Cascading-failure-aware Disaster Recovery in Optical Cloud Networks

Guilherme S. Ramalho*, Keiko O. Fonseca*, Daniel F. Pigatto*, Carlos Natalino‡,
Paolo Monti‡, Gustavo B. Figueiredo†, and Juliana de Santi*
* Federal University of Technology - Paraná, Brazil. Email: guilhermeramalho@alunos.utfpr.edu.br
{keiko, pigatto, jsanti}@utfpr.edu.br
† Federal University of Bahia, Brazil. Email: gustavo@dcc.ufba.br
‡ Chalmers University of Technology, Gothenburg, Sweden. Email: {carlos.natalino, mpaolo}@chalmers.se

***Abstract*—In optical cloud networks, data center nodes process cloud services requested by users. Disasters can damage critical infrastructure elements and possibly trigger overloading or malfunctioning of nearby apparatuses generating cascading failures with severe consequences for the operation of the entire infrastructure. Providing optical connectivity services in the presence of post-disaster cascading failures is crucial. This paper proposes a restoration strategy that combines the ability to recover service after a disaster event while leveraging information about potentially correlated cascading failures. Simulation results show how this strategy successfully reduces the chances of a service being disrupted multiple times by a disaster event and its cascading failures.**

***Index Terms*—disaster, restoration, relocation**

I. Introduction

Optical Cloud Networks (OCNs) encompass distributed Data Centers (DCs) providing clients during a period of time with processing and storage capabilities while being interconnected by high-bit-rate and low-latency optical connectivity services. OCNs are vulnerable to disasters, natural (e.g., earthquakes, flooding) or human-made (e.g., fiber cuts) [1]. Disasters cause the failure of multiple links and/or nodes and may lead to correlated cascading failures, which happen sometime afterward and are a consequence of the disaster that can damage links/nodes and, thus, overload and disrupt other links/nodes, as well as power disruptions in cascade [2]. As cloud-based services are becoming increasingly important, network and cloud operators must implement strategies to guarantee the resilience of their infrastructures in the presence of disasters, including their correlated aftermaths.

OCNs survivability can be achieved by either protection or restoration strategies [1]. Protection methods reserve backup resources before failures. Restoration strategies, on the other hand, provide an acceptable trade-off between resource usage and reliability guarantees by re-provisioning interrupted services based on the post-disaster available resources. Moreover, restoration performance can be improved by cloud service relocation [3], [4], i.e., migrating a cloud service to a different DC if the original one providing the service is not reachable anymore due to a lack of transport resources.

This paper proposes a restoration strategy for disaster recovery in OCNs. There are a number of studies aimed at improving the disaster recovery performance of restoration strategies [1]. However, the strategy proposed in the paper is the first that jointly takes advantage of service relocation in combination with cascading failures risk awareness for disaster recovery in OCN networks. The rationale is to restore disrupted services using links/DC that reduce the risk of being re-disrupted by another failure stemming from the same disaster that disrupted them in the first place. Simulation results indicate that the proposed strategy reduces the number of services affected by cascading failures, while not impacting the performance in terms of blocking ratio and number of relocations when compared to a strategy that does not leverage cascading failure awareness.

II. System Model and Proposed Strategy

Cascading failures are events that affect elements of the network that are close to the epicenter or a disaster. For example, Fig. 1 illustrates the epicenter of a disaster that may propagate and disrupt the areas around the epicenter. The closeness (physically or in terms of dependencies) to the epicenter defines how likely an element may be impacted (e.g., 73%, 15%, and 5% in Fig. 1) [2].

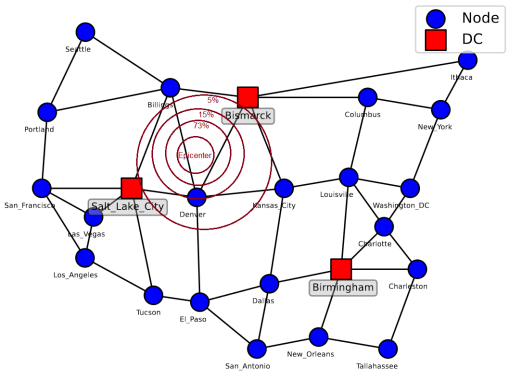

Fig. 1. USA topology with an example of disaster including the potential cascading failures.

This work has been supported by VINNOVA as part of the project "Smart city concepts in Curitiba – low-carbon transport and mobility in a digital society", and by UTFPR - DIREC, Edital No. 11/2021.

978-3-903176-54-6

Assuming a scenario where requests from users (to a DC) have independent arrivals and departures, the performance of the network is controlled by the provisioning and restoration strategies. A service request $r=<s, h, tu, pu>$ that specifies the source node (s), the holding time (h), and the number of the transport and processing units requested (tu and pu, respectively). Since our proposed restoration approach works independently from the provisioning strategy, the provisioning can use e.g. closest-DC or load-balancing strategies.

A. The Path Restoration with Failure Probability Awareness

We propose a strategy called **P**ath **R**estoration with Failure **P**robability **A**wareness (**PRPA**), which restores services considering the trade-off between the transport resources used by the restoration path and the risk of being affected by cascading failure. We formally introduce the proposed PRPA strategy in Algorithm 1. It uses an auxiliary graph G to represent the post-disaster OCN, and it is executed whenever a set S of provisioned services are disrupted due to a disaster.

Algorithm 1: The PRPA strategy

Input: Graph $G = (V, E)$ for post-disaster OCN; set S of disrupted services.
Output: Each service $s_i \in S$ restored or dropped.

```
1  for each service s_i ∈ S | s_i^rt > s_i^dt do
2      K ← k-shortest-paths for s_i (source to cur. DC)
3      if K ≠ ∅ then
4          selPath ← min_cost (K) [Eq. (1)]
5          Restore s_i on selPath
6      else
7          selPath = ∞
8          for DC_n ∈ N_DC | DC_n^pu ≥ s_i^pu do
9              K ←k-shortest-paths from source to DC_N
10             if K ≠ ∅ then
11                 relPath ← min_cost (K) [Eq. (1)]
12                 if cost(relPath) < cost(selPath) then
13                     selPath ← relPath
14         if selPath ≠ ∞ then
15             relocateAndRestore s_i on selPath
16         else
17             drop s_i
```

The algorithm iterates over each service $s_i \in S$, testing whether or not the remaining service time (rt) for s_i is greater than its restoration time (dt) (line 1). If this is not the case, the service is dropped (i.e., there is not enough time to restore it). Then, the algorithm computes the set K of shortest routes for s_i from its source to the DC currently used (line 2). If there exist routes with enough transport resources for s_i (line 3), then the one with the lowest cost (defined by (1) explained in the following) is selected (line 4). Otherwise, the algorithm checks whether it is possible to connect s_i to a different DC_n with processing capacity ($DC_n^{pu} \geq s_i^{pu}$) enough to provision s_i (lines 7–15). The path with the lowest cost (according to (1)) to a DC is adopted and s_i is relocated and restored using $selPath$ (line 15). Otherwise, the service is dropped (line 17).

To calculate the cost of adopting a path for restoration, the following cost function is used:

$$C(p_i) = (prob \times \alpha) + \left(\frac{hops}{maxHops} \times (1 - \alpha) \right) \quad (1)$$

where $C(p_i)$ is the cost utilization of path p_i; $prob \in [0, 1]$ is the probability of p_i being affected by a cascading event; α determines the weight for $prob$; $hops$ is the number of hops of p_i; $maxHops$ is the highest number of hops among the k candidate paths; $(1 - \alpha)$ determines the weight of hops to $C(p_i)$; k=10 routes with the least number of hops. Thus, a path with a low probability of failure and a small number of hops will have a low cost, making it a potential candidate to accommodate a restored service. The α parameter allows us to set the importance given to the two involved metrics.

III. Performance evaluation

We conduct simulations to assess the benefits of the proposed PRPA strategy. The knowledge of cascading failure risk is used only during the restoration phase (Sec. II-A), and it does not influence how cloud services are initially provisioned in the network. The closest available DC strategy is used for service provisioning. As a baseline for comparison, we use the Path Restoration with service Relocation (PRwR) algorithm [3] that is agnostic to the risk of cascading failures. If restoring the service to its current DC, the PRwR selects DCs based on the shortest-path criteria.

The simulation considers the USA topology (Fig. 1), with 24 nodes, 43 links with 80 wavelengths in each direction, and full wavelength conversion capability. Three DC nodes are selected based on their connectivity and are equipped with $1,800$ processing units. Three hundred simulation runs were carried out for each point in the curves presented in this section. Each simulation run involved 100,000 cloud services following a Poisson process. The holding time of each cloud service and disaster duration are exponentially distributed with a mean equal to, respectively, 86,400 and 43,200 units. The requests source are uniformly distributed among all non-DC nodes in the network. Each request demands the bandwidth corresponding to one wavelength channel. The number of processing units required by each cloud service is uniformly chosen in the interval $[1, 5]$. Fig. 1 shows one of the ten disaster zones considered (epicenter and cascades with the considered probabilities)[1], each one repeated twice during the simulation time. The time between two consecutive epicenter disasters is uniformly distributed through the number of service requests, whereas the time between two consecutive cascades is 3,600 time units. Three cascading events are considered per epicenter, and their occurrence depends on their relative proximity to the epicenter with 73%, 15%, and 5% probability [2]. The service reconfiguration time is 1,800 time units.

[1]Other zones available on GitHub https://github.com/GSRamalho/python-simple-anycast-wdm-simulator/zones

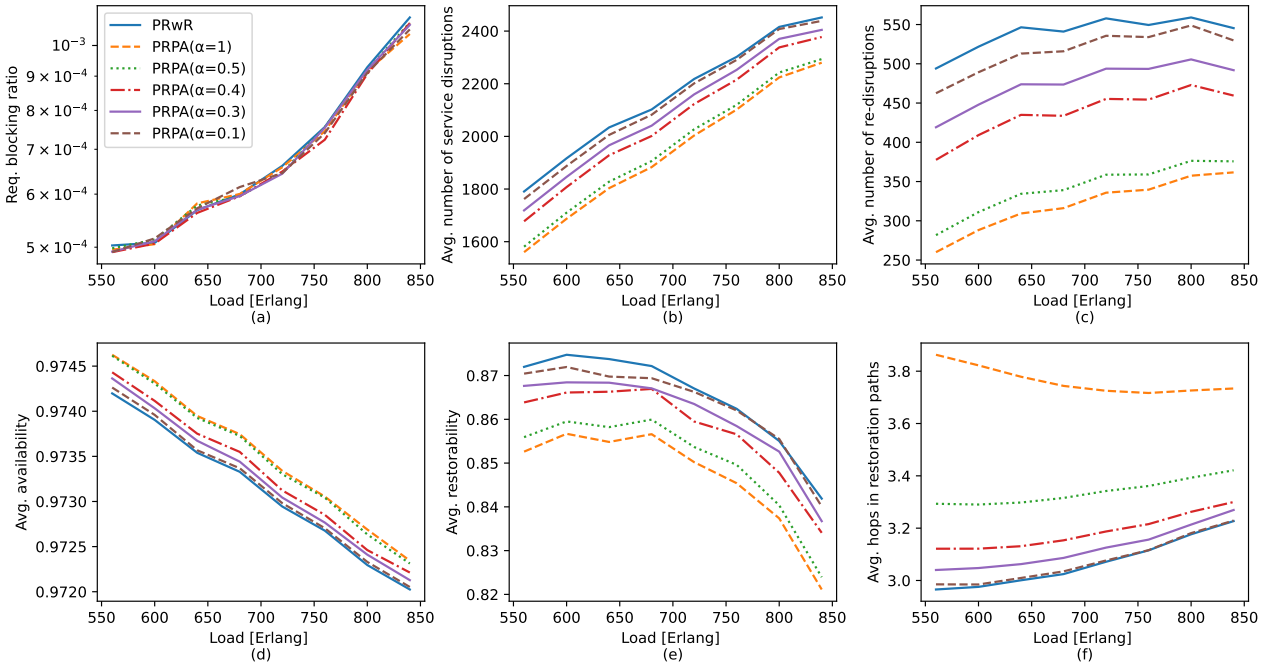

Fig. 2. Results of the PRPA strategy with different α values.

Figure 2 shows the results of the simulations as a function of the network load. To verify the impact of the number of hops and cascading failure probability in the restoration decisions, the proposed PRPA algorithm was executed considering different weights for α – see (1). Fig. 2(a) shows that the blocking ratio presented by all compared algorithms is very similar, indicating that our strategy does not impact the blocking ratio. Fig. 2(b) show the average number of service disruptions per simulation campaign (i.e., 100,000 arrivals). The performance of PRPA is defined by α, where higher values of α lead to higher impact of the cascading failure risk to the decision. PRPA with α=0.5 reduces the number of service disruptions by around 11%. The reason for the lower number of disruptions can be explained by analyzing Fig. 2(c), where we have the number of services that were disrupted more than once. In this case, PRPA with α=0.5 reduces the number of services that are re-disrupted by more than 40%. The lower number of disruptions achieved by PRPA contributes to higher availability (i.e., ratio between the sum of the uptime of services and the sum of their holding time values) than PRwR, as illustrated in Fig. 2(d). Fig. 2(e) shows that, although availability achieved by PRPA is higher than the one achieved by PRwR, the restorability is lower. This is explained by the fact that the number of disruptions in PRPA is lower, leading to a lower number of opportunities to restore services. Moreover, as Fig. 2(f) shows, the restoration paths used by PRPA are slightly longer than the ones used by PRwR. This is explained by the fact that, by avoiding links with potential cascading effects, PRPA needs to deviate from the surroundings of the disaster epicenter.

IV. Conclusion

This paper introduced PRPA, a disaster restoration strategy for OCNs. PRPA leverages the knowledge of the risk links and DSCs being down due to cascading failures. Simulation results show how PRPA is able to outperform benchmark strategy that does not consider cascading failures information during the recovery operations. The total number of service disruptions, and especially the number of re-disruptions, are substantially reduced while maintaining blocking ratio values similar to non-cascading-failure-aware methods. Another added benefit is in terms of better service availability levels thanks to the reduction in the number of disruptions. On the other hand, the consideration of potential cascading failures leads to a slightly lower restorability, mainly caused by the use of longer restoration paths.

In the future work, it would be interesting to extend this study with additional cascading event modeling. The investigation of the monetary consequences of disasters is also an interesting topic. Finally, investigating the disruption of other network elements (e.g., DCs) can also require new strategies.

References

[1] J. Rak, R. Girão-Silva, T. Gomes, G. Ellinas, B. Kantarci, and M. Tornatore, "Disaster resilience of optical networks: State of the art, challenges, and opportunities," Optical Switching and Networking, vol. 42, p. 100619, 2021, DOI: 10.1016/j.osn.2021.100619.

[2] C. Colman-Meixner, M. Tornatore, and B. Mukherjee, "Cloud-network disaster recovery against cascading failures," in GLOBECOM, 2015, DOI: 10.1109/GLOCOM.2015.7417558.

[3] C. Natalino, L. Wosinska, S. Spadaro, J. C. W. A. Costa, C. R. L. Frances, and P. Monti, "Restoration in optical cloud networks with relocation and services differentiation," Journal of Optical Communications and Networking, vol. 8, no. 2, pp. 100–111, 2016, DOI: 10.1364/JOCN.8.000100.

[4] R. Goścień, "Traffic-aware service relocation in software-defined and intent-based elastic optical networks," Computer Networks, p. 109660, 2023, DOI: 10.1016/j.comnet.2023.109660.