Towards Optimally Resilient Topologies against Optimal Attacks

Martin Backhaus
Telematics/Computer Networks Research Group
Technische Universität Ilmenau, Germany
martin.backhaus@tu-ilmenau.de

Guenter Schaefer
Telematics/Computer Networks Research Group
Technische Universität Ilmenau, Germany
guenter.schaefer@tu-ilmenau.de

***Abstract*—By overlay topology optimization with regard to the minimum number of overlay edges that need to breakdown until all communication between two chosen nodes ceases, the robustness against optimal attacks can be maximized and the network can still perform its operation despite (multiple) link-failures. This article presents an approach to construct optimally resilient topologies on the basis of a metric used by an optimal attacker: the minimum overlay cut. We developed a bilevel Integer Linear Program (ILP) formulation for exactly calculating optimal overlay topologies on small problem instances consisting of two-layer networks. Since bilevel optimization is very hard to perform in practice on bigger networks, we propose an approximation algorithm as well, whose quality can be assessed on smaller instances. An evaluation of a typical VPN networking scenario shows that compared to the obvious approach of realizing maximal robustness towards attackers by a fully meshed overlay, our approach reaches the same desired connectivity with far fewer overlay edges.**

***Index Terms*—overlay networks, minimum overlay cut, resilient VPN topology, bilevel ILP, approximation**

I. Introduction

Availability of network functionalities and robustness of network connections have always been an issue in research and practice. Emerging technologies like Internet of things [1] and 5G [2] do not only strengthen the importance of availability and robustness, but increase the relevance of these topics in wireless communications with low mobility as well. As those networks gain importance, their vulnerability increases as well – depending on their purposes. In case of critical infrastructures where information security is accomplished for a selected number of nodes using VPN overlay networks [3], an ongoing function of the network is a worthwhile objective.

Since a reduced number of nodes participate in establishing and maintaining connections to one another, we are dealing with a two-layered network (or overlay networks). One advantage of such networks can be compensation of bad underlay properties, i.e., insufficient recovery in case of link outages due to slow underlay rerouting. If an overlay is maintained for other reasons already, it can help to compensate for link outages.

There are metrics to assess overlay network properties on the scale of two communicating nodes: Maximum Flow [4], Minimum Cut [5], [6] and Maximum Disjoint Paths [6]. Maximum Flow focuses on capacity that can pass through the network between two given nodes and omits robustness towards outages. The metric for Maximum Disjoint Paths allows to calculate the maximum number of paths (and the paths themselves) from source to target node that are edge disjoint in the underlying transport network and has been studied in our previous work [7]. At a first glance, the Minimum Cut works from an attackers perspective only, because a minimal (and therefore optimal) quantity of underlay edges (and the edges themselves) that need to fail in order to prevent communication between two nodes completely are being computed for that particular metric (referred to as "optimal attack").

As the MaxFlow-MinCut-Theorem loses its validity [6] in two-layered networks, Maximum Flow can no longer be used to derive Minimum Cut or Maximum Disjoint Paths – which means that (in cases of overlay networks) their computation gets more complex and their values are indifferent. Comparing the two latter metrics, state-of-the-art research declares "Minimum Cut $\geq$ Maximum Disjoint Paths" with an arbitrarily large gap [6]. Hence, the minimum cut is more desirable concerning robustness, but it does not yield any constructive output as the metric for maximum disjoint paths does. In this paper, we study the problem to obtain robust topologies that realize the minimum cut with the least possible amount of overlay edges. Reducing the number of involved overlay edges is desirable due to effort for maintenance, connection setup and a less complex overlay routing. The latter is of particular importance in order to use the reduced amount of edges – the overlay is required, to react to link outages autonomously.

The task of optimal topology design with respect to optimality of a further metric requires bilevel optimization [8], which is computationally hard [9], since that depicts one optimization problem nested into another one. Therefore, we designed an approximation algorithm in addition to the exact bilevel formulation of the problem of finding an overlay topology with the least amount of edges, that realizes the maximal possible minimum cut. The approximation is – by design – still optimal with respect to the biggest value for the minimum cut, but it tends to yield more edges than necessary.

In particular, we provide the following contributions:

1) An exact bilevel Integer Linear Program (ILP) formulation for optimally resilient topologies against an optimal attack strategy based on the minimum overlay cut.

This work was supported by the Deutsche Forschungsgemeinschaft (DFG) under grant number SCHA 309/3-1.

2) A corresponding approximation algorithm.
3) Evaluation of approximation quality for small networks.
4) Evaluation of needed quantity of overlay edges using big VPN networks in a wireless scenario.

The remainder of this paper is organized as follows: First, we give an overview of the state-of-the-art and motivate requirements for our resulting approach (Sec. II). Then Sec. III will explain our bilevel ILP formulation for an overlay consisting of a minimal quantity of edges realizing the maximum possible minimum cut. Our corresponding approximation will be presented in Sec. IV. The evaluation in Sec. V studies approximation quality and the required number of edges for our approach. To sum up, Sec. VI presents some concluding thoughts and directions for future research.

II. Requirements and Related Work

In order to enable an overlay network to quickly react to failures or attacks of a link in the underlying transport network, an approach should be able to: 1) reduce the number of needed overlay edges, while 2) maintaining maximal robustness, and 3) yield reasonable computational effort. Therefore, an ideal approach should generate an overlay topology consisting of a preferably small number of overlay edges that still realize a maximum minimal cut. Thus, lesser edges facilitate overlay routing and optimal robustness is being maintained at the same time. Since an optimal solution involves bilevel linear programming, an approximation is needed in order to cope with bigger problem instances. This approximation should maintain optimal robustness, but may use more overlay edges than the optimal solution.

Our previous work [7] already tackled overlay topology design. We focused on the maximal number of disjoint paths, which is less desirable with respect to robustness. Other state-of-the-art approaches do not explicitly address overlay topology design. One notable example is [6], where robust lightpath embedding in WDM networks is studied. They use metrics derived from the minimum overlay cut, but focus on designing an embedding from scratch – thus, optimization of overlay topologies is not considered. In [10] resilience in parallel link networks is studied using a game theoretical framework. However those networks are different as parallel links are completely independent, which is not the case for overlay networks. Another game theoretic approach can be found for interdependent networks [11]. They depict a special kind of networking scenario, where two networks work on their own, but interconnect with each other for added resilience. Since overlay networks need to rely on somewhat fixed underlays and yield flexibility within the overlay topology, only, the scenarios are completely different.

To the best of our knowledge, this is the first paper to use the attackers perspective metric (minimum overlay cut) to construct optimal overlay topologies with the option of overlay rerouting to increase robustness towards link outages.

III. Exact Bilevel ILP Formulation

A detailed description of exact solutions to the problem of finding optimal topologies can be found in the following section. After giving a few basic presumptions, we present a novel bilevel ILP formulation for one communication demand that yields optimal solutions for small instances. As mentioned earlier, a bilevel problem consists of two nested problems: an upper level and a lower level problem. The upper problem is constrained to optimality of the lower level problem.

A. Presumptions

Before giving a bilevel ILP formulation, we need to mention some presumptions and constraints. They are similar to the ones already covered in our previous work [7].

1) Overlay control: In order to construct optimal topologies, we need to control the overlay completely – including connection setup and routing. This represents the flexibility of overlay networks that we will use in order to gain resilience.

2) Underlay knowledge: It is well known that underlay knowledge is hard or even impossible to get if communication takes place inside the public Internet. However, in some cases – e.g. provider networks – underlay knowledge can be obtained. Without underlay knowledge, overlay robustness is achievable with high effort (e.g. only through a full mesh overlay). Therefore, we assume knowledge of the underlay topology. This assumption is not unrealistic, e.g., in commercial VPN services the transport network provider can also offer installation and operation of VPNs as a commercial service.

B. Notation and Bilevel ILP Formulation

To clarify the ILP formulation, we give a couple of notations that have already been used in our previous work [12], [7]. A formal description of a generic overlay consists of four parts: a graph $\underline{G} = (\underline{V}, \underline{E})$ as underlay with nodes and edges, similarly $\overline{G} = (\overline{V}, \overline{E})$ for the overlay, a function $\pi : \overline{V} \mapsto \underline{V}$ mapping overlay to underlay nodes and another function $\phi : \overline{E} \mapsto \underline{P}$ mapping overlay edges to an element of $\underline{P}$ (the set of all possible underlay paths) represented by a tuple of all nodes visited when traversing the path (including source and target node). An important distinction is, that the 4-tuple $G = (\underline{G}, \overline{G}, \pi, \phi)$ is called *overlay network*, whereas *overlay graph* refers to $\overline{G}$, only. If readability is improved, variables (no matter if representing nodes or edges) will be underlined if they belong to the underlay graph (or overlined respectively). Note furthermore, that both $\overline{e}$ and (i, j) identify overlay edges $\in \overline{E}$ (analogously, underlined for underlay edges), so either of those can be indexes (if $\overline{e} = (i, j)$).

For most scenarios, $\overline{V} \subseteq \underline{V}$ holds and π is the identity function (assuming suitable node naming) and can thus be omitted in case of communication networks, because communicating instances are most likely to be relevant if they reside on different nodes.

Furthermore, the function $\mathrm{dep}(\cdot)$ maps underlay edges $\underline{e}$ to the set of all overlay edges $\overline{e}$ that use $\underline{e}$. It can be derived easily with the help of π and ϕ. Note, that edge outages are usually symmetric. This is taken into account for the computation of

$\mathrm{dep}(\cdot)$. It can further be used with a different domain, i.e., overlay edges $\overline{e}$, in which case $\mathrm{dep}(\cdot)$ maps to all underlay edges used by an overlay edge $\overline{e}$.

The bilevel ILP's input is the overlay network itself, a source node s and a target node t. The goal is to find an overlay topology having minimal quantity of edges that realizes the maximal minimum cut. One possible formulation is as follows:

$$\max \sum_{e \in \underline{E}} C \cdot \underline{y}_e - \sum_{e \in \overline{E}} \overline{I}_e \tag{1}$$

$$\text{s. t. } \overline{f}_e \leq \overline{I}_e \qquad , \forall e \in \overline{E} \tag{2}$$

$$\overline{I}_e \leq \overline{f}_e + 1 - \varepsilon \qquad , \forall e \in \overline{E} \tag{3}$$

$$\sum_{(v,j) \in \overline{E}} \overline{f}_{v,j} = \sum_{(i,v) \in \overline{E}} \overline{f}_{i,v}, \forall v \in \overline{V} \setminus \{s,t\} \tag{4}$$

$$\overline{I}_e \in \{0,1\} \qquad , \forall e \in \overline{E} \tag{5}$$

$$\overline{f}_e \in \mathbb{R}^+ \qquad , \forall e \in \overline{E} \tag{6}$$

$$\overline{x}_v \in \{0,1\} \quad , \forall v \in \overline{V} \tag{7}$$

$$\underline{y}_e \in \{0,1\} \quad , \forall e \in \underline{E} \tag{8}$$

$$(\overline{x}, \underline{y}) \in \Phi(\overline{I}) \tag{9}$$

Essentially, (5) creates binary variables $\overline{I}_e$ that indicate, whether or not an overlay edge e is being used for the resulting overlay ($\overline{I}_e = 1$) or not ($\overline{I}_e = 0$). The objective function (1) is twofold. On one hand, there should be a maximal number of edges that need to breakdown within the resulting topology in order to prevent communication from s to t ($\sum_{e \in \underline{E}} C \cdot \underline{y}_e$) and on the other hand, we should use the minimum amount of overlay edges to achieve that ($-\sum_{e \in \overline{E}} \overline{I}_e$). Hereby, C is a large constant (e.g. $C = |\overline{V}|^2$) to clarify, that the goal of being maximally robust is in favor compared to using the minimal amount of edges. Constraints (2), (3) and (4) are dispensable in a sense, as they do not affect the optimal solution. However, they are useful to shape the resulting overlay in a suitable manner and contribute to faster solving of the bilevel ILP. By introduction of continuous flow (6), we can demand network flow to conserve itself – equivalent to Kirchhoff's current law. (2) demands flow to reside on chosen edges, only. Usage of ε within (3) turns $\leq$ into $<$ and the constraint basically states $\overline{I}_e - 1 < \overline{f}_e$, which means that if an edge is chosen ($\overline{I}_e = 1$), there must be some significant amount of flow. We did not detect any numerical problems when using $\varepsilon = 0.01$. Finally, we introduce variables for the lower level problem (7), (8) and the connection to the lower level problem Φ (9).

$$\Phi(\overline{I}) = \underset{\overline{x}, \underline{y}}{\operatorname{argmin}} \left\{ \sum_{e \in \underline{E}} \underline{y}_e \;\middle|\; \right. \tag{10}$$

$$\overline{x}_s = 1, \tag{11}$$

$$\overline{x}_t = 0, \tag{12}$$

$$\left. 1 - \overline{I}_{i,j} + \sum_{e \in \mathrm{dep}((i,j))} \underline{y}_e \geq \overline{x}_i - \overline{x}_j \;\; \forall (i,j) \in \overline{E} \right\} \tag{13}$$

The lower level problem itself is similar to the ILP for the minimum cut for overlay networks given in [5], including a few minor simplifications. One mayor difference is, that not every edge is relevant for computing the minimum cut – only those chosen by the upper level problem should be taken into account. This is why (according to state-of-the-art bilevel programming papers), we denote $\overline{I} = \left\{ \overline{I}_e \;\middle|\; e \in \overline{E} \right\}$ and pass it as argument to the lower level problem. Therefore, variables I_e from the upper level problem are just constants within the lower level problem.

To explain the minimum cut ILP from scratch, variables $\overline{x}_v$ indicate, if an overlay node v belongs to S ($\overline{x}_v = 1$) or T ($\overline{x}_v = 0$). Remember, that $S \cap T = \emptyset$ and $S \cup T = \overline{V}$ hold for every cut. With (11) and (12) we assign s to S and t to T. Binary variables $\underline{y}_e$ indicate, if an underlay edge needs to break down to create a cut ($\underline{y}_e = 1$) or not ($\underline{y}_e = 0$). Examining constraint (13), without the term $1 - \overline{I}_{i,j}$ we see that the overlay edge from i to j needs to break down if $i \in S$ and $j \in T$. Since overlay edges do not break down by themselves, the sum $\sum_{e \in \mathrm{dep}((i,j))} \underline{y}_e$ ensures, that any of the underlay edges leading to the desired breakdown (and thus at least one variable $\underline{y}_e$) will be chosen. Through extension of the left hand side by $1 - \overline{I}_{i,j}$, a breakdown is only necessary if the overlay edge (i,j) is chosen. If not, the constraint is trivially fulfilled. Finally, the quantity of underlay breakdowns gets minimized in (10). With $\overline{x} = \{\overline{x}_e \mid e \in \overline{E}\}$ and $\underline{y} = \left\{ \underline{y}_e \;\middle|\; e \in \overline{E} \right\}$ being lower level variables (as indicated by the subscript of "argmin"), $\Phi(\overline{I})$ will yield the minimum amount of underlay edges that need to breakdown for inhibiting communication from s to t, when $\overline{I}$ is the set of chosen overlay edges.

C. Practical Considerations and Multiple Demands

As stated earlier, bilevel optimization yields enormous runtimes due to NP-hardness. In our case, the upper level goal of maximize the minimum cut and meanwhile minimizing the number of required edges are in direct contrast. This is why we only managed to obtain optimal results for small network instances of at most 10 overlay nodes. For solving bilevel optimization problems we employed the method shown in [13]. Essentially, they present a branch and bound algorithm that iteratively adds constraints to the problem and solves the high point problem (omitting the inner objective function), as well as the inner optimization problem (with outer variables set to constants) repeatedly. As the authors present a mathematical explanation and no out-of-the-box implementation, we experienced huge performance improvements by carefully translating mathematical notations to actual algorithms (including representation of matrices and check for trivial constraints). Runtimes for a 10 node instance decreased from several hours to less than 10 minutes (fixed s and t). Solving of any modified ILP's throughout the process is done with `Gurobi` [14].

Some modifications can be made to enhance performance of the illustrated problem even further. As already mentioned, constraints (2), (3) and (4) depict such modifications. One additional enhancement can be made, by forcing the minimum cut to be a certain value. Let min_c be the precomputed minimal number of edges that need to break down (computed on the overlay and underlay network by simply employing the method from [5]), then we can introduce the following constraint to the upper level problem: $\sum_{e \in \underline{E}} \underline{y}_e = min_c$. As the minimum cut can not be bigger and should not be smaller, we restrict the space of feasible solutions of the upper level problem significantly. It simplifies the objective function also. Another modification is to restrict failing underlay edges to chosen overlay edges with the following constraint: $\underline{y}_e \leq \sum_{e' \in \mathrm{dep}(e)} \overline{I}_{e'}$. If added to the upper level problem, an underlay edge e does not need to break down, if there is no overlay edge e' chosen that uses e.

IV. Approximation

In the following section, we present an algorithm to approximate the optimal overlay topology that can be calculated using our bilevel ILP formulation.

A. Presumptions

Our approximation basically needs the same presumptions as the bilevel formulation and can be obtained from Sec. III. Additionally, we can distinguish what kind of overlay network is at hand and to what extend it can be controlled. For example, in an Internet-like scenario, every connection can be controlled freely, i.e., the potential overlay graph is fully meshed. Another scenario, such as WDM, does not provide the same level of flexibility, so in this case: present overlay edges can either be used or dismissed. That varying amount freedom needs to be reflected in an algorithm as well. For the exact calculation via the bilevel approach, this consideration was not necessary, as we computed it for smaller instances of WDM networks, only.

B. Algorithm for Approximating Optimal Overlays

Algorithm 1 Approximation of an optimal topology

```
Input: underlay $\underline{G}$, overlay $\overline{G} = (\overline{V}, \overline{E})$, source $s$, target $t$
Output: overlay topology $\overline{G}_F$
    initialization:
 1: $\overline{G}_F = (\emptyset, \emptyset)$
 2: optional: $\overline{G}$ = fullmesh graph with nodes from $\overline{V}$
 3: $\overline{G'} = \overline{G}$
 4: x = optimalValue of $\mathrm{MinOverlayCut}(G = (\underline{G}, \overline{G}, \pi, \phi))$
 5: x' = 0
    loop:
 6: while $x' < x$ do
 7:     $\overline{G}_D$ = $\mathrm{AdaptWeights}(\overline{G'}, \overline{G}_F)$
 8:     $\overline{P}$ = $\mathrm{Dijkstra}(\overline{G}_D, s, t)$
 9:     $\overline{G}_F = \overline{G}_F \cup \overline{P}$
10:     $x'$ = optimal value of $\mathrm{MinOverlayCut}(\underline{G}, \overline{G}_F)$
11:     $\underline{E'}$ = optimal edge set of $\mathrm{MinOverlayCut}(\underline{G}, \overline{G}_F)$
12:     $\overline{G'} = \overline{G} \backslash \underline{E'}$
13: end while
14: return $\overline{G}_F$
```

A pseudocode representation of our approximation can be seen in algorithm 1. The basic idea is to iteratively design an overlay topology ($\overline{G}_F$) for transporting potential traffic from s to t, that grows until the maximal possible minimum overlay cut is realized. Initially, the topology is empty. Depending on the scenario, the minimum overlay cut has to be computed differently. As mentioned earlier, if the scenario yields complete freedom of overlay edge setup, we need to set $\overline{G}$ to a fully meshed graph consisting of nodes from $\overline{V}$. That way, the computed minimum cut can not get better and the desired freedom is being reflected. In case of an different scenario, $\overline{G}$ has to be the overlay graph without modifications. But still, it might be possible that some edges in $\overline{G}$ turn out to be unnecessary. Within the algorithm, $\overline{G'}$ reflects remaining possibilities of communication between s to t. Initially, $\overline{G'} = \overline{G}$ since all options are still available. With x as the best achievable value for the minimum cut in mind, x' will be the minimum cut that is currently achieved by $\overline{G}_F$ and thus initially set to 0.

Iterations in algorithm 1 continue as long as $x' < x$, i.e., $\overline{G}_F$ does not realize the maximal minimum cut. The main components of one iteration are: 1) A graph search from s to t to extend $\overline{G}_F$ and 2) The validation of the current minimum cut that $\overline{G}_F$ realizes so far. Before a graph search is being performed, we need to determine edge weights to adjust how favorable certain edges are. Therefore, $\mathrm{AdaptWeights}(\overline{G'}, \overline{G}_F)$ assigns edges from $\overline{G'}$ a weight, so that edges already in use by $\overline{G}_F$ are cheap, and other edges (that would be newly acquired) are expensive (we omitted a separate pseudocode formulation as this procedure is simple). This way, growth of $\overline{G}_F$ is being reduced to a necessary minimum. The graph search itself works as one would expect: $\mathrm{Dijkstra}(\overline{G}_D, s, t)$ invokes Dijkstra's algorithm on $\overline{G}$ and returns the shortest path from s to t. With a new overlay path $\overline{P}$ obtained and joined to $\overline{G}_F$ (in a simple way of joining node and edge sets), the current minimum cut (x') is checked by computing it using $\overline{G}_F$ in conjunction with the given underlay graph. If $x' = x$ we can stop and return right away after line 10 (omitted in the pseudocode description for brevity). Otherwise, we need to update all remaining possibilities to communicate from s to t (which are maintained in $\overline{G'}$) by simply deleting the set of cut edges from the latest check for minimum cut compliance (the ILP for minimum cut yields the quantity of edges and the edges themselves). Please note, that removing underlay edges from an overlay network requires removing any overlay edge using at least one of the removed underlay edges.

The approximation quality can further be enhanced by checking, if every overlay edge is indispensable. Therefore, the edge in question is being removed and the minimum cut is computed again. If the minimum cut gets smaller, the edge was indispensable. This is being done with every edge identified as part of an approximatively optimal topology by algorithm 1. Since the order of checking edges might influence the final result, the order is chosen at random. Repeating the process and trying different orders to obtain better results might be possible, but would contribute to longer runtimes.

C. Practical Considerations

By design, the approximation always creates overlay topologies realizing the maximal possible minimum cut and is optimal to that extent. As a consequence, approximation quality has to be considered regarding the number of edges used, only.

Although the algorithm features repeated computations of the minimum cut, which is being done with an ILP, it yields good runtimes on far bigger instances than our bilevel formulation was able to cope with. Especially the modification mentioned last of checking every overlay edge introduces a lot more computations, but since it is being done on topologies already narrowed by algorithm 1, we did not observe any impacts on runtime that would prevent practical applicability.

If a topology for multiple demands is desired, we simply obtain one topology for each demand and join the results into a single topology. That way, each demand will be fulfilled and edges will be reused. Having computed one topology, an appropriate overlay routing protocol needs to take care of path choices – which we will not focus on in this work. Tackling all demands at once makes the problem a lot more complex – especially when computing optimal solutions for comparison. However, in our future work we plan to further investigate topology generation for multiple demands.

V. Evaluation

We evaluated our approach of optimally resilient VPN topologies concerning topology size and approximation quality. This section shows the scenario and metrics being used for evaluation and presents results. The scenario specifies the underlay topology being used as well as their sizes. Furthermore, the scenario describes the embedding of overlay nodes and edges into the underlay as well. Any given parameter configuration is evaluated in 32 runs (each with a separately generated underlay and overlay with suitable parameters) for statistically significant results.

A. Scenarios

The following scenarios are the basis for our evaluation. Both have different properties concerning link capacities and placement of overlay nodes.

1) A WDM scenario consisting of a 24 node underlay used in [6], where a robust embedding of lightpaths is described, will be used as well. Only a fraction of the involved nodes will be overlay nodes, chosen randomly. Present overlay edges are chosen uniformly among all involved overlay nodes.

2) A wireless mesh network scenario (called *Wireless*, already used in our previous work [12], [7]) according to the generator NPART presented in [15] and capacities chosen on the basis of the path loss in the free-space propagation model [16] – only assuring that there is at least 30% nominal capacity available at every link, otherwise: there is no connection. These topologies represent the underlay, presuming radio connections do not interfere with each other – e.g. when using directional radio technology. Overlay nodes are placed randomly. One possible use case for an overlay in this scenario could be the installation of a VPN.

For evaluation of the approximation quality, we chose small instances of the WDM scenario (6 - 10 nodes). As mentioned earlier, bilevel programming involves enormous computational effort – thus, we used small network instances. We will use the other scenario to find out how many edges are not needed for an optimal topology and compare the quantity of required edges to the case of a fully meshed overlay that trivially accomplishes maximal robustness.

The setting of multiple demands reflects the idea, that although all nodes are equally important to the overall network function (peer-to-peer character of overlays) we always find a few nodes that provide unique services or contain important information in practical scenarios. Therefore, we introduce 2 important nodes (type A) and all other nodes are less important (type B). Clearly, all edges between types A and B will be necessary. All other connections between type B nodes are less likely – in our case, a type B node has connections to 5% of all other nodes of the same type. That specific demand model was already contained in our previous work [7].

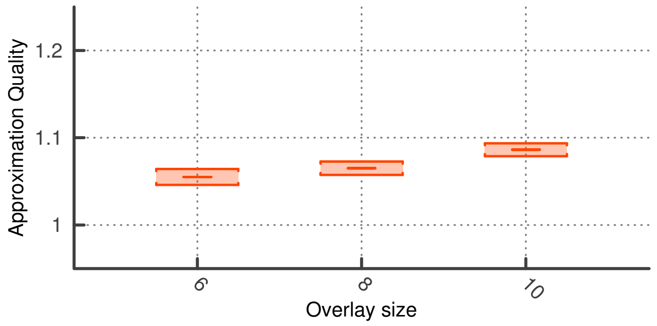

Fig. 1. Approximation quality for different overlay sizes. WDM 24 nodes.

B. Metrics

Since this paper deals with approximation algorithms, one obvious metric is the quality of approximation. When dealing with minimization problems, it is defined as z_{approx}/z_{opt} to obtain metric values ≥ 1 (note that a value of 1 is best). Hereby z_{approx} is the quantity of overlay edges in the topology generated by our approximation algorithm and z_{opt} the respective quantity by the bilevel ILP. The approximation quality will be used for our evaluation when running both bilevel ILP and approximation algorithm on smaller instances, only. Since the approximation at hand is hard to analyze theoretically, we restrict the evaluation to empirical results. In general, the size of the resulting topology will be an important figure of merit, as well as the actual value of the minimum cut. The latter is reflecting the resilience gained by using our approach. Concerning the size of resulting topologies, we normalized edge quantities to the size of a full mesh to determine the ratio of indispensable overlay edges. Throughout this paper, illustrated confidence intervals show 95% significance.

C. Results

First, we assess the quality of approximation for smaller instances, for which bilevel ILP computation was still possible within acceptable time. Therefore, we took small overlay WDM scenario instances and checked all node pairs. The conjecture of degrading quality with increasing overlay sizes seems obvious. But overall, a quality close to 1 is desired.

Fig. 1 shows confidence intervals for approximation qualities with different overlay sizes 6, 8 and 10 nodes. Larger instances were impossible to compute within reasonable time. The aforementioned conjecture is confirmed, as the qualities get worse with growing overlay sizes. Overall, they are ranging well below 1.1 which can be considered positive. We would like to examine the approximation quality even further, but unfortunately, larger instances are computationally intense.

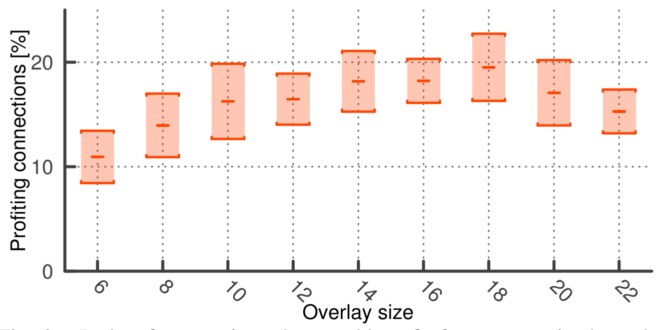

Fig. 2. Ratio of connections that would profit from our optimal overlay (compared to maximum disjoint path approach). WDM 24 nodes.

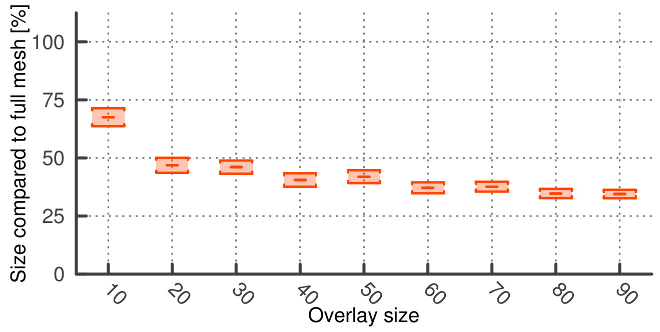

Fig. 3. Size of optimal overlay topology normalized by the size of a fully meshed overlay. Wireless with 100 nodes. Multiple demand setting.

Even with 3 different overlay sizes, one could assume that the quality of approximation worsens with growing overlays.

Now we investigate the resilience of the resulting overlay topologies. We examine the average ratio of connections that would improve their resilience compared to using the maximum disjoint paths approach from our previous work [7] (i.e., where "minimum cut $\geq$ maximum disjoint paths" holds).

Therefore, Fig. 2 shows the ratio of improved connections for several overlay sizes within our WDM scenario. A significant statistical impact of the overlay size is not proven, as almost every confidence interval overlaps, which can be due to the small size of the underlying topology and the accompanied restrictions to different overlay sizes. However, as values are ranging from 10.9% to 19.5% of connections that would improve their robustness on average, an overall gain of using the novel, minimum cut based approach is verified (i.e., robustness towards link outages increases by at least one link). That holds for WDM topologies examined in this paper. Using the wireless scenario, we did not detect any improvements. This can be due to missing features of topology generation, i.e. asymmetries of paths within the underlying network, that would occur naturally (due to interference patterns for wireless topologies) and as mentioned earlier, the gap between the maximum number of paths and the minimum cut can be arbitrarily large. Nonetheless, employing other scenarios is the only way to evaluate our approach with respect to the quantity of required overlay edges on bigger topologies, where reducing installed overlay connections is particularly important.

Moving on to the multiple demand setting, Fig. 3 illustrates the size of our resulting topologies compared to a naive, fully meshed overlay topology within our wireless scenario. A growing number of participating overlay nodes leads to comparatively smaller topologies, as the effort for a full mesh grows quadratically. Our approach achieves the very same robustness with e.g. 34% of all possible overlay edges (90 overlay nodes). Hence, a huge amount of overlay edges is not installed, meanwhile robustness is still maximal (and optimal).

VI. Conclusion and Future Work

Robustness of VPN overlay networks can be increased by calculating the minimum overlay cut, although it looks like it serves for attacking purposes only. This paper introduces two ways of exploiting the minimum overlay cut in order to create more robust overlay networks: a bilevel ILP formulation and an approximation. Even though bilevel ILPs yield enormous runtimes, we managed to solve small instances, which enabled evaluating the approximation quality of our algorithm. Furthermore, our evaluation shows, that only a fraction of the overlay edges needed to fully mesh all participating overlay nodes is sufficient to maintain maximal robustness. This offers significant practical advantages, as 1) an overlay routing protocol needs to take less edges into account and 2) maintenance effort for overlay edges is being reduced.

In our future work, a routing algorithm on overlay networks will be employed to measure how long an overlay rerouting takes. We also plan on calculating optimal topologies for multiple demands directly instead of obtaining one topology for each pair of communicating nodes and joining.

References

[1] F. Samie, L. Bauer, and J. Henkel, "Iot technologies for embedded computing: A survey," in *IEEE/ACM/IFIP CODES*, 2016.

[2] A. Gupta and R. K. Jha, "A survey of 5g network: Architecture and emerging technologies," *IEEE Access*, 2015.

[3] M. Rossberg, G. Schaefer, and T. Strufe, "Distributed automatic configuration of complex ipsec-infrastructures," *J. Netw. Syst. Manage.*, 2010.

[4] Y. Zhu and B. Li, "Overlay networks with linear capacity constraints," in *Quality of Service – IWQoS 2005*, 2005.

[5] L. Tang and P. Zhang, "Approximating minimum label st cut via linear programming," in *LATIN*, 2012. [Online]. Available: http://link.springer.com/chapter/10.1007%2F978-3-642-29344-3_55

[6] K. Lee, E. Modiano, and H.-W. Lee, "Cross-layer survivability in wdm-based networks," *IEEE/ACM Transactions on Networking*, 2011.

[7] M. Backhaus and G. Schaefer, "Towards construction of efficient and optimally resilient vpn topologies by exactly calculating maximum disjoint paths," in *IEEE ICC Proceedings*, 2017. Accepted and in press.

[8] F. Girlich, M. Rossberg, and G. Schaefer, "On the resistance of overlay networks against bandwidth exhaustion attacks," *Telecommunication Systems*, 2015.

[9] S. Dempe, *Foundations of Bilevel Programming*. Kluwer Academic Publishers, 2002.

[10] E. Altman, A. Singhal, C. Touati, and J. Li, *Resilience of Routing in Parallel Link Networks*, 2016.

[11] J. Chen and Q. Zhu, "Interdependent network formation games," *CoRR*, 2016.

[12] M. Backhaus and G. Schaefer, "Backup paths for multiple demands in overlay networks," in *IEEE GIIS Proceedings*, 2016.

[13] P. Xu and L. Wang, "An exact algorithm for the bilevel mixed integer linear programming problem under three simplifying assumptions," *Computers & Operations Research*, 2014.

[14] Gurobi Optimization, Inc., "Gurobi optimizer reference manual," 2016. [Online]. Available: http://www.gurobi.com

[15] B. Milic and M. Malek, "Npart - node placement algorithm for realistic topologies in wireless multihop network simulation," in *Simutools*, 2009.

[16] T. Rappaport, *Wireless Communications: Principles and Practice*, 2nd ed. Upper Saddle River, NJ, USA: Prentice Hall PTR, 2001.