these overlay edges may be disconnected as a result of a single hardware fault in that link, possibly disconnecting the overlay network. The affected overlay links are said to *share* the *risk* of the failure of the underlying physical link, hence they are referred to in the literature as a *shared risk link group* (SRLG). An SRLG-based model for overlay networks was extensively studied in recent years;[7] see, e.g., [14] for a useful introduction to this notion and [2] for a discussion of this concept in the context of MPLS. The SRLG model hides the actual structure of the underlying network, in the sense that many different underlying networks can yield the same sets of SRLG. (For certain purposes, this is an advantage of the model.) An even more general notion is that of *Shared Risk Resource Group* However, sometimes this model abstracts away too much information, making certain computational goals (such as, e.g., flow computations) harder to achieve.

In contrast to the SRLG model, we present the alternative model of *deep connectivity*, which allows us to simultaneously consider all three components: the overlay network, the underlying network, and the mapping (the routing scheme). Note that all three should be considered: For example, if the underlying network is not connected, then neither can the overlay network be. The routing scheme also affects the connectivity properties as different routing schemes may yield significantly different overlay link dependencies. In some cases, routing is constrained to shortest paths, whereas in other cases it can be very different. In *policy based* routing schemes (see, e.g. [16]), for example, some underlying edges are not allowed to be used for routing from u to v, which may cause the underlying path implementing the overlay link (u, v) to be much longer than the shortest (u, v)-path.

[7] Note that a common underlying link is not the only possible shared risk; overlay links sharing a node may form a shared risk link group too.

1.2 The Deep Connectivity Model

The underlying network is modeled by a (simple, undirected, connected) graph G whose vertex set $V(G)$ represents the network nodes, and whose edge set $E(G)$ represents the communication links between them. Some nodes of the underlying network are designated as *peers*; the set of peers is denoted by $\mathcal{P} \subseteq V(G)$. The overlay network, modeled by a graph H, spans the peers, i.e., $V(H) = \mathcal{P}$ and $E(H) \subseteq \mathcal{P} \times \mathcal{P}$; H typically represents a "virtual" network, constructed on top of the peers in the underlying communication network G.

An edge (u, v) in the overlay graph H may not directly correspond to an edge in the underlying graph G (that is, $E(H)$ is not necessarily a subset of $E(G)$). Therefore, communication over a (u, v) edge in H should often be routed along some multi-hop path connecting u and v in G. This is the role of a *routing scheme* $\rho : \mathcal{P} \times \mathcal{P} \to 2^{E(G)}$ that maps each pair (u, v) of peers to some simple path $\rho(u, v)$ connecting u and v in G. A message transmitted over the edge (u, v) in H is physically disseminated along the path $\rho(u, v)$ in G. We then say that (u, v) is *implemented* by $\rho(u, v)$. For the sake of simplicity, the routing scheme ρ is assumed to be symmetric, i.e., $\rho(u, v) = \rho(v, u)$. More involved cases do exist in reality: the routing scheme may be asymmetric, or may map some overlay edge into multiple routes; the simple model given here suffices to show interesting differences between the various connectivity measures.

When a message is routed in H from a peer $s \in \mathcal{P}$ to a non-neighboring peer $t \in \mathcal{P}$ along some multi-hop path $\pi = (x_0, x_1, \ldots, x_k)$ with $x_0 = s$, $x_k = t$, and $(x_i, x_{i+1}) \in E(H)$, it is physically routed in G along the concatenated path $\rho(x_0, x_1)\rho(x_1, x_2) \cdots \rho(x_{k-1}, x_k)$. In some cases, when the overlay graph H is known, it will be convenient to define the routing scheme over the edges of H, rather than over all peer pairs.

The notion of *deep connectivity* grasps the connectivity in the overlay graph H, while taking into account its implementation by the underlying paths in G. Specifically, given two peers $s, t \in \mathcal{P}$, we are interested in three different parameters, each capturing a specific type of connectivity. In order to define these parameters, we extend the domain of ρ from vertex pairs in $\mathcal{P} \times \mathcal{P}$ to collections of such pairs in the natural manner, defining $\rho(F) = \bigcup_{(u,v) \in F} \rho(u, v)$ for every $F \subseteq \mathcal{P} \times \mathcal{P}$. In particular, given an (s, t)-path π in H, $\rho(\pi) = \bigcup_{e \in \pi} \rho(e)$ is the set of underlying edges used in the implementation of the overlay edges along π.

- The *edge-removal deep connectivity* of s and t in H with respect to G and ρ, denoted by $\mathrm{ERDC}_{G,\rho}(s, t, H)$, is defined as the size of the smallest subset $F \subseteq E(G)$ that intersects with $\rho(\pi)$ for every (s, t)-path π in H; namely, the minimum number of underlying edges that should be removed in order to disconnect s from t in the overlay graph.
- The *path-disjoint deep connectivity* of s and t in H with respect to G and ρ, denoted by $\mathrm{PDDC}_{G,\rho}(s, t, H)$, is defined as the size of the largest collection C of (s, t)-paths in H such that $\rho(\pi) \cap \rho(\pi') = \emptyset$ for every $\pi, \pi' \in C$ with $\pi \neq \pi'$, i.e., the maximum number of overlay paths connecting s to t whose underlying implementations are totally independent of each other.

- The *flow deep connectivity* of s and t in H with respect to G and ρ, denoted by $\mathrm{FDC}_{G,\rho}(s,t,H)$, is defined as the maximum amount of flow[8] that can be pushed from s to t in G restricted to the images under ρ of the (s,t)-paths in H, assuming that each edge in $E(G)$ has a unit capacity. Intuitively, if s and t are well connected, then it should be possible to push a large amount of flow between them.

Example: To illustrate the various definitions, consider the underlying network G depicted in Fig. 1(a), and the overlay network H depicted in Fig. 1(b). The routing scheme ρ assigns each of the 6 overlay edges adjacent to the extreme S and T a simple route consisting of the edge itself. For the remaining three overlay edges, the assigned routes are as follows:

$$\rho(U_1,U_4) = (U_1,M_1,M_2,U_2,U_3,U_4)$$
$$\rho(M_1,M_4) = (M_1,M_2,D_2,D_3,M_3,M_4)$$
$$\rho(D_1,D_4) = (D_1,D_2,D_3,U_3,U_4,D_4)\ .$$

The route $\rho(M_1,M_4)$ is illustrated by the dashed line in Fig. 1(a). Note that in the original (underlying) network G, the connectivity of the extreme nodes S and T is 3 under all three definitions. In contrast, the values of the three connectivity parameters for the extreme nodes S and T in the overlay network H under the routing scheme ρ are as follows:

- The edge-removal deep connectivity of s and t in H w.r.t. G and ρ is $\mathrm{ERDC}_{G,\rho}(s,t,H) = 2$.
  Indeed, disconnecting the underlying edge (D_2,D_3) plus any edge of the upper underlying route will disconnect S from T.
- The path-disjoint deep connectivity of s and t in H w.r.t. G and ρ is $\mathrm{PDDC}_{G,\rho}(s,t,H) = 1$.
  Indeed, any two of the three overlay routes connecting S and T share an underlying edge.
- The *flow deep connectivity* of s and t in H w.r.t. G and ρ is $\mathrm{FDC}_{G,\rho}(s,t,H) = 3/2$.
  This is obtained by pushing 1/2 flow unit through each of the three overlay routes.

For each deep connectivity (s,t)-parameter $X_{G,\rho}(s,t,H)$, we define the corresponding *all-pairs* variant $X_{G,\rho}(H) = \min_{s,t\in\mathcal{P}} X_{G,\rho}(s,t,H)$. When G and ρ are clear from the context, we may remove them from the corresponding subscripts.

1.3 Contribution

Our model for overlay networks makes it convenient to explore the discrepancies between the different deep connectivity notions. Classical results from graph

[8] In the setting of undirected graphs, flow may be interpreted in two different ways depending on whether two flows along the same edge in opposite directions cancel each other or add up. Here, we assume the latter interpretation which seems to be more natural in the context of overlay networks.

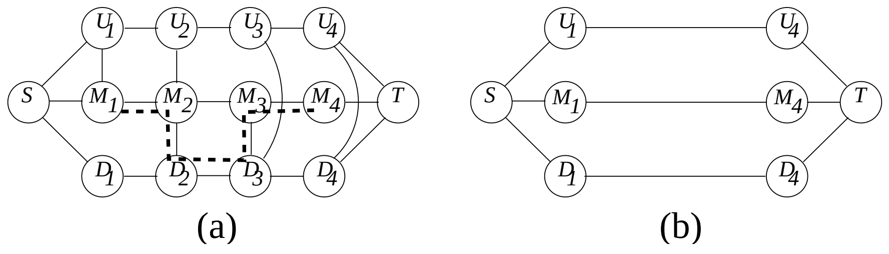

Fig. 1. (a) The underlying graph G. (b) The overlay network H.

theory, e.g., the fundamental min-cut max-flow theorem [8, 7] and Menger's theorem [13] state that the three connectivity parameters mentioned above are equivalent when a single layer network is considered. Polynomial time algorithms that compute these parameters (in a single layer network) were discovered early [8, 5, 6] and have since become a staple of algorithms textbooks [3, 11]. As mentioned above, previous results [12, 4], when translated to our model, have shown that in multi-layer networks, two of the three connectivity parameters are computationally hard and even hard to approximate. Our first technical contribution is to expand on these negative results by showing that the path-disjoint deep connectivity property cannot be approximated to any finite ratio when attention is restricted to simple paths in the underlying graph.

On the positive side, we show that the flow deep connectivity parameters can be computed in polynomial time (Sect. 3), thus addressing the motivation of [12] for studying the disjoint paths property in overlay networks. Then, we address the issue of constructing a "good" overlay graph for a given underlying graph and routing scheme. As opposed to the difficulty of approximating the value of the parameters, we show that the related construction problem can sometimes be well approximated. Specifically, in Sect. 4, we investigate the problem of constructing 2-edge removal deeply connected overlay graphs with as few as possible (overlay) edges. This problem is shown to be NP-hard, but we show that a logarithmic-approximation for it can be obtained in polynomial-time.

2 Hardness of Approximation

As mentioned earlier, Lee et al. [12] established hardness of approximation results for the problems of computing the parameters $\mathrm{ERDC}_{G,\rho}(s,t,H)$, $\mathrm{PDDC}_{G,\rho}(s,t,H)$, and $\mathrm{ERDC}_{G,\rho}(H)$. For completeness, we observe that the all-pairs variant $\mathrm{PDDC}_{G,\rho}(H)$ of the path-disjoint deep connectivity parameter is also hard to approximate, establishing the following theorem, which is essentially a corollary of Theorem 4 in [12] combined with a result of Håstad [10].

Theorem 1. *Unless NP = ZPP, the problem of computing the parameter* $\mathrm{PDDC}_{G,\rho}(H)$ *cannot be approximated to within a ratio of* $|E(H)|^{1/2-\epsilon}$ *for any* $\epsilon > 0$.

We now turn to show that the following natural variants of the PDDC parameters cannot be approximated to within any finite ratio. Let $\mathrm{SPDDC}_{G,\rho}(s,t,H)$ denote the size of the largest collection C of (s,t)-paths in H such that all paths $\pi \in C$ are implemented by simple paths $\rho(\pi)$ in G and $\rho(\pi) \cap \rho(\pi') = \emptyset$ for every $\pi, \pi' \in C$ with $\pi \neq \pi'$; let $\mathrm{SPDDC}_{G,\rho}(H) = \min_{s,t \in \mathcal{P}} \mathrm{SPDDC}_{G,\rho}(s,t,H)$. Note that these parameters are merely a restriction of the PDDC parameters to simple paths (hence the name, which stands for *simple* path-disjoint deep connectivity).

The inapproximability of the $\mathrm{SPDDC}_{G,\rho}(s,t,H)$ parameter is proved by reducing the set packing problem to the problem of distinguishing between the case $\mathrm{SPDDC}_{G,\rho}(s,t,H) = 0$ and the case $\mathrm{SPDDC}_{G,\rho}(s,t,H) \geq 1$. In fact, the vertices $s,t \in V(H)$ that minimize $\mathrm{SPDDC}_{G,\rho}(s,t,H)$ in this reduction are known in advance, thus establishing the impossibility of approximating the all-pairs parameter $\mathrm{SPDDC}_{G,\rho}(H)$ as well. The proofs of the following two theorems are deferred to the full version.

Theorem 2. *Unless P = NP, the problem of computing the parameter* $\mathrm{SPDDC}_{G,\rho}(s,t,H)$ *cannot be approximated to within any finite ratio.*

Theorem 3. *Unless P = NP, the problem of computing the parameter* $\mathrm{SPDDC}_{G,\rho}(H)$ *cannot be approximated to within any finite ratio.*

3 Efficient Algorithm for FDC

In this section we develop a polynomial time algorithm that computes the flow deep connectivity parameter $\mathrm{FDC}(s,t,H)$ (which clearly provides an efficient computation of the parameter $\mathrm{FDC}(H)$ as well). Consider some underlying graph G, routing scheme ρ, overlay graph H, and two vertices $s,t \in V(H)$. Let $\mathcal{P}$ denote the collection of all simple (s,t)-paths in H. For each path $p \in \mathcal{P}$ and for each edge $e \in E(G)$, let $\psi(p,e)$ be the number of appearances of the edge e along the image of p under ρ. We begin by representing the parameter $\mathrm{FDC}(s,t,H)$ as the outcome of the following linear program:

$$\begin{array}{ll} \max \sum_{p \in \mathcal{P}} x_p \quad \text{s.t.} & \\ \quad \sum_{p \in \mathcal{P}} \psi(p,e) \cdot x_p \leq 1 & \forall e \in E(G) \\ \quad x_p \geq 0 & \forall p \in \mathcal{P} \end{array}$$

The variable x_p represents the amount of flow pushed along the image under ρ of the path p for every $p \in \mathcal{P}$. The goal is to maximize the total flow pushed along the images of all paths in $\mathcal{P}$ subject to the constraints specifying that the sum of flows pushed through any edge is at most 1. This linear program may exhibit an exponential number of variables, so let us consider its dual program instead:

$$\begin{array}{ll} \min \sum_{e \in E(G)} y_e \quad \text{s.t.} & \\ \quad \sum_{e \in E(G)} \psi(p,e) \cdot y_e \geq 1 & \forall p \in \mathcal{P} \\ \quad y_e \geq 0 & \forall e \in E(G) \end{array}$$

The dual program can be interpreted as fractionally choosing as few as possible edges of G so that the image under ρ of every path p in $\mathcal{P}$ traverses in total at least one edge. We cannot solve the dual program directly as it may have an exponential number of constraints. Fortunately, it admits an efficient separation oracle, hence it can be solved in polynomial time (see, e.g., [9]).

Given some real vector $\boldsymbol{y}$ indexed by the edges in $E(G)$, our separation oracle either returns a constraint which is violated by $\boldsymbol{y}$ or reports that all the constraints are satisfied and $\boldsymbol{y}$ is a feasible solution. Recall that a violated constraint corresponds to some path $p \in \mathcal{P}$ such that $\sum_{e \in E(G)} \psi(p, e) \cdot y_e < 1$. Therefore our goal is to design an efficient algorithm that finds such a path $p \in \mathcal{P}$ if such a path exists.

Let $w(e) = \sum_{e' \in \rho(e)} y_{e'}$ for every edge $e \in E(H)$ and let H' be the weighted graph obtained by assigning weight $w(e)$ to each edge $e \in E(H)$. The key observation in this context is that the (weighted) length of an (s,t)-path p' in H' equals exactly to $\sum_{e \in E(G)} \psi(p, e) \cdot y_e$, where p is the path in H that corresponds to p' in H'. Therefore, our separation oracle is implemented simply by finding a shortest (s,t)-path p^* in H': if the length of p^* is smaller than 1, then p^* corresponds to a violated constraint; otherwise, the length of all (s,t)-paths in H' is at least 1, hence $\boldsymbol{y}$ is a feasible solution. This establishes the following theorem.

Theorem 4. *The parameters* $\mathrm{FDC}_{G,\rho}(s,t,H)$ *and* $\mathrm{FDC}_{G,\rho}(H)$ *can be computed in polynomial time.*

4 Sparsest 2-ERDC Overlay Graphs

In this section we are interested in the following problem, referred to as the *sparsest* 2-ERDC *overlay graph* problem: given an underlying graph G, a peer set $\mathcal{P} \subseteq V(G)$, and a routing scheme $\rho : \mathcal{P} \times \mathcal{P} \to 2^{E(G)}$, construct the sparsest overlay graph H for $\mathcal{P}$ (in terms of number of overlay edges) satisfying $\mathrm{ERDC}_{G,\rho}(H) \geq 2$. Of course, one has to make sure that such an overlay graph H exists, so in the context of the sparsest 2-ERDC overlay graph problem we always assume that $\mathrm{ERDC}_{G,\rho}(K_{\mathcal{P}}) \geq 2$, where $K_{\mathcal{P}}$ is the complete graph on $\mathcal{P}$. This means that a trivial solution with $\binom{n}{2}$ edges, where $n = |\mathcal{P}|$, always exists and the challenge is to construct a sparser one.

4.1 Hardness

We begin our treatment of this problem with a hardness result.

Theorem 5. *The sparsest* 2-ERDC *overlay graph problem is NP-hard.*

Proof. The assertion is proved by a reduction from the *Hamiltonian path* problem. Consider an n-vertex graph G_0 input to the Hamiltonian path problem. Transform it into an instance of the sparsest 2-ERDC overlay graph problem as follows: Construct the underlying graph G by setting $V(G) = V(G_0) \cup \{x, y\}$

and $E(G) = E(G_0) \cup \{(x,y)\} \cup \{(v,x),(v,y) \mid v \in V(G_0)\}$ and let $\mathcal{P} = V(G_0)$. Define the routing scheme ρ by setting

$$\rho(u,v) = \begin{cases} (u,v) & \text{if } (u,v) \in E(G_0) \\ (u,x,y,v) & \text{otherwise.} \end{cases}$$

This transformation is clearly polynomial in n.

We argue that G_0 admits a Hamiltonian path if and only if there exists an overlay graph H for $\mathcal{P}$ so that $|E(H)| = n$ and $\text{ERDC}_{G,\rho}(H) \geq 2$. To that end, suppose that G_0 admits a Hamiltonian path π. If π can be closed to a Hamiltonian cycle (in G_0), then take H to be this Hamiltonian cycle. Otherwise, take H to be the cycle consisting of π and a virtual edge connecting between π's endpoints. In either case, H clearly has n edges and by the design of ρ, H satisfies $\text{ERDC}_{G,\rho}(H) = 2$.

Conversely, if there exists an overlay graph H for $\mathcal{P}$ so that $|E(H)| = n$ and $\text{ERDC}_{G,\rho}(H) \geq 2$, then H must form a Hamiltonian cycle C in $\mathcal{P} \times \mathcal{P}$. This cycle can contain at most one virtual edge as otherwise, the removal of (x,y) breaks two edges of C which means that $\text{ERDC}_{G,\rho}(H) < 2$. By removing this virtual edge, we are left with a Hamiltonian path in G_0. □

4.2 Constructing Sparse 2-ERDC Overlay Graphs

On the positive side, we develop a polynomial time logarithmic approximation algorithm for the sparsest 2-ERDC overlay graph problem. Our algorithm proceeds in two stages: First, we take T to be an arbitrary spanning tree of $\mathcal{P} \times \mathcal{P}$. Subsequently, we aim towards (approximately) solving the following optimization problem, subsequently referred to as the *overlay augmentation* problem: augment T with a minimum number of $\mathcal{P} \times \mathcal{P}$ edges so that the resulting overlay graph H satisfies $\text{ERDC}_{G,\rho}(H) \geq 2$.

We will soon explain how we cope with this optimization problem, but first let us make the following observation. Denote the edges in $\rho(T)$ by $\rho(T) = \{e_1, \ldots, e_\ell\}$ and consider some overlay graph H such that $E(H) \supseteq T$ and some $1 \leq i \leq \ell$. Let $F_i(H)$ be the collection of connected components of the graph obtained from H by removing all edges $e \in E(H)$ such that $e_i \in \rho(e)$. Fix

$$\kappa_i(H) = |F_i(H)| - 1 \quad \text{and} \quad \kappa(H) = \sum_{i=1}^{\ell} \kappa_i(H)\,.$$

We think of $\kappa(H)$ as a measure of the distance of the overlay graph H from being a feasible solution to the overlay augmentation problem (i.e., $\text{ERDC}(H) \geq 2$).

Proposition 1. *An overlay graph $H \supseteq T$ satisfies* $\text{ERDC}(H) \geq 2$ *if and only if* $\kappa(H) = 0$.

Proof. If $\text{ERDC}(H) \geq 2$, then $F_i(H)$ must consist of a single connected component for every $1 \leq i \leq \ell$, thus $\kappa(H) = 0$. Conversely, if $\kappa(H) = 0$, then necessarily

$|F_i(H)| = 1$ for every $1 \leq i \leq \ell$, which means that H does not disconnect by the removal of any edge $e_i \in \rho(T)$. It is also clear that the removal of any edge in $E(G) - \rho(T)$ does not disconnect H as the tree T remains intact. Therefore, $\mathrm{ERDC}(H) \geq 2$. $\square$

Consider some edge $e \in (\mathcal{P} \times \mathcal{P}) - E(H)$ and let $H \cup \{e\}$ denote the overlay graph obtained from H by adding the edge e. By the definition of the parameter κ, we know that $\varDelta_i(e, H) = \kappa_i(H) - \kappa_i(H \cup \{e\})$ is either 0 or 1 for any $1 \leq i \leq \ell$. Fixing $\varDelta(e, H) = \kappa(H) - \kappa(H \cup \{e\})$, we observe that: $\varDelta(e, H) = \sum_{i=1}^{\ell} \varDelta_i(e, H)$ referred to as Property ($\star$).

We are now ready to complete the description of our approximation algorithm. Starting from $H = T$, the algorithm gradually adds edges to H as long as $\kappa(H) > 0$ according to the following greedy rule. At any intermediate step, we add to H an edge $e \in (\mathcal{P} \times \mathcal{P}) - E(H)$ that yields the maximum $\varDelta(e, H)$. When $\kappa(H)$ decreases to zero, the algorithm terminates (recall that this means that $\mathrm{ERDC}(H) \geq 2$).

The analysis of our approximation algorithm relies on the following proposition.

Proposition 2. *The parameter κ can be computed in polynomial time. Moreover, for every two overlay graphs H_1, H_2 such that $E(H_1) \subseteq E(H_2)$, we have*

(1) *$\kappa(H_1) \geq \kappa(H_2)$; and*
(2) *$\varDelta(e, H_1) \geq \varDelta(e, H_2)$ for every edge $e \in \mathcal{P} \times \mathcal{P}$.*

Proof. The fact that κ can be computed efficiently and the fact that $\kappa(H_1) \geq \kappa(H_2)$ are clear from the definition of κ, so our goal is to prove that $\varDelta(e, H_1) \geq \varDelta(e, H_2)$ for every $e \in \mathcal{P} \times \mathcal{P}$. By Property ($\star$), it suffices to show that $\varDelta_i(e, H_1) \geq \varDelta_i(e, H_2)$ for every $1 \leq i \leq \ell$. If $\varDelta_i(e, H_2) = 0$, then this holds vacuously, so suppose that $\varDelta_i(e, H_2) = 1$. This means that $e_i \notin \rho(e)$ and the endpoints of e belong to different connected components in $F_i(H_2)$. But since $E(H_1) \subseteq E(H_2)$, it follows that the endpoints of e must also belong to different connected components in $F_i(H_1)$, hence $\varDelta_i(e, H_1) = 1$ as well. $\square$

Proposition 2 implies that the overlay augmentation problem falls into the class of *submodular cover* problems (cf. [17, 1]) and our greedy approach is guaranteed to have an approximation ratio of at most $\ln(\kappa(T)) + 1 = O(\log N)$, where $N = |V(G)|$. More formally, letting $\hat{H}$ be a sparsest overlay graph such that $\hat{H} \supseteq T$ and $\mathrm{ERDC}(\hat{H}) \geq 2$, it is guaranteed that the overlay graph H generated by our greedy approach satisfies $|E(H) - T| \leq O(\log N) \cdot |E(\hat{H}) - T|$.

To conclude the analysis, let H^* be an optimal solution to the sparsest 2-ERDC overlay graph problem. Clearly, $|E(H^*)| > n - 1 = |T|$. Moreover, since $E(H^*) \cup T$ is a candidate for $\hat{H}$, it follows that $|E(H^*) \cup T| \geq |\hat{H}|$, thus $|E(H^*)| \geq |E(\hat{H}) - T|$. Therefore,

$$\begin{aligned} |E(H)| &\leq O(\log N) \cdot |E(\hat{H}) - T| + |T| \\ &\leq O(\log N) \cdot |E(H^*)| + |E(H^*)| \\ &= O(\log(N) \cdot |E(H^*)|) \end{aligned}$$

which establishes the following theorem.

Theorem 6. *The sparsest* 2-ERDC *overlay graph problem admits a polynomial-time logarithmic-approximation.*

References

[1] J. Bar-Ilan, G. Kortsarz, and D. Peleg. Generalized submodular cover problems and applications. *Theor. Comput. Sci.*, 250(1-2):179–200, Jan. 2001.

[2] Cisco IOS Software Releases 12.0 S. Mpls traffic engineering: Shared risk link groups (srlg). www.cisco.com/en/US/docs/ios/12_0s/feature/guide/fs29srlg.html.

[3] T. H. Cormen, C. E. Leiserson, R. L. Rivest, and C. Stein. *Introduction to Algorithms.* The MIT Press, 2 edition, 2001.

[4] D. Coudert, P. Datta, S. Perennes, H. Rivano, and M.-E. Voge. Shared risk resource group: Complexity and approximability issues. *Parallel Processing Letters*, 17(2):169–184, June 2007.

[5] E. A. Dinic. Algorithm for Solution of a Problem of Maximum Flow in a Network with Power Estimation. *Soviet Math Doklady*, 11:1277–1280, 1970.

[6] J. Edmonds and R. M. Karp. Theoretical improvements in algorithmic efficiency for network flow problems. *J. ACM*, 19(2):248–264, Apr. 1972.

[7] P. Elias, A. Feinstein, and C. Shannon. A note on the maximum flow through a network. *Information Theory, IEEE Transactions on*, 2(4):117–119, Dec 1956.

[8] L. R. Ford and D. R. Fulkerson. Maximal flow through a network. *Canadian Journal of Mathematics*, 8:399–404, 1956.

[9] M. Grotschel, L. Lovasz, and A. Schrijver. Geometric algorithms and combinatorial optimization. *Combinatorica*, 1981.

[10] J. Hastad. Clique is hard to approximate within $n^{1-\epsilon}$. *Foundations of Computer Science, IEEE Annual Symposium on*, 0:627, 1996.

[11] J. Kleinberg and E. Tardos. *Algorithm Design.* Addison Wesley, 2006.

[12] K. Lee, E. Modiano, and H.-W. Lee. Cross-layer survivability in wdm-based networks. *Networking, IEEE/ACM Transactions on*, 19(4):1000 –1013, aug. 2011.

[13] K. Menger. Zur allgemeinen kurventheorie. *Fund. Math*, 10(95-115):5, 1927.

[14] Network Protection Website. Network protection techniques, network failure recovery, network failure events. www.network-protection.net/shared-risk-link-group-srlg/.

[15] S. Ramamurthy and B. Mukherjee. Survivable wdm mesh networks. part i-protection. In *INFOCOM '99. Eighteenth Annual Joint Conference of the IEEE Computer and Communications Societies. Proceedings. IEEE*, volume 2, pages 744 –751 vol.2, mar 1999.

[16] White Paper, Cisco Systems Inc., 1996. Policy-based routing. www.cisco.com/warp/public/cc/pd/iosw/tech/plicy_wp.pdf.

[17] L. Wolsey. An analysis of the greedy algorithm for the submodular set covering problem. *Combinatorica*, 2:385–393, 1982. 10.1007/BF02579435.