Storage Protection with Connectivity and Processing Restoration for Survivable Cloud Services

Citation for the original published paper (version of record):
Natalino Da Silva, C., Rostami, A., Monti, P. (2021). Storage Protection with Connectivity and Processing Restoration for Survivable Cloud Services. Proceedings - International Conference on Computer Communications and Networks, ICCCN, 2021-July.
http://dx.doi.org/10.1109/ICCCN52240.2021.9522324

Storage Protection with Connectivity and Processing Restoration for Survivable Cloud Services

Carlos Natalino*, Ahmad Rostami†, and Paolo Monti*
*Department of Electrical Engineering, Chalmers University of Technology, SE-412 96 Gothenburg, Sweden
E-mail: {carlos.natalino, mpaolo}@chalmers.se
† Ericsson Research, Farogatan 6, SE-164 83 Stockholm, Sweden.

***Abstract*—The operation and management of software-based communication systems and services is a big challenge for infrastructure and service providers. The challenge is mainly associated with the more significant number of configurable elements and the higher dynamicity in the software-based systems than the classical ones. On the other hand, the modularity and programmability in software-based networks enabled by technologies like Software-Defined Networking (SDN) and Network Function Virtualization (NFV) provide new opportunities for operators to realize advanced network and service management strategies beyond the classical techniques.**

In our work, we elaborate on these new opportunities and propose a novel strategy for the management of survivable cloud services. In particular, we leverage the flexibility of SDN and NFV to combine proactive protection and reactive restoration mechanisms, and we put forward a novel strategy for enhancing the survivability of cloud services. Through comprehensive evaluations, we demonstrate that the proposed strategy offers significant benefits in terms of availability and restorability of services while reducing, at the same time, the overhead caused by the relocation of cloud services in case of failures.

***Index Terms*—Software Defined Networking (SDN), Virtual Network Function (VNF), Cloud services, Resiliency, Protection, Restoration, Availability, Service Relocation.**

I. Introduction

The softwarization of communication infrastructures is changing the way we create and manage networking services. More specifically, complementary technologies such as Software-Defined Networking (SDN) and Network Function Virtualization (NFV) collectively enable the flexible and dynamic creation of advanced network and service management strategies. These changes bring about several benefits for network operators and providers, as well as for the consumers of networking services. In particular, an operator can efficiently share its resources among several verticals, a provider can bring new services to the market faster, and, at the same time, consumers can request and get a variety of flexible services (virtually) on the fly [1] [2].

Besides the benefits just described, network softwarization also poses new challenges for managing software-based networks and services. For instance, it is expected that services provisioned over softwarized platforms can offer at least the same level of availability and resiliency as those provided over conventional networks. Fulfilling such requirements might be challenging since software-based environments deal with a more heterogeneous set of manageable entities and interfaces, making it more difficult to control the network using classical tools and methods. Nonetheless, SDN/NFV technologies open up new possibilities for designing and developing innovative architectures and methods for managing and orchestrating networks and services [3]. In particular, SDN and NFV concepts can be leveraged to build resource-optimized, cost-efficient, and survivable cloud services.

A cloud service comprises the joint orchestration of connectivity, Processing Units (PUs), and Storage Units (SUs). Connectivity is provisioned between a client node (i.e., where the service originates) and a destination Data Center (DC) where PUs and SUs (i.e., sometimes also referred to as Information Technology (IT) resources) are used to instantiate the required Virtual Network Functions (VNFs). Traditionally, there are two main categories of survivable strategies: *protection* and *restoration*. Protection strategies provision proactively redundant backup resources that are used only upon the occurrence of a failure. For a cloud service, this translates into having to duplicate connectivity, PUs, and SUs. As a result, the high resiliency guaranteed by protection strategies comes at the expense of low resource efficiency [4]. On the other hand, restoration strategies do not pre-provision backup resources and rely on spare (unused) resources in the network to recover a service after the occurrence of a failure. Depending on the failure scenario, restoring a cloud service translates into allocating (on the fly) new connectivity resources between the client node and the destination DC and/or new IT resources in the destination DC. This results in a higher resource efficiency at the cost of lower resiliency (i.e., compared to protection) since there are no guarantees backup resources will be available when needed.

Cloud services are anycast in nature (i.e., the service VNFs can be instantiated in any DC as long as the service requirements are met). Therefore, one way to improve the resiliency performance of restoration strategies is to combine them with the service relocation concept [5]. More specifically, when restoration fails because of the lack of connectivity (i.e., towards the destination DC), and/or IT resources (i.e., within the destination DC), a cloud service can be relocated to a

This work is supported by VINNOVA (Sweden's innovation agency) within the framework of the EUREKA cluster CELTIC-NEXT project AI-NET-PROTECT (2020-03506). Part of the work was developed while Ahmad Rostami was with Ericsson Research. He is now with the Corporate Research of Robert Bosch GmbH, Gerlingen 70839, Germany (e-mail: ahmad.rostami@de.bosch.com).

different DC that presents better reachability conditions, and/or that has better chances to restore the failed VNFs. Despite the improved restorability performance, Restoration with Relocation (RwR) has the detrimental effect of introducing an additional service downtime due to the (possibly extended) relocation process of the service SUs [6]. On the other hand, since SUs are pretty cheap compared to connectivity and PUs [7], the relocation time of a RwR procedure can be drastically reduced by proactively protecting the SUs of a service. In contrast, connectivity and PUs can still be allocated on the fly after a failure (i.e., to retain the resource efficiency benefits typical of RwR [7]). A smart resiliency strategy could then orchestrate, during the service provisioning phase, the replication of the service's SUs over different DCs. Upon the occurrence of a failure, it could recover the failed service by allocating backup connectivity and PUs only when and where needed.

This paper uses the intuition just described to introduce the Storage protection with COnnectivity and processing REstoration (SCORE) strategy. The proposal uses an intelligent approach that combines the protection and the RwR concepts to achieve high resource efficiency and survivability. For a given cloud service, SCORE orchestrates the provisioning of connectivity, PUs, and SUs in a primary DC, together with additional backup SUs in a secondary DC. Primary and backup SUs are continuously synchronized through a replication process. Upon the occurrence of a failure, the following procedure is triggered. SCORE first tries to recover the service at the primary DC, i.e., by either restoring the failed connectivity path and/or by allocating the required amount of PUs[1]. If the previous attempt fails, SCORE tries to restore the service at the secondary DC where SUs are already provisioned, i.e., by allocating a new connectivity path to it and by activating the required amount of PUs. If none of the previous attempts is successful, SCORE tries one last restoration attempt by relocating the service in its entirety (i.e., the whole set of SU) to an entirely new DC. The paper presents a heuristic for the resilient provisioning of cloud services (i.e., according to the SCORE intuition) and anInteger Linear Programming (ILP) formulation for recovery of cloud services affected by a failure.

Results derived from simulating a scenario with single link failures show that SCORE can significantly reduce the cost of protecting a cloud service, i.e., blocking probability is reduced compared to traditional protection strategies. At the same time, SCORE reduces the need for service relocation, which results in a lower relocation downtime and a lower relocation overhead performance for the recovered services. Service availability and restorability performance are also improved since SCORE allows for more services to be restored.

The rest of this paper is organized as follows. A review of the literature is presented in Sec. II. Sec. III details the control and management architecture considered in this work and introduces the central intuition behind SCORE. The details of the SCORE strategy are described in Sec. IV. Sec. V presents the performance evaluation results. Finally, Sec. VI provides a few concluding remarks.

II. Literature Review

The adoption of software-based (virtualized) solutions for high-performance networking has been the subject of many studies. One of the challenges is integrating different network segments, e.g., access, metro, and core, to provide end-to-end services that can meet specific performance requirements [8]. Another challenge is integrating different domains, e.g., fixed and mobile networks and data centers, so that connectivity and compute resources are managed by a single orchestrator [9]. Also, virtualized networking should achieve performance (in terms of e.g., bit rate and latency) comparable to the one achieved by dedicated hardware [10]. The authors in [11] and [12] provide an in-depth review of the challenges and possibilities brought by SDN and NFV, respectively.

As these softwarized networks support many essential services with a variety of requirements, resiliency becomes a significant concern for infrastructure providers [13]. On the one hand, virtualization allows fast restoration of malfunctioning services through replication and/or relocation. On the other, many more components can trigger a failure (e.g., hardware or software failure). Several strategies have been studied to address the complexity of providing resiliency in softwarized networks, In this section, we first review a few works that optimize the service replication and relocation procedures. Then, we review some studies that design resilient network infrastructures considering that the network can perform these replication and relocation procedures. Finally, we review works that propose provisioning strategies that leverage these replication and relocation capabilities to improve network resiliency.

The VNF relocation and SUs replication concepts have been the subject of several studies. The work in [14] proposes a strategy to replicate SUs over different DCs, guaranteeing that each available replica maintains data integrity upon a failure event while reducing the overhead (in terms of response time) caused by the services being replicated. The works [6], [15] analyze how to improve service relocation performance by reducing both the number of network resources used during the relocation and the relocation downtime. These works show that, although the overhead of replication and relocation procedures have been significantly reduced over the last years, the service downtime due to service relocation is still significant. Mitigating the relocation downtime is one of the goals of the proposed SCORE strategy.

The work in [16] proposes a design model to dimensions network and DC resources. It exploits service relocation to reduce the number of resources required to serve a defined set of service requests. The work shows that higher cost savings are achieved by deploying more DCs. In [17] the authors propose a static model to place DCs over a network topology and to initially place content replicas over these DCs. The

[1] In the presence of a failure of the SUs, SCORE skips this step moving directly to the next one.

authors also propose a dynamic content placement strategy that reduces the risk of content loss in the presence of disaster events, showing that relocation can help mitigating content loss in case of disasters. On the other hand, none of the works just described deal with the dynamic provisioning of cloud services, comprising connectivity, PUs and SUs. The survey in [18] provides an in-depth analysis of the works considering cloud network infrastructures.

Service replication and relocation have also been studied in the context of dynamic provisioning. The work in [5] proposes the Connectivity Restoration with Service Relocation (CR+SR) strategy, where the service relocation capability is added to the path restoration strategy. Results show that by adding service relocation to restore cloud services, availability and restorability can be improved without penalizing the blocking probability. Unlike the strategies mentioned above, our strategy combines service replication and relocation to enable partially protected services to survive diverse link and node failure events. Moreover, we evaluate the proposed strategy under random link failures.

III. Orchestration of Survivable Cloud Services

This section introduces the network architecture considered in this work and how the SCORE strategy leverages the architecture capabilities to improve cloud services' resiliency.

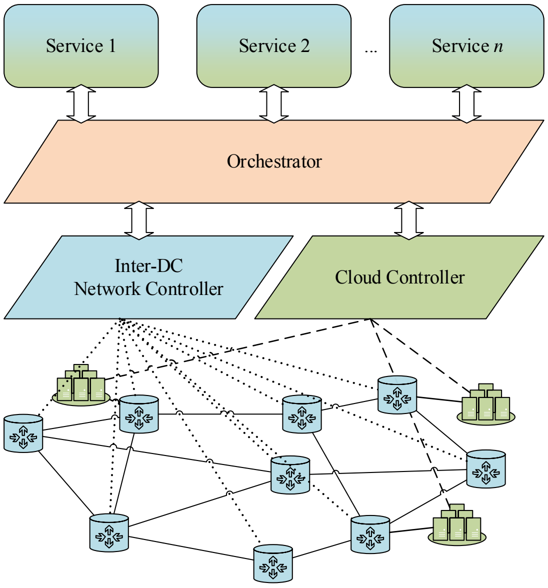


Fig. 1. Network architecture: a transport network and a cloud controller are connected to an orchestration layer responsible to manage the end-to-end services.

Figure 1 illustrates how the architecture is organized. It comprises four levels: the physical infrastructure, the infrastructure-specific controllers, an orchestration layer, and a service layer. The physical infrastructure includes both the inter-DC network and the cloud domains. The connectivity resources are managed by an inter-DC network controller, while intra-DC resources, i.e., PUs and SUs, are managed by a cloud controller. Both the (inter-DC) network and the (intra-DC) cloud controllers interact over their northbound interface with an orchestrator in charge of cloud services operations. In this architecture, service requests are placed at the northbound interface of the orchestrator. Services require the provisioning of resources in the inter-DC network and/or cloud domains. The orchestrator is the entity responsible for deciding how to map a service request into the infrastructure resources. The domain-specific controllers are in charge of provisioning the resource in each of the technology-specific domains (i.e., according to the orchestrator's instructions).

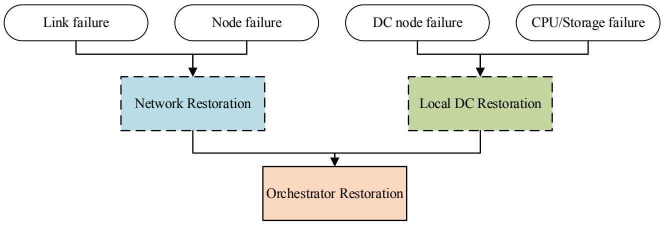


Fig. 2. Service restoration options as a function of the failure scenarios.

Traditionally, resiliency strategies do not leverage on the interaction between different network domains [19]–[22]. On the other hand, a coordinated, multi-domain provisioning approach might improve resource efficiency. Upon a failure event, resiliency is ensured by triggering domain-specific restoration procedures (i.e., the dashed rectangles in Fig. 2) (possibly) leading to inefficient use of resources in the physical infrastructure. For instance, restoring a link failure considering only network restoration may leave some services un-recovered (e.g., due to the lack of connectivity resources). On the other hand, an orchestrated restoration procedure (Fig. 2) could consider relocating the IT resources of the disrupted services to a different DC with enough resources to support the service execution and still reachable by the client node. Therefore, leveraging on the orchestration of connectivity and IT resources can unveil new and more resource-efficient opportunities to restore services.

The strategy proposed in this paper relies on multi-domain orchestration to define provisioning and failure recovery strategies that combine protection and restoration procedures to enhance the survivability of cloud services. One key aspect is SUs replication to ensure that a service is protected against single DC failures, as well as to shorten the restoration downtime potentially.

The envisioned provisioning procedure works as follows. Upon receiving a request to provision a cloud service, the orchestrator needs to select: *(i)* a primary DC where the necessary PUs and SUs can be allocated; *(ii)* a backup DC where the backup SUs can be allocated; and *(iii)* a path in the inter-DC network with enough resources to connect the client node with the primary DC. The proposed strategy assumes that primary and backup DCs are allocated a connectivity path that allows primary and backup SUs to maintain a consistent replica of the service data over the two DCs. If any of

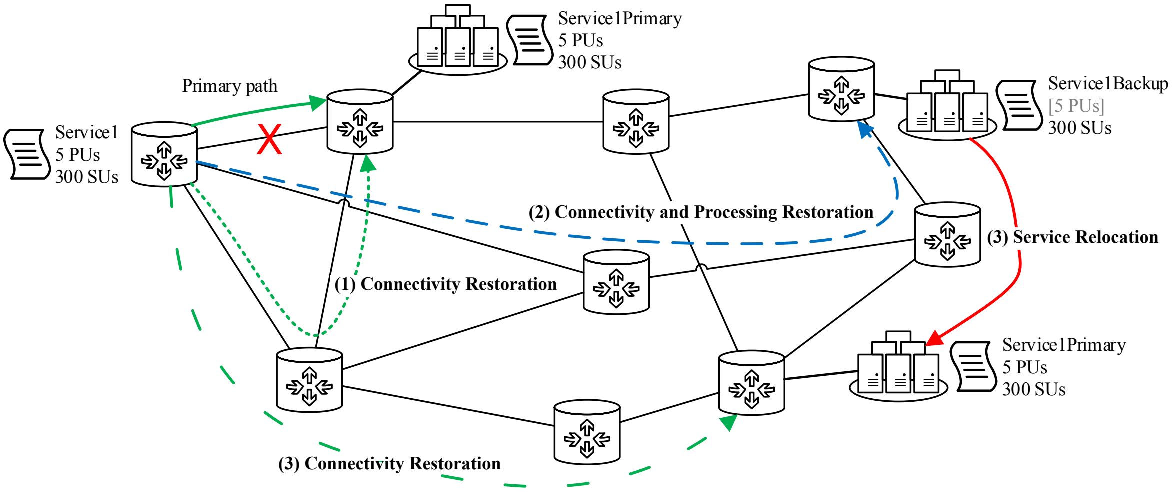


Fig. 3. Illustrative example: the three ways to restore a cloud service affected by a link failure using SCORE: (1) Connectivity Restoration (CR), (2) Connectivity and Processing Restoration, and (3) Connectivity Restoration with Service Relocation (CR+SR).

these requirements are not met, the cloud service request is blocked. The proposed service provisioning procedure is expected to require double the SUs compared to a traditional (unprotected) provisioning strategy. However, SUs are usually cheaper than PUs [7], they consume little electricity [23], and enable the proposed strategy to be effective against more disruptive failure scenarios, e.g., DC or disaster failure [24]. Finally, the backup SUs can be leveraged during the restoration procedure as follows.

To restore disrupted services, SCORE relies on spare (unused) connectivity and PUs as well as on the backup SUs deployed during the service provisioning phase. For each disrupted service, the restoration procedure first checks if it is possible to find a path with enough free connectivity resources to connect the client node to the active (primary) DC. The service is restored using this path if such a path exists. Otherwise, the procedure checks *(i)* if it is possible to find a new path with enough connectivity resources from the client node to the DC hosting the backup SUs, and *(ii)* if this DC has enough PUs to process the service. If the above conditions are met, the service is restored in a non-revertive way at the backup DC. Note that with the two recovery options mentioned above, the service downtime is limited to reconfiguring the connectivity paths. Finally, if none of the previous options is possible, SCORE considers relocating the failed services. During service relocation, either the primary or the backup copies of the service data are relocated to another DC with enough PUs and SUs, and which can be connected to the client node. Service relocation is considered the last option due to its inherent overhead and the downtime incurred by transferring data between DCs.

Figure 3 presents an example describing the set of options SCORE can use for restoring a cloud service disrupted by a link failure. At first, the orchestrator tries a conventional *connectivity restoration* procedure between the client node and the primary DC. If the service affected by the failure can be restored, the procedure ends. If connectivity restoration is unsuccessful, the orchestrator attempts to establish a path between the client node and the backup DC. Additionally, the backup DC needs to have enough free PUs to process the recovered service. If successful, this operation, referred to as *connectivity and processing restoration*, will recover the service with no relocation downtime incurred. If none of the above procedures is successful, the orchestrator tries a third (and last) option, i.e., *connectivity restoration with service relocation*. In this case, other DCs in the network, different from the primary or backup DCs already assigned to the service, are considered. For this operation to be successful, connectivity and IT resources should be available, and service relocation should be possible. SCORE uses service relocation as the last option because of the usually extended relocation downtime, which negatively impacts the service availability performance. If this last attempt also fails, the service is dropped.

SCORE offers survivability against other failures not reported in Fig. 3. For instance, similar operations can be done in the presence of PUs, SUs, or DC failures. In the next section, we introduce a heuristic that can be used for provisioning a cloud service and an ILP formulation that can be used to compute a solution for recover a cloud service after a failure according to the SCORE intuition.

IV. Storage Protection with Connectivity and Processing Restoration (SCORE)

This section presents a detailed description of the SCORE strategy. Sec. IV-A introduces a heuristic for the provisioning of cloud services following the intuition explained in the previous section. Sec. IV-B presents an ILP formulation implementing the SCORE idea. The notation used by the heuristic and the ILP model is summarized in Table I.

Algorithm 1: Provisioning with storage protection heuristic according to the SCORE concept.

```
Data: $G(N, E)$ and $s$
1  $route$ = getRoute($s$, null, $true$, $true$, $true$) ;
2  if $route \neq null$ then
3      $secRoute$ = getRoute($s$, $\neq route^{dst}$, $true$, $false$, $true$) ;
4      if $secRoute \neq null$ then
5          assignPrimaryRoute($s$, $route$);
6          assignSecondaryDC($s$, $secRoute$);
7          enableLiveReplication($s$, $route^{dst}$, $secRoute^{dst}$);
8      else
9          blockService($s$);
10 else
11     blockService($s$);
```

A. Provisioning with Storage Protection

This heuristic uses a procedure named *getRoute* to select a connectivity path (i.e., among a set of *k* candidates) from a client to a DC node that meets the cloud service connectivity requirements. The *getRoute* procedure receives as input the following parameters: *(i)* the description of the cloud service to be considered s (i.e., as defined in Table I); *(ii)* the DC affinity constraint (e.g., the DC to be chosen must be different from a given one, $null$ if no affinity is specified); *(iii)* a flag (i.e., $true$ / $false$) to enforce a minimum number of connectivity resources on the connectivity path from s^{src} and the considered DC; *(iv)* a flag (i.e., $true$ / $false$) to make sure that the considered DC has at least s^{pu} PUs; *(v)* a flag (i.e., $true$ / $false$) to make sure that the considered DC has at least s^{su} SUs. The procedure returns the shortest connectivity path (in terms of the number of hops) found, or $null$ if a connectivity path cannot be found. Note that the procedure can also be easily modified to return the path with the shortest length.

The objective of this provisioning procedure is to be able to select the following: *(i)* a primary DC node with enough PUs and SUs, where the service will be hosted; *(ii)* a secondary DC node with enough SUs to host the backup SUs replica; *(iii)* a path in the inter-DC network with enough capacity to connect the client node to the primary DC.

Algorithm 1 details the provisioning procedure. Initially, the algorithm searches for an available route from the client node to the closest DC with enough PUs, SUs, and connectivity resources (line 1). If a primary route is available (line 2), the algorithm searches for the closest DC, different from the primary one, and with enough SUs (line 3). By selecting the two closest DCs, the algorithm potentially shortens the restoration paths used upon a failure. If a secondary DC is found (line 4), the heuristic allocates the primary (line 5) and backup resources (line 6). It then enables the live replication between the primary and the secondary DCs (line 7). If there are not enough resources (i.e., either connectivity or IT) to accommodate it, the service is blocked (lines 8 and 11).

B. ILP Formulation for Service Restoration

During the restoration of a set of disrupted cloud services, SCORE has three main objectives. They are, in order of importance: *(i)* maximize the value of the average service availability, *(ii)* maximize the number of services restored, and *(iii)* minimize the number of connectivity resources used by the services' restoration paths. The amount of IT resources used by the disrupted services is not part of the objective, as their quantity does not change during the restoration process. These three objectives are considered in the proposed optimization model, which is formally described next.

Objective function:

$$\text{Minimize} \quad \alpha \cdot \sum_{\forall s \in S} \left(n^{rt}_{s,s^{pdc}} - \sum_{\forall i \in N_{DC}} A^s \cdot n^{rt}_{s,i} \right) \quad (1)$$

$$+\beta \cdot \sum_{\forall s \in S} \sum_{\forall k \in N_{DC} | k \neq s^{pdc} \wedge k \neq s^{bdc}} A^s_k + \gamma \cdot \sum_{\forall s \in S} \sum_{\forall (i,j) \in E} x^s_{ij}$$

Subject to:

$$\sum_{\forall i \in N | (i,j) \in E} x^s_{ij} - \sum_{\forall i \in N | (j,i) \in E} x^s_{ji} = \begin{cases} -A^s, \text{ if } j = s^{src} \\ A^s_j, \text{ if } j \in N_{DC} \\ 0, \text{ otherwise} \end{cases}, \quad (2)$$

$$\forall s \in S, \forall j \in N.$$

$$\sum_{\forall d \in N_{DC}} A^s_d \leq 1, \quad \forall s \in S. \quad (3)$$

$$A^s = \sum_{\forall d \in N_{DC}} B^s_d, \quad \forall s \in S. \quad (4)$$

$$A^s_d + B^s_d \leq 1, \quad \forall s \in S, \forall d \in N_{DC}. \quad (5)$$

$$\sum_{s \in S} A^s_d \cdot s^{pu} \leq d^{pu}, \quad \forall d \in N_{DC}. \quad (6)$$

$$\sum_{s \in S} (A^s_d \cdot s^{su} + B^s_d \cdot s^{su}) \leq d^{su}, \quad \forall d \in N_{DC}. \quad (7)$$

$$\sum_{s \in S} x^s_{ij} \leq \lambda_{ij}, \quad \forall (i,j) \in E. \quad (8)$$

$$A^s_n = R^s_{s^{pdc} \to n} + R^s_{s^{sec} \to n}, \quad (9)$$

$$\forall s \in S, n \in N_{DC} | n \neq s^{pdc} \wedge n \neq s^{sec}.$$

$$\sum_{s \in S | m = s^{pdc} \vee m = s^{sec}} R^s_{m \to n} \leq L, \quad \forall m, n \in N_{DC}. \quad (10)$$

Each term in (1) represents one of the three SCORE's objectives. The first term maximizes the value of the average service availability. This is done by minimizing the total service downtime of the disrupted services computed as the difference between the normalized remaining service time and the normalized relocation time. The second term minimizes the number of service relocations. This is done by computing the number of services that are restored using a DC that is different from the primary or secondary DCs used before the failure. The third term minimizes the number of connectivity

TABLE I
INPUT AND VARIABLES.

Definition	Description
$G(N, E)$	A graph representing a transport network consisting of $\lvert N\rvert$ nodes and $\lvert E\rvert$ links
N	Set of network nodes ($N = N_{DC} \cup N_c$), consisting of DCs ($N_{DC} \subseteq N$) and client nodes ($N_c \subseteq N$)
N_{DC}	Set of DCs nodes, where $d \in N_{DC}$ has d^{su} available SUs and d^{pu} available PUs
E	Set of links, where link $(i, j) \in E$ has λ_{ij} connectivity resources
S	Set of active services disrupted by a failure, where $s \in S$ is characterized by a client node s^{src}, a primary and a secondary DCs $s^{pdc}, s^{sec} \in N_{DC}$, and a required amount of SUs and PUs s^{su} and s^{pu}, respectively
L	Number of connectivity resources available for relocation between each DC pair
α	Constant representing the weight for the restored service time in the ILP
β	Constant representing the weight for the number of relocations in the ILP
γ	Constant representing the weight for the number of wavelengths used in the ILP
x^s_{ij}	1 when $s \in S$ is using link $(i, j) \in E$ as its primary path, 0 otherwise
A^s	1 if $s \in S$ can be successfully restored, 0 otherwise
A^s_i	1 when $s \in S$ is using $i \in N_{DC}$ as its primary DC, 0 otherwise
B^s_i	1 when $s \in S$ is using $i \in N_{DC}$ as the backup DC for its SUs, 0 otherwise
$R^s_{i\rightarrow n}$	1 when $s \in S$ is being relocated from i to n with $i, n \in N_{DC}$, 0 otherwise

resources used for the restoration paths while recovering the disrupted services. This is done by computing the number of connectivity resources used over all the links in the network. The three terms in (1) are weighted by α, β and γ, used to decide the relative importance of each terms. We normalize all the time-related quantities in the interval [1,10000] to allow a better setting of the objective function's weights.

The flow conservation constraints are defined in (2), where the restored services and their primary DCs are also computed. Constraints (3) define that each service can select at most one primary DC, while constraints (4) ensure that a restored service must have both a primary and a secondary DC. Constraints (5) ensure that, for each restored service, primary and secondary DCs must be different. Constraints (6) and (7) ensure that the number of PUs and SUs, respectively, do not exceed the total number available at each DC. Specifically for the SUs, (7) computes the number of SUs necessary for primary (working) and backup purposes. Constraints (8) make sure that the amount of connectivity resources used in each link does not exceed the total amount available. Constraints in (9) and (10) ensure that the number of service relocations triggered by the solution does not exceed the limit set by the number of connectivity resources reserved for this purpose. A service relocation is required if a service is restored to a DC different from its primary or secondary ones according to (9). The number of relocations between each DC pair is constrained by (10).

V. Performance Assessment

This section presents the performance assessment work done for the SCORE strategy introduced in Sec. IV. We use a custom-built Java-based simulator that implements a multi-domain architecture comprising an inter-DC network and a cloud computing domain (Sec. III). Our simulator assumes that the orchestrator has full knowledge of how resources are used in both domains, i.e., we consider fully transparent domain controllers. To solve the ILP formulations presented in the previous section, our simulator interfaces with the Gurobi solver [25]. A Red Hat Enterprise Linux (RHEL) workstation with one 8-core Intel Xeon CPU clocked at 3 GHz and 64 GB of RAM is used to run the simulations.

The SCORE strategy is benchmarked against three approaches taken from the literature. The Dedicated Path Protection (DPP) strategy relies on the proactive assignment of backup resources. It provisions a new cloud service by choosing the first and second closest DCs (in terms of hops) with enough IT resources, and a link-disjoint path that connects the client node to them. The other two benchmark strategies rely purely on restoration. A new cloud service is provisioned using a path to the closest DC (in terms of hops) and without reserving any backup resource. Connectivity Restoration (CR) [20] aims to maximize service availability by relying solely on the network controller capabilities, i.e., it has no relocation capabilities. Connectivity Restoration with Service Relocation (CR+SR) [5] aims to maximize service availability by using both path restoration and service relocation to recover the disrupted services.

Each benchmarked strategy is evaluated according to the following metrics: *(i)* blocking probability, i.e., the number of unsuccessfully accommodated services over the total number of cloud service requests; *(ii)* availability, i.e., the observed service uptime in the system over the total service time; *(iii)* restorability, i.e., the number of successfully restored services over the total number of services disrupted by failures; and *(iv)* number of relocations, i.e., the number of service relocations performed over the total number of successfully restored services.

In the following, a description of the simulation scenario is provided, followed by a discussion of the performance results. Finally, we present a sensitivity analysis for the number of connectivity resources reserved for relocation.

A. Simulation Scenario

Figure 4 shows the NSF topology considered for the simulations. For this study, the connectivity resources are modeled in terms of optical wavelengths, thus simulating an optical network infrastructure. All (fiber) links are bidirectional, with 80 wavelengths in each direction. Given the optical nature of the network resources, we refer to connectivity paths as lightpaths. We consider that all nodes have wavelength conversion capabilities, i.e., no wavelength continuity constraints are enforced. We assume to have 3 non-neighboring DCs colocated at the nodes with the highest degree (i.e., nodes 1, 6, and 9). Fiber links connecting the DCs to their respective network nodes have enough connectivity capacity to carry the required traffic, i.e., they are not the bottleneck of the system.

Both DPP and SCORE require live replication between SUs in the primary and secondary DCs. This is ensured by reserving two lightpaths (i.e., one primary and one backup) between each DC pair, i.e., we assume in-band replication. For the strategies leveraging on service relocation (i.e., CR+SR and SCORE) it is crucial to make sure that data can be transferred among DCs when needed. For this reason, with CR+SR and SCORE L lightpaths between each DC pair are reserved for service relocation purposes. Unless otherwise specified, we assume that L is equal to 5% of the number of wavelength resources available in a fiber link, i.e., L=4. At the beginning of each simulation, a set of k=10 shortest paths is computed between each client and DC node.

This study assumes a single link failure scenario. The link failure rate is exponentially distributed with an average of $231{\cdot}10^{-5}$ [1/sec], while the Mean Time to Repair (MTTR) follows an exponential distribution with an average 4320 [sec]. The link failures are uniformly distributed over all the links in the inter-DC network. Upon the occurrence of a failure, the simulator solves an ILP instance of the problem comprising all the disrupted cloud services. Each optimization instance is solved optimally, i.e., no gaps are allowed.

The arrival rate of the cloud services follows a Poisson distribution. The holding time of each cloud service is exponentially distributed with an average of 60 hours. The mean time between arrivals is set according to the load value chosen for the specific experiment. The client node of a service request is chosen uniformly among all the client nodes in the network.

Each service requires one bi-directional lightpath with the capacity of a single wavelength between the client node and the service DC(s), according to the strategy being considered. The number of PUs requested by each service follows a normal distribution with values {1, 2, 4, 8, 12, 16, 24, 32, 40} and average (μ^{pu}) equal to 16. The number of SUs also follows a normal distribution in the range from 10 [GB] to 1 [TB], with a 10 [GB] step and average (μ^{su}) equal to 500 [GB]. For both PU and SU distributions, the standard deviation is set to half the average value.

The dimensioning of the IT resources at each DC is defined according to the average number of PUs and SUs per request, defined as:

$$r^{\star} = \lceil \rho^{\star} \cdot \mu^{\star} \cdot degree(d) \rceil, \, \star \in \{su, pu\}, \, \forall d \in N_{DC}, \quad (11)$$

where $\rho^{\star}$ represents the dimensioning factor of a particular resource, $\mu^{\star}$ represents the average number requested for a particular resource, $degree(d)$ represents the degree of the node where DC d resides, and $r^{\star}$ represents the number of resource units placed at d. We assume $\rho^{pu} = 1.2$ and $\rho^{su} = 2.0$, which represents the case where IT resources are not the bottleneck in the system. The parameters α, β and γ are set to 10^4, 10^5 and 1, respectively.

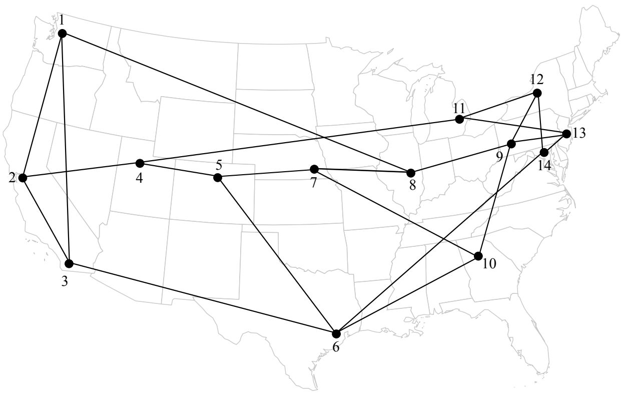


Fig. 4. NSF network topology comprising 14 nodes and 22 links. DCs are located at nodes 1, 6 and 9.

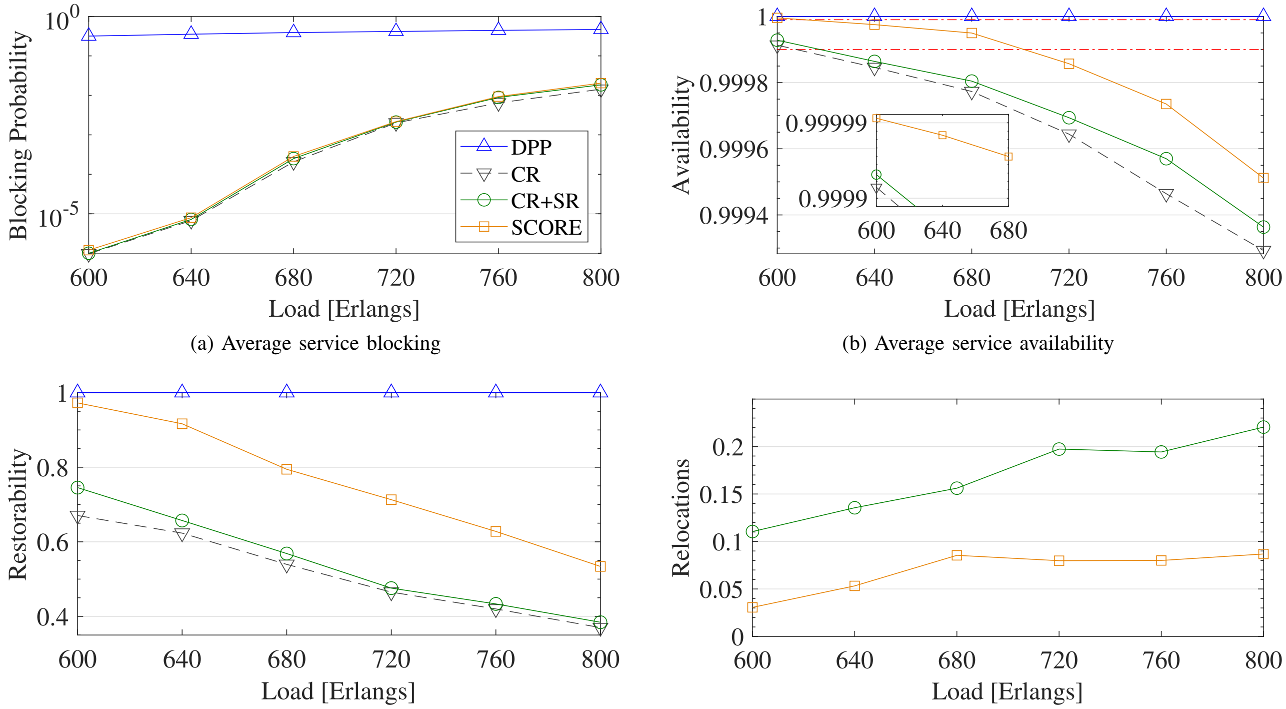
The results are averaged over 200 experiments with 1 million cloud service request arrivals for each experiment. The confidence interval is 5% or lower (i.e., for blocking probability values in medium to high loads), with 95% of confidence.

B. Simulation Results

Figure 5 presents the results of the simulation work made with the NSF network topology. The blocking probability results presented in Fig. 5a demonstrate the poor performance provided by DPP. By proactively protecting each service, DPP increases the blocking probability by at least one order of magnitude compared to the restoration-based approaches. When CR+SR) and SCORE are used service relocation causes a slight increase in the blocking probability compared to CR. The reason is threefold. First, similarly as in [5], service relocation leads to better service restorability performance. With more resources being occupied, it becomes more difficult to accommodate new service requests. Second, with CR+SR and SCORE, the connectivity resources available in the inter-DC network are slightly less due to the need to reserve L lightpaths for relocation purposes. Third, with SCORE one additional primary and backup lightpaths for live replication purposes need to be reserved between each DC pair, which also impacts the blocking probability performance.

Figure 5b shows that DPP is always able to guarantee 100% availability in the single link failure scenario considered in this work. Compared to the pure restoration-based approaches (i.e., CR and CR+SR), SCORE can drastically improve the availability performance. The inset in Fig. 5b highlights the results for four and five 9's availability. More specifically, SCORE achieves five 9's at the lowest load, while the other strategies can offer only four 9's. Moreover, SCORE provides four 9's availability up to 680 Erlangs, while the other strategies can only offer four 9's at the lowest load conditions.

Figure 5c presents the restorability results. DPP protects services against single link failures and can recover all the



(a) Average service blocking

(b) Average service availability

(c) Average service restorability

(d) Average number of relocations per recovered service

Fig. 5. Simulation results for the NSF network topology with 3 DCs. L=4 is equal to 5% of the number of wavelength resources available in a fiber link. The inset in figure (b) highlights the results for four and five 9's availability.

TABLE II
IMPACT OF L ON THE PERFORMANCE OF THE RECOVERY STRATEGIES UNDER EXAM. THREE CASES: 5%, 10% AND 15% OF WAVELENGTH RESOURCES ARE RESERVED FOR RELOCATION PURPOSES BETWEEN EACH DC PAIR. THE LOAD IS EQUAL TO 720 [ERLANG].

Strategy	Blocking Prob.			Availability			Restorability			Relocations		
	5%	10%	15%	5%	10%	15%	5%	10%	15%	5%	10%	15%
DPP	0.4157	0.4157	0.4157	1	1	1	1	1	1	N.A.	N.A.	N.A.
CR	0.0020	0.0020	0.0020	0.99964	0.99964	0.99964	0.465	0.465	0.465	N.A.	N.A.	N.A.
CR+SR	0.0021	0.0033	0.0038	0.99969	0.99948	0.99939	0.475	0.545	0.592	0.197	0.316	0.390
SCORE	0.0021	0.0024	0.0036	0.99985	0.99966	0.99965	0.712	0.720	0.722	0.079	0.148	0.191

disrupted ones by switching to the backup path/DC. The hybrid protection/restoration approach used by SCORE enables it to restore at least 10% more services than CR and CR+SR, achieving an improvement of up to 20% in low load conditions.

Figure 5d shows the average number of relocations needed to recover a service with the relocation-based strategies (CR+SR and SCORE). SCORE shows a significant reduction compared to CR+SR. While the number of relocations required by CR+SR continues to increase with the load value, the number of relocations performed by SCORE stabilizes for medium and high load conditions. This shows how the use of a secondary DC carrying a copy of the service's SUs effectively reduces the number of relocations.

C. Sensitivity analysis

In the last part of the performance results analysis, we focus on the impact that the number of lightpaths reserved for service relocation has on the performance of the recovery strategies discussed so far. The results are presented in Table II. The amount of wavelength reserved on each link for relocation purposes is varied between 5% and 15% with increments of 5%each. In our scenario (i.e., 80 wavelengths per fiber link) it translates in 4, 8, and 12 lightpaths available between each DC pair for service relocation, respectively. Results for DPP and CR are also shown for reference purposes, even if the value of L does not impact their performance.

There is a trade-off to be considered when setting the value of L. Increasing L's value reduces the number of wavelengths available for provisioning new cloud services. On the one hand, a high L's value can potentially improve the chances disrupted services have to be restored if relocation is needed.

This trade-off is clear when looking at the combined values of the availability, restorability, and relocations performance in Table II. While availability decreases when L increases, the restorability and relocations performance follow an opposite trend. With more resources reserved for relocation, there are more chances of relocating services. SCORE prioritizes services with longer remaining holding time, but with more wavelength resources available to relocate, services with a shorter remaining lifetime can also be restored. This is demonstrated by the increase in the relocations and restorability performance when L=10% and L=15% However, relocating services with short remaining time occupies resources that could otherwise be used to provision new incoming cloud service requests, an aspect demonstrated by the increase in blocking probability with increasing values of L.

On the contrary, with fewer resources reserved for relocation, there are less chances for relocating services. In this case, services with shorter remaining lifetimes are more likely to be dropped than to be relocated. This allows resources to be assigned to (disrupted and/or new) longer-lasting services, positively impacting the availability performance in the long term.

VI. Conclusions

The paper proposes a new strategy called Storage protection with COnnectivity and processing REstoration (SCORE), which leverages the joint orchestration of connectivity and cloud resources enabled by SDN and NFV technologies. The objective is to improve the survivability of cloud services while maximizing the efficiency in which both connectivity and IT resources are used. The proposed strategy uses a hybrid provisioning procedure based on protecting the SUs of cloud services while restoring their connectivity and processing resources. The hybrid nature of the SCORE strategy increases the restorability of services and, at the same time, drastically reduces the number of service relocations required in case of failures, collectively resulting in a significant improvement in the cloud service availability performance.

References

[1] M. R. Raza, A. Rostami, L. Wosinska, and P. Monti, "A slice admission policy based on big data analytics for multi-tenant 5G networks," *Journal of Lightwave Technology*, vol. 37, no. 7, pp. 1690–1697, 2019.

[2] M. R. Raza, M. Fiorani, A. Rostami, P. Öhlen, L. Wosinska, and P. Monti, "Dynamic slicing approach for multi-tenant 5G transport networks [invited]," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 10, no. 1, pp. A77–A90, 2018.

[3] R. Mijumbi, J. Serrat, J. l. Gorricho, S. Latre, M. Charalambides, and D. Lopez, "Management and orchestration challenges in network functions virtualization," *IEEE Communications Magazine*, vol. 54, no. 1, pp. 98–105, Jan 2016.

[4] I. M. Araújo, C. Natalino, H. Chen, M. De Andrade, D. L. Cardoso, and P. Monti, "Availability-guaranteed service function chain provisioning with optional shared backups," in *16th International Conference on the Design of Reliable Communication Networks (DRCN)*, 2020.

[5] C. Natalino, L. Wosinska, S. Spadaro, J. ao C. W. A. Costa, C. R. L. Francês, and P. Monti, "Restoration in optical cloud networks with relocation and services differentiation," *J. Opt. Commun. Netw.*, vol. 8, no. 2, pp. 100–111, Feb 2016.

[6] T. Wood, K. K. Ramakrishnan, P. Shenoy, J. V. der Merwe, J. Hwang, G. Liu, and L. Chaufournier, "CloudNet: Dynamic pooling of cloud resources by live WAN migration of virtual machines," *IEEE/ACM Transactions on Networking*, vol. 23, no. 5, pp. 1568–1583, Oct 2015.

[7] C. Natalino, P. Monti, L. França, M. Furdek, L. Wosinska, C. R. Francês, and J. W. Costa, "Dimensioning optical clouds with Shared-Path Shared-Computing (SPSC) protection," in *IEEE 16th International Conference on High Performance Switching and Routing (HPSR)*, July 2015.

[8] V. Lopez, J. M. G. Josa, V. Uceda, F. Slyne, M. Ruffini, R. Vilalta, A. Mayoral, R. Muñoz, R. Casellas, and R. Martínez, "End-to-end service orchestration from access to backbone," *J. Opt. Commun. Netw.*, vol. 9, no. 6, pp. B137–B147, Jun 2017.

[9] A. Rostami, P. Ohlen, K. Wang, Z. Ghebretensae, B. Skubic, M. Santos, and A. Vidal, "Orchestration of ran and transport networks for 5g: An sdn approach," *IEEE Communications Magazine*, vol. 55, no. 4, pp. 64–70, April 2017.

[10] T. Wood, K. K. Ramakrishnan, J. Hwang, G. Liu, and W. Zhang, "Toward a software-based network: integrating software defined networking and network function virtualization," *IEEE Network*, vol. 29, no. 3, pp. 36–41, May 2015.

[11] D. Kreutz, F. M. V. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE*, vol. 103, no. 1, pp. 14–76, Jan 2015.

[12] R. Mijumbi, J. Serrat, J. L. Gorricho, N. Bouten, F. D. Turck, and R. Boutaba, "Network function virtualization: State-of-the-art and research challenges," *IEEE Communications Surveys Tutorials*, vol. 18, no. 1, pp. 236–262, Firstquarter 2016.

[13] B. Han, V. Gopalakrishnan, G. Kathirvel, and A. Shaikh, "On the resiliency of virtual network functions," *IEEE Communications Magazine*, vol. 55, no. 7, pp. 152–157, July 2017.

[14] T. Wood, H. A. Lagar-Cavilla, K. K. Ramakrishnan, P. Shenoy, and J. Van der Merwe, "PipeCloud: Using causality to overcome speed-of-light delays in cloud-based disaster recovery," in *Proceedings of the 2nd ACM Symposium on Cloud Computing*, ser. SOCC '11. New York, NY, USA: ACM, 2011, pp. 17:1–17:13.

[15] U. Mandal, P. Chowdhury, M. Tornatore, C. U. Martel, and B. Mukherjee, "Bandwidth provisioning for virtual machine migration in cloud: Strategy and application," *IEEE Transactions on Cloud Computing*, vol. PP, no. 99, pp. 1–1, March 2016.

[16] C. Develder, J. Buysse, B. Dhoedt, and B. Jaumard, "Joint dimensioning of server and network infrastructure for resilient optical grids/clouds," *IEEE/ACM Transactions on Networking*, vol. 22, no. 5, pp. 1591–1606, Oct 2014.

[17] S. Ferdousi, F. Dikbiyik, M. F. Habib, M. Tornatore, and B. Mukherjee, "Disaster-aware datacenter placement and dynamic content management in cloud networks," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 7, no. 7, pp. 681–694, July 2015.

[18] C. Colman-Meixner, C. Develder, M. Tornatore, and B. Mukherjee, "A survey on resiliency techniques in cloud computing infrastructures and applications," *IEEE Communications Surveys Tutorials*, vol. 18, no. 3, pp. 2244–2281, 2016.

[19] M. Xia, M. Tornatore, C. U. Martel, and B. Mukherjee, "Risk-aware provisioning for optical wdm mesh networks," *IEEE/ACM Transactions on Networking*, vol. 19, no. 3, pp. 921–931, June 2011.

[20] J. Ahmed, C. Cavdar, P. Monti, and L. Wosinska, "Hybrid survivability schemes achieving high connection availability with a reduced amount of backup resources," *IEEE/OSA Journal of Optical Communications and Networking*, vol. 5, no. 10, pp. A152–A161, Oct 2013.

[21] S. Secci and S. Murugesan, "Cloud networks: Enhancing performance and resiliency," *Computer*, vol. 47, no. 10, pp. 82–85, Oct 2014.

[22] N. Shahriar, S. R. Chowdhury, R. Ahmed, A. Khan, S. Fathi, R. Boutaba, J. Mitra, and L. Liu, "Virtual network survivability through joint spare capacity allocation and embedding," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 502–518, March 2018.

[23] J. Baliga, R. W. A. Ayre, K. Hinton, and R. S. Tucker, "Green cloud computing: Balancing energy in processing, storage, and transport," *Proceedings of the IEEE*, vol. 99, no. 1, pp. 149–167, Jan 2011.

[24] R. D. S. Couto, S. Secci, M. E. M. Campista, and L. H. M. K. Costa, "Network design requirements for disaster resilience in IaaS clouds," *IEEE Communications Magazine*, vol. 52, no. 10, pp. 52–58, October 2014.

[25] I. Gurobi Optimization, "Gurobi optimizer reference manual," 2017. [Online]. Available: http://www.gurobi.com