Resilient routing in SDN networks

Mathieu Leconte, Stefano Paris, Jérémie Leguay

Routage résilient dans les réseaux SDN

Mathieu Leconte,   Stefano Paris et Jérémie Leguay

Mathematical and Algorithmic Sciences Lab - Paris Research Center, Huawei Technologies France.

Les réseaux SDN (software-defined networking) permettent à un contrôleur centralisé de décider du routage. Afin d'établir des routes fiables, il est souvent nécessaire de trouver plusieurs chemins dans le réseau ne partageant pas les mêmes ressoures à risque, communément appelé SRLG pour Shared Risk Link Group. Tout en assurant cette fiabilité, l'objectif est aussi de minimiser un coût, qui intègre des indicateurs de congestion ou de latence.

Ce problème peut être modélisé par un programme linéaire en nombres entiers (ILP). Nous proposons ici une méthode efficace pour sa résolution qui utilise une relaxation fractionnaire bien choisie, dont nous montrerons qu'elle mène en fait la plupart du temps à une solution entière. La résolution de ce problème relaxé utilise une méthode de génération de colonnes (CG), où chaque colonne représente un chemin dans le réseau avec une notion de coût modifié prenant en compte les SRLGs ; les nouvelles colonnes peuvent être obtenues par un algorithme efficace de programmation dynamique qui étend les algorithmes classiques de calcul de plus court chemin. Afin de limiter l'explosion combinatoire potentielle, nous présentons également une heuristique qui accélére le calcul d'une solution résiliente tout en préservant de très bonnes performances. Les résultats numériques montrent que notre approche donne une solution de très bonne qualité dans un temps de calcul raisonnable sur des instances de réseau réalistes.

Mots-clefs : Réseaux logiciel, routage, calcul de chemins, fiabilité, SRLGs.

1 Introduction

Future communication networks are expected to support multiple applications with a broad range of routing constraints and Quality-of-Service (QoS) requirements. An efficient organization of such a system may be obtained by some level of centralized control as proposed in the SDN paradigm, whereby a centralized controller monitors a part of the network and centrally decides routing policies.

A critical requirement for many applications is the reliability of connectivity, which translates into the survivability of routing against different failure scenarios. A standard way to model failure scenarios is through SRLGs [SCT01], which indicate sets of links that may fail together. A fundamental building block for optimizing the allocation of network resources is the computation of a set of end-to-end survivable paths (i.e., paths that belong to different SRLGs) and minimize an appropriate cost function that reflects objectives like congestion and latency.

Unfortunately, the problem of routing under SRLG-disjointness constraints is hard. Therefore exact approaches typically do not scale to large networks. The most common approach to simplify the problem relies on graph transformation techniques to replace SRLGs by virtual network nodes [DS08], but this technique only works in small scenarios. Therefore, heuristic approaches have been proposed [OMSY02, XXQL03, SGF$^+$11] to scale up with the network size. While these approaches focus on finding a set of survivable paths, they provide no guarantee in terms of cost with respect to the optimum.

In this work, we propose an efficient and systematic approach for solving the survivable path computation problem. Our approach also provides a lower-bound on the feasible minimum-cost solution, which allows us to verify the very good performance of our scheme in practice.

2 Formulation of the reliable routing problem

We consider a network graph $G = (V, E)$, where V is the set of vertices (or nodes) and E the set of edges (or links). Each edge $e \in E$ is associated with a cost c_e that typically represents multiple objectives like the price for using link capacity or the transmission delay of the link ; we write $\mathbf{c} = \{c_e\}_{e \in E}$. Shared risks among edges are represented by a set SRLG of subsets of edges in E. Each set $S \in$ SRLG indicates the edges that may fail together.

A demand is a request for sending traffic from a given source node s to a given destination node d. A path p in G is a sequence of consecutive edges. The cost of the path is the sum of all edge costs, namely $c_p := \sum_{e \in p} c_e$. We denote by $\mathcal{P}$ the set of all paths valid for the considered demand. Our goal is to select for each demand $k > 1$ paths that minimize the cost metric and do not share any risk.

The reliable routing problem can thus be formulated as the following Integer Linear Program (ILP) :

$$\begin{aligned} \underset{\mathbf{X} \in \{0,1\}^{\mathcal{P}}}{\text{minimize}} \quad & \sum_{p \in \mathcal{P}} X_p c_p, \\ \text{such that} \quad & \sum_{p \in \mathcal{P}} X_p = k, \\ & \forall S \in \text{SRLG}, \quad \sum_{p \cap S \neq \varnothing} X_p \leq 1, \end{aligned} \tag{1}$$

where the binary variables $\mathbf{X} := \{X_p\}$ indicate whether the paths p are selected for routing the demand. The second constraint of problem (1) indicates that selected paths in the set $P \subseteq \mathcal{P}$ must belong to different SRLGs. The above problem is known as the Minimum-Cost SRLG-Diverse Routing (MCSDR) problem [Hu03] and is NP-complete.

In the next section, we show a decomposition of the problem (1) which allows us to design a column-generation method with a simple and effective dynamic programming algorithm for solving the MCSDR problem.

3 Algorithmic approach and fast heuristics

Instead of solving directly the ILP (1), our approach solves the fractional relaxation of the problem where variables $X_p \in [0,1]$ and uses dual variables to price SRLGs that have been already traversed to avoid their further use. The relaxed problem is a Linear Program (LP), hence we can maintain a restricted basis of active paths P and add new variables using a column generation (CG) method. The restricted LP can be solved efficiently using standard LP solvers, which will compute optimal values for $\{X_p\}_{p \in P}$ and dual variables $\gamma := \{\gamma_S\}_{S \in \text{SRLG}}$ associated with the SRLG-constraints. We denote by SRLG(p) the set of SRLGs crossed by path p, i.e., SRLG$(p) = \{S \in \text{SRLG} : S \cap p \neq \varnothing\}$. For generating new paths, we need to compute shortest paths with respect to a modified cost notion, that can be defined considering the Lagrangian of the relaxed problem (1) :

$$L(\mathbf{X}, \gamma) = \sum_{p \in \mathcal{P}} X_p c_p + \sum_{S \in \text{SRLG}} \gamma_S \left(\sum_{p \cap S \neq \varnothing} X_p - 1 \right) = \sum_{p \in \mathcal{P}} X_p \left(c_p + \sum_{S \in \text{SRLG}(p)} \gamma_S \right) - \sum_{S \in \text{SRLG}} \gamma_S. \tag{2}$$

The best column to add to the basis corresponds to the path :

$$p^* \in \underset{p \in \mathcal{P}}{\arg\min}\ \widetilde{c}_p := \sum_{e \in p} c_e + \sum_{S \in \text{SRLG}(p)} \gamma_S. \tag{3}$$

3.1 Shortest path with SRLG-cost

Unfortunately the second term in equation (3) is not separable across edges in p. Therefore, we have extended the classical Dijkstra algorithm to find a path minimizing the modified cost $\widetilde{c}_p$. At each vertex v we need to store multiple paths from s and not only the path with the smallest cost, as different paths may cross different SRLGs and we do not known yet which SRLGs will be encountered in the remaining section of the path to the destination node d. In other words, for the cost $\widetilde{c}_p$, subsections of shortest paths are not necessarily shortest paths. Our solution is to discover all the *non-dominated* paths p from s with costs $\widetilde{c}_p < \widetilde{c}^*$, where $\widetilde{c}^*$ is the minimum modified cost of paths from s to d. A path p' from s to v dominates another path p from s to v if

$$\widetilde{c}_{p'} \leq \widetilde{c}_p - \textstyle\sum_{S \in \text{SRLG}(p) \setminus \text{SRLG}(p')} \gamma_S. \tag{4}$$

Indeed, whichever path p'' we may add, this condition ensures that $\widetilde{c}_{p' \cup p''} \leq \widetilde{c}_{p \cup p''}$.

In general, the complexity of this method can be exponential in the number of SRLGs. However, we only need to consider SRLGs S with dual variables $\gamma_S > 0$ and such SRLGs are likely to be few. In practice, we observe that the restricted LPs lead to very few SRLGs with positive cost. We show hereafter a pseudo-code for this procedure. The set OUT_v contains the edges outgoing from a vertex $v \in V$.

Shortest path with SRLG-cost – $\mathbf{sp}^{\mathbf{SRLG}}(G, \mathbf{c}, \mathcal{S}, \gamma)$.

```
Inputs : a graph $G = (V, E)$ with edge-costs $(c_e)$ ; a set of SRLGs $\mathcal{S} \subseteq$ SRLG with costs $(\gamma_S)$, with $\gamma_S > 0$.
Output : a path $p$ from $s$ to $d$ with minimum cost $\widetilde{c}_p$.
Procedure :
Perform a shortest path exploration with the SRLGs in $\mathcal{S}$ being monitored :
  — For each node $v \in V$, we maintain predecessor edges $e^U(v)$ and associated cumulative costs $\widetilde{c}^U(v)$ for some subsets $U \subseteq \mathcal{S}$, initially for none.
  — We maintain a heap of exploration moves, where a move is a tuple $(e, U, \widetilde{c})$ of an edge $e$, a set $U \subseteq \mathcal{S}$, and an associated modified cost $\widetilde{c}$. The heap is ordered so that it is quick to extract the move with the smallest cost.
  — We initialize the heap with a fake move $(\varnothing \to s, \varnothing, 0)$.
  — Until $d$ is explored (i.e., a move $(e, U, c)$ with vertex $d$ as head of $e$ is extracted), do the following :
      1. Extract from the heap the move $(e, U, \widetilde{c})$ with the minimum value of $\widetilde{c}$. Let $v$ be the head of $e$.
      2. If there exists $U' \in \mathcal{S}$ with an associated cost $\widetilde{c}^{U'}(v)$ in memory such that $\widetilde{c}^{U'}(v) \leq \widetilde{c} - \sum_{S \in U \setminus U'} \gamma_S$, then the current path is dominated : skip the rest of this step and go back to step 1.
         Else,
         (a) store in memory $e^U(v) \leftarrow e$ and $\widetilde{c}^U(v) \leftarrow \widetilde{c}$;
         (b) for all $e' \in \mathrm{OUT}_v$, add the move $(e', U \cup \mathrm{SRLG}(e') \cap \mathcal{S}, \widetilde{c} + c_{e'} + \sum_{S \in \mathrm{SRLG}(e') \setminus U} \gamma_S)$ to the heap.
  — Reconstruct the path $p$ by moving along all the predecessor edges in reverse direction :
    start with $p = \varnothing$, $v = d$, and $U$ is the only value for which there is an edge $e^U(d)$ in memory ; until $v = s$, do the following :
      1. let $e = e^U(v)$ ; add $e$ at the beginning of $p$ ;
      2. set $v$ to be the tail of $e$ ; replace $U$ by the subset $U' \subseteq U$ with the minimum value $\widetilde{c}^{U'}(v)$ in memory.
  — return $p$.
```

Note that it is perfectly possible to modify the method to work with bi-directional search.

3.2 Column-Completion Heuristic (CCH)

In many real cases, cost-optimality in (1) can be slightly compromised in order to quickly obtain a feasible set of paths. In this case, we can speed up the method by trying to complement every new column p_1 with a set $\{p_2, \ldots, p_k\}$ of $k-1$ SRLG-diverse paths (also SRLG-disjoint with p_1) and we output the first feasible set $\{p_1, \ldots, p_k\}$ obtained in this way. This yields a recursive column-completion method, where we declare failure if any LP is solved to completion. Note that the method may not find the minimum-cost solution, yet it still finds a feasible solution whenever the original algorithm succeeds. As we show in the numerical results, this method provides a substantial speed-up gain for a marginal cost increase.

4 Numerical Results

To evaluate our algorithm we generate large-scale networks composed of 1000 nodes and around 4000 edges, and we randomly selected source and destination nodes. SRLGs are randomly generated by clustering nodes in 100 domains and assigning all edges with at least one endpoint in a domain to the same SRLG. Fig. 4 shows a realization of such networks, where nodes in the same domain are depicted using the same color. We postpone more extensive simulations to longer versions of the paper. Note that, though the SRLGs in the simulations presented are generated in a very specific manner which would allow more efficient algorithms to be used, our method is generic and does not relies on any assumptions on the SRLGs.

We compare our algorithm with and without CCH against the fractional solution of problem (1), which may be non-integral and thus infeasible, and a greedy successive shortest path (SSP) approach, which

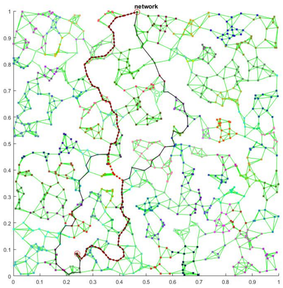

FIGURE 1: A figure

metrics	SSP	CCH	LP relaxation
success probability	0.84882	0.998	0.998
normalized avg. cost	1.0548	1.0498	1
avg. nb. of LPs	0	0.34155	6.8287
avg. nb. of normal S.P. calls	2	2.3415	1
avg. nb. of S.P. calls with SRLGs	0	0.34155	6.8287
max. nb. of active SRLGs	N/A	6	9

TABLE 1: numerical results

computes the first path over the original network, whereas the second one on a residual network where edges that belong to the SRLGs used by the first path are removed.

Table 4 shows the numerical results obtained with about 900 realizations of the network where the fractional relaxation of (1) is feasible. In all but 2 of the tested instances (i.e., in 99.8% of cases), the fractional relaxation is actually integral. Thus, our algorithms succeeded in almost all cases. On the contrary, the greedy algorithm only succeeded in less than 85% of the instances. In all the cases where greedy succeeds, CCH will find the same solution, which may however differ from that obtained by solving exactly the fractional relaxation. We find that the column-completion heuristics only costs an excess 5% on average (we normalized cost w.r.t. the fractional relaxation), while it significantly reduces the complexity of the algorithm.

In the particular scenario of Fig. 4, we show the solutions obtained by SSP and CCH. This scenario is chosen to show why SSP may fail : the first path it finds (dashed red line) uses critical SRLGs, which then makes it impossible to find a second SRLG-disjoint path. On the contrary, the restricted LPs adequately weight critical SRLGs to find an SRLG-diverse pair of paths (solid black lines).

Références

[DS08] Pallab Datta and Arun K Somani. Graph transformation approaches for diverse routing in shared risk resource group (srrg) failures. *Computer Networks*, 52(12) :2381–2394, 2008.

[Hu03] Jian Qiang Hu. Diverse routing in optical mesh networks. *IEEE Transactions on Communications*, 51(3) :489–494, 2003.

[JSMR01] Alpar Juttner, Balazs Szviatovski, Ildikó Mécs, and Zsolt Rajkó. Lagrange relaxation based method for the qos routing problem. In *IEEE INFOCOM 2001*, 2001.

[OMSY02] Eiji Oki, Nobuaki Matsuura, Kohei Shiomoto, and Naoaki Yamanaka. A disjoint path selection scheme with shared risk link groups in gmpls networks. *IEEE Communications letters*, 6(9) :406–408, 2002.

[OP08] Sebastian Orlowski and Michał Pióro. On the complexity of column generation in survivable network design with path-based survivability mechanisms. *Warsaw University of Technology, Tech. Rep*, 2008.

[SCT01] John Strand, Angela L Chiu, and Robert Tkach. Issues for routing in the optical layer. *IEEE Communications Magazine*, 39(2) :81–87, 2001.

[SGF$^+$11] José Silva, Teresa Gomes, Luís Fernandes, Carlos Simoes, and José Craveirinha. An heuristic for maximally srlg-disjoint path pairs calculation. In *ICUMT conference*, 2011.

[XXQL03] Dahai Xu, Yizhi Xiong, Chunming Qiao, and Guangzhi Li. Trap avoidance and protection schemes in networks with shared risk link groups. *Journal of Lightwave Technology*, 21(11) :2683, 2003.