Optimal Design of Virtual Networks for Resilient Cloud Services

Isil Burcu Barla*†, Dominic A. Schupke*, Marco Hoffmann* and Georg Carle†
*Nokia Siemens Networks, Munich, Germany
†Munich Technical University, Munich, Germany
Emails: isil.barla.ext@nsn.com, schupke@ieee.org, marco.hoffmann@nsn.com, carle@in.tum.de

***Abstract*—Currently, accessing global cloud services has its limitations in end-to-end quality of experience (QoE) due to the independent operation of cloud infrastructures and communication networks. The novel virtual network design approaches for resilient cloud services that are presented in this paper can realize end-to-end availability and latency guarantees by combining the control of network and cloud resources. We formulate design models as linear optimization problems to realize resilience either at the virtual or the physical layer. On the basis of extensive simulations, we analyze the effect on the overall performance when cost factors and influential parameters of the network virtualization environment are varied. We compare the proposed models in detail and show that they outperform prior approaches. Finally, we provide a discussion about the implementation and applicability of the proposed models.**

***Index Terms*—Virtual network, resilience, cloud network, ILP, cross-stratum optimization.**

I. Introduction

Businesses and applications are more and more based on cloud technologies, with infrastructure, software and platform as a service being important service types. Hence, guaranteeing end-to-end quality of experience (QoE) for cloud services is of high importance, especially for business-critical applications. According to a survey conducted in 2011 with over 3700 companies worldwide, the primary concern of businesses adopting cloud services is reliability, and performance ranks third in the list of concerns [1]. Cloud providers therefore offer solutions to address these concerns. However, existing solutions focus on the performance and connectivity inside the cloud, thereby insufficiently addressing the communication networks, which can be an important cause of service impairments such as unacceptable latencies and outages. Offering end-to-end QoE guarantees for cloud services is difficult today, as communication networks and cloud domains are typically operated by different entities. Moreover, today the services are usually requested from a single cloud provider. In case of complete datacenter (DC) failures, depending on the geographical diversity and availability of the resources of the cloud provider, recovery of the services may cause long outages. A promising solution for these problems is the concept of *Network Virtualization* with combined control of network and IT resources. This enables an overall view on the available resources of various physical domains, thereby allowing optimized operation of cloud networks.

Network virtualization is considered as a key enabler for next generation networks and the future Internet [2], [3]. The difference to current virtualization techniques in networks like Virtual Private Networks (VPNs) and overlay networks is that network virtualization enables operation of isolated *Virtual Networks* (VNets). A VNet can have isolated network elements, links and IT resources. In a network virtualization environment new business roles can be established, which realize different tasks and trading virtual resources between them [4]. These resources can be network resources and/or IT resources. Hence, new control mechanisms and interfaces are necessary to realize the setup and operation of these heterogeneous VNets. There are already several suggestions in the literature for possible realizations of combined control of IT and network resources using virtualization [5], [6]. There are also some commercial offers from e.g. Amazon [7], where a VNet is deployed for the connectivity to the cloud although it still lacks resilience and end-to-end QoE guarantees. Hence, as a solution, we propose novel *virtual network design models* to enable optimal provisioning of cloud services with end-to-end availability and latency guarantees. In our network virtualization model, we define two main 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 as shown in Fig.1. As shown in the figure, there might be multiple PIPs and VNOs in the environment.

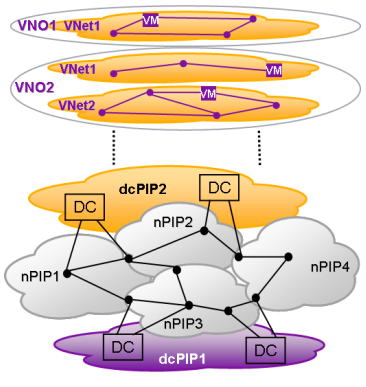


Fig. 1: Virtual Network Environment (VNO:Virtual Network Operator, VNet: Virtual Network, VM: Virtual Machine, PIP: Physical Infrastructure Provider, DC: Datacenter)

A PIP, as the owner of the physical infrastructure, is in the position to monitor all of its physical and virtual resources. It can own both physical network and IT resources, or only network resources. In the remainder of this paper, the former is referred to as "dcPIP", and the latter "nPIP".

A VNO can operate one or several VNets, which are mapped onto the physical infrastructure of one or more PIPs. A VNet can consist of both virtual network and cloud resources. Interfaces and information sharing between the VNO and the PIP depend on their business models, and the contract between them [8]. 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 end-to-end VNet for single or bundled service requests, taking into account the VNet cost and service latency. There can be two types of VNOs; one just renting a certain connectivity from the PIPs and not having any network operation functions and the second one renting a complete VNet, operating and managing it. For the former, the network optimization and resilience provisioning have to be realized in the physical domain. In this paper, we focus on the latter, which requires network management and operation functions and know-how available at the VNO, and which provides the opportunity to design a more cost and latency efficient VNet in certain cases.

There are two fundamental alternatives for providing resilience: providing resilience in the underlay by the PIP *(PIP-Resilience)*, or in the overlay by the VNO *(VNO-Resilience)*, each having its advantages and drawbacks. Since we focus on the evaluation of these advantages and drawbacks, hybrid models are out of the scope of this paper. We formulate models for these alternatives as mixed-integer linear problems (MILPs) and evaluate their performance via extensive simulations.

The remainder of the paper is organized as follows: Section II introduces the related work and Section III our resilient VNet models for cloud services over networks. In Section IV, we present the simulation framework with the used simulation parameters. Section V evaluates the results and presents a framework for the efficient utilization of our models by future VNOs and PIPs for various parameter and cost settings. In Section VI, we discuss the implementation and applicability of the proposed models. Finally, Section VII concludes the paper with a discussion of the main results and an outlook.

II. Related Work

A qualitative comparison of PIP-Resilience and VNO-Resilience is provided in [9]. In [10], we introduce a quantitative comparison and resilient VNet design models for unicast services, where the VNOs need to request additional cloud connectivity separately in case of cloud services, which might result in sub-optimal solutions. In this paper, we introduce new models, which provide a resilient VNet design for end-to-end cloud services. Our models offer combined optimization of the network and the cloud domains for individual services in case of delay or cost optimized VNet.

We design the VNets for a given set of services by optimizing the routing in the virtual layer, while simultaneously mapping the VNet onto the physical network. In contrast, related work treats optimization of the routing and of the mapping separately. In resilient overlay networks [11], [12], a fixed mapping of the virtual links onto the physical layer is assumed, and only service routing is optimized. References [13], [14] offer solutions for survivable VNet embedding, which consider that the VNet is already designed and given. Hence, optimal VNet design for the VNO is out of scope of these papers. Similarly, the literature available for optimal server selection and routing of anycast services in the physical layer for intra and inter-DC networks [15]-[17] lack the treatment of the resilient network design in the virtual layer. In this paper, we focus on the interface between the PIP and the VNO, which needs to establish an optimal and cost-efficient VNet to offer services to its customers using the resources provided by the PIPs. We show that our models outperform the approach of fixed mapping in terms of cost and applicability.

III. Resilient Virtual Network Design Models for Cloud Services

This section introduces the models for VNO-Resilience and PIP-Resilience. In both models, each anycast demand from a service source node to the cloud is routed to k servers, where a single one is operational at a time (k-1:1 redundancy). The primary and disaster recovery (DR) sites are chosen per service from the set of all available and suitable DCs. The DCs are modeled by their network connection points. Our resilient virtual network design models offer resilience in presence of both DC and network failures.

Now we first introduce the general virtual network design model for cloud services, which is the basis for both VNO-Resilience and PIP-Resilience. Then, we explain the differences of the two models. In the following a list of the sets, parameters and variables used in both models is provided.

- *Sets:*
  - S: Set of the service nodes
  - C: Set of the DC connection nodes
  - V: Set of all the virtual nodes with $S \cup C = V$ and $S \cap C = \{\}$
  - L: Set of the virtual link candidates, where there is at least one link between all node pairs in S and from each node in S to all nodes in C
  - D: Set of the all possible unicast realizations of the requested anycast services, where $|D| = |S| \cdot |C|$ and $d = (s, c) \in D$ with $s \in S$ and $c \in C$
  - D_s: Set of the all possible unicast realizations of the requested anycast service having the source node $s \in S$ with $|D_s| = |C|$ and $D_s \subseteq D$
  - E_l: Set of the endpoints of a virtual link $l \in L$
  - Z: Set of virtual link pairs $(l, k) \in L^2$, which are not disjoint
  - E: Set of the edges in the physical network topology
  - N: Set of the nodes in the physical network topology
  - P_l: Set of the physical edges $e \in E$, on which the virtual link $l \in L$ is mapped

- R: Set of DC connection node pairs $(c_1, c_2) \in C^2$ with $c_1 \neq c_2$, which are located in the same availability region of the physical topology

- *Parameters:*
  - k: Number of the DCs, which will be selected for each anycast service with $k \in \{1, ..., |C|\}$
  - b_d: Requested bandwidth for the service $d \in D$
  - n_d: Requested network node resources for the service $d \in D$
  - r_d: Requested server resources for the service $d \in D$
  - t_l: Physical length of the virtual link $l \in L$
  - λ_l: Fixed setup cost for having a new virtual link $l \in L$ in the VNet
  - θ_l: Setup cost per unit capacity of a virtual link $l \in L$
  - μ_v: Fixed setup cost for having a virtual network node $v \in V$ in the VNet
  - η_v: Setup cost per unit capacity of a virtual network node $v \in V$
  - ϕ_c: Fixed setup cost for having a new virtual machine in the VNet, which is connected to node $c \in C$
  - φ_c: Setup cost per unit capacity of a virtual machine connected to node $c \in C$
- *Variables:*
  - $a_{s,c}$: Binary variable taking the value of 1 if a virtual machine is placed into the DC connected to node $c \in C$ to satisfy the anycast demand with source $s \in S$, 0 otherwise
  - $\beta_{d,l}$: Binary variable taking the value of 1 if the link $l \in L$ is used for the demand $d \in D$ and if demand $d = (s, c)$ is chosen as one of the realizations of the anycast service with source $s \in S$, 0 otherwise
  - $\delta_{d,v}$: Binary variable taking the value of 1 if the node $v \in V$ is used for the demand $d \in D$ and if demand $d = (s, c)$ is chosen as one of the realizations of the anycast service with source $s \in S$, 0 otherwise
  - γ_l: Binary variable taking the value of 1 if the link $l \in L$ is included to the VNet, 0 otherwise
  - α_v: Binary variable taking the value of 1 if the node $v \in V$ is included to the VNet, 0 otherwise
  - y_c: Binary variable taking the value of 1 if a virtual machine on the DC connected to node $c \in C$ is included to the VNet, 0 otherwise
  - u_l: Used capacity on link $l \in L$ with $u_l \in [0, \infty)$
  - ω_v: Used capacity on node $v \in V$ with $\omega_v \in [0, \infty)$
  - z_c: Used capacity on DC connected to node $c \in C$ with $z_c \in [0, \infty)$

The MILP takes as input (i) the undirected physical network graph, (ii) available DCs, (iii) DC connection nodes, and (iv) a set of virtual links and nodes to generate an optimized VNet with the routing of the given set of services. It maps the services onto a VNet multigraph $G_l = (V, L)$ that in turn is mapped onto a physical simple graph $G = (N, E)$. Nodes in V are mapped onto a subset of nodes in N. The set of virtual nodes is the union of the service nodes and the DC connection nodes. The initial virtual link set is generated by establishing a virtual link between all service node pairs and links from each service node to all DC connection nodes. To maintain linearity, instead of using one virtual link with different mappings, we generate a separate virtual link between the same end-nodes for each mapping and include it to L. In the simulations up to 20 mappings are considered per virtual link.

The main constraints used in the virtual network design model for cloud services are given in the following. Eq. (1) ensures that $k \in \{1, ..., |C|\}$ server locations are chosen for an anycast service with source s. $k = 1$ means that there is no DC resilience in the virtual layer. Increasing k increases the level of protection. Eq. (2) is the link-flow constraint, and (3) ensures that a node is flagged as "used" for a service if it is the source or the target of that service and if it is chosen as a realization of the anycast service with source s. Eq. (4), (5) and (6) state that if a virtual link, node or virtual machine (VM) carries the traffic of any service, it is part of the resulting VNet, otherwise not. Additionally, (7), (8) and (9) provide upper bounds for γ_l, α_v and y_c, respectively, which ensures that a virtual link, node or VM to be part of the resulting VNet only if it is actually used. These bounds are only necessary for calculating the VNet cost in delay optimization to obtain meaningful cost values but do not restrict the optimality. Finally, (10), (11) and (12) are the constraints for calculating the required virtual link, node and VM capacities, respectively.

We omit the inclusion of unicast services in this paper since we focus on the combined optimization of network and IT resources. However, unicast service requests can be easily included into the model by extending the service set and by adding the unicast flow constraint as shown in [10].

$$\sum_{c \in C} a_{s,c} = k \quad \forall s \in S \tag{1}$$

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

$$\delta_{d,v} = a_{s,c} \quad \forall d = (s, c) \in D,\ v \in \{s, c\} \tag{3}$$

$$\gamma_l \geq \beta_{d,l} \quad \forall l \in L,\ d \in D \tag{4}$$

$$\alpha_v \geq \delta_{d,v} \quad \forall v \in V,\ d \in D \tag{5}$$

$$y_c \geq a_{s,c} \quad \forall c \in C,\ s \in S \tag{6}$$

$$\gamma_l \leq \sum_{d \in D} \beta_{d,l} \quad \forall l \in L \tag{7}$$

$$\alpha_v \leq \sum_{d \in D} \delta_{d,v} \quad \forall v \in V \tag{8}$$

$$y_c \leq \sum_{s \in S} a_{s,c} \quad \forall c \in C \tag{9}$$

$$u_l \geq \sum_{d \in D} \beta_{d,l}\, b_d \quad \forall l \in L \tag{10}$$

$$\omega_v \geq \sum_{d \in D} \delta_{d,v}\, n_d \quad \forall v \in V \tag{11}$$

$$z_c \geq \sum_{s \in S} a_{s,c}\, r_d \quad \forall c \in C \text{ with } d = (s, c) \tag{12}$$

There are two objective functions defined for different optimization objectives, namely VNet cost minimization and propagation delay minimization. The cost of the VNet consists of link cost, network node cost and VM cost as given in (13), (14) and (15), respectively. Each of these costs has two parts, namely the fixed setup cost for having a new link, node or VM in the VNet and the capacity dependent cost depending on the requested capacity of a link, node or VM. For sufficiently simple PIP-VNO business relationships, a linear cost model is assumed. For cost minimization, the overall cost of the VNet is minimized. For propagation delay minimization, the total length of the routes for each service is minimized. Assuming that the network is designed for normal load conditions, we only consider the propagation delay of the physical routes as latency metric for a service. The two objective functions for cost minimization and delay minimization are provided in (16) and (17), respectively.

$$\varepsilon_l = \lambda_l \gamma_l + \theta_l u_l \quad \forall l \in L \tag{13}$$

$$\varepsilon_v = \mu_v \alpha_v + \eta_v \omega_v \quad \forall v \in V \tag{14}$$

$$\varepsilon_c = \phi_c y_c + \varphi_c z_c \quad \forall c \in C \tag{15}$$

$$\min\ \varepsilon,\ \varepsilon = \sum_{l \in L} \varepsilon_l + \sum_{v \in V} \varepsilon_v + \sum_{c \in C} \varepsilon_c \tag{16}$$

$$\min \quad \sum_{d \in D} \sum_{l \in L} \beta_{d,l}\, t_l \tag{17}$$

A. VNO-Resilience

In VNO-Resilience, the VNet is designed for a given set of services, which are routed in the virtual layer to k different server locations. We assume $k = 2$ as a typical number. Both the VMs and the paths leading to the VMs have to be physically disjoint, such that in case of a failure at the primary site, the DR site can be used by re-routing the service inside the VNet. Hence, we need to add the diversity constraints for these paths and VMs to the model. (18) and (19) are the diversity constraints for link and node-diversity respectively for the connection paths. Additionally, in case of node-diversity, node-disjointness in the physical layer has to be ensured via the set Z. Moreover, (20) ensures that the primary and backup sites are located in different availability regions. The diversity constraints can be easily extended for multiple and regional failures by generating the set Z accordingly.

$$\beta_{d_1,l} + \beta_{d_2,k} \leq 1 \quad \forall s \in S,\ (d_1, d_2) \in D_s^2,\ (l, k) \in Z \tag{18}$$

$$\delta_{d_1,v_1} + \delta_{d_2,v_2} \leq 1 \quad \forall s \in S,\ (d_1, d_2) \in D_s^2,\ (v_1, v_2) \in (V \setminus \{s\})^2 \tag{19}$$

$$a_{s,c_1} + a_{s,c_2} \leq 1 \quad \forall s \in S,\ (c_1, c_2) \in R \tag{20}$$

Fig.2a shows the realization of VNO-Resilience for a single service. For both primary and DR VMs, the connection nodes of the corresponding DCs as well as the paths connecting them to the source node of the anycast service, e_p and e_r, are part of the VNet. The paths e_p and e_r can be composed of multiple virtual links and nodes and they have to be physically disjoint.

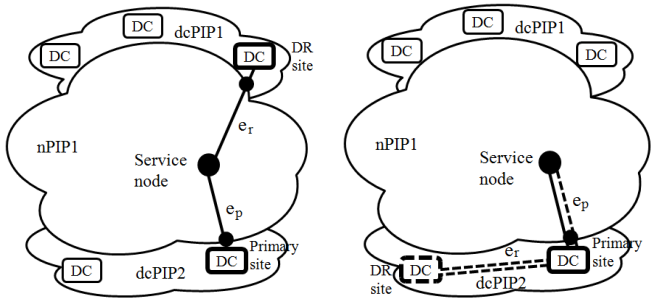
B. PIP-Resilience

In the case of PIP-Resilience, providing resilience is the responsibility of the PIPs. The services are routed on a single path in the virtual layer to the primary server site, i.e. k is equal to 1. This virtual path is protected by the corresponding nPIP(s), where each virtual link has a 1:1 protection mapping on the physical layer. Since 1:1 protected path mapping is provided as an input to the MILP, diversity constraints are unnecessary. Moreover, the dcPIP owning the primary site is responsible for providing DC resilience. The realization of PIP-Resilience for a single service is shown in Fig.2b. Similarly to the VNO-Resilience case, a single DR site is assumed. From the VNO perspective, only the connection path to the primary site, the network connection node of the primary site and the primary site VM are visible. Upon failure of the primary site, the service is re-routed to the DR site in the physical layer, where the VNet and the routing of the services in the VNet remain unchanged.

In PIP-Resilience, the DR site(s) for each server candidate and their resilient physical connection path(s) are pre-calculated. This information is incorporated in the fixed cost factor of the corresponding primary server. Different strategies can be used to select a DR server site. It can be chosen randomly, or such that it offers the shortest interconnection path. These strategies are referred to as *Random DC selection* and as *Shortest delay strategy* in the remainder of this paper. Once the DR site is chosen, their interconnection path is calculated as the shortest disjoint path pair connecting the two sites. Finally, the physical length t_l of a virtual link $l \in L$ is set as the total length of the physical working and protection paths of l.

IV. Simulation Framework and Parameters

We performed simulations to provide insights into the applicability and efficiency of the proposed models compared with



(a) VNO-Resilience: Each service node is connected in the virtual layer to two datacenter (DC) sites of two different PIPs (a single PIP is also possible).

(b) PIP-Resilience: Each service node is connected to a single DC site in the virtual layer. Resilience is internally provided per PIP.

Fig. 2: Proposed resilience models with disaster recovery (DR) sites

prior approaches and finally to present a quantitative analysis for the effect of different parameters and cost factors on the performance of the models. In this section the simulation framework will be introduced and different parameter settings will be presented. The simulations are performed using a virtual network simulation tool developed in Java. The optimization problems are implemented using the IBM Concert library and solved with CPLEX 12.3. The resulting optimal VNets for different settings are then simulated to determine the maximum propagation delay, which can be guaranteed for that VNet, and the VNet cost. The simulation results are within a $\pm 5\%$ confidence interval at a confidence level of 95%.

We compare the performance of our models with two models where resilient routing is provided in the VNet and the virtual link mapping is fixed as the shortest path mapping. The first model, namely the Shortest Path Mapping (SPM) model, uses the set of service nodes and DC connection nodes as the initial virtual node set like in VNO-Resilience and PIP-Resilience. For the second model, namely the SPM with Additional Nodes (SPMwAN) model, we use an extended initial virtual node set, in which a virtual node corresponding to each physical node is included. Thus, the optimal VNet can include some virtual nodes, which are not used for services but just for routing purposes.

The simulations are performed using two physical network topologies, namely the NobelUS and NobelEU networks [18]. NobelUS has 14 nodes and 21 edges. NobelEU has 28 nodes and 41 edges. At each simulation run, for the given physical network topology, the DCs are placed in the network. The simulator takes as input the number of dcPIPs, number of DCs per PIP and the DC placement strategy, which places the DCs randomly or as far as possible from each other. For both cases, we divide the physical topology map into equal-sized rectangular availability regions. A failure in one region is assumed not to affect the other regions when the size of the regions is adjusted accordingly [19]. Availability regions enable DC resilience against natural disasters like hurricanes, tsunamis, earthquakes, floods etc, where DCs of a single dcPIP are placed such that each DC is in a different availability region. We used 12 regions for the NobelEU and 6 for the NobelUS topology.

For the given physical network and selected DC locations, we generate random anycast service requests, where the number of services is given as an input parameter, and the service nodes are chosen randomly from the physical topology. Then an optimal resilient VNet is calculated according to the selected resilience method. Note that for NobelUS, the worst-case duration of solving the MILP is around 1 minute for VNO-Resilience and 0.2 seconds for PIP-Resilience. Depending on the simulation aim, the corresponding value, e.g. cost of the optimal VNet or the maximum delay occurring in the VNet is computed. We continue to generate random services and run the models until a required confidence level is reached for the mean of the result values. This mean value corresponds to the used DC set. Afterwards, the same loop is repeated for a new random DC set, until the required confidence level is reached for the results of different DC sets.

TABLE I: Cost settings used in the simulations

Cost setting	Link Cost	Node Cost	VM Cost
(L,1,1)	Length	1	1
(1,1,1)	1	1	1
(L,A,A)	Length	Average	Average
(1,1,A)	1	1	Average
(1,A,1)	1	Average	1

In the remainder of this section we provide the necessary settings and formulations of VNet cost with VNO-Resilience and PIP-Resilience. As given in (13)-(15), the cost of a VNet has three parts. Each of them consists of the fixed cost of placing VNet components and the capacity-related cost depending on the size of the components. We used five cost settings, which are listed in Table I.

"Length" means that the resulting physical length of the virtual link in kilometers, t_l, is used as the cost factor instead of a fixed value. In case of 1, the cost is constant per type of links, nodes and VMs. "Average" means that the average shortest path length of the topology is used as the cost factor. Cost setting (L,1,1), (1,1,A) and (1,A,1) are used to evaluate the effect of the dominance of each cost factor. In setting (1,1,1), all cost factors are equal and in (L,A,A) comparable to each other. The difference of (L,A,A) compared with (1,1,1) is that the link cost depends again on the physical path length. Hence, these cost settings provide a complete list for all possible cost factor options.

In VNO-Resilience these cost factors are directly used. However, in PIP-Resilience the resilience cost needs to be included to the cost of the links and VMs. For virtual nodes no resilience is provided, and hence, the cost of the nodes remains unchanged. In PIP-Resilience, if t_l is used as the cost factor for the links, resilience cost is implicitly included, since t_l is the total length of the primary and backup path mappings for l. However, if a fixed value is used like in settings (1,1,1), (1,1,A) and (1,A,1), the additional cost of providing resilience at the physical layer should be included in the cost of a virtual link by introducing a resilience premium r as given in (21).

$$\varepsilon_{l,PIP,Fixed} = (\lambda_l \gamma_l + \theta_l u_l) r \tag{21}$$

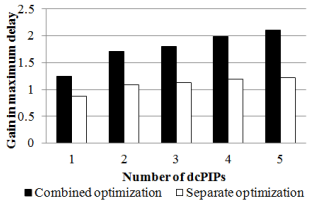
If DC resilience is provided by the PIP, the cost of resilience consists of the second DC site usage and the cost of the physical paths connecting the two sites. The fixed cost remains the same since neither the second VM nor the connection path is part of the VNet. The capacity dependent cost of the resilient VM is given in (22) and (23) for the length-dependent link cost and fixed link cost cases, respectively.

$$\varphi_{c,PIP,Length} = 2\varphi_c + a_p \tag{22}$$

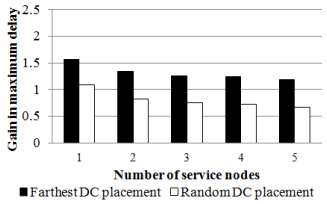
$$\varphi_{c,PIP,Fixed} = 2\varphi_c + \lambda r \tag{23}$$

V. Simulation Results

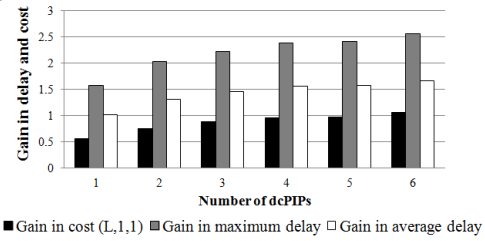
In this section, the simulation results are presented. First, the proposed models are compared with prior approaches and sep-



(a) Combined vs. separate optimization over number of dcPIPs (2 DCs each, farthest DCs placement, 3 service nodes)



(b) Farthest vs. random DC placement over number of service nodes (1 dcPIP, 2 DCs each)



(c) Effect of varying the number of dcPIPs (2DCs each, farthest DC placement, 1 service node)

Fig. 3: Gain of VNO-Resilience over PIP-Resilience ($= \frac{x_{\text{PIP-Resilience}} - x_{\text{VNO-Resilience}}}{x_{\text{VNO-Resilience}}}$) for different models and settings

arate optimization. Then, the proposed models' performance is evaluated under different parameters and cost factors.

We compare the performance of our VNO-Resilience model with prior approaches. Our simulations show that in around 50% of the simulation runs SPM fails to find a resilient VNet solution, while this value is only 0.02% for VNO-Resilience. These simulations are performed for 1 dcPIP and 2 DC located randomly and varying the number of the service nodes between 1 and 10. For these simulations and for the delay gain simulations presented in the remainder of this section, the NobelEU topology is used, and delay optimization is applied.

If using additional nodes is allowed for SPM, namely using SPMwAN, the simulations show that on average the resulting VNets include a higher number of virtual links and nodes compared with VNO-Resilience, where the difference is around 45% for the number of virtual links, and 40% for the number of virtual nodes. Hence, allowing additional nodes enables the prior approach to find solutions. This however increases the setup and maintenance costs of the VNet significantly.

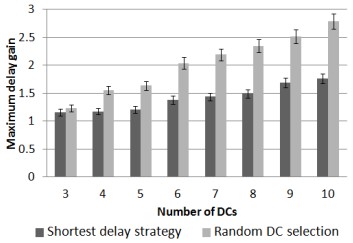
Fig.3a compares the gain of VNO-Resilience using the proposed combined optimization models with using the separate optimization models introduced in [10]. In the latter, the VNet design is optimized for unicast services, and then the VNet is connected to one or two DC sites by adding virtual links to it to minimize the delay to the cloud. In the former, the VNet design optimizes network and cloud resources in combination, and it works on the service level for DC selection. Hence, combined optimization chooses a primary and a DR site per service, whereas separate optimization uses the same primary and DR sites for all services. The maximum delay, which is an important performance parameter besides the average delay performance for certain applications, occurring in the VNet for PIP-Resilience and VNO-Resilience is compared using the two different optimization approaches. The maximum delay is decreased for both models with combined optimization, while for VNO-Resilience this delay gain is around 50% for 5 dcPIPs and for PIP-Resilience less than 30% for the used settings. Hence, combined optimization increases the maximum delay gain compared with separate optimization as shown in Fig.3a. With 5 dcPIPs, PIP-Resilience results in 80 ms maximum round-trip delay with combined optimization only due to the propagation and it is reduced to 25ms for VNO-Resilience.

Fig.3b shows the effect of placing the DCs into the regions randomly vs. choosing the farthest regions on the delay gain for the maximum delay, which can be guaranteed in VNO-Resilience and PIP-Resilience. The reason for farthest DC placement would be for the dcPIP to have access to different parts of the physical topology and to increase DC resilience. Simulation results show that the farthest DC placement of the nodes increases the maximum delay values for PIP-Resilience around 30% and for VNO-Resilience around 5-10%. Hence, the delay difference between the PIP-Resilience and VNO-Resilience is increased with farthest DC placement as shown in Fig.3b. Therefore, if the dcPIPs want to offer resilience for delay sensitive services, they should consider placing their DCs in a more random fashion. Moreover, it is also shown that the delay gain decreases with increasing number of service nodes. Hence, the delay gain is a more important decision parameter for smaller VNets.

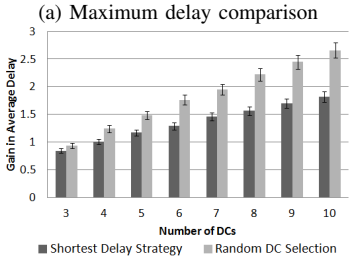
Fig.3c shows the effect of the number of the dcPIPs in the network on the cost, maximum and average delay gain of VNO-Resilience compared with PIP-Resilience. Increasing the number of dcPIPs increases the number of the DC options for VNO-Resilience and for the primary site choice in PIP-Resilience without affecting the choice of the DR site for PIP-Resilience. Thus, the gain in average delay is expected to grow with increasing number of dcPIPs. The simulation results validate this conclusion. Moreover, it is shown that the gain in maximum delay, which can be guaranteed in the VNet, reaches 200% already for 2 dcPIPs and increases with increasing number of dcPIPs, reaching 250% for 6 dcPIPs.

The effect of the DR site selection strategy on the maximum and average delay gain is presented in Fig.4a and 4b, respectively. Simulations performed for maximum and average delay gain show a similar trend. For both cases, the delay gain of the VNO-Resilience is reduced by almost 50% for 5 DCs and by 75% for 10 DCs using Shortest Delay Strategy instead of Random DC Selection. Hence, from the point of view of the dcPIPs, the choice of this strategy affects the delay performance of PIP-Resilience drastically. Therefore, shortest delay strategy should be preferred for delay-sensitive services to increase the competitiveness of the PIP-Resilience offer.

The trend of the cost and delay gain is similar for cost setting (L,1,1) as shown in Fig.3c. This is due to the fact that in cost setting (L,1,1), the virtual link cost is the dominant factor, which depends on the physical length of the virtual links. Hence, optimizing for cost setting (L,1,1) is aligned with delay optimization, and for this cost setting VNO-Resilience results always in cheaper VNets compared with PIP-Resilience.



(a) Maximum delay comparison



(b) Average delay comparison

Fig. 4: Delay gain of VNO-Resilience over PIP-Resilience ($= \frac{x_{\mathrm{PIP-Resilience}} - x_{\mathrm{VNO-Resilience}}}{x_{\mathrm{VNO-Resilience}}}$) for the Shortest Delay and Random DC Selection Strategies

Using cost setting (1,1,1), the cost gain depends on the choice of the resilience premium. For the simulations, the resilience premium is taken as 2. Note that for cost settings (1,1,1), (1,1,A) and (1,A,1) using cost optimization, the delay is not minimized and takes a random value depending on the selected DC site. Hence, if there are many DCs available, cost optimization for these cost settings might result in much higher delay values compared with delay optimization. If the service delay is important, delay constraints should be added to the MILP by cost optimization. This results in similar VNet costs with acceptable delay characteristics.

Cost setting (L,A,A) results in comparable cost values for PIP-Resilience and VNO-Resilience, while PIP-Resilience has slightly lower values. In this cost setting, again the physical length of the virtual links is used as the link cost factor and, hence, it also optimizes for the delay implicitly. For cost setting (1,1,A), PIP-Resilience results in lower VNet cost compared with VNO-Resilience due to the higher number of VMs involved in VNO-Resilience. The cost gain of PIP-Resilience decreases with increasing number of service nodes since it causes the capacity-dependent cost of the VM to be the dominant factor compared with its initial setup cost. For one service node there is a difference of 30%, which decreases to 8% for 5 service nodes according to our simulations performed with 1 dcPIP and 2 DCs, NobelUS topology and cost optimization. For cost setting (1,A,1), where the virtual node cost is the dominant factor, PIP-Resilience always results in cheaper VNets since the number of virtual nodes used in VNO-Resilience and their capacity is much higher compared with PIP-Resilience. The difference in cost lies at around 40%.

VI. Implementation and Applicability

In this paper, two novel resilient VNet design models for cloud services are introduced, which are modeled as mixed-integer linear problems. We evaluated their performance in terms of VNet setup cost and service latency under different parameters in the previous section. In this section, we will discuss the implementation possibilities and applicability of the proposed models.

The introduced models are abstracting the VNet design from the underlying technology, and hence, they are technology independent. These models can be implemented in systems using, e.g., IP over WDM, GMPLS or OpenFlow. New interfaces are required that support the needed information exchange and control between the different roles. Regarding the interfaces, a virtual network architecture as proposed in [8] can be used. The information exchange level depends on the business model of the VNO and the PIP and on their contract. It can be expected that PIPs refrain from sharing detailed physical topological information with VNOs, while VNOs need a certain level of information in order to be able to design VNets. Our proposal for an appropriate level of information exchange for both roles consists of the availability information of the virtual link candidates, the virtual nodes adjacent to the virtual link candidates, optimization related properties of the virtual link candidates and the disjointness information.

The availability information of the virtual link candidates contains the available bandwidth if bandwidth constraints are applied. Similarly, the amount of node resources can be shared in presence of node resource constraints. The connectivity information of the virtual link candidates is sufficient for building the virtual network and routing the services. If the services need to be transmitted to/from a certain location, this can be ensured by specifying the node location while requesting virtual nodes. In our example optimization related information of the virtual links is the cost and the delay information. The cost of each virtual network element candidate should be specified by the PIP to the VNO. Additionally, end-to-end delay information of each virtual link can be made available to the VNO without giving the actual physical mapping. In the simulations, we considered only the propagation delay of the virtual links, however, the model can be directly applied for end-to-end delay calculation if this information is available from the PIP. Finally, the disjointness information of the virtual link candidates should be given to the VNO such that it can provision resilience in its VNet. This can be realized by defining Shared Risk Link Groups (SRLGs) containing the virtual link candidates sharing the same failure risk. If e.g. physical edge disjointness is requested, all virtual links sharing the same physical edge are grouped since they would all be affected by the failure of this physical edge. Similarly, SRLGs for node or sub-network disjointness can be formed. Another option is building a set of virtual link pairs, which are not physically disjoint, and providing this information to the VNO as modeled in the MILPs. Using this information the VNOs can then form their VNets using the proposed MILPs. As seen from the simulation results the proposed models can be applied in realistic physical topologies for various scenarios. Moreover, even though only the results for NobelUS and

NobelEU topologies are presented, we expect the complexity of the problem not to be affected by larger networks if the same number of paths between each node pair are used in the model. Thus, the models are applicable for any kind of physical topologies.

As discussed above, in VNO-Resilience, either the VNO should have knowledge about the disjointness properties of the physical equipment, or this information should be signaled on request from the PIP to the VNO depending on their interface. Similarly, PIP-Resilience might involve communication and information exchange between peer PIPs if the paths need to span multiple PIP domains. In this case, either the PIPs have to coordinate the resilience design among each other, or a third party can be used to combine the resources of the PIPs and lease a resilient VNet to the VNO.

Another important point is choosing the location of the physical DCs. As mentioned in Section IV, the distance of the primary and backup sites should be decided according to the fault and disaster types, against which the PIPs want to provide protection. In certain cases placing the servers in different buildings might be sufficient, whereas recovery in case of natural disasters like hurricanes, tsunamis, earthquakes, floods etc. would require larger physical distances and possibly different networks. Moreover, the state synchronization strategy for the primary and DR sites should be decided on according to the service specific needs [20]. Depending on the required failover time of the requested services and the physical distances between the two DC sites either shared systems can be used where the load is shared on both servers during normal operation or standby systems where the traffic is redirected to the backup site only in case of failure.

VII. Conclusion and Outlook

In this paper, we propose two novel resilient virtual network design models for cloud services providing combined optimization of communication networks and datacenter (DC) resources. Resilience in cloud services will increase the end-to-end quality of experience (QoE) and the business value for operators and service providers by enabling them to offer higher quality services. In the first model, called PIP-Resilience, resilience is provided in the physical layer by the physical infrastructure providers (PIPs). In the second model, called VNO-Resilience, resilience is provided in the virtual layer by the virtual network operators (VNOs). We formulate our models as mixed-integer linear problems optimizing the average delay and cost of the virtual network.

Our resilience models simultaneously optimize the mapping of the virtual network (VNet) and the routing inside it for a given set of cloud service requests. We show that our models outperform prior approaches, where the mapping is limited to the shortest paths or the optimization of VNet design and cloud connections is performed separately. For the former we show that the prior approach fails to find a resilient VNet solution in around 50% of the cases, while our model almost always finds a solution. For the latter, we show that combined optimization reduces the maximum delay occurring in the VNet by 30% and 50% for PIP-Resilience and VNO-Resilience, respectively. We performed extensive simulations using realistic topologies to quantitatively analyze the effect of different cost settings and network virtualization environment parameters on the performance of the models. We show that already for two cloud infrastructure providers available in the network, the delay gain of VNO-Resilience reaches 200% in terms of the maximum delay, which can be guaranteed in the VNet. Our simulation results provide a framework to the future VNOs to decide between providing resilience by themselves or having it from the PIPs according to the actual cost factors offered by the PIPs and their delay preferences. Finally, we provide a discussion for the implementation possibilities of the proposed models.

In this paper we have focused on the two fundamental cases, in which resilience is either provided by the PIP or VNO. Our work can be extended for hybrid resilience realizations. Furthermore, the effect of both unicast and anycast service requests in the network and SRLG failures are to be evaluated.

References

[1] Symantec, Virtualization and Evolution to the Cloud Survey, 2011.

[2] M. K. Chowdhury, R. Boutaba, A survey of network virtualization, Elsevier Computer Networks, 54(5), 2010.

[3] A. Khan et al., "Network virtualization: a hypervisor for the internet?," IEEE Communications Magazine 50(1): 136-143, 2012.

[4] P. Papadimitriou et al., Implementing network virtualization for a future internet, In 20th ITC Specialist Seminar, Vietnam, May 2009.

[5] P. Vicat-Blanc et al., Bringing Optical Networks to the Cloud: An Architecture for a Sustainable Future Internet, Lecture Notes in Computer Science, 2011, vol 6656, The Future Internet, p. 307-320.

[6] J. Van der Merwe et al., "Towards a ubiquitous cloud computing infrastructure," IEEE Workshop on Local and Metropolitan Area Networks, 2010.

[7] Amazon AWS, Online: http://aws.amazon.com/directconnect/

[8] S. Meier et al., "Provisioning and Operation of Virtual Networks," Electronic Communications of the EASST, vol. 37, Mar. 2011.

[9] I. B. Barla, D. A. Schupke, G. Carle, "Analysis of Resilience in Virtual Networks," 11th Würzburg Workshop on IP: Joint ITG and Euro-NF Workshop "Visions of Future Generation Networks," 2011.

[10] I. B. Barla, D. A. Schupke, G. Carle, "Resilient Virtual Network Design for End-To-End Cloud Services," IFIP NETWORKING 2012, Prague, May 2012.

[11] S. Koo, G. Sahin, S. Subramaniam, "Cost Efficient LSP Protection in IP/MPLS-over-WDM Overlay Networks", IEEE ICC 2003, Anchorage, USA, May 2003.

[12] A. Capone, J. Elias, F. Martignon, "Models and Algorithms for the Design of Service Overlay Networks", IEEE Transactions on Network and Service Management, vol.5, no.3, p.143-156, 2008.

[13] K. Lee, E. Modiano, "Cross-Layer Survivability in WDM-Based Networks," INFOCOM 2009, IEEE, April 2009.

[14] H. Yu, "Survivable Virtual Infrastructure Mapping in a Federated Computing and Networking System under Single Regional Failures," IEEE GLOBECOM 2010, Dec. 2010.

[15] J. W. Jiang et al., "Joint VM placement and routing for data center traffic engineering," IEEE INFOCOM 2012, March 2012.

[16] J. Xu et al., "Survivable virtual infrastructure mapping in virtualized data centers," IEEE International Conference on Cloud Computing, 2012.

[17] M. Alicherry, T.V. Lakshman, "Network aware resource allocation in distributed clouds," IEEE INFOCOM 2012, March 2012.

[18] NobelUS and NobelEU Topologies, Online:http://sndlib.zib.de

[19] T. Weems, "How far is Far Enough," Disaster Recovery Journal, Spring 2003, Vol. 16, Issue 2.

[20] K. Schmidt, High Availability and Disaster Recovery: Concepts, Design, Implementation, Springer, Berlin, Germany, September 2006.