HAL
archives-ouvertes.fr

Resilient Virtual Network Design for End-to-End Cloud Services

Isil Barla, Dominic Schupke, Georg Carle

▶ **To cite this version:**

Isil Barla, Dominic Schupke, Georg Carle. Resilient Virtual Network Design for End-to-End Cloud Services. Robert Bestak; Lukas Kencl; Li Erran Li; Joerg Widmer; Hao Yin. 11th International Networking Conference (NETWORKING), May 2012, Prague, Czech Republic. Springer, Lecture Notes in Computer Science, LNCS-7289 (Part I), pp.161-174, 2012, NETWORKING 2012. <10.1007/978-3-642-30045-5_13>. <hal-01531116>

HAL Id: hal-01531116
https://hal.inria.fr/hal-01531116

Submitted on 1 Jun 2017

HAL is a multi-disciplinary open access archive for the deposit and dissemination of scientific research documents, whether they are published or not. The documents may come from teaching and research institutions in France or abroad, or from public or private research centers.

L'archive ouverte pluridisciplinaire **HAL**, est destinée au dépôt et à la diffusion de documents scientifiques de niveau recherche, publiés ou non, émanant des établissements d'enseignement et de recherche français ou étrangers, des laboratoires publics ou privés.

Distributed under a Creative Commons Attribution| 4.0 International License

Resilient Virtual Network Design for End-To-End Cloud Services

Isil Burcu Barla[1,2], Dominic A. Schupke[1], and Georg Carle[2]

[1] Nokia Siemens Networks, Munich, Germany,
{isil.barla.ext, dominic.schupke}@nsn.com
[2] University of Munich, Munich, Germany,
carle@in.tum.de

Abstract. Network virtualization with combined control of network and IT resources enables network designs for end-to-end cloud services with latency and availability guarantees. Even though providing such QoE guarantees is of high importance for cloud services, it is mostly not possible today if the services traverse different domains. To addresses this problem, firstly, we introduce novel resilient design methods for virtual networks minimizing the cost or the latency of the virtual network. We realize the routing of the services and the mapping of the virtual network simultaneously. Secondly, we provide two fundamental cloud connection architectures, which provide end-to-end resilience for cloud services in the presence of both network and datacenter failures. Using extensive simulations, we evaluate the performance of the proposed architectures in terms of cost of the virtual networks and maximum end-to-end delay that they can guarantee for cloud services.

Keywords: network virtualization, resilience, cloud services, latency

1 Introduction

Cloud services are more and more utilized by businesses and for private applications, where latency and availability guarantees are of high importance especially for business critical applications. The performance of the cloud services is the key metric to measure the acceptability of that service by the end-users and hence it directly impacts the revenue for service providers [1]. Moreover, resilience of the cloud services in the presence of both *DataCenter* (DC) and network failures is as well a key point especially for the "cloudified" businesses, where a DC failure might cause a service outage in the range of days.

Cloud providers are aware of these problems and try to address them by offering service level agreements to their customers. However, these agreements only cover the performance and connectivity inside the cloud and exclude the telecommunication networks, which might actually cause excessive latencies and even service outages. Thus, today it is essentially impossible to provide quality of experience guarantees in an end-to-end fashion for cloud services. One solution for this problem is using the concept of *Network Virtualization* with combined control for network and IT resources.

Network virtualization is proposed as a key enabler for the next generation networks and the future Internet [3, 4]. Unlike current virtualization techniques like Virtual Private Networks (VPNs) and overlay networks, network virtualization enables operation of isolated network slices. A slice consists of isolated computational resources inside network and DC nodes, as well isolated network resources between them. In a virtual network environment new business models realizing different tasks are expected to be established and as a result, as well trading of virtual resources between them [5]. Note that these resources can be network resources and/or IT resources. In such an environment new control mechanisms and interfaces are necessary to realize the setup and operation of the *Virtual Networks* (VNets). For possible realizations of combined control of IT and network resources using virtualization, there are already several suggestions in the literature [17]. There are as well some commercial offers like from Amazon [6], where a virtual network is offered together with the cloud services. Still, such solutions currently lack redundancy and QoE guarantees, which are the main reasons for hesitation of businesses to adopt cloud solutions according to a survey done in 2011 with 3700 enterprises worldwide [7]. Therefore, in this paper we propose novel architectures to enable provisioning of cloud services with end-to-end availability and latency guarantees. In our network virtualization model, we define two business roles, namely the *Virtual Network Operator* (VNO), operating a VNet on the physical substrate, and the *Physical Infrastructure Provider* (PIP) owning the physical substrate.

A PIP is the owner of the physical infrastructure, which can be e.g. a transport network, wireless access network, IT resources like processing or storage units or any combination of them. The PIP is in the position to monitor all of its physical and virtual resources and has the knowledge of the usage and physical location of its virtual resources. Given a PIP with existing resources, its incentive would be optimizing the utilization of its resources to maximize its revenue according to a chosen strategy, e.g. minimizing the used capacity or load balancing, by allocating the virtual resources accordingly. In this paper we focus on the PIPs owning transport networks and IT resources. In the remainder of the paper, if there is a need for distinguishing, the PIPs owning only network resources will be called nPIPs and the ones having only DC or a combination of both will be called dcPIPs. A VNO can operate one or several VNets, which are mapped onto the physical infrastructure of possibly one or more PIPs. A VNet can consist of both virtual network and IT resources. The interfaces and information sharing between the VNO and the PIP would depend on their internal business models and the contract between them [2]. Without loss of generality, we assume that for the VNet setup the available virtual resources of the PIPs are advertised to the VNO. The VNO can negotiate with various PIPs and compute an optimal VNet according to its specific needs. These can be e.g. either minimizing the cost of the VNet or optimizing the VNet design to provide minimum latency for the running services.

In such a virtual network environment there are two fundamental resilience architecture options, namely resilience can be provided solely by the PIP, *PIP-*

Resilience, or only by the VNO, *VNO-Resilience*. Applying resilience at different layers has several advantages and drawbacks. In [8], we address this question and compare the PIP and VNO-Resilience cases in a qualitative manner in terms of network resource usage, service level resilience adaptability and network setup and operation complexity. On the one hand, a VNO can utilize the available resources of different PIPs to reach an overall optimal design regarding both resilience and performance considerations. On the other hand, PIP-Resilience can offer a simpler signaling interface between the VNO and the PIP. It also provides lower system complexity in terms of the concurrent actions taken in case of a failure. Furthermore, in PIP-Resilience, the recovery action is transparent to the VNet and hence the virtual topology remains unchanged. Finally, the VNO-Resilience can offer VNet setup at service level granularity. Note that hybrid mechanisms are out of the scope of this paper due to our aim of investigating the resilience design choices affecting the delay performance and cost of the VNets.

In this paper, we propose novel solutions for end-to-end cloud services with availability and latency guarantees under both network and DC failures. We provide end-to-end solutions both for VNO and PIP-Resilience cases. Firstly we introduce novel resilient VNet design methods, which route the requested services and map the VNets onto the physical substrate simultaneously. We model our resilient VNet designs as two sets of *Mixed Integer Linear Problems* (MILPs) for VNO and PIP-Resilience cases by minimizing the delay of all the possible services in the VNet and the cost of the VNet, while providing network resilience, respectively. Afterwards, we combine these VNets with resilient cloud connection models for VNO and PIP-Resilience, which provide resilience in the presence of both physical network and DC failures for end-to-end cloud services. We evaluate the performance of our VNet designs in terms of cost and delay and we show their efficiency and applicability compared to traditional shortest paths mapping approaches. Finally, using extensive simulations we compare the two end-to-end solutions in terms of their delay performances. The remainder of the paper is organized as follows: Section 2 gives a short summary of the related work, in Section 3 the resilient VNet designs and in Section 4 the cloud connection models are introduced and their performances are evaluated. Finally, Section 5 concludes the paper with a discussion of the results and an outlook.

2 Related Work

Regarding the VNet design there are mainly two types of works in the literature. The first one is on routing the services in a VNet according to quality of service or availability requirements [9, 10]. It is assumed that the VNet is already existing and mapped onto the physical substrate. However, in this paper we deal with the problem of designing a new VNet for a VNO according to the given requirements. Unlike the overlay or VPN services where the customer needs to pay only per usage, a VNO would need to pay for the setup and maintenance of its VNet.

Therefore, it is very important from the beginning to design a cost-efficient VNet, which can offer the required service quality.

The second type of work in the literature offer solutions for the mapping of a VNet onto the physical substrate [11, 12]. Moreover, there are as well proposed algorithms for mapping survivable VNets [13, 14]. However, all of these works assume that the virtual topology is already given and in case of survivable mapping, the VNet has to be even bi-connected so that the mapping can be realized at all. Our work does not have such a limitation and designs a VNet with the given requirements. In [16], the authors deal with the VNet design problem for the case of the overlay networks by minimizing the cost of the overlay network. However, they do not consider resilience and use direct shortest path mappings. In [15], resilient VPN designs are realized but again assuming direct mapping of the virtual links on shortest physical paths. We extend this approach by allowing several mapping choices for a virtual link and show that our approach outperforms the shortest path mapping model in terms of feasibility and delay performance. Thus, there is extensive work available for mapping a given VNet on the physical substrate and routing a set of services in a VNet, which is already mapped onto the physical substrate. However, to the best of our knowledge this is the first paper, which considers both mappings simultaneously to realize cost-efficient and latency-optimized resilient VNet designs.

Finally, we provided a qualitative comparison of PIP and VNO-Resilience cases in a virtual network environment in [8]. However, to the best of our knowledge, this is the first paper proposing resilient VNet designs and cloud connections for these two fundamental cases and conducting a quantitative study for the cost and latency evaluation of these models.

3 Resilient Virtual Network Design

In this section, the resilient VNet design models are introduced. It is assumed that a set of requested services and the physical network are provided. The models are given in the form of MILPs and the optimization is performed for minimizing the maximum latency or the cost of the VNet. The first subsection introduces the simple model without any resilience considerations. In the following subsections VNO-Resilience and PIP-Resilience models are introduced. Finally, in Subsection 3.4 we evaluate the performance of the different models using different parameter settings and comparing to *Shortest Path Mapping* (SPM) and *SPM with Additional Nodes* (SPMwAN) models, where each virtual link is directly mapped onto the physical shortest path between its end-nodes.

3.1 Simple Model without Resilience

In the *Simple Model* (SM) the virtual links are mapped onto single paths in the physical network and the services are routed in the VNet on $i \in \{1, .., r\}$ routes. In SM, the *service nodes*, i.e. the end-nodes of the given services, are directly used as the virtual nodes of the resulting VNet. The virtual links have k-shortest

paths[3] mapping possibilities. However, to maintain linearity instead of using one virtual link with several possible mappings, we generate a new virtual link for each mapping and add it to the list of all the possible virtual links. The result of the optimization problem is a VNet, which consists of only the links and nodes that are used to route any of the given services. In the following, the sets, parameters and variables used in the MILPs are briefly introduced.

- *Sets:*
  - V: Set of the all virtual node candidates
  - L: Set of the all virtual link candidates
  - D: Set of the requested services
  - E_l: Set of the endpoints of link $l \in L$
  - N: Set of virtual links $(j,k) \in L^2$, which share at least one physical edge
- *Parameters:*
  - b_d: Requested bandwidth for the service $d \in D$
  - c_d: Requested node resources for the service $d \in D$
  - s_l: Physical length of link $l \in L$
  - λ_l: Fixed setup cost for having a new link $l \in L$
  - θ_l: Setup cost per unit capacity for link $l \in L$
  - μ_v: Fixed setup cost for having a new node $v \in V$
  - η_v: Setup cost per unit capacity for node $v \in V$
- *Variables*
  - $\beta_{i,d,l}$: Binary variable taking the value of 1 if the link $l \in L$ is used for the i^{th} route of the demand $d \in D$, 0 otherwise
  - $\delta_{i,d,v}$: Binary variable taking the value of 1 if the node $v \in V$ is used for the i^{th} route of the demand $d \in D$, 0 otherwise
  - γ_l: Binary variable taking the value of 1 if the link $l \in L$ is in the resulting VNet, 0 otherwise
  - α_v: Binary variable taking the value of 1 if the node $v \in V$ is in the resulting VNet, 0 otherwise
  - $u_l \in [0,\infty]$: Used capacity on link $l \in L$
  - $\omega_v \in [0,\infty]$: Used capacity on node $v \in V$

The constraints for SM are given in the following. Eq. (1) is the link-flow constraint. Eq. (2) makes sure that a node is flagged as "used" for a service if it is the source or the target of that service. Eq. (3) and (4) state that a virtual link or node is part of the resulting VNet if it carries the traffic of any service, respectively. Finally, Eq. (5) and (6) are the constraints for link and node capacity, respectively.

$$\sum_{l:v\in E_l} \beta_{i,d,l} = \begin{cases} 1 & \text{if } v = s \text{ or } v = t \\ 2\delta_{i,d,v} & \text{otherwise} \end{cases} \quad \forall d = (s,t) \in D,\ v \in V,\ i \in \{1,..,r\} \tag{1}$$

[3] When all simple paths between two nodes are listed in ascending order according to their lengths, k-shortest paths between these two nodes are the first k paths in the list.

$$\delta_{i,d,v} = 1 \quad \forall d = (s,t) \in D,\ \forall v \in (s,t),\ i \in \{1,..,r\} \tag{2}$$

$$\gamma_l \geq \beta_{i,d,l} \quad \forall l \in L,\ \forall d \in D,\ \forall i \in \{1,..,r\} \tag{3}$$

$$\alpha_v \geq \delta_{i,d,v} \quad \forall v \in V,\ \forall d \in D,\ \forall i \in \{1,..,r\} \tag{4}$$

$$u_l \geq \sum_{i \in \{1,..,r\}} \sum_{d \in D} \beta_{i,d,l}\, b_d \quad \forall l \in L \tag{5}$$

$$\omega_v \geq \sum_{i \in \{1,..,r\}} \sum_{d \in D} \delta_{i,d,v}\, c_d \quad \forall v \in V \tag{6}$$

There are two objective functions defined for different optimization objectives, namely VNet cost minimization and delay minimization. Note that the cost of the VNet constitutes of the link cost and the node cost, where each of them has again two parts, namely the fixed setup cost for having a new link or node in the VNet and the capacity dependent cost depending on the requested capacity on that link or node. To achieve simplicity in the PIP-VNO business relationships, a linear cost model is assumed. In cost minimization the total cost of the VNet and in propagation delay minimization the total length of the routes for each service are minimized. We only consider the propagation delay in the physical path as the latency metric for a service since the network is designed for normal load conditions. Thus, the queueing delay is negligible and the main latency is caused by the propagation of the signal over physical distances. Expressions (7) and (8) show the objective functions for VNet cost minimization and delay minimization of the services, respectively.

$$\min \quad \Big(\sum_{l \in L} (\lambda_l \gamma_l + \theta_l u_l) + \sum_{v \in V} (\mu_v \alpha_v + \eta_v \omega_v) \Big) \tag{7}$$

$$\min \quad \sum_{d \in D} \sum_{i \in \{1,..,r\}} \sum_{l \in L} \beta_{i,d,l}\, s_l \tag{8}$$

3.2 VNO-Resilience

For VNO-Resilience, 1:1 protection routing is used in the virtual layer, where the working and protection paths of a service have to be physically disjoint. Hence, the number of the routes r is 2. To provide resilience additional diversity constraints are introduced to the model. The constraint given in (9) ensures that the virtual working and protection paths of a service do not contain any two virtual links, which share common edges in the physical layer. Equation (10) provides node-diversity, where the working and protection paths are not allowed to share any nodes other than the end-nodes. In case of a physical link or node failure, the affected services are re-routed by the VNO on their pre-calculated protection paths.

$$\beta_{1,d,j} + \beta_{2,d,k} \leq 1 \quad \forall d \in D,\ (j,k) \in N \tag{9}$$

$$\delta_{1,d,j} + \delta_{2,d,k} \leq 1 \quad \forall d = (s,t) \in D,\ (j,k) \in V \setminus \{s,t\} \tag{10}$$

3.3 PIP-Resilience

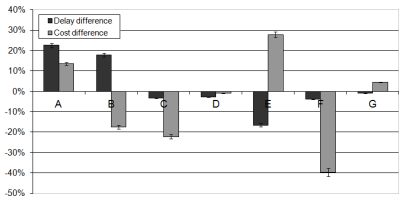
In the case of PIP-Resilience, providing resilience is the responsibility of the PIP(s). The services are routed on single paths in the VNet layer, where each virtual link is mapped on two disjoint physical paths in the physical layer. The disjointness criteria can be defined as link-disjoint or node-disjoint. For PIP-Resilience, SM is directly applied where the number of virtual routes is set to 1. However, instead of k-shortest physical path mapping for the virtual links, k-shortest disjoint path pairs[4] mapping is used. Therefore, the VNO sees only a simple network, which is protected in the physical layer. The re-routing in case of a failure is realized in the physical layer by the corresponding PIP, i.e. the virtual topology remains unchanged and ideally the services are not disrupted.

3.4 Performance Evaluation of the Resilient Virtual Network Design Options

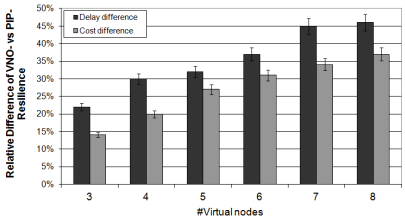
In this section the two proposed models are compared in terms of the VNet cost and maximum service delay they provide for different cost parameters and optimization functions. Moreover, the performance of the models is evaluated against the SPM and SPMwAN models, where both of them use direct shortest path mapping. As the resilience strategy they both utilize the VNO-Resilience, i.e. the services are routed on two virtual paths, which are physically disjoint. In SPM, like in the VNO and PIP-Resilience cases, the virtual node set consists of the service nodes. In SPMwAN, however, the virtual node set is extended, where the VNO can use as well additional virtual nodes for routing purposes, which are not the source or target of any of the services. Similarly, the virtual link set is as well extended to cover the possible links between all the node pairs in the new node set.

For the performance evaluation, we used two test networks, namely the NobelUS and NobelEU [18] networks. For VNet generation, first, we select a certain number of service nodes randomly from the physical network, where there is uniform demand between all of them. Then we solve the optimization problem for different resilience models, where each of them result in a different VNet. They are then compared regarding their delay/cost performances until a confidence level of 95% and $\pm 5\%$ confidence interval is reached. Link diversity option is used for the simulations. However, our results show that node diversity option results in comparable delay and cost values as link-diversity. To evaluate the effect of different cost factors and optimization functions on the resulting cost and delay, we distinguish between seven cases as shown in Fig.1a. For this analysis the NobelUS network is used. Results for 3-node VNets are shown due to their significance and applicability in real life scenarios. The cost and delay differences shown in the figure are the relative differences of the two models, which are calculated by taking the difference of PIP and VNO-Resilience value and

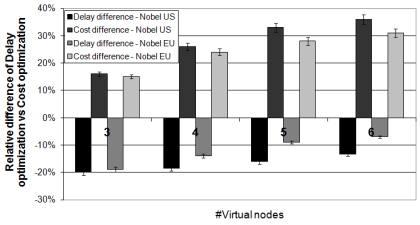
[4] K-shortest disjoint path pairs are the first k disjoint path pairs when all the disjoint path pairs are listed in ascending order according to their total length.



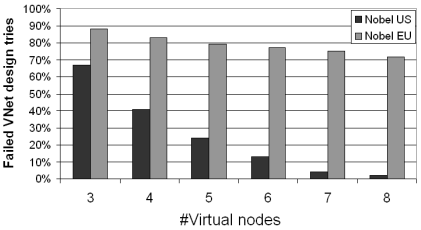
(a) Effect of different cost and optimization function settings denoted by A-G



(b) VNet Design for VNO-Resilience vs PIP-Resilience (cost optimization)



(c) Delay Minimization vs. Cost Minimization

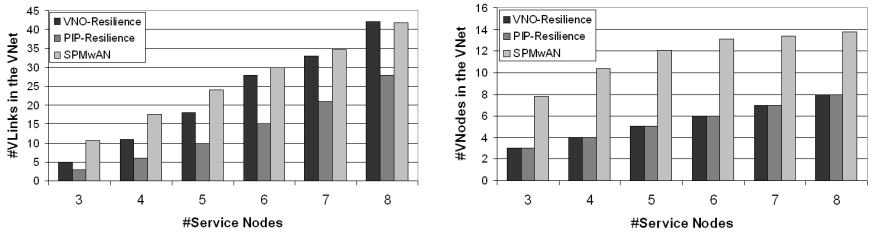


(d) % of failed VNet designs with SPM

Fig. 1: VNet design performance comparisons for VNO-Resilience, PIP-Resilience and Shortest Path Mapping (SPM)

dividing it to the VNO-Resilience value. In cases A-F, cost optimization is used and the cost factors for link and node costs are varied. The link cost can be defined as a fixed value or might depend on the length of the physical path it is mapped on. In the latter case, we use the flag "length." The cases are defined as a quadruples; {the fixed link setup cost, the capacity dependent link setup cost, the fixed node setup cost, the capacity dependent node setup cost}. The cases are defined as A={length,length,1,1}, B={length,length,2000,2000}, C={1,1,1,1}, D={1,100,1,1}, E={100,1,1,1} and F={1,1,100,100}. The cases are chosen to investigate the effect of each individual cost component in case of fixed and length-dependent cost factors. In case B, the node cost factor is taken as 2000, which is a value in the range of average virtual length link for the used test network. Note that the length corresponds to the total physical length of the virtual link, i.e. in VNO-Resilience it is the length of the single physical path and in PIP-Resilience it is the sum of the lengths of the two disjoint physical paths for each virtual link. Hence, the protected virtual links are in general more expensive than the unprotected ones. Similarly, for fixed link cost values we introduce a resilience cost factor for PIP-Resilience. Its value is taken as 2 for the simulations. Finally, in cases C-F we investigate the effect of the cost component with the weight 100, where the rest is kept minimum.

Cases B,C and E show that when the node cost is in the range of the link cost or higher, VNO-Resilience results in higher VNet cost compared to PIP-



(a) Comparison of the number of virtual nodes

(b) Comparison of the number of virtual links

Fig. 2: VNet design performance comparisons for VNO-Resilience, PIP-Resilience and Shortest Path Mapping with Additional Nodes

Resilience. This effect is caused by the higher virtual node capacity usage in VNO-Resilience due to the two-paths routing in the VNet. In cases A and B, the cost of the link depends on the physical length of the link and hence cost optimization is aligned with delay optimization. In these cases, VNO-Resilience results in 20% lower delay than PIP-Resilience. Wherever the delay optimization function is used, the VNO and PIP-Resilience result in comparable delay and cost values. However, if we compare the results of delay optimization and cost optimization for VNO-Resilience with the same cost factors as in case A, it is observed that the delay minimization option results always in lower delay but higher VNet cost. Increasing the number of the service nodes, decreases the delay difference of the two optimization functions but increases the cost difference as shown in Fig.1c. Hence, the appropriate optimization function should be chosen according to both the number of service nodes and the cost factors.

In the remainder of the results the cost factors are always taken as in case A. For cost optimization the delay and cost differences of PIP and VNO-Resilience increases with increasing VNet size as shown in Fig.1b. As can be seen for larger VNets, a VNet with PIP-Resilience costs on average 35% more than a VNet with VNO-Resilience. Moreover, PIP-Resilience results in 45% higher VNet-latency than VNO-Resilience for these settings. These results are obtained for the NobelUS network.

In SPM direct shortest path mapping is used and hence it is not always possible to find disjoint paths to route the services and the design cannot be performed. Fig.1d shows the ratio of the VNet design tries, which failed to find a solution during the simulations. Note that with increasing VNet size, the probability to find a solution for SPM is increasing. However, for NobelEU network, even for 8-nodes VNets in over 70% of the tries, no solution could be found. Moreover, even if a solution is found for SPM, it always results in higher maximum delay compared to VNO-Resilience. This difference decreases with increasing VNet size but is still over 20% for 8-nodes VNet on the test network NobelUS.

SPMwAN results in comparable latency as the VNO-Resilience and solves the problem faced by SPM. However, firstly it less scalable for larger physical

networks. For our test networks, VNO and PIP-Resilience simulations find a solution in a time interval of seconds but for SPMwAN, the simulation lasts for several minutes or even hours. Second, the resulting VNet has more virtual links and nodes compared to the PIP and VNO-Resilience cases as shown in Fig.2a and 2b. The virtual link numbers of SPMwAN and VNO-Resilience come closer for higher service node numbers. However, SPMwAN always has a higher number of nodes independent of the VNet size. Hence, especially for a high node cost factor, the network cost is drastically higher for SPMwAN.

4 Enhanced Datacenter Connection Models

In this section, the DC connection models for VNO and PIP-Resilience cases are introduced. In both cases, the VNet is connected to one primary and one backup DC to serve all the cloud services within the VNet. The design aim of both models is providing resilience in presence of both network and DC failures.

4.1 VNO-Resilience

In VNO-Resilience, all elements of the DC connection model, namely the DCs and the links and nodes connecting them to the VNet, are chosen by the VNO. To provide resilience in the presence of DC failures, the two DCs should be located in geographically disjoint locations. In our model, we divide the physical network into *availability regions*, where a failure in one region does not affect any other region. While choosing the DCs, the region information of the DCs should be provided to the VNO to guarantee disjointness. Moreover, to provide network resilience in case of single link failures, we choose the three virtual links, namely the two connecting the DCs with the VNet and the one connecting the two DCs, all mutually disjoint. Hence, the physical disjointness information of the available virtual links should be also provided to the VNO by the PIPs.

Fig.3a shows the DC connection model for VNO-Resilience. In normal operation, the services are routed via the link l_p to the *Virtual Machine* (VM) located in the primary DC. In case of a failure in the primary DC, the services will be rerouted to the backup DC using the second connection node and the backup link l_b. Similarly, in case of a failure on l_p, the traffic is rerouted on l_b to the backup DC. The link between the two DCs, l_c, is established for synchronization, data migration and failure routing purposes. In the special case, where both l_p and the backup DC fail simultaneously, the primary DC can be still reached using the path l_b and l_c. This case will be referred to as the *worst-case scenario*. Note that, the VNO can choose the two DCs from the same or different dcPIP(s) to optimize the performance of the cloud services in terms of latency.

The VNO-Resilience model becomes non-scalable with increasing number of DCs due to the large number of possible DC-connection node combinations. Therefore, we introduce a heuristic, where the primary DC and its connection node, node 1, are chosen first according to the maximum end-to-end delay it provides. However, the path l_p is not fixed but rather a candidate path list

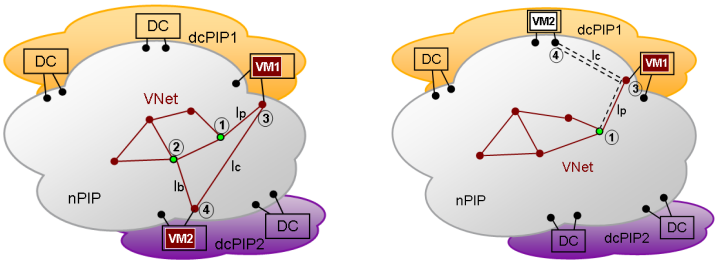


Fig. 3: Datacenter Connection Models: (a) VNO-Resilience, (b) PIP-Resilience

is created holding the k-shortest paths between the primary DC and node 1. Afterwards, the backup DC and its connection node, node 2, are chosen to minimize the end-to-end delay considering both the VNet delay and the routing on l_p, l_b and l_c, where all of these links are mutually physically disjoint. The end-to-end delay performance difference of the optimal case and the heuristic remain in $\pm 5\%$ interval for the NobelUS network with different DC and dcPIP settings and it is hence negligible.

4.2 PIP-Resilience

In the PIP-Resilience case, as shown in Fig.3b, the connection to the VNet is established over one virtual link. The primary DC with the connection link, l_p, is chosen by the VNO. In this model, providing resilience is the responsibility of the PIPs. Therefore, l_p is mapped in the physical network on two disjoint paths. Moreover, the dcPIP, owning the primary DC is responsible to provide resilience against DC failures and, thus, provides the connection to a DC it chooses from its own domain, which serves as the backup DC. Note that this connection link, l_c, has to be as well mapped onto two disjoint physical paths. This link might be owned directly by the dcPIP if it has relevant resources or has to be leased from an nPIP to connect the two DCs. In PIP-Resilience model, l_c and the backup DC are not visible to the VNO and all the recovery actions taken in case of any network or DC failure are transparent to the VNO. In case of a DC failure, the services are redirected to the backup DC and in case of a network failure, the protection paths are used to route the services in the physical layer.

For PIP-Resilience model to be realizable, each dcPIP has to own at least two DCs, which are located in geographically disjoint locations. For a dcPIP having a single DC, providing resilience is impossible. Moreover, the choice of the backup DC would depend on internal strategy of the dcPIP and on the contract with the VNO. The internal strategy of a dcPIP might be e.g. load balancing among its DCs or providing the highest performance for the services. Finally, on the one hand, for VNO-Resilience, the number of the virtual links and nodes, which have to be established and maintained is higher than the PIP-Resilience as shown in Fig.3a and 3b. On the other hand, for PIP-Resilience the virtual network links

have to be mapped on two physically disjoint paths, where for VNO-Resilience single path mapping is sufficient.

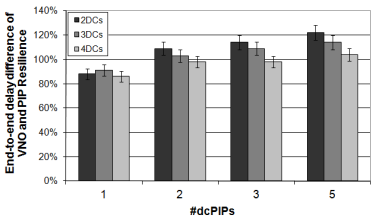
4.3 Delay Performance Evaluation of the End-To-End System

The end-to-end maximum delay performance is evaluated by combining corresponding VNet designs with the DC connection models. We compare the delay performance of the models in terms of VNet size, number of available DCs, number of different dcPIP domains, location of the DCs, different DC connection model preferences and different failure cases. The DCs can be placed either randomly, with the option "random", or to obtain maximum distance between them, namely with the option "away." Note that for both cases, the DCs of a dcPIP are located in different availability regions. Finally, it is assumed that in PIP-Resilience, the dcPIP chooses the backup DC randomly from its domain. In the simulations random VNets are generated for the test networks with DCs located randomly on them. Note that the chosen test networks are realistic topologies covering large physical areas. This enables end-to-end resilience design even in case of disasters and makes the problem more interesting by possibly enabling having multiple PIPs. For each VNet and DC set, the cloud connections are designed using the two models and the maximum end-to-end latency observed in both cases is compared until the confidence level of 95% with $\pm 5\%$ confidence interval is reached for the result.

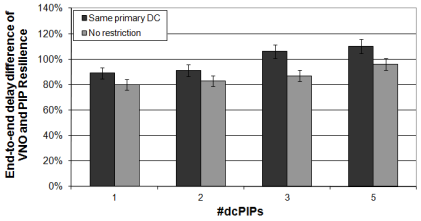
Fig.4a shows the effect of the number of the different dcPIP domains and number of DCs each domain possesses on the end-to-end maximum delay difference of PIP and VNO-Resilience. The results are obtained using 3-nodes VNets mapped on the NobelEU network with random DCs. For this simulation the DC location option is "away" and the same primary DC is used for PIP and VNO-Resilience. It is observed that the PIP-Resilience results always in higher end-to-end delay compared to VNO-Resilience and this difference increases with increasing number of dcPIPs. However, for a certain number of dcPIPs, increasing the number of DCs per dcPIP decreases the relative delay difference, since the dcPIPs' DC selection options increase as well.

The simulations performed with the NobelUS network show that if in PIP-Resilience the primary DC is selected freely to minimize the latency, the relative delay difference is decreased by 10% compared with the same primary DC selection scenario as shown in Fig.4b. Moreover, comparing Fig.4a and 4b, it is seen that a larger physical network results in higher relative delay difference. For NobelUS network, with 1 dcPIP and 2 DCs, the absolute maximum end-to-end round-trip delay of the PIP-Resilience is around 112 ms for random 3-nodes VNets. For NobelEU network, the maximum round-trip delay of 3-nodes VNets is 107 ms and of 5-nodes VNets 117 ms. However, the relative delay difference of the PIP and VNO-Resilience remains almost constant for different VNet sizes.

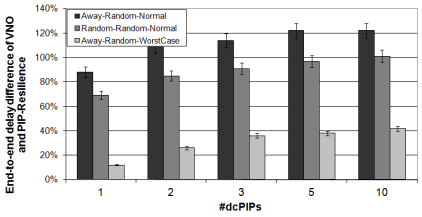
Finally, different DC location and protection options are compared using the NobelEU network and 3-nodes VNets as shown in Fig.4c. In all cases PIP-Resilience results in higher maximum delay compared to VNO-Resilience. This difference goes beyond 120% if more than 5 dcPIPs are available for the "away"



(a) Delay performance comparison for different # DCs and dcPIPs



(b) Effect of using the same primary DC



(c) Effect of random and worst-case options

Fig. 4: Performance comparisons of DC connection models

DC location option. When the DCs are placed randomly, the relative delay difference is reduced by around 20%. Finally, in worst-case scenario, the relative delay difference is drastically decreased and reaches 40% for 10 dcPIPs.

5 Conclusion

In this paper, we propose novel solutions for enabling the provisioning of cloud services with end-to-end availability and latency guarantees. The problem is separated into resilient Virtual Network (VNet) design and cloud connection design parts, where each are of different nature but enable together end-to-end resilient cloud services. First, we introduce two fundamental resilient VNet designs, one at the Virtual Network Operator (VNO) layer, namely VNO-Resilience, and the other one at the Physical Infrastructure Provider (PIP) layer, namely PIP-Resilience, in form of mixed-integer linear problems. We show that the proposed models outperform the models, where the mapping is done using shortest paths, in terms of efficiency and applicability. With direct shortest path mapping, in more than 80% of the cases no solution can be found for small VNets. Allowing additional virtual nodes solves this problem but results in relatively higher cost compared to the proposed models. Different cost factor values and optimization functions are discussed and it is shown how the model decision should be made according to the actual cost factor values and the cost vs. delay requirements.

In the second half of the paper, we introduce two DataCenter (DC) connection models for the designed VNets, which allow end-to-end reliability in presence

of DC and network failures. We perform an end-to-end maximum delay performance analysis and our simulation results show that the relative delay difference of the two models can reach 120% for the test networks and PIP-Resilience results always in higher end-to-end delay compared to VNO-Resilience.

In this paper, we obtain the end-to-end system by designing the VNets and cloud connections sequentially. As future work, the designs can be optimized together for both parts to achieve maximum efficiency and performance. Moreover, in the delay-optimization MILP, we minimize the total delay for all the services within the network. Another option would be minimizing the maximum delay. Finally, capacity constraints can be added to the MILP models.

References

1. Greenberg, A. et al.: The Cost of a Cloud: Research Problems in Data Center Networks. In: ACM SIGCOMM CCR, January 2009.
2. Meier, S., et al.: Provisioning and Operation of Virtual Networks. In: Electronic Communications of the EASST, vol. 37, Mar. 2011.
3. Chowdhury, M. K., Boutaba R.: A survey of network virtualization. In: Elsevier Computer Networks, 54(5), 2010.
4. Tutschku, K., et al.: Network virtualization: Implementation steps towards the future internet. In: KiVS, Kassel, Germany, March 2009.
5. Papadimitriou, P., et al.: Implementing network virtualization for a future internet. In: 20th ITC Specialist Seminar, Hoi An, Vietnam, May 2009.
6. Amazon AWS, Online: http://aws.amazon.com/directconnect/
7. Symantec, Virtualization and Evolution to the Cloud Survey, 2011.
8. Barla, I.B., Schupke, D.A., Carle, G.: Analysis of Resilience in Virtual Networks. In: 11th Wuŕzburg Workshop on IP: Joint ITG and Euro-NF Workshop "Visions of Future Generation Networks," August 2011.
9. Anderson, D., et al.: Resilient overlay networks. In: Proceedings of 18th ACM Symposium on Operating Systems Principles, 2001.
10. Li, Z., Mohapatra, P.: QRON: QoS-aware routing in overlay networks. In: IEEE Journal on Selected Areas in Communications, vol. 22, no. 1, pp. 2940, 2004.
11. Zhu, Y., Ammar, M.: Algorithms for Assigning Substrate Network Resources to Virtual Network Components. In: Proc. IEEE INFOCOM, Mar. 2006.
12. Chowdhury, N. M. M. K., Rahman, M. R., Boutaba, R.: Virtual Network Embedding with Coordinated Node and Link Mapping. In: IEEE INFOCOM, 2009.
13. Modiano, E., Narula-Tam, A.: Survivable lightpath routing: A new approach to the design of WDM-based networks. In: IEEE J. Selected Areas in Communications, vol. 20, pp. 800809, May 2002.
14. Rahman, M. R., Aib, I., Boutaba, R.: Survivable Virtual Network Embedding. In: Lecture Notes in Computer Science, 2010(4), April 2010:40-52.
15. Maliosz, M., Cinkler, T.: Configuration of Protected Virtual Private Networks. In: DRCN 2001, Budapest, 2001.
16. Kamel, M. et al.: Optimal topology design for overlay networks. In: Lectures Notes in Computer Science, Springer, Vol. 4479/2007, pp714725, 2007.
17. Koponen, T. et al: Onix: A Distributed Control Platform for Large-scale Production Networks. In: Proc. OSDI, October 2010.
18. Nobel US and Nobel EU Topologies, `http://sndlib.zib.de`