A failure avoidance oriented approach for virtual network reliability enhancement

Shuopeng Li, Mohand Yazid Saidi, Ken Chen

A Failure Avoidance Oriented Approach for Virtual Network Reliability Enhancement

Shuopeng LI, Mohand Yazid SAIDI, and Ken CHEN

L2TI, Institut Galilée, Université Paris 13, Sorbonne Paris Cité, Villetaneuse, France

{li.shuopeng, saidi, ken.chen}@univ-paris13.fr

***Abstract*—Network virtualization allows the co-existing of logical networks (virtual networks) on physical network (substrate networks). Virtual Network (VN) reliability is a critical problem for end-users and service providers. It aims to ensure service continuity even upon failure. As more and more VNs are created over substrate networks (SN), the failure of a single SN component may lead to the failure of many VNs. Thus, the VN reliability issue is becoming more and more critical. VN reliability can be enhanced in two ways: (1) by failure recovery (post-failure) with protection and/or restoration methods; (2) by failure avoidance with the selection of most reliable components at the network topology setting phase. Traditional virtual network embedding (VNE) methods are mainly focused on bandwidth optimization. In this paper, we focus on the reliability issue. We propose VNE methods which take into account the failure probability of SN components with a failure-avoidance approach, in order to minimize the VN failure probability. Our heuristics are based on the use of Steiner Minimal Tree (SMT). Simulations results confirm that our heuristics provide better reliability against traditional VNE with bandwidth as sole target, and, in case of failure of a SN component, reduce the number of affected VN.**

I. Introduction

Network virtualization [1] provides new functionalities beyond traditional services. An important step of network virtualization is to establish virtual network (VN) above substrate network (SN). This is referred as *virtual network embedding* (VNE). The VNE problem aims to find a mapping from the VN to SN in a way that objectives are optimized and constraints are satisfied.

With network virtualization, VN can be set in a flexible manner. A VN can be embedded with various mapping topologies to the underlying SN, in order to satisfy various criteria (bandwidth utilization, reliability, etc.). This is quite different from the setting of a traditional (physical) network: once the mapping to the underlying network is set, the topology evolves rarely due to expensive cost of modification.

Network fails for various reasons, like bottlenecks, software attacks, natural disasters, etc. A failure of a single physical element (node, link) may result in major disruption involving several VN services. To avoid such scenarios which induce severe penalties (monetary and reputational) for the service provider (SP), this latter should provide reliable VNs capable to deal with failures [2].

Various techniques are developed for reliability. These techniques can be grouped in two categories: (1) failure-recovery and (2) failure-avoidance.

Failure-recovery techniques [3][4] consist to repair the affected VNs after the failure occurrence by determining a backup routing. *Protection* mechanisms pre-compute backup paths before the failure occurs, whereas *restoration* performs backup path computation upon failure occurrence. Generally, protection pre-reserves resources for the backup paths to guarantees enough resources upon failure and to ensure service continuity.

Failure-avoidance techniques [5] try to provide a primary routing which is determined in a way that failures affect as little as possible the network. Such techniques generally determine routes which bypass the network components which are statistically the most vulnerable to failures.

In this paper, we aims at proposing a novel failure-avoidance oriented approach for VN embedding. Considering that several VN can share a single SN component, our goal is to set up VNs which are *natively* more reliable, in order to minimize the frequency of the activation of failure recovery mechanisms. Actually, the latter would induce extra costs and delay. Our solution is complementary to, and compatible with, failure recovery mechanisms (restoration/protection).

Technically speaking, our approach is based on a primary consideration on VNE with infinite bandwidth. It happens when the physical resources are much larger than the logical requests. We modeled this VNE problem as a classical SMT (Steiner Minimal Tree) problem. This starting point allows us to consider the more realistic scenario of VNE with limited bandwidth. This problem is formulated as an Integer Linear Program (ILP) problem for which we propose SMT-based heuristics.

The rest of this paper is organized as follows. Section II presents the problem and the related work. Section III and IV present our solutions. The evaluation results are shown in section V. Section VI concludes this paper.

II. Position of problem and Related Work

A. Position of problem

With the following example (Figure 1), we show that different VNE strategies lead to different use of SN resources and, consequently, different level of reliability. The virtual network consists of 3 nodes $\{A, B, C\}$ and 2 virtual links $\{A - B, B - C\}$ (Figure 1(a)). Both virtual links demand

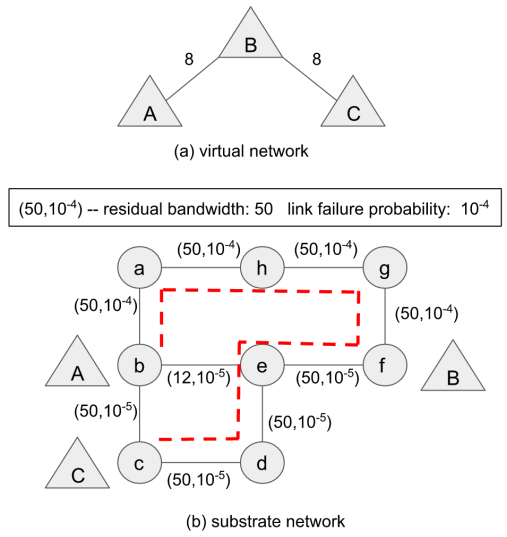

Fig. 1: virtual network embedding

8 units of bandwidth. The substrate network (Figure 1(b)) is composed of 8 nodes and 9 links. Each link is associated with a pair of values (residual bandwidth, link failure probability). We assume that the node mapping is: $b \rightarrow A$, $f \rightarrow B$ and $c \rightarrow C$. If the main criterion is bandwidth optimization, a plausible mapping is given by the dotted line. A mapping taking into account the failure probability is given in Figure 3. The VN failure probability of the former mapping is $1-(1-10^{-4})^4(1-10^{-5})^3$, which is higher than the latter $1-(1-10^{-5})^4$. This example suggests that the failure of SN components should be specifically taken into account in VNE problem for reliability. This is the focus of our paper, actually, the second mapping is provided by one of our heuristics.

B. Related work and our direction

Existing VNE framework [6] mainly concentrate on bandwidth oriented solution. A typical VNE problem can be solved in 2 stages, node mapping and then link mapping [7][8]. To address the lack of cooperation, one stage VNE solution are proposed in [9][10]. At our best investigation effort, we have found few VNE method dealing with failure avoidance. In [5], SN and VN is composed of access nodes and backbone nodes. Bee colony heuristic is applied to embed backbone nodes in order to avoid older equipements.

A probabilistic model is proposed in [11] to deal with Shared Risk Link Group (SRLG) failure. They formulated the problem as Non-linear Program and propose several approximations to solve it. In elastic optical networks[12], virtual links are mapped to the most reliable joint link path (primary and backup). All the possible mapping orders are computed, so it is difficult to apply in large networks. The work of [5] combines VNE and failure avoidance.

Since existing solutions are designed for different underlying networks, there is few avoidance oriented method for VNE. In this paper, we propose a novel failure avoidance method which embeds VN to a tree-like topology. Our method is compatible with any failure recovery method and it is efficient.

III. Infinite Bandwidth Probabilistic Solution

In the rest of the paper, as we are interested solely in VN failure frequency (recovery process activation frequency), we assume that a VN survives only if none of its substrate links is affected by a failure. As our method is compatible with restoration/protection, we assume that failures are eventually recovered, so that the affected VNs are not withdrawn. Besides, we focus solely on the failure of links. Actually, the joint consideration of both node and link mapping would create an additional degree of difficulty of VNE.

Below, we first introduce our network model and then maximize the reliability of VN when the link capacities are much larger than VN requests.

A. Network model

A Substrate Network is modeled as a connected undirected graph G^S=(N^S, L^S), where N^S is the set of substrate nodes and L^S is the set of substrate links. Each substrate node $n^s \in N^S$ is associated with a CPU capacity $cpu(n^s)$ and a geographic location $loc(n^s)$. A substrate link l_{mn} between node m and n is associated with a bandwidth capacity $bw(l_{mn})$ and a failure probability P_{mn}. We assume that link failures are independent. For the case of infinite bandwidth probabilistic solution, we set the bandwidth capacities of links to infinite.

Similar to the substrate network, virtual network is also modeled as an undirected graph $G^V = (N^V, L^V)$. N^V, L^V, $cpu(n^v)$, $loc(n^v)$ and $bw(l^v)$ have the same signification as those in substrate network. We denoted $M()$ as the node mapping function. The embedding substrate nodes form N^S_M. Link mapping stage embeds the virtual links to a subset of substrate links denoted by L^S_M.

B. Objective function

Given a VN request G^V, we define a set of binary variables X_{mn} denoting whether l_{mn} is used to embed G^V:

$$X_{mn} = \begin{cases} 1, & l_{mn} \in L^S_M \\ 0, & l_{mn} \notin L^S_M \end{cases}, \forall l_{mn} \in L^S \tag{1}$$

The surviving probability $PS(X)$ of a VN is given by:

$$PS(X) = \prod_{l_{mn} \in L^S} (1 - P_{mn} X_{mn}) \tag{2}$$

To optimize the reliability, (2) should be maximized. Instead of maximizing the non linear function (2), it is more easy to look for a new linear function that is optimal for the same solution X^* as (2).

To determine such linear function, we apply logarithm function to (2):

$$\begin{aligned} log\ PS(X) &= \sum_{l_{mn} \in L^S} log(1 - P_{mn} X_{mn}) \\ &= \sum_{l_{mn} \in L^S} log(1 - P_{mn}) X_{mn} \end{aligned} \tag{3}$$

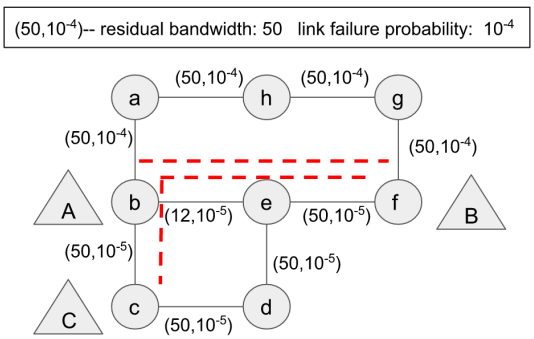

Fig. 2: Steiner tree solution

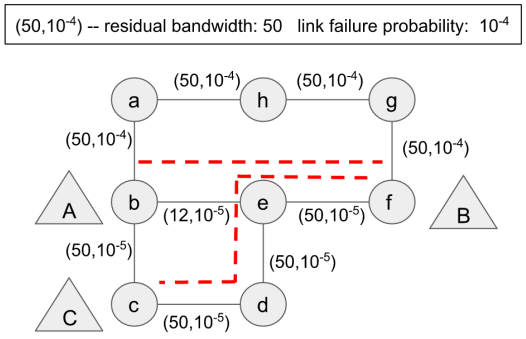

Fig. 3: Limited bandwidth solution

as $log(1 - P_{mn}X_{mn})$ is equal to $log(1 - P_{mn})X_{mn}$ for any l_{mn}. Recall that (2) and (3) are maximized for the same solution X^*. Instead of maximizing (3), we chose to minimize $-log\ PS(X)$. The final objective function is described below:

$$\min \sum_{l_{mn} \in L^S} [-log(1 - P_{mn})]X_{mn} \tag{4}$$

By associating a cost $C_{mn} = -log(1 - P_{mn})$ to each link l_{mn}, we deduce that the optimal solution of (4) corresponds to the least cost subs-network (in terms of C_{mn}) which spans all the nodes in N_M^S.

C. Steiner minimal tree solution

At VNE link mapping stage, we recall that the substrate nodes (N_M^S) which support virtual nodes via a node mapping function M are known. N_M^S is a subset of N^S. For the case of substrate links with infinite bandwidth capacities, the objective can be reduced to find a connected sub-graph spanning N_M^S with minimal cost C_{mn}. This is a classical NP-hard problem, called *Steiner minimal tree problem in graph (SMT)*, where N_M^S is the set of Steiner nodes.

Some efficient heuristics have been proposed. Among them we cite KMB [13]. KMB first constructs a complete graph of Steiner nodes where the links correspond to the shortest paths in substrate network. Finding the minimal spanning tree on the complete graph and deleting the loops (if necessary), we determine the final KMB tree which is at worst twice more costly than the Steiner tree.

Given SMT topology, each virtual link can be embedded to one and only one substrate path. This link mapping minimize the VN failure probability. For our example in Figure 1, Steiner tree solution is shown in Figure 2. Recall that the residual bandwidth constraint is ignored. The VN is mapped to $\{c - b, b - e, e - f\}$. The failure probability is $1 - (1 - 10^{-5})^3$.

IV. Limited Bandwidth Probabilistic Solution

In this section, we consider limited bandwidth capacities and add the admission control of bandwidth. Since failure probability and bandwidth are often orthogonal criteria, we try to optimize the probability of VN failure without breaking the bandwidth constraint.

For VN request with (1) constant number of links and (2) small bandwidth demands, the solution to the limited bandwidth probabilistic model corresponds also to Steiner tree, thus the limited bandwidth version is also NP-hard. Note that the solution of the limited bandwidth version is not a tree because of admission control on links. Below, we formulate the problem as an ILP and then give some efficient heuristics to solve it.

A. Exact ILP formulation

Our formulation doesn't allow path splitting. The formulation is shown below:

Variables:

- f_{mn}^{AB}: A binary variable denoting that virtual link l^{AB} is routed on the substrate link l_{mn}.
- X_{mn}: See Equation (1).

Objective:

Minimize:

$$\sum_{l_{mn} \in L^S} [-log(1-P_{mn})]X_{mn} + \epsilon \sum_{l_{mn} \in L^S} \sum_{l^{AB} \in L^V} (f_{mn}^{AB} + f_{nm}^{AB}) \tag{5}$$

Subject to:

$$\sum_{n \in N^S} (f_{mn}^{AB} - f_{nm}^{AB}) = \begin{cases} 1, & m = M(A) \\ 0, & \text{otherwise} \\ -1, & m = M(B) \end{cases}, \forall l^{AB} \in L^V \tag{6}$$

$$\sum_{l^{AB} \in L^V} bw(l^{AB})(f_{mn}^{AB} + f_{nm}^{AB}) \leq R_{mn}, \quad \forall l_{mn} \in L^S \tag{7}$$

$$\sum_{l^{AB} \in L^V} (f_{mn}^{AB} + f_{nm}^{AB}) \leq \beta X_{mn} \tag{8}$$

- The objective function (5) minimize the probability of substrate network failure (Formula 4). The second term avoids the path repetition. ϵ is a small constant.
- Equation (6) is the flow conservation constraint.
- Inequality (7) is the capacity constraint.
- β is a constant greater than the maximum number of VN links. Inequality (8) guarantees that if there exists an amount of flow on l_{mn}, X_{mn} would be 1.

Algorithm 1: baseline heuristic

Input : virtual network request $G^V(N^V, L^V)$
Output: link mapping solution

1 **begin**
2 **foreach** l^v *in* L^V **do**
3 compute probability-based shortest path cost c^v on G^S without bandwidth constraint;
4 **end**
5 **foreach** l^v *in ascending order cost* **do**
6 Determine a probability-based shortest path π for l^v. All the substrate links in π must verify the constraints of bandwidth
7 **if** $\pi = NULL$ **then**
8 free pre-allocated resources;
9 **return** no solution;
10 **end**
11 **else**
12 **foreach** l^s *in* π **do**
13 pre-allocate bandwidth resource;
14 set probability cost C_{l^s} of link l^s to 0;
15 add π to the solution;
16 **end**
17 **end**
18 **end**
19 **return** solution;
20 **end**

For the mapping problem described in Figure (1), ILP determines the solution shown in Figure (3). The two virtual links cannot cross $\{b-e\}$ at the same time because of bandwidth constraint ($8+8>12$). $B-C$ is embedded to $\{c-d-e-f\}$, whereas $A-B$ is mapped to $\{a-e-f\}$. The probability of VN failure is equal to $1-(1-10^{-5})^4$, which is lower than that of Figure 1. This solution minimizes the VN failure probability and respects the bandwidth constraint.

B. Heuristics

The ILP formulation is not scalable since it treats a NP-hard problem. Here, we propose heuristics which are inspired by KMB algorithm. Our heuristics use the shortest paths between Steiner nodes. However, only the shortest paths for pairs of Steiner nodes which support extremities of an existing virtual link are computed. Note that the resources allocated to a substrate path has incidence on the computation of paths. Thus, the sequence of link mappings impacts the final solution. Below we propose 2 heuristics to provide reliability for VNs.

1) baseline: in the basic Algorithm 1, we sort the virtual links according to their costs without taking into account the bandwidth constraint (line 3). Thus, virtual links are mapped one by one in their ascending order of cost (line 5). Substrate paths are computed so that they minimize the probability-based cost while verifying the bandwidth constraint (line 6). If no substrate path is determined to accommodate the traffic of a virtual link, we consider that the link mapping has no solution (line 7-9).

Algorithm 2: reinforced heuristic

Input : virtual network request $G^V(N^V, L^V)$
Output: link mapping solution

1 **begin**
2 $UL^V \leftarrow L^V$;
3 $cost \leftarrow \infty$; $\pi^* \leftarrow NULL$; $l^* \leftarrow NULL$;
4 **while** $UL^V \neq \emptyset$ **do**
5 **foreach** l^v *in* L^V **do**
6 Determine a probability-based shortest path π for l^v. All the substrate links in π must verify the constraints of bandwidth
7 **if** $\pi = NULL$ **then**
8 free pre-allocated resources;
9 **return** no solution;
10 **end**
11 **if** $C(\pi, l^v) < cost$ **then**
12 $cost = C(\pi, l^v)$;
13 $\pi^* = \pi$.
14 $l^* = l^v$;
15 **end**
16 **foreach** l^s *in* π^* **do**
17 pre-allocate bandwidth resource;
18 set probability cost C_{l^s} of link l^s to ε;
19 **end**
20 delete l^* from UL^V;
21 add (l^*, π^*) to solution;
22 **end**
23 **end**
24 **return** solution;
25 **end**

Similar to the Steiner tree, it is better to re-use substrate links. Thus, once a virtual link is mapped to a substrate path, the costs of substrate links on this path are set to zero (line 14) so that the next mappings privilege these links.

Note that the complexity of our basic heuristic is $O(|L^V|\ |N^S| log |N^S|)$, where $|N^S|$ is the cardinality of substrate nodes set and $|L^V|$ is the cardinality of virtual link set.

2) reinforced: the mapping performed by baseline heuristic is solely driven by links concentration and has no consideration on bandwidth optimization. Thus, at each iteration, it tends to re-use substrate links which have already been used for previous set of virtual links to set up the current virtual link, because this will not induce additional failure cost. As a result, compared to bandwidth-oriented VNE, the baseline heuristic leads to longer links paths consuming more bandwidth resources.

The reinforced heuristic (see Algorithm 2) is developed to correct this drawback. We no longer set a fixed order among substrate links. At each iteration, the least costly virtual link is mapped. In addition, we design a new cost function, $C(\pi, l^v)$, which takes also into account the bandwidth, along with the failure probability:

$$C(\pi, l^v) = C_s(\pi) - \mu \frac{bw(l^v) - \overline{(L^V)}}{\overline{(L^V)}}$$

$\overline{(L^V)}$ is the average of bandwidth demands of a VN. μ is a positive constant which controls the importance of bandwidth. This cost is inversely proportional to the bandwidth demand, so virtual links with high bandwidth demands are privileged. Virtual links with low demands are more likely to re-use the mapped links, which reduce the side effect.

Furthermore, if we have 2 shortest paths with the same cost, one of them re-uses some substrate links and the other not. To better balance the load, we propose to avoid the re-use of substrate links which don't reduce the failure probability. To achieve this purpose, the cost of a used link (line 18) is set to a small constant ε instead of 0. The complexity of reinforced heuristic is $O(|L^V|^2 \ |N^S|log|N^S|)$.

V. Performance Evaluation

We implemented a discrete event simulator to evaluate the performances of our proposals. The optimization problem is solved by IBM CPLEX library. Since we are basically interested in link mapping, all the evaluated methods work with the same node mapping described in [7].

A. Evaluation environment

The substrate network (50 nodes, 120 links) is generated by GT-ITM tool [14]. The CPU capacity of each node is randomly chosen in [100, 150]. The bandwidth capacity is randomly selected in [100, 150]. The failure probability of substrate links follows a uniform distribution over the interval $[0,\ 2 \times 10^{-5}]$.

The virtual networks are also generated by GT-ITM tool. The virtual nodes of each VN follow a uniform distribution between 3 and 10. The virtual nodes are interconnected with probability 0.4. The CPU and bandwidth demands are uniformly chosen in [0, 20].

The VN request arrival process is Poisson with arrival rate $\lambda \in (1 \dots 8)$ per 100 time units. The life time of each VN follows an exponential distribution with an average of 1000 time units. Each simulation lasts for 100000 time units.

B. Comparison

Through the numerical studies, we mainly want to assess the advantage as well as shortcoming of our methods versus traditional bandwidth-oriented VNE methods. We compared the following methods:

(i) $baseline$: our baseline heuristic (Section IV-B1),
(ii) $reinforced$: our reinforced heuristic (Section IV-B2),
(iii) $exact$: the ILP solution provided by CPLEX which acts as the bottom line reference,
(iv) bw: a basic shortest path method for virtual network embedding which optimizes the bandwidth. It is our comparison reference.

More precisely, we want to examine the following points:

(i) What is the VN acceptance ratio, which is the overall indicator of resources utilization.
(ii) What is the probability of failure for each method: a VN provider should give this assessment.
(iii) What is the impact of a failure: a VN provider should have a clear idea on it to assess the failure risk.

For this, we chose the following metrics:

- *Acceptance ratio of VN request*: the number of the accepted VN requests over the number of the total arrived VN requests.
- *Total revenue*: The revenue of a VN corresponds to the weighted sum of bandwidth and CPU:

$$\mathscr{R} = \beta \sum_{l^v \in L^V} bw(l^v) + \alpha \sum_{n^v \in N^V} cpu(n^v)$$

  where α and β correspond respectively to the unit revenue for cpu and bandwidth.
- *VN failure probability*: A VN survives if none of its virtual link fails. We deduce the failure probability P_{G^v} of a VN $(G^V) : P_{G^v} = 1 - \prod_{l_{mn} \in L^S_{G^V}} (1 - P_{mn})$, where $L^S_{G^V}$ is the set of substrate links on which G^V is mapped.
- *Average number of affected VNs*: A single substrate link failure affects all of the virtual networks using the failed link. This metric counts the number of affected VNs for each failure event.

C. Result analysis

In terms of acceptance ratio, our heuristics are surely less efficient than the bandwidth-oriented method. The good news is that the difference is quite limited unless in case of heavy load. Figure 4 and 5 show that, at low network loads ($\lambda < 4$), our methods achieve more than 95% of mapping task compared to bw. This percent falls to 80% for heavy load ($\lambda > 4$).

Our approach of failure-avoidance does work. Actually, Figure 6 shows that there is a significant reliability enhancement carried by our method versus the bandwidth-oriented VNE, from about 60% for low network load till 20% for heavy load.

Our heuristics also limit the failure impact. Actually, as given by Figure 7, the number of impacted VN produced by bandwidth method in case of failure is always higher than those obtained with our method.

Now, let us take a closer look at the difference between our two methods. As shown by Figures 6 and 7, $reinforced$ outperforms $baseline$. Our $reinforced$ method takes advantage of the dynamic metric model and it is closer to $exact$ (the reference optimal value) than $baseline$. We also want to point out that for the acceptance ratio and revenue in Figures 4 and 5, our two heuristics get the same performance as that provided by $exact$.

VI. Conclusion

Virtual networks are being more and more used as a major component for network architecture. A single component of SN may be used to support a large number of VNs, and, consequently, its failure may have a large impact on the reliability of these VNs. Failure recovery strategies can be applied to VN protection. However, the activation of these strategies induces

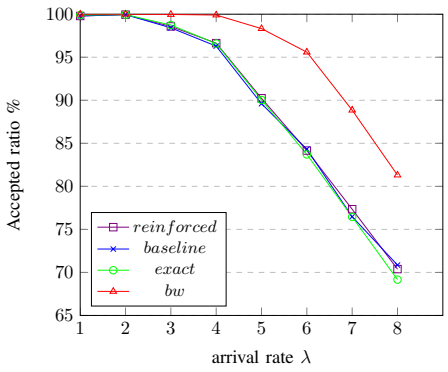

Fig. 4: Acceptance ratio

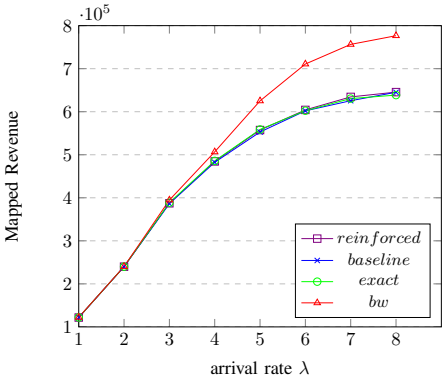

Fig. 5: Mapped Revenue

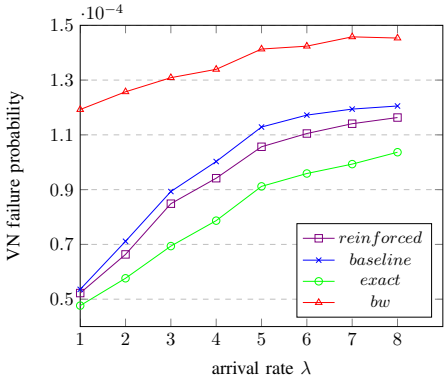

Fig. 6: VN failure probability

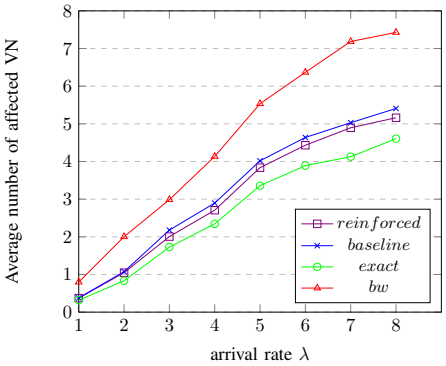

Fig. 7: Average number of affected VN

cost and delay, and, there is always communication disruptions before recovery.

In this paper, we presented our approach of a novel VNE scheme, which takes into account the failure probability of the underlying SN components, with the strategy of avoiding the worst components if possible. In this way, our method tries to provide a *natively* more reliable virtual network by minimizing its failure probability. Simulations results confirm that our method does achieve our design goal, against traditional VNE with bandwidth as sole target. By applying our method, at the price of a slightly lower acceptance ratio, a VN provider can exhibit a better reliability for each single VN instantiation. In case of failure of a SN component, the number of affected VN will also be reduced.

References

[1] T. Anderson, L. Peterson, S. Shenker, and J. Turner, "Overcoming the internet impasse through virtualization," *Computer*, no. 4, pp. 34–41, 2005.

[2] C. Colman-Meixner, C. Develder, M. Tornatore, and B. Mukherjee, "A survey on resiliency techniques in cloud computing infrastructures and applications," *Communications Surveys & Tutorials, IEEE*, vol. PP, no. 99, p. 1, 2016.

[3] A. Jarray and A. Karmouch, "Cost-efficient mapping for fault-tolerant virtual networks," *IEEE Transactions on Computers*, vol. 64, pp. 668–681, Mar. 2015.

[4] S. Herker, A. Khan, and X. An, "Survey on survivable virtual network embedding problem and solutions," in *ICNS 2013, The Ninth International Conference on Networking and Services*, pp. 99–104, 2013.

[5] O. Soualah, I. Fajjari, N. Aitsaadi, and A. Mellouk, "Pr-vne: Preventive reliable virtual network embedding algorithm in cloud's network," in *Global Communications Conference (GLOBECOM), 2013 IEEE*, pp. 1303–1309, IEEE, 2013.

[6] A. Fischer, J. F. Botero, M. Till Beck, H. De Meer, and X. Hesselbach, "Virtual network embedding: A survey," *Communications Surveys & Tutorials, IEEE*, vol. 15, no. 4, pp. 1888–1906, 2013.

[7] M. Yu, Y. Yi, J. Rexford, and M. Chiang, "Rethinking virtual network embedding: substrate support for path splitting and migration," *ACM SIGCOMM Computer Communication Review*, vol. 38, no. 2, pp. 17–29, 2008.

[8] N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual network embedding with coordinated node and link mapping," in *INFOCOM*, pp. 783–791, IEEE, 2009.

[9] J. Lischka and H. Karl, "A virtual network mapping algorithm based on subgraph isomorphism detection," in *Proceedings of the 1st ACM workshop on Virtualized infrastructure systems and architectures*, pp. 81–88, ACM, 2009.

[10] X. Cheng, S. Su, Z. Zhang, H. Wang, F. Yang, Y. Luo, and J. Wang, "Virtual network embedding through topology-aware node ranking," *ACM SIGCOMM Computer Communication Review*, vol. 41, no. 2, pp. 38–47, 2011.

[11] H. W. Lee, E. Modiano, and K. Lee, "Diverse routing in networks with probabilistic failures," *IEEE/ACM Transactions on Networking*, vol. 18, pp. 1895–1907, Dec. 2010.

[12] H. Yang, L. Cheng, G. Luo, J. Zhang, Y. Zhao, H. Ding, J. Zhou, and Y. Wang, "Survivable virtual optical network embedding with probabilistic network-element failures in elastic optical networks," *Optical Fiber Technology*, vol. 23, pp. 90–94, 2015.

[13] L. Kou, G. Markowsky, and L. Berman, "A fast algorithm for steiner trees," *Acta informatica*, vol. 15, no. 2, pp. 141–145, 1981.

[14] GT-ITM *http://www.cc.gatech.edu/projects/gtitm/*.