

ABNO-driven Content Distribution in the Telecom Cloud

Lluís Gifre^{1*}, Massimo Tornatore^{2,3}, Luis M. Contreras⁴, Biswanath Mukherjee³, and Luis Velasco¹

¹Universitat Politècnica de Catalunya (UPC), Barcelona, Spain.

²Dipartimento di Elettronica Informazione e Bioingegneria, Politecnico di Milano, Italy

³Department of Computer Science, University of California, Davis, USA

⁴Telefónica Investigación y Desarrollo (TID), Madrid, Spain

*Corresponding author: lgifre@ac.upc.edu

Abstract—As current sustained traffic growth is expected to strain capacity of today's metro network, novel content distribution architectures where contents are placed closer to the users are being investigated. In that regard, telecom operators can deploy datacenters (DCs) in metro areas, thus reducing the impact of the traffic going from users to DCs. However, effective solutions to distribute contents to those metro DCs and to synchronize contents among them need to be investigated. In this paper, a hierarchical content distribution architecture for the telecom cloud is investigated: core DCs placed in geographically distributed locations, are interconnected through permanent “per content provider” (CP) virtual network topology (CP-VNT); additionally, metro DCs are placed close to users, and need also to be interconnected with the core DCs. CP's data is replicated in the core DCs through the CP-VNT, while metro-to-core (M2C) anycast connections are established periodically for metro DCs to synchronize contents. Since failures in the interconnection network might disconnect the CP-VNTs, in this paper recovery mechanisms are proposed to reconnect topologies and anycast connections. Topology creation, anycast provisioning and recovery problems are first formally stated and modelled as Integer Linear Programs (ILP). Solving these models, however, becomes impractical for real-sized scenarios, so heuristic algorithms are also proposed. Exhaustive simulation results show that the investigated architecture provides significant improvements in both supported traffic and restorability compared to the case of a single core DC. Workflows to implement the algorithms within the Applications-based Network Operations (ABNO) architecture and extensions for PCEP are proposed. Finally, the architecture is experimentally validated in our ABNO-based iONE test-bed.

Keywords: Datacenter interconnection, Optical networks, Content distribution.

1 INTRODUCTION

Cloud-based contents are experiencing an exponential growth in users and data volume. In [1], Cisco forecast a yearly global growth of 23% in datacenter (DC)-to-user traffic, 29% in DC-to-DC traffic, and 22% in the intra-DC traffic. Likewise, Alcatel-Lucent highlights that metro traffic will increase 560% by 2017 mainly driven by IP video and cloud traffic [2], growing two times faster than core network traffic. They conclude that placing contents in DCs closer to users can result in a 41% bandwidth reduction in the metro networks. The resulting distributed DC architecture should come together with the extension of the core network functionalities towards the metro, as proposed in [3].

Robust telecom infrastructures are needed to minimize the impact of failures on the services [4]; this is especially relevant in the case of video distribution. In fact, prolonged service outages and disruptions have negatively impacted the confidence of important companies toward cloud environments. Telecom operators are in a vantage position to capitalize on highly-reliable cloud services since they own the infrastructure to support on-demand delivery of processing, storage, and network resources, which allow them developing a *telecom cloud* infrastructure [5]. That connected cloud infrastructure, consisting of datacenters and telecom networks, can provide highly-reliable ultra-low latency cloud services by placing computing resources closer to the end-users thus, increasing quality of experience. Network operators can use the telecom cloud for many purposes, such as to distribute live-TV [6]. In addition, content providers (CP) can take advantage of the telecom cloud to increase content availability by replicating and hosting contents in different geographically-distributed DCs to prevent data losses in the event of failures or disasters [7], [8]. Distributed DCs can synchronize replicated contents using any available approach including distributed databases, such as Apache Cassandra [9]. Regardless of content replication, it is of paramount importance deploying survivable DC connectivity since otherwise contents might become outdated or even unavailable.

In this paper, we investigate a hierarchical content distribution architecture where geographically-distributed core DCs manage contents in the core, whereas metro DCs are placed closer to the users. Core DCs are interconnected to create a per-CP virtual network topology (CP-VNT) that must be survivable against failures. In contrast to core DCs, that are always connected through the CP-VNT, the metro DCs periodically request metro-to-core (M2C) connectivity to any of the core DCs so as to synchronize contents. Therefore, to implement the architecture, the following optimization

problems need to be solved: CP-VNT creation, M2C *anycast* connection provisioning, and service recovery in the event of a network failure.

Since telecom cloud infrastructures need the transfer of massive amounts of data between DCs, the high-capacity connectivity provided by optical networks is needed. The traditional fixed-grid Wavelength Division Multiplexing (WDM) or the recently available flexgrid technology [10] can be used to create express Gb/s optical connections, also called *lightpaths*. Authors in [11] compare the performance transparent/translucent vs. opaque optical networks and conclude that for medium-scale networks translucency improves network performance. However, higher degree of opacity may be needed for larger-scale networks.

As for the control plane, the IETF is currently standardizing the Applications-based Network Operations (ABNO) [12] architecture to control telecom networks. ABNO includes a controller able to implement workflows and a Path Computation Element (PCE) [13] module to deal with Label Switched Path (LSP) computation; internally, the ABNO modules can use the PCE Protocol (PCEP) [14] while offering a north bound interface for external systems, such as the Network Management System (NMS), to request connectivity. Based on the capabilities of the ABNO architecture to implement complex workflows, authors in [15], [16] proposed re-optimizing the network while in operation.

1.1 Related work

Several research studies related to the cloud and network cross-stratum orchestration have been lately carried out. The authors in [17] study the cost savings obtained by inter-connecting the DCs belonging to a federation. In [18], the authors propose an architecture to provide dynamic bandwidth on demand over WDM; in case of failure, dynamic restoration is used. A specific module running on top of ABNO, named as Application Service Orchestrator (ASO), was first proposed in [19] to coordinate DC and network elastically aiming at improving resource utilization and save costs. In [20], [21] the ASO was used to adapt the specific semantic of cloud to that of the network and it was experimentally assessed in [22].

With respect to anycast connections, the authors in [23] consider both available network resources and IT resource requirements to compute an optimal route for dynamic connections. In [24] an enhanced control plane following a hierarchical PCE architecture is proposed for jointly perform network routing and IT server provisioning decisions applicable to anycast scenarios for IT resource selection. Routing and spectrum allocation algorithms are proposed in [25] for serving anycast connections over elastic optical networks; to select the destination of the anycast connection, a set of candidate paths is computed.

As for connectivity recovery, authors in [26] propose a fast spanning tree *reconnection* mechanism to handle single link failures in Ethernet-based networks. In [27], authors studied the problem of shared path protection for anycast connections against single link failures. Multi-path restoration was proposed in the context of flexgrid networks in [28] to improve restorability, i.e. the percentage of traffic restored with respect to the total affected by a failure. Authors in [29] proposed a bulk restoration mechanism where restoration requests are jointly computed aiming at finding a global optimal solution.

1.2 Contributions

The contribution of this paper is many-fold: *i)* firstly, Integer Linear Program (ILP) formulations and algorithms for the identified optimization problems are devised. CP-VNTs are created as tree topologies where optical links are used only once so any single link failure disconnects CP-VNTs into two connected graph components. A general method to reconnect CP-VNTs under single and multiple failures is devised. Augmented network graphs to reduce problems' complexity are proposed and used for anycast provisioning and CP-VNT reconnection; *ii)* the performance of the hierarchical content distribution architecture is evaluated through simulation and the appropriateness of different configurations is studied; *iii)* workflows to implement the algorithms within the ABNO architecture and extensions for PCEP are proposed; in particular, support for anycast path computation is studied; *iv)* workflows and algorithms are experimentally assessed in our ABNO-based iONE test-bed.

The rest of this paper is organized as follows. In section 2 we describe the proposed hierarchical content distribution architecture and we identify the optimization problems to be solved. These problems are formally stated, mathematically formulated as ILPs, and algorithms are eventually proposed for solving them in section 3. Next, in section 4, an ABNO-based architecture is proposed to be used in the proposed architecture, the workflows to implement the proposed algorithms on an ABNO-based control plane are developed and protocol specifics, including the proposed extensions, are studied. Section 5 is devoted to extensively evaluate the efficiency and to experimentally assessing the proposed content distribution architecture. Finally, section 6 draws the main conclusions.

2 HIERARCHICAL CONTENT DISTRIBUTION ARCHITECTURE

The content distribution architecture under study is hierarchical: content replicas are first distributed over a set of core DCs and then, to reduce the traffic in the core, a set of metro DCs are placed closer to the users. Fig. 1 shows the

architecture for a single CP. A CP private Virtual Network Topology (CP-VNT) is created by establishing *soft-permanent* [30] Core-to-Core (C2C) virtual links connecting core DCs; this CP-VNT allows core DCs to be permanently synchronized, i.e. any modification performed in the contents in one core DC is propagated to the rest of the core DCs through the CP-VNT. In contrast, contents in metro DCs are synchronized periodically setting-up Metro-to-Core (M2C) connections to any of the DCs belonging to the CP-VNT (anycast connections).

To illustrate how the architecture operates, let us assume a scenario where a distributed database is deployed for research purposes and users (e.g., a research center, a hospital, and a university) interact with contents. In the example, user 1, user 2 and metro DC-1 are placed in geographical area 1, while user 3 and metro DC-6 are in geographical area 2. User 1 interacts with the contents in metro DC-1 and uploads new content; user 2 will see the updated content immediately. Conversely, user 3 will have access to the updated content only when contents are synchronized to the core DCs and metro DC-6 synchronizes its contents with the core. To that end, after a synchronization event is triggered, a M2C connection is set-up between metro DC-1 and any of the core DCs in the CP-VNT to synchronize contents. Likewise, a M2C connection needs to be set-up from metro DC-6, so that its contents are synchronized and user 3 will have access to the updated content.

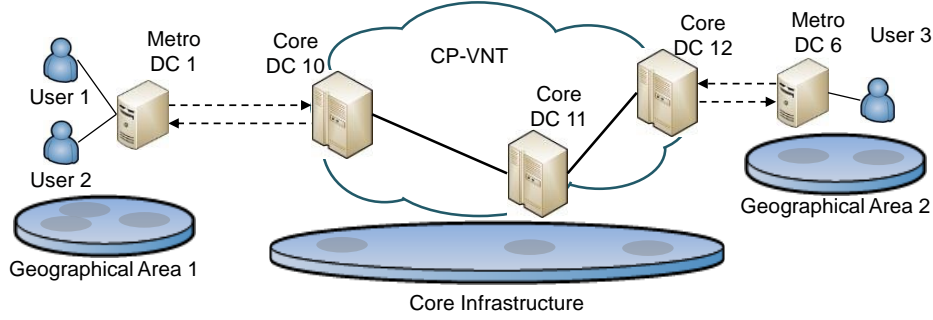


Fig. 1. Hierarchical Telecom Cloud architecture

The proposed architecture brings benefits, including the reduction of traffic in the core as well as its inherently high availability against DC failures. However, as a result of considering several core DCs, a CP-VNT topology needs to be created among them. A failure affecting the CP-VNT might disconnect it creating several subsets of connected core DCs (connected components) whose contents become outdated. Therefore, the CP-VNT topology should be designed to cope with network failures. In that regard, *pre-planned reconnection* (similar to protection) and *dynamic reconnection* (similar to restoration) schemes [31] can be considered to *reconnect* the CP-VNT topology in the event of a failure. In light of the significantly lower amount of resources used when dynamic reconnection is implemented compared to pre-planned reconnection, and the short recovery times that can be achieved (in the order of hundreds of ms. [32]), in this paper we assume the CP-VNTs are created as a tree topology and dynamic reconnection is used. Similarly, restoration will be used to recover M2C anycast connections from failures.

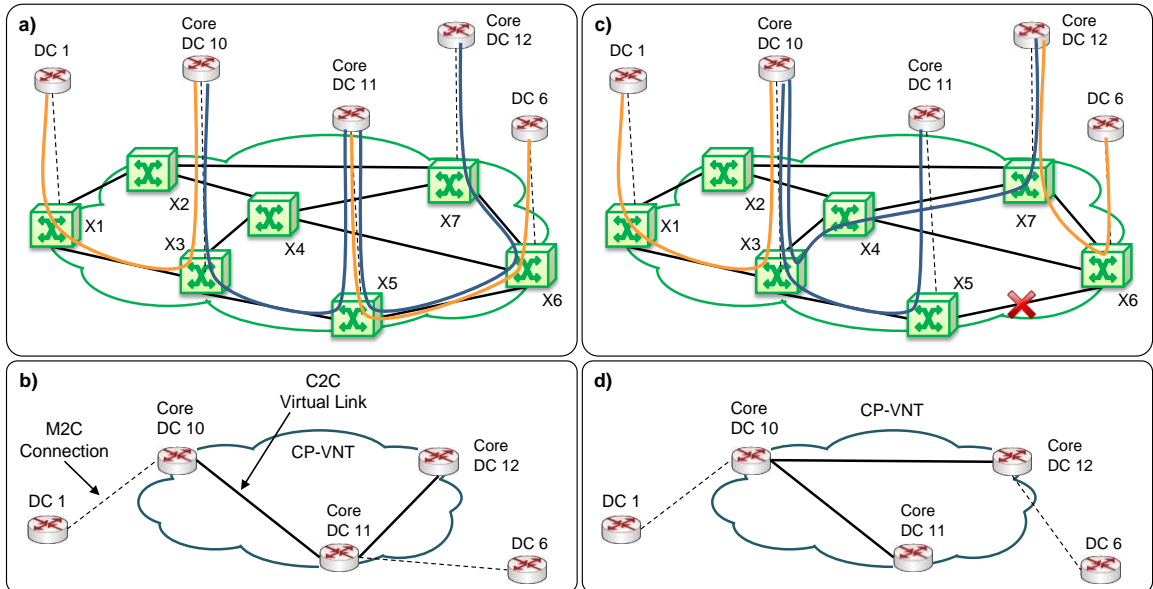


Fig. 2. CP-VNT with two M2C anycast connections. Physical (a) and virtual (b) topologies under normal conditions. Physical (c) and virtual (d) topologies after a failure in link X5-X6.

Fig. 2 presents an example with 3 core DCs (labelled as 10, 11 and 12) and 2 metro DCs (labelled as 1 and 6); we assume that a Layer 2 (L2) switch connects each DC to the optical transport network. Core DCs are connected among them by a set of C2C virtual links supported by lightpaths set-up over the optical transport network (blue lines in Fig. 2a) forming a CP-VNT (Fig. 2b). When a metro DC needs to synchronize its local contents with those in the CP-VNT, it establishes a dynamic M2C connection to any of the core DCs belonging to the CP-VNT. For example, in Fig. 2a, two M2C connections (orange lines) are established: 1-10 and 6-11.

In the event of a failure, C2C as well as M2C connections can be affected; in Fig. 2c a link failure has torn-down connections 6-11 and 11-12, thus breaking the original virtual network into two connected graph components ($\{10, 11\}$ and $\{12\}$). Therefore, the CP-VNT needs to be urgently reconnected into a single connected topology; only after the CP-VNT is reconnected keeping contents updated in the core, M2C connections will be recovered. In Fig. 2c the CP-VNT has been reconnected by creating the new C2C virtual link 10-12. Fig. 2d shows the new CP-VNT topology after the reconnection. Now, also affected M2C connections can be restored; in Fig. 2c the failed anycast connection 6-11 has been restored by creating the connection 6-12, thus *reconnecting* metro DC6 to the CP-VNT.

Hence, three problems have been identified: CP-VNT creation, M2C anycast provisioning, and CP-VNT and M2C reconnection. The next section formally defines each problem and proposes mathematical models and algorithms for solving them.

3 MATHEMATICAL FORMULATIONS AND ALGORITHMS

3.1 CP-VNT_CREATE Problem

This problem consists in finding a tree connecting all core DCs selected by a particular CP. Our approach consists in creating an *auxiliary full-mesh* graph connecting all involved core DCs. Then, a Minimum Spanning Tree (MST) [33] is computed over the auxiliary graph. The links in the resulting tree are the lightpaths that are set-up using a node-link formulation [34]; we assume an opaque optical core network, so contiguity and continuity constraints are relaxed. To improve the availability of the CP-VNT, each optical link is used just once on each CP-VNT; therefore, a single link failure would disconnect CP-VNTs into two connected graph components at the most.

The CP-VNT_CREATE problem can be stated as follows:

Given:

- the optical network topology represented by a graph $G(N_o, L)$, where N_o is the set of optical nodes and L the set of optical links.
- the number of available slices η_l in each optical link $l \in L$,
- the CP c that requests the CP-VNT,
- the number of slices χ_l for the virtual links of each VNT.

Output: the route for each bidirectional lightpath.

Objective: minimize the number of optical resources required.

The sets and parameters for this problem (many of them are common for the other problems) are:

Topology:

- N set of all nodes in the network, e.g., optical nodes in the network and Ethernet switches in the DCs, index n .
- N_o subset of N containing the optical nodes in the network.
- N_c subset of N containing the switches in the core DCs.
- L set of optical links connecting two nodes, index l .
- L_o subset of L containing the optical links connecting two optical nodes.

Content Providers:

- C set of CPs, index c .
- $N_c(c)$ subset of N_c containing the core DCs for CP c .
- E set of candidate virtual links connecting two switches in $N_c(c)$, index e .
- $OD(e)$ the subset of $N_c(c)$ containing the endpoints of candidate virtual link e .

Other parameters:

- δ_{nl} 1 if link l is incident to node n ; 0 otherwise.
- χ_1 number of slices required for each C2C connection.
- χ_2 number of slices required for each M2C connection.
- η_l number of available spectrum slices in link l .

The decision variables are:

- x_{el} binary, 1 if optical link l is used to support virtual link e in the CP-VNT; 0 otherwise.
- y_e binary, 1 if virtual link e is selected; 0 otherwise.

Then, the formulation for the CP-VNT_CREATE problem is as follows:

$$(CP-VNT_CREATE) \quad \text{Min} \quad \sum_{e \in E} \sum_{l \in L} x_{el} \quad (1)$$

subject to:

$$\sum_{e \in E} y_e = |N_C(c)| - 1 \quad (2)$$

$$\sum_{e \in (S, \bar{S})} y_e \geq 1 \quad \forall S \subset N_C(c) \quad (3)$$

$$\sum_{l \in L} \delta_{nl} \cdot x_{el} = y_e \quad \forall e \in E, n \in OD(e) \quad (4)$$

$$\sum_{l \in L} \delta_{nl} \cdot x_{el} \leq 2 \quad \forall e \in E, n \in N \setminus OD(e) \quad (5)$$

$$\sum_{l' \in L, l' \neq l} \delta_{nl'} \cdot x_{el'} \geq \delta_{nl} \cdot x_{el} \quad \forall e \in E, n \in N \setminus OD(e), l \in L \quad (6)$$

$$\sum_{e \in E} \chi_l \cdot x_{el} \leq \eta_l \quad \forall l \in L \quad (7)$$

$$\sum_{e \in E} x_{el} \leq 1 \quad \forall l \in L_O \quad (8)$$

The objective function (1) minimizes the total number of optical links used to deploy the CP-VNT. Constraints (2) and (3) define the tree topology by restricting the CP-VNT to have $|N_C(c)|-1$ virtual links and ensuring connectivity of the nodes belonging to the CP-VNT, respectively. Note that constraint (3) is applied to any subset S of $N_C(c)$. Constraints (4)-(6) route the virtual links through the optical network. Constraint (4) ensures that one lightpath for each demand is created with end nodes equal to the source and destination of demand. Constraint (5) and (6) guarantee that the route of each lightpath is connected and it does not contain any loop. Constraint (7) ensures that selected optical links have enough optical resources, while constraint (8) guarantees that each optical link is used once at the most.

The CP-VNT_CREATE problem requires a quadratic number of candidate virtual links to be considered with respect to the number of core DCs, i.e. $|E| \sim |N_C(c)|^2$. Thus, the number of variables is $O(|N_C(c)|^2 \cdot |L|)$ and the number of constraints is $O(2^{|N_C(c)|} + |N_C(c)|^2 \cdot |N| \cdot |L|)$. For instance, the number of variables and constraints for the 24-node 43-link US topology used in Section 4, is approximately 10^3 and 10^5 , respectively. The size of the problem can be reduced and solved to optimality using large scale optimization techniques, such as column generation [35]. In this paper, however, a heuristic algorithm is proposed.

The algorithm in Table I will be reused for the CP-VNT_RECONNECT problem, so the *mode* parameter allows to specify the desired behavior; for the CP-VNT_CREATE algorithm a value of *CREATE* is expected. The algorithm starts creating a *full-mesh* graph $G(N_C(c), E)$ and assigning a metric to each link in E as a function of number of optical hops for the lightpath supporting that virtual link (lines 1-3). Then, a MST is computed over $G(N_C(c), E)$ (line 6). The links in the tree are iteratively routed over the optical network (lines 9-16) and optical links already used (set L_R) are not considered for routing new lightpaths (line 10). If no path is found for a virtual link, it is removed from the *full-mesh* graph and the routing restarts (lines 11-13); otherwise, the path is added to the *LSP* set and sets are updated (lines 14-16). The solution in *LSP* is eventually returned (line 17).

Table I. CP-VNT_CREATE Heuristic Algorithm

INPUT: $G(N, L)$, c , N_C , χ_l , <i>mode</i>	
OUTPUT: <i>LSP</i>	
1:	$G(N_C(c), E) \leftarrow \text{Create_Full_Mesh_Undirected}(N_C(c))$
2:	for each $e=(a, b)$ in E do
3:	$e.\text{metric} \leftarrow \text{Shortest_Path}(G(N_C(c), E), a, b) $
4:	while TRUE do
5:	$L_R \leftarrow \emptyset$; $LSP \leftarrow \emptyset$
6:	$E_T \leftarrow \text{MST}(G(N_C(c), E))$;
7:	if $E_T = \emptyset$ then return \emptyset
8:	while $E_T \neq \emptyset$ do
9:	$e=(a, b) \leftarrow \text{argmax}_{e.\text{metric}}(E_T)$
10:	$p \leftarrow \text{Shortest_Path}(G(N, L \setminus L_R), a, b, \chi_l)$
11:	if $p = \emptyset$ then
12:	$E \leftarrow E \setminus \{e\}$
13:	break
14:	$LSP \leftarrow LSP \cup \{p\}$
15:	if $\text{mode}=\text{CREATE}$ then $L_R \leftarrow L_R \cup p$
16:	$E_T \leftarrow E_T \setminus \{e\}$
17:	if $E_T = \emptyset$ then return LSP

3.2 M2C-PROV Problem

The M2C-PROV problem aims at finding an anycast connection between a selected metro DC and one of the core DCs in the target CP-VNT. The problem can be stated as follows:

Given:

- the network topology represented by a graph $G(N, L)$,
- the number of available slices in each link $l \in L$,
- the CP c that requests the CP-VNT,
- the number of slices for anycast connection.

Output: the route for the bidirectional lightpath.

Objective: minimize the number of optical resources required.

The approach we devised to solve this problem consists in building a graph with links connecting the metro DC to all the core DCs in the CP-VNT. As before, the metric of each link is a function of number of optical hops for the lightpath supporting that link. The graph is augmented adding one *dummy* node and links connecting each core DC to the dummy node; see Fig. 3a, where the dummy node is labelled as D . Then, a shortest path is computed between the source metro DC and the dummy node. The resulting path contains the optimal route for the M2C anycast connection to be established (Fig. 3b).

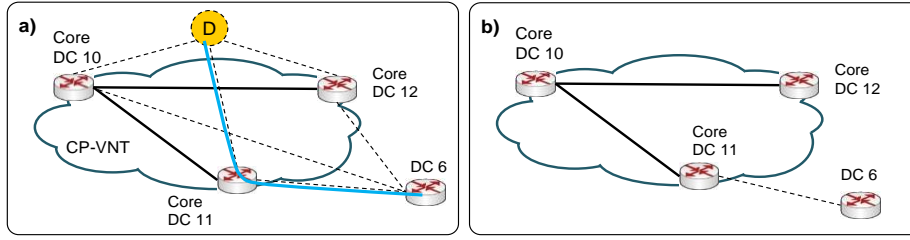


Fig. 3. M2C provisioning.

3.3 RECONNECT Problem

The RECONNECT problem aims at reconnecting the CP-VNT topologies and the M2C anycast connections affected by a network failure that might involve several nodes and/or links. Recall that reconnecting the CP-VNTs is of first priority reduce contents in the core becoming outdated.

As for the M2C-PROV problem, our approach for solving this problem consists in creating an augmented graph where *dummy* nodes and links are added. For the CP-VNT reconnection problem, one dummy node for each connected component is added and connected to every core DC in that connected component. Fig. 4a illustrates a CP-VNT topology where virtual links 10-11 and 11-12 have failed. Three dummy nodes have been added (labelled as $D1$, $D2$, and $D3$ in Fig. 4b). Then, finding a tree topology connecting these dummy nodes will reconnect the connected components into a connected CP-VNT topology. We adapt the approach proposed for the CP-VNT creation problem to compute the set of links for the tree and to find feasible lightpaths to support the virtual links (Fig. 4c), eventually reconnecting the CP-VNT (Fig. 4d).

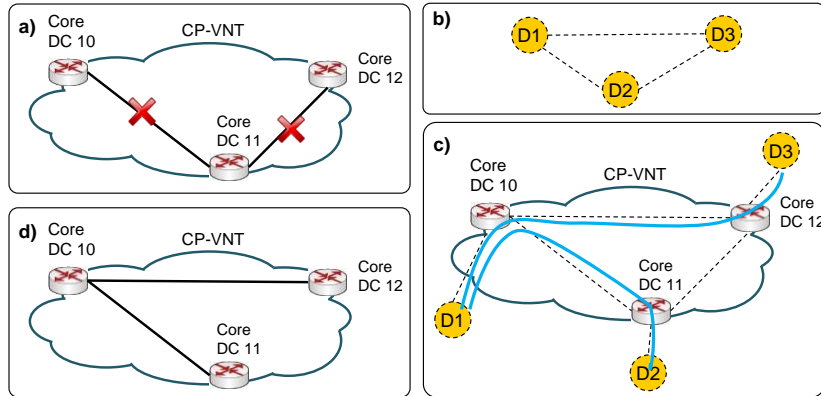


Fig. 4. CP-VNT reconnection.

Affected M2C anycast connections are only recovered provided that the associated CP-VNT was either not affected by any failure or already reconnected. Apart from that, the approach is similar to the M2C-PROV problem. Then, the RECONNECT problem can be stated as follows:

Given:

- the network topology represented by a graph $G(N, L)$ where the failed elements (optical links or nodes) have been removed,
- the number of available slices η_l in each link l ,
- the set of disconnected CP-VNTs, each of them represented by a graph $G_c(N_C(c), E(c))$, where $E(c)$ is the set of operational virtual links after removing those affected by the failure. It is worth noting that even in the case of a single optical link failure, several virtual links might be affected,
- the set of CP-VNTs not affected by any failure,
- the set D with the failed M2C anycast connections,
- the number χ_1, χ_2 of slices required for each connection type,

Output: the route for the bidirectional LSPs used to reconnect the CP-VNTs and to recover the failed M2C connections.

Objective: maximize the number of connected CP-VNTs and (secondarily) the recovered M2C demands.

In addition to the sets and parameters previously defined, new ones are defined:

CP-VNT Reconnection

$N_D(c)$ set of dummy nodes corresponding to the connected components in VNT of CP c .

A includes the subset A_C of candidate C2C virtual links connecting pairs of dummy nodes in $N_D(c)$ for CP-VNTs and the subset A_M of M2C virtual links connecting the source of affected anycast connections to the dummy node for its target CP-VNT, index a .

$A_C(c)$ subset of A_C containing the candidate C2C virtual links belonging to the VNT of CP c .

$OD(a)$ subset of N containing the endpoints of virtual link a .

α weight of disconnected CP-VNTs in the objective function.

M2C Recovery

D set of M2C anycast connections to be reconnected, index d .

$D(c)$ subset of D containing connections belonging to CP c .

$A_M(d)$ the specific M2C virtual link in A_M for anycast connection d .

δ_c 1 if VNT for CP c is already connected; 0 otherwise.

The decision variables are:

x_{al} binary. 1 if virtual link a is supported by optical link l ; 0 otherwise.

y_a binary. 1 if virtual link a is routed; 0 otherwise.

z_c binary. 1 if VNT of CP c is reconnected; 0 otherwise.

Then, the ILP formulation is as follows:

$$(RECONNECT) \quad \text{Max} \quad \alpha \cdot \sum_{c \in C} z_c + \sum_{d \in D} \sum_{a \in A_M(d)} y_a \quad (9)$$

subject to:

$$\sum_{a \in A_C(c)} y_a \leq |N_D(c)| - 1 \quad \forall c \in C \quad (10)$$

$$\sum_{a \in (S, \bar{S})} y_a \geq 1 \quad \forall c \in C, S \subset N_D(c) \quad (11)$$

$$\sum_{a \in A_C(c)} y_a \geq (|N_D(c)| - 1) \cdot z_c \quad \forall c \in C \quad (12)$$

$$\sum_{l \in L} \delta_{nl} \cdot x_{al} = y_a \quad \forall c \in C, a \in A_C(c), n \in OD(a) \quad (13)$$

$$\sum_{l \in L} \delta_{nl} \cdot x_{al} \leq 2 \quad \forall c \in C, a \in A_C(c), n \in N \setminus OD(a) \quad (14)$$

$$\sum_{l' \in L, l' \neq l} \delta_{nl'} \cdot x_{al'} \geq \delta_{nl} \cdot x_{al} \quad \forall c \in C, a \in A_C(c), n \in N \setminus OD(a), l \in L \quad (15)$$

$$y_a \leq z_c + \delta_c \quad \forall c \in C, d \in D(c), a \in A_M(d) \quad (16)$$

$$\sum_{l \in L} \delta_{nl} \cdot x_{al} = y_a \quad \forall d \in D, a \in A_M(d), n \in OD(a) \quad (17)$$

$$\sum_{l \in L} \delta_{nl} \cdot x_{al} \leq 2 \quad \forall d \in D, a \in A_M(d), n \in N \setminus OD(a) \quad (18)$$

$$\sum_{l' \in L, l' \neq l} \delta_{nl'} \cdot x_{al'} \geq \delta_{nl} \cdot x_{al} \quad \forall d \in D, a \in A_M(d), n \in N \setminus OD(a), l \in L \quad (19)$$

$$\sum_{c \in C} \sum_{a \in A_C(c)} \chi_1 \cdot x_{al} + \sum_{d \in D} \sum_{a \in A_M(d)} \chi_2 \cdot x_{al} \leq \eta_l \quad \forall l \in L \quad (20)$$

The ILP combines a MST formulation to reconnect the failed CP-VNTs with a node-link formulation to set-up the underlying lightpaths. The multi-objective function (9) maximizes both, the number of connected CP-VNTs and

recovered M2C anycast connections, where the α parameter gives priority to the former. Constraints (10)-(15) deal with CP-VNT reconnection and are simple extensions to constraints (2)-(6) presented in section 3.1. Constraints (16)-(20) are responsible for recovering the M2C anycast demands. Constraint (16) forces to activate a virtual link related to a M2C connection provided that the corresponding CP-VNT was not affected by failures or it was reconnected. Constraints (17)-(19) routes the activated links through the optical network. Finally, constraint (20) ensures that optical resources used in a link does not exceed its capacity.

The size of the RECONNECT problem is bigger than that of the CP-VNT_CREATE one and it needs to be solved in real time, e.g., less than a second. In view of that, a heuristic algorithm is proposed. Our approach consists in splitting the problem into two sub-problems: CP-VNT_RECONNECT and M2C-RECONNECT. The algorithms are solved sequentially, firstly CP-VNTs are reconnected and then M2C anycast connections are restored.

Table II presents the pseudo-code of the CP-VNT_RECONNECT heuristic algorithm. It makes use of the CP-VNT_CREATE algorithm to reconnect CP-VNTs by using one dummy node for each connected component of the CP-VNT. The algorithm begins finding, for each disconnected VNT, the set of connected components (line 3), and pre-computing the set of dummy nodes and the links connecting them to the core DCs (lines 4-11). Next, solutions for failed VNTs are generated by randomly sorting the list of CP-VNTs and the best solution is stored (lines 12-19). The best solution found is eventually returned (line 20).

Table II. CP-VNT RECONNECT Heuristic Algorithm

INPUT: $G(N, L)$, C , $G(N_c, E)$, $iter$, χ_1	
OUTPUT: LSP	
1:	$N_D \leftarrow \emptyset; E_D \leftarrow \emptyset$
2:	for each c in C do
3:	$P \leftarrow \text{Find_Connected_Components}(G(N_c(c), E(c)))$
4:	$N_{D,c} \leftarrow \emptyset; E_{D,c} \leftarrow \emptyset$
5:	for each $p(N_p, E_p)$ in P do
6:	$v \leftarrow \text{Create_Dummy_Node}()$
7:	$N_{D,c} \leftarrow N_{D,c} \cup \{v\}$
8:	for each n in N_p do
9:	$E_{D,c} \leftarrow E_{D,c} \cup \{(n, v)\}$
10:	$N_D \leftarrow N_D \cup \{N_{D,c}\}$
11:	$E_D \leftarrow E_D \cup \{E_{D,c}\}$
12:	$bestSolution \leftarrow \emptyset$
13:	for $it = 1..iter$ do
14:	$solution \leftarrow \emptyset; C' \leftarrow \text{Random_Sort}(C)$
15:	for each c in C' do
16:	$LSP_c \leftarrow \text{CPVNT_CREATE}(G(N \cup N_{D,c}, L \cup E_{D,c}), N_{D,c}, \chi_1, \text{RECONNECT})$
17:	if $LSP_c \neq \emptyset$ then $solution.paths \leftarrow solution.paths \cup LSP_c$
18:	$cost \leftarrow cost + \text{Cost_Function}(LSP_c, N_{D,c})$
19:	$bestSolution \leftarrow \text{Select_Best}(bestSolution, solution)$
20:	return $bestSolution.paths$

Table III lists the pseudo-code of the heuristic algorithm for the M2C-RECONNECT sub-problem. It uses the M2C-PROV algorithm to reconnect M2C connections. Solutions are generated by randomly sorting the M2C connection list and keeping the best solution, which is eventually returned.

Table III. M2C-RECONNECT Heuristic Algorithm

INPUT: $G(N, L)$, $iter$, D , N_c , χ_2	
OUTPUT: LSP	
1:	$bestSolution \leftarrow \emptyset$
2:	for $it = 1..iter$ do
3:	$solution \leftarrow \emptyset$
4:	$D' \leftarrow \text{Random_Sort}(D)$
5:	for each d in D' do
6:	$p \leftarrow \text{M2C_PROV}(G(N, L), d.src, N_c(d.c), \chi_2)$
7:	if $p \neq \emptyset$ then $solution.paths \leftarrow solution.paths \cup \{p\}$
8:	$cost \leftarrow cost + \text{Cost_Function}(p, L)$
9:	$bestSolution \leftarrow \text{Select_Best}(bestSolution, solution)$
10:	return $bestSolution.paths$

4 WORKFLOWS AND PROTOCOL EXTENSIONS

In this section we focus on implementing the hierarchical content distribution architecture, i.e., deploying the algorithms for identified problems on the control and management planes. In that regard, we assume that the ABNO architecture is used.

The following components are considered (Fig. 5): *i*) the controller that receives network-level requests and implements the workflows to coordinate the rest of components; *ii*) an active stateful front-end PCE (fPCE) [13], [36] with LSP initiation capabilities [37] able to issue requests to the Provisioning Manager (PM), and delegating complex computations to the back-end PCE (bPCE); *iii*) a PM dealing with new connections' set-up; *iv*) a Topology Module storing the network topology in the Traffic Engineering Database (TED) and the set of already established lightpaths in the LSP Database (LSP-DB); and *v*) a bPCE responsible for dealing with computationally intensive algorithms [38], such as CP-VNT creation and reconnection algorithms. The modules in the ABNO architecture interoperate by means of PCEP [14]. Furthermore, two additional components are considered: cloud resource managers, deployed in each DC, issue application-level service requests, and the ASO, which translates them into network-level requests for the network control plane and stores service-related data.

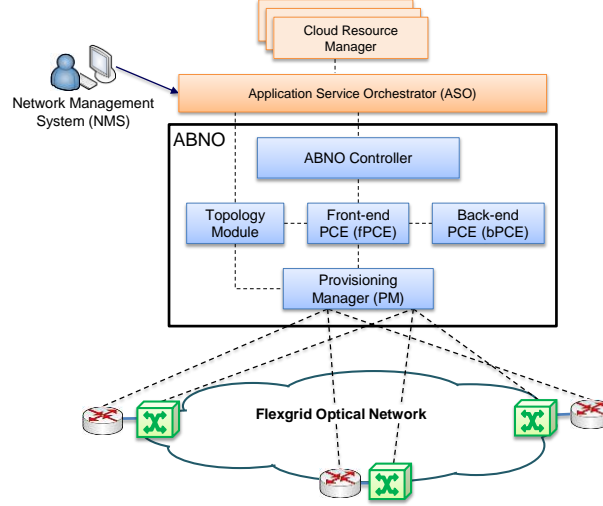


Fig. 5. Considered control and management architecture.

Next, workflows for the identified problems as well as related PCEP issues are analyzed and extensions are proposed.

4.1 CP-VNT_CREATE Workflow

The workflow (Fig. 6) is triggered by the operator through the NMS. Its goal is two-fold, first to add the set of core DCs belonging to the CP-VNT and the metro DCs authorized to connect to the CP-VNT to the ASO's database and second to set-up the CP-VNT itself. For the latter, the ASO issues a CP-VNT creation request message (labelled as 1 in Fig. 6) to the controller with the set of core DCs to be connected. The controller issues a PCRequest (PCReq) message (2) to the fPCE, which delegates the CP-VNT topology computation to the bPCE (3). The bPCE runs the CP-VNT_CREATE algorithm and replies the solution in a PCReply (PCRep) message (4). The fPCE issues a PCInitiate (PCInit) message to the PM so as to implement the related LSPs and waits for the PCReport (PCRpt) messages confirming the implementation of the lightpaths (5). When the CP-VNT topology has been created, the fPCE sends back a PCReply (PCRep) message to the controller (6) with the lightpaths implemented, which replies to the ASO (7) that updates its services database and informs the NMS.

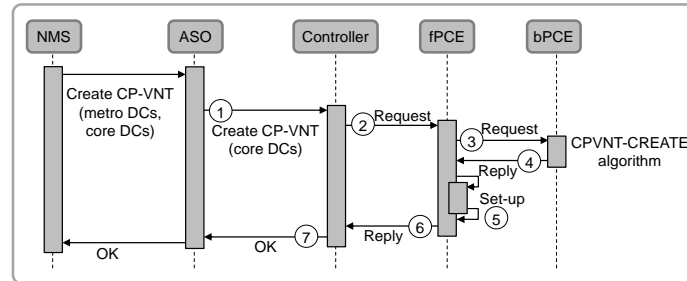


Fig. 6. CP-VNT_CREATE workflow.

PCReq messages (2) and (3) include one Request Parameters (RP) object for each candidate virtual link connecting pairs of core DCs (specified by the ASO) together with its corresponding Point-to-Point (P2P) END-POINT and BANDWIDTH objects. To group all the candidate virtual links belonging to the CP-VNT, a Synchronization VECTOR (SVEC) object is also included. The topology type, a tree in our case, is defined by adding an Objective Function (OF) object. Explicit Route Object (ERO) and BANDWIDTH objects are used for the candidate virtual links selected to be part of the CP-VNT. NO-PATH objects are used for those virtual links not in the CP-VNT. If no feasible solution was found,

all requests are replied with NO-PATH objects.

4.2 M2C-PROV Workflow

When a metro DC needs to synchronize contents with the core DCs, the local cloud resource manager issues a M2C connection request to the ASO (Fig. 7).

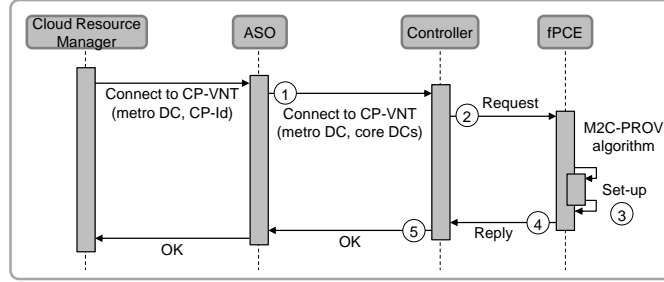


Fig. 7. M2C-PROV workflow.

The ASO checks whether the metro DC belongs to the CP and issues a message to the controller with the request (message 1 in Fig. 7). The controller creates a PCReq message and forwards it to the fPCE (2), which computes the route for the connection using the M2C-PROV algorithm and issues an initiation request for the LSP to the PM. When the LSP has been set-up, the fPCE replies to the controller (4) that in turn replies to the ASO (5). Finally, the ASO updates its database and replies to the cloud resource manager confirming the availability of the requested M2C connection.

Note that for the anycast M2C path computation the set of candidate core DCs belonging to the target CP-VNT has to be specified. To this end, we propose extending the Point-to-MultiPoint (P2MP) END-POINT object defined in [39] with a new leaf type to select one-of-many destinations.

4.3 RECONNECT Workflow

The RECONNECT workflow, reproduced in Fig. 8, is initiated by the cloud resource managers, issuing CP-VNT and M2C reconnection requests to the ASO, when their services are affected by a failure. As a result, the ASO might receive multiple reconnection requests.

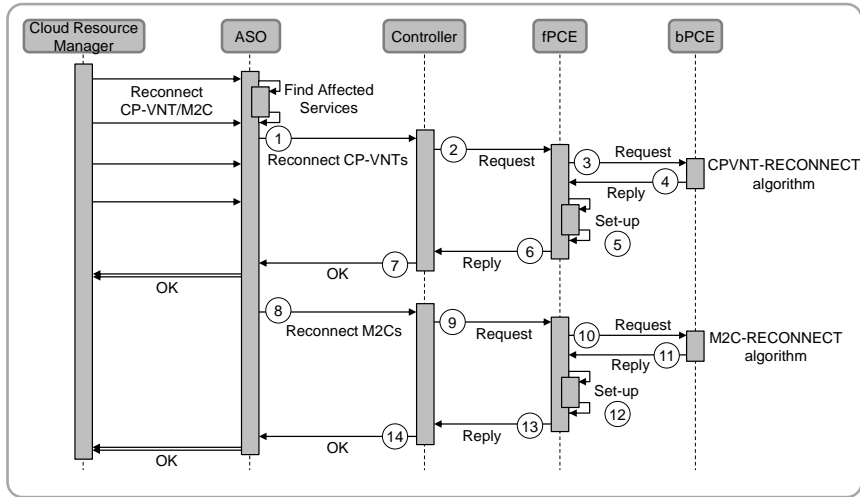


Fig. 8. RECONNECT workflow.

To accelerate the reconnection process, after receiving a number of reconnection requests [29] the ASO can infer the source of the failure (out of the scope of this paper) and find the affected services. Owing to the fact that the CP-VNTs need to be reconnected before the M2C connections, the ASO groups services and issues separate requests to the controller for CP-VNT and for M2C reconnections. It first issues a CP-VNT reconnection request to the controller (labelled as 1 in Fig. 8), which sends a PCReq message to the fPCE (2). CP-VNT reconnection is delegated to the bPCE, so the fPCE forwards the message to the bPCE (3) that runs the CPVNT-RECONNECT algorithm. Once a solution has been found the bPCE sends a PCRpt message (4) to the fPCE. Next, the fPCE issues a request to the PM to implement the lightpaths needed for recovery (5). The PM sends a PCRpt after each lightpath has been set-up. When all the related PCRpt messages have been received, the fPCE informs the controller (6), which in turn informs the ASO (7). The ASO updates its service database, notifies the cloud resource managers, and requests the controller to recover those

M2C anycast connections whose target CP-VNT is connected (8); the rest of the workflow is similar as before.

Since multiple CP-VNTs could need to be jointly reconnected (see Table II), we propose using a hierarchy of *SVEC* objects in PCReq message (2) and (3): the outer pair of *SVEC* and *OF* objects forces a joint computation of all CP-VNT reconnections, while one *SVEC* object per CP-VNT groups the virtual links belonging to that CP-VNT.

In addition, since some CP-VNT's virtual links could remain operational, both, candidate and operational virtual link have to be included in the PCReq messages. To that end, a *Symbolic-Path-Name* TLV is included in each *RP* object for the bPCE to know whether the related LSP already exists.

5 ILLUSTRATIVE RESULTS AND EXPERIMENTAL ASSESSMENT

In this section illustrative simulation results evaluating the proposed algorithms over a realistic network topology are presented and next, the feasibility of the proposed architecture is experimentally assessed in a real environment.

5.1 Simulation Results

The hierarchical architecture and the algorithms presented in the previous sections have been evaluated on an ad-hoc event-driven simulator based on OMNeT++ [40]. To that end, the 24-node 43-link Large Carrier US topology [41] is used, where metro DCs were placed in every node location while 6 core DCs were strategically placed in New York, Florida, Illinois, Texas, Washington, and California. Each DC is connected to the optical transport network through a L2 switch equipped with 100Gb/s transponders.

A dynamic network environment was simulated, where CP-VNTs are initially created and then 100Gb/s M2C anycast connection requests arrive following a Poisson process. The holding time of the M2C connection requests is exponentially distributed with a mean value equal to 1 h. For each CP-VNT, a metro DC is randomly chosen with equal probability (uniform distribution) among all metro DCs. Different loads are created by changing the arrival rate while keeping the mean holding time constant.

Let us first analyze the M2C provisioning performance. Fig. 9a shows the blocking probability as function of the offered load for each CP. Each plot in the graph is for a fixed number of core DCs, ranging from 1 to 5. As observed, the effects of increasing the number of core DCs has a direct effect on the blocking probability; assuming a target 1% blocking probability, 66% more traffic can be served when CP-VNTs consist in 2 core DCs compared to one single DC. The rationale behind this is related to the length of the M2C connections (Fig. 9b); the number of hops to reach the core DCs decreases when more core DCs are considered, especially when the selected core DCs are geographically distributed.

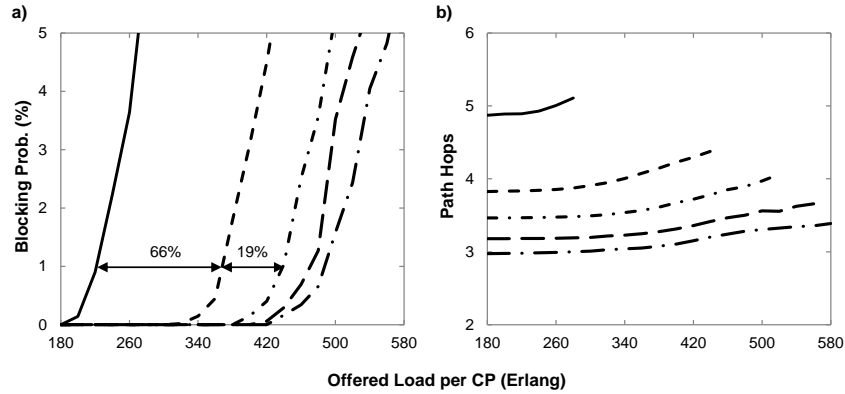


Fig. 9. M2C connection provisioning results for 5 CP-VNTs

Next, let us analyze the effect of increasing the number of CPs, while keeping the total load constant. We selected the load (440 Er) for which having 3 core DCs returns around 1% blocking (see Fig. 9a). As shown in Fig. 10, increasing the number of CPs has an almost linear effect on blocking, proportional to the number of CP-VNTs to be created. Since more CPs for the same total load entails reducing the load per CP, a network provider should verify the forecast load for that CP to advice about the right size of the CP-VNT.

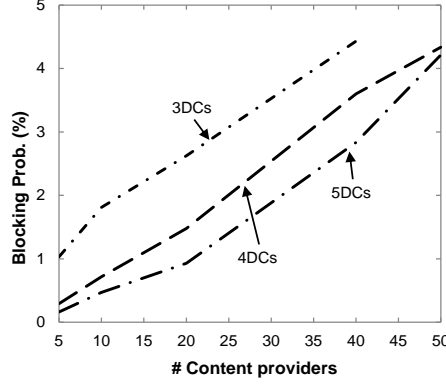


Fig. 10. Blocking prob. when increasing the number of CPs

The restorability of the proposed hierarchical content distribution architecture was evaluated considering single optical link failures, single optical node failures, and double adjacent optical node failures. As illustrated in Fig. 11, increasing the number of DCs in the CP-VNT significantly increases restorability; e.g., for an offered load of 300 Erlang, a 99% restorability can be obtained using 2 core DCs under single link failure scenarios, while for single node failures 3 core DCs have to be deployed per CP-VNT to obtain an equivalent restorability. In the 2 adjacent node failures scenario under the same offered load, 95% of restorability can be obtained only when 5 core DCs are deployed per CP-VNT.

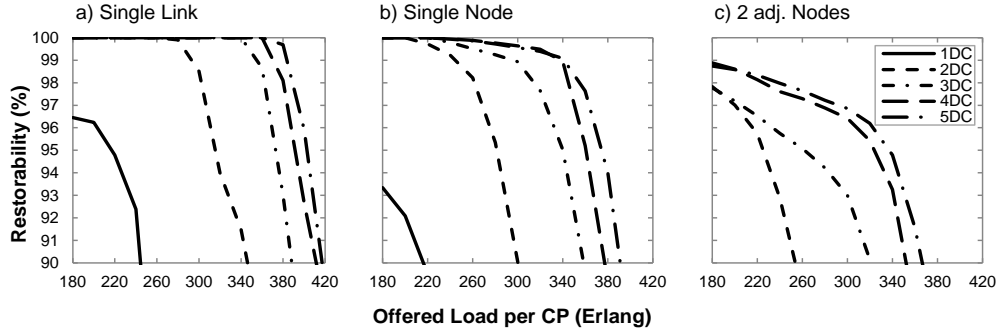


Fig. 11. Restorability

Table IV summarizes the increments of traffic load for a 90% of restorability with respect to 1 core DC per CP-VNT, e.g., by deploying 3 core DC CP-VNTs, the traffic load can be doubled under the 2 adjacent node failures scenario while maintaining the rate of restorability of affected demands.

Table IV. Load increment (%) w.r.t. 1DC for 90% restorability

	Failure Type		
	1 Link	1 Node	2 Adj. Nodes
2DC	41.8	38.9	64.3
3DC	59.0	65.7	107.8
4DC	68.4	74.5	128.6
5DC	70.9	81.0	137.7

5.2 Experimental Assessment

Once the performance of the proposed architecture has been studied, let us now focus on its experimental validation. Experiments were carried out in the UPC's iONE test-bed that follows the architecture depicted in Fig. 5. ASO and all the ABNO components were implemented in C++ for Linux. ASO issues XML-encoded messages to the controller through its HTTP REST API. The network shown in Fig. 2 was assumed, where the optical nodes are in the IP subnetwork 10.0.0.X, L2 switches for the DCs are in 10.0.1.X. Control and management modules run in the IP subnetwork 172.16.103.X. Specifically, ASO runs in .1, the controller in .2, fPCE in .3, bPCE in .4, and PM in .5. An emulated data plane was deployed for the experiments.

5.2.1 CP-VNT_CREATE Workflow

Fig. 12 shows the relevant messages for the CP-VNT_CREATE workflow. The messages are identified with the same sequence numbers used in the workflow description. Fig. 13 shows PCReq message (3) and PCRep message (4) details. The PCReq message contains a set of RP, END-POINT, and BANDWIDTH objects for each candidate link; an SVEC object groups the requests for a joint computation. The PCRep message indicates that 2 virtual links were

selected since ERO and BANDWIDTH objects are included, whereas the other virtual link is discarded. Note in Fig. 12 that one single PCInit message was used for the two LSPs to be created, whereas two individual PCRpt reported each individual set-up.

No.	Source	Destination	Protocol	Length	Info
①	509 172.16.103.1	172.16.103.2	HTTP/XML	507	POST /abno/CPVNT_CREATE HTTP/1.1
②	511 172.16.103.2	172.16.103.3	PCEP	220	Path Computation Request
③	513 172.16.103.3	172.16.103.4	PCEP	196	Path Computation Request
④	514 172.16.103.4	172.16.103.3	PCEP	408	Path Computation Reply
	516 172.16.103.3	172.16.103.5	PCEP	444	Initiate
⑤	518 172.16.103.5	172.16.103.3	PCEP	240	Path Computation LSP State Report (PCRpt)
	520 172.16.103.5	172.16.103.3	PCEP	276	Path Computation LSP State Report (PCRpt)
⑥	522 172.16.103.3	172.16.103.2	PCEP	408	Path Computation Reply
⑦	523 172.16.103.2	172.16.103.1	HTTP/XML	1277	HTTP/1.1 200 OK

Fig. 12. CP-VNT_CREATE message list.

The control plane contribution to the CP-VNT_CREATE workflow processing time was less than 9 ms., including messages exchange and CP-VNT_CREATE algorithm in the bPCE.

③ Path Computation Element communication	④ Path Computation Element communication
▶ Path Computation Request Header	▶ Path Computation Reply Header
▶ SVEC object	▶ RP object
▶ OBJECTIVE FUNCTION object (OF)	▶ EXPLICIT ROUTE object (ERO)
▶ RP object	▶ BANDWIDTH object
▶ END-POINT object	▶ RP object
▶ BANDWIDTH object	▶ NO-PATH object
▶ RP object	▶ RP object
▶ END-POINT object	▶ EXPLICIT ROUTE object (ERO)
▶ BANDWIDTH object	▶ BANDWIDTH object
▶ RP object	
▶ END-POINT object	
▶ BANDWIDTH object	

Fig. 13. CP-VNT_CREATE PCReq and PCRep messages.

5.2.2 M2C-PROV Workflow

Fig. 14 shows the relevant messages for the M2C-PROV workflow. Fig. 15 shows the P2MP END-POINT object included in PCReq message (2); we used the proposed leaf type 5 labelled as *New anycast leave to add* to request an anycast connection from metro DC 1 (10.0.1.1) to any of the core DCs belonging to the target CP-VNT: 10, 11, and 12. In the PCRep message (4), an ERO object specifies the route from metro DC 1 to core DC 10 through the optical layer. Note that since the request was for an anycast connection, the route to one single destination is provided.

The control plane contribution to the M2C-PROV workflow processing time was less than 3 ms., including messages exchange and M2C-PROV algorithm in the fPCE.

No.	Source	Destination	Protocol	Length	Info
①	904 172.16.103.1	172.16.103.2	HTTP/XML	232	POST /abno/M2C_PROV HTTP/1.1
②	906 172.16.103.2	172.16.103.3	PCEP	124	Path Computation Request
	908 172.16.103.3	172.16.103.5	PCEP	252	Initiate
③	910 172.16.103.5	172.16.103.3	PCEP	252	Path Computation LSP State Report (PCRpt)
④	912 172.16.103.3	172.16.103.2	PCEP	212	Path Computation Reply
⑤	914 172.16.103.2	172.16.103.1	HTTP/XML	639	HTTP/1.1 200 OK

Fig. 14. M2C-PROV message list.

② Path Computation Element communication Protocol	④ Path Computation Element communication Protocol
▶ Path Computation Request Header	▶ Path Computation Reply Header
▶ RP object	▶ RP object
▼ END-POINT object	▼ EXPLICIT ROUTE object (ERO)
Object Class: END-POINT OBJECT (4)	Object Class: EXPLICIT ROUTE OBJECT (ERO) (7)
0011 = Object Type: Point-to-MultiPoint IPv4 (3)	0001 = Object Type: 1
▶ Flags	▶ Flags
Object Length: 24	Object Length: 120
Leaf Type: New anycast leave to add (5)	▶ SUBOBJECT: Unnumbered Interface ID: 10.0.1.1:1
Source IPv4 Address: 10.0.1.1 (10.0.1.1)	▶ SUBOBJECT: Label Control
Destination IPv4 Address: 10.0.1.10 (10.0.1.10)	▶ SUBOBJECT: Label Control
Destination IPv4 Address: 10.0.1.11 (10.0.1.11)	▶ SUBOBJECT: Unnumbered Interface ID: 10.0.0.1:3
Destination IPv4 Address: 10.0.1.12 (10.0.1.12)	▶ SUBOBJECT: Label Control
▶ BANDWIDTH object	▶ SUBOBJECT: Label Control
	▶ SUBOBJECT: Unnumbered Interface ID: 10.0.0.3:110
	▶ SUBOBJECT: Label Control
	▶ SUBOBJECT: Label Control
	▶ SUBOBJECT: IPv4 Prefix: 10.0.1.10/32
	▶ BANDWIDTH object

Fig. 15. M2C-PROV PCReq and PCRep messages.

5.2.3 RECONNECT Workflow

Fig. 16 shows the relevant messages for the RECONNECT workflow, where messages (1)-(7) are related to CP-VNT reconnection and messages (8)-(14) to M2C anycast connection recovery. Fig. 17 shows the details of PCReq message (2). The first SVEC object is used to group all the CP-VNTs to be reconnected so as to do a joint computation, as specified in the associated OF object. One SVEC object is used to group the virtual links of each individual CP-VNT.

No.	Source	Destination	Protocol	Length	Info
①	1358 172.16.103.1	172.16.103.2	HTTP/XML	533	POST /abno/CPVNT_RECONNECT HTTP/1.1
②	1360 172.16.103.2	172.16.103.3	PCEP	228	Path Computation Request
③	1362 172.16.103.3	172.16.103.4	PCEP	228	Path Computation Request
④	1363 172.16.103.4	172.16.103.3	PCEP	288	Path Computation Reply
⑤	1365 172.16.103.3	172.16.103.5	PCEP	276	Initiate
⑥	1366 172.16.103.5	172.16.103.3	PCEP	276	Path Computation LSP State Report (PCRpt)
⑦	1368 172.16.103.3	172.16.103.2	PCEP	288	Path Computation Reply
⑧	1370 172.16.103.2	172.16.103.1	HTTP/XML	938	HTTP/1.1 200 OK
⑨	1636 172.16.103.1	172.16.103.2	HTTP/XML	236	POST /abno/M2C_RECONNECT HTTP/1.1
⑩	1638 172.16.103.2	172.16.103.3	PCEP	136	Path Computation Request
⑪	1639 172.16.103.3	172.16.103.4	PCEP	128	Path Computation Request
⑫	1640 172.16.103.4	172.16.103.3	PCEP	212	Path Computation Reply
⑬	1642 172.16.103.3	172.16.103.5	PCEP	252	Initiate
⑭	1644 172.16.103.5	172.16.103.3	PCEP	252	Path Computation LSP State Report (PCRpt)
⑮	1646 172.16.103.3	172.16.103.2	PCEP	212	Path Computation Reply
⑯	1648 172.16.103.2	172.16.103.1	HTTP/XML	643	HTTP/1.1 200 OK

Fig. 16. RECONNECT message list.

The control plane contribution to the RECONNECT workflow processing time was less than 12 ms., including messages exchange and algorithms. The CP-VNT reconnection sub-workflow took 9 ms., while the M2C recovery took 3 ms.

Path Computation Element communication Protocol	
▶ PATH COMPUTATION REQUEST MESSAGE Header	
▶ SVEC object	Groups all CP-VNTs to reconnect
▶ OBJECTIVE FUNCTION object (OF)	
▶ SVEC object	Specifies the links of each CP-VNT
▶ RP object	
▶ RP object	
▶ END-POINT object	
▶ BANDWIDTH object	
▶ RP object	
▶ END-POINT object	
▶ BANDWIDTH object	

Fig. 17. CPVNT-RECONNECT PCReq message (2).

6 CONCLUDING REMARKS

An architecture for hierarchical content distribution on the telecom cloud was proposed, where core and metro DCs were considered. Core DCs were geographically distributed and interconnected to create a specific CP-VNT topologies for each CP. Contents in the core DCs are automatically synchronized using the connectivity provided by the CP-VNT, whereas metro DCs synchronize their contents with core DCs periodically. To that end, M2C anycast connections are dynamically requested.

Three main problems were identified for operating this architecture: the CP-VNT creation, the M2C connection provisioning, and the reconnection of both CP-VNTs and M2C connections after a network failure. These problems were formally stated, mathematically formulated, and then heuristic algorithms were eventually proposed for solving them in real time. In particular, an exact algorithm for finding anycast shortest paths and efficient algorithms for creating CP-VNTs and reconnecting connected components of disconnected CP-VNTs were proposed.

Exhaustive simulations were carried out to evaluate the performance of the architecture. The results showed that high efficiency can be achieved when increasing the number of core DCs in the VNT; more than 66% load increment was observed. Furthermore, significant improvement in the network restorability for different types of failures was shown when the number of DCs in the CP-VNTs was increased.

Aiming at experimentally assessing the proposed architecture an ABNO-based control plane architecture was considered. The ASO module was responsible for service management and issues requests to the ABNO. As a result of the complexity of problems to be solved and the stringent time in which they need to be solved, a back-end PCE was used. Workflows were developed for the identified problems and the PCEP feasibility for those workflows was studied. PCEP extensions for M2C anycast connection provisioning and joint CP-VNTs reconnection and M2C connection recovery were proposed. Experimental assessment was carried out in our experimental iONE test-bed.

ACKNOWLEDGEMENTS

The research leading to these results has received funding from the European Community's Seventh Framework Programme FP7/2007-2013 under grant agreement n° 317999 IDEALIST project and from the Spanish MINECO SYNERGY project (TEC2014-59995-R).

REFERENCES

- [1] "Cisco Global Cloud Index: Forecast and Methodology, 2013-2018," Cisco white paper, October 2014 [Online]. Available: http://www.cisco.com/c/en/us/solutions/collateral/service-provider/global-cloud-index-gci/Cloud_Index_White_Paper.pdf.
- [2] "Metro Network Traffic Growth: An Architecture Impact Study," Alcatel-Lucent Bell Labs strategic white paper, December 2013 [Online]. Available: <http://resources.alcatel-lucent.com/asset/171568>.
- [3] L. Velasco, P. Wright, A. Lord, and G. Junyent, "Saving CAPEX by Extending Flexgrid-based Core Optical Networks towards the Edges," *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, vol. 5, pp. A171-A183, 2013.
- [4] C. Colman-Meixner, F. Dikbiyik, M. F. Habib, M. Tornatore, C.-N. Chuah, B. Mukherjee, "Disaster-survivable cloud-network mapping," *Springer Photonic Network Communications*, vol. 27, pp. 141-153, 2014.
- [5] L. M. Contreras, V. López, O. Gonzalez de Dios, A. Tovar, F. Muñoz, A. Azañón, J. P. Fernandez-Palacios, and J. Folgueira, "Toward Cloud-Ready Transport Networks," *IEEE Communications Magazine*, vol. 50, pp. 48-55, 2012.
- [6] A. Asensio, L. M. Contreras, M. Ruiz, V. López, L. Velasco, "Scalability of Telecom Cloud Architectures for Live-TV Distribution," in *Proc. OSA Optical Fiber Communication Conference (OFC)*, 2015.
- [7] S. Savas, F. Dikbiyik, M. F. Habib, M. Tornatore, B. Mukherjee, "Disaster-aware service provisioning with manycasting in cloud networks," *Springer Photonic Network Communications*, vol. 28, pp. 123-134, 2014.
- [8] F. Dikbiyik, M. Tornatore, B. Mukherjee, "Minimizing the Risk From Disaster Failures in Optical Backbone Networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 32, pp. 3175-3183, 2014.
- [9] The Apache Cassandra Project [Online]. Available: <http://cassandra.apache.org/>.
- [10] O. Gerstel, M. Jinno, A. Lord, S.J.B. Yoo, "Elastic optical networking: a new dawn for the optical layer?" *IEEE Communications Magazine*, vol. 50, pp. s12-s20, 2012.
- [11] B. Ramamurthy, H. Feng, D. Datta, J. P. Heritage, and B. Mukherjee, "Transparent vs. opaque vs. translucent wavelength-routed optical networks," in *Proc. Optical Fiber Communication Conference (OFC)*, 1999.
- [12] D. King and A. Farrel, "A PCE-based Architecture for Application-based Network Operations," *IETF draft draft-farrkingel-pce-abno-architecture-14*, work in progress, December 2014.
- [13] A. Farrel, J. Vasseur, and J. Ash, "A Path Computation Element (PCE)-Based Architecture," *IETF RFC 4655*, 2006.
- [14] J.P. Vasseur and J.L. Le Roux, "Path Computation Element (PCE) Communication Protocol (PCEP)," *IETF RFC 5440*, 2009.
- [15] L. Velasco, D. King, O. Gerstel, R. Casellas, A. Castro, and V. López, "In-Operation Network Planning," *IEEE Communications Magazine*, vol. 52, pp. 52-60, 2014.
- [16] Ll. Gifre, F. Paolucci, L. Velasco, A. Aguado, F. Cugini, P. Castoldi, and V. López, "First Experimental Assessment of ABNO-driven In-Operation Flexgrid Network Re-Optimization," *IEEE/OSA Journal of Lightwave Technology (JLT)*, 2015.
- [17] L. Velasco, A. Asensio, J. Ll. Berral, E. Bonetto, F. Musumeci, V. López, "Elastic Operations in Federated Datacenters for Performance and Cost Optimization," *Elsevier Computer Communications*, vol. 50, pp. 142-151, 2014.
- [18] A. Mahimkar, A. Chiu, R. Doverspike, M. D. Feuer, P. Magill, E. Mavrogiorgis, J. Pastor, S. L. Woodward, J. Yates, "Bandwidth on Demand for Inter-Data Center Communication," in *Proc. 10th ACM Workshop on Hot Topics in Networks (HotNets-X)*, 2011.
- [19] L. Velasco, A. Asensio, J.Ll. Berral, V. López, D. Carrera, A. Castro, and J.P. Fernández-Palacios, "Cross-Stratum Orchestration and Flexgrid Optical Networks for Datacenter Federations," *IEEE Network Magazine*, vol. 27, pp. 23-30, 2013.
- [20] L. Velasco, A. Asensio, J.Ll. Berral, A. Castro, V. López, "Towards a Carrier SDN: An example for Elastic Inter-Datacenter Connectivity," (Invited Paper) *OSA Optics Express*, vol. 22, pp. 55-61, 2014.
- [21] A. Asensio and L. Velasco, "Managing Transfer-based Datacenter Connections," *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, vol. 6, pp. 660-669, 2014.
- [22] Ll. Gifre, F. Paolucci, O. González de Dios, L. Velasco, L. M. Contreras, F. Cugini, P. Castoldi, and V. López, "Experimental Assessment of ABNO-driven Multicast Connectivity in Flexgrid Networks," (Invited Paper) *IEEE/OSA Journal of Lightwave Technology (JLT)*, 2015.
- [23] M. Gharbaoui, B. Martini, and P. Castoldi, "Anycast-Based Optimizations for Inter-Data-Center Interconnections [Invited]," *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, vol. 4, pp. B168-B178, 2012.
- [24] J. Buysse, M. De Leenheer, L. M. Contreras, J. Aznar, J. Rodriguez, G. Landi, C. Develder, "NCP+: an integrated network and IT control plane for cloud computing," *Elsevier Optical Switching and Networking*, vol. 11, pp. 137-152, 2014.
- [25] K. Walkowiak, A. Kasprzak, and M. Klinkowski, "Dynamic routing of anycast and unicast traffic in Elastic Optical Networks," in *Proc. IEEE International Conference on Communications (ICC)*, 2014.
- [26] J. Qiu, Y. Liu, G. Mohan, K. C. Chua, "Fast spanning tree reconnection mechanism for resilient Metro Ethernet networks," *Computer Networks*, vol. 55, pp. 2717-2729, 2011.
- [27] K. Walkowiak and J. Rak, "Shared Backup Path Protection for Anycast and Unicast Flows Using the Node-Link Notation," in *Proc. IEEE International Conference on Communications (ICC)*, 2011.
- [28] F. Paolucci, A. Castro, F. Cugini, L. Velasco, and P. Castoldi, "Multipath Restoration and Bitrate Squeezing in SDN-based Elastic Optical Networks," (Invited Paper) *Springer Photonic Network Communications*, vol. 28, pp. 45-57, 2014.
- [29] A. Castro, R. Martínez, R. Casellas, L. Velasco, R. Muñoz, R. Vilalta, and J. Comellas, "Experimental Assessment of Bulk Path Restoration in Multi-layer Networks using PCE-based Global Concurrent Optimization," *IEEE/OSA Journal of Lightwave Technology (JLT)*, vol. 32, pp. 81-90, 2014.
- [30] "Architecture for the Automatically Switched Optical Network," *ITU-T Recommendation G.8080*, 2012.

- [31] J. Zhang, and B. Mukherjee, "A Review of Fault Management in WDM Mesh Networks: Basic Concepts and Research Challenges," *IEEE Network*, vol. 18, pp. 41-48, 2004.
- [32] L. Velasco, F. Agraz, R. Martínez, R. Casellas, S. Spadaro, R. Muñoz, and G. Junyent, "GMPLS-based Multi-domain Restoration: Analysis, Strategies, Policies and Experimental Assessment," *IEEE/OSA Journal of Optical Communications and Networking (JOCN)*, vol. 2, pp. 427-441, 2010.
- [33] S. Pettie and V. Ramachandran, "An optimal minimum spanning tree algorithm," *Journal of the ACM*, vol. 49, pp. 16-34, 2002.
- [34] L. Velasco, A. Castro, M. Ruiz, and G. Junyent, "Solving Routing and Spectrum Allocation Related Optimization Problems: from Off-Line to In-Operation Flexgrid Network Planning," *IEEE/OSA Journal of Lightwave Technology (JLT)*, vol. 32, pp. 2780-2795, 2014.
- [35] M. Ruiz, M. Pióro, M. Zotkiewicz, M. Klinkowski, and L. Velasco, "Column Generation Algorithm for RSA Problems in Flexgrid Optical Networks," *Springer Photonic Network Communications*, vol. 26, pp. 53-64, 2013.
- [36] E. Crabbe, J. Medved, I. Minei, and R. Varga, "PCEP Extensions for Stateful PCE," *IETF draft draft-ietf-pce-stateful-pce-10*, work in progress, October 2014.
- [37] E. Crabbe, I. Minei, S. Sivabalan, and R. Varga, "PCEP Extensions for PCE-initiated LSP Setup in a Stateful PCE Model," *IETF draft draft-ietf-pce-pce-initiated-lsp-02*, work in progress, October 2014.
- [38] Ll. Gifre, L. Velasco, N. Navarro, and G. Junyent, "Experimental Assessment of a High Performance Back-end PCE for Flexgrid Optical Network Re-optimization," in *Proc. OSA Optical Fiber Communication Conference (OFC)*, 2014.
- [39] Q. Zhao, D. King, F. Verhaeghe, T. Takeda, Z. Ali, and J. Meuric, "Extensions to the Path Computation Element Communication Protocol (PCEP) for Point-to-Multipoint Traffic Engineering Label Switched Paths," *IETF RFC 6006*, 2010.
- [40] OMNeT++ [Online]. Available: <http://www.omnetpp.org/>.
- [41] S. Huang, M. Xia, C. Martel, and B. Mukherjee, "A Multistate Multipath Provisioning Scheme for Differentiated Failures in Telecom Mesh Networks," *IEEE/OSA Journal of Lightwave Technology*, vol. 28, pp. 1585-1596, 2010.