

A scalable model for multi-period virtual network mapping for resilient multi-site data centers

Ting Wang¹, Brigitte Jaumard¹, Chris Develder²

¹*CSE, Concordia University, Montréal (QC) H3G 1M8 Canada*

²*INTEC – IBCN, Ghent University – iMinds, Ghent, Belgium*

Abstract

In the currently dominant cloud computing paradigm, applications are being served in data centers (DCs) which are connected to high capacity optical networks. For cost efficiency reasons, in both DC and optical network domains, virtualization of the physical hardware is exploited. In a DC, it means that multiple so-called virtual machines (VMs) are being hosted on the same physical server. Similarly, the network is partitioned into separate virtual networks, thus providing isolation between distinct virtual network operators (VNOs). Thus, the problem of virtual network mapping arises: how to decide which physical resources to allocate for a particular virtual network? In this paper, we study that problem in the context of cloud computing with multiple DC sites. This introduces additional flexibility, due to the anycast routing principle: we have the freedom to decide at what particular DC location to serve a particular application. We can exploit this choice to minimize the required resources when solving the virtual network mapping problem.

This paper builds on our earlier work and solves a resilient virtual network mapping problem that optimally decides on the mapping of both network and data center resources, considering time-varying traffic conditions and protecting against possible failures of both network and DC resources. We consider the so-called VNO resilience scheme: rerouting under failure conditions is provided in the virtual network layer. To minimize physical resource capacity requirements, we allow reuse of both network and DC resources: we can reuse the same resources for the rerouting under failure scenarios that are assumed not to occur simultaneously. Since we also protect against DC failures, we allocate backup DC resources, and account for synchronization between primary and backup DCs. To deal with the time variations in the volume and geographical pattern of the application traffic, we investigate the potential benefits (in terms of overall resource capacity requirements) of reconfiguring the virtual network mapping from one period to the next.

Previously, we developed a model to solve the multi-period traffic case one step at a time: given the virtual network mapping for time t , we determine the (possibly changed) mapping for $t + 1$. Compared to that previous work, we now (i) define a truly multi-period model path formulation exploiting column generation, and (ii) demonstrate its scalability on a nation-wide network with hourly varying traffic.

Keywords: Network Virtualization, End-to-End Resilience, Cloud Computing, Anycast Resilience, Multi-period Virtual Network Mapping.

1. Introduction

Increasingly, businesses are relying on cloud computing: applications and content are being served in data centers (DCs). Given the high bandwidth capacity and low latency of the underlying optical networking technology, users typically do not care about very much about the exact location of these data centers, and service providers can exploit anycast routing: to serve a new request, they basically have the freedom to pick any of the available data centers. As a result, this anycast principle can be exploited for resiliency purposes []: if either the server infrastructure (in the DC), or the (optical) network is affected by a failure, backup could be provided at a different DC location compared to failure-free operation. Earlier work has considered the potential benefits of exploiting anycast on the required network capacity under static traffic (e.g., [1]).

This paper considers time-varying traffic: we investigate the potential savings of reconfiguring traffic routing (working and/or backup routes) from one time period to the next, assuming a time-slotted approach where the traffic patterns change from one period to the next but some traffic does survive multiple periods. This topic has been investigated in the past, but not thoroughly. For instance, He and Poo [2] propose a sub-reconfiguration technique in order to rearrange the paths for WDM (Wavelength

Division Multiplexing) networks, using pre-computed alternate backup paths. They report a 10% bandwidth saving with simulation experiments using OPNET. Other studies look at differentiated protection schemes, e.g., [3] or [4], with either pre-emption or multiple protection paths, but without backup re-configuration. But these works considered “classical” traffic, i.e., not the cloud computing scenario with anycast routing. The current paper builds on our earlier work [5], which to our knowledge was the first to consider resilient multi-period anycast traffic routing. Yet, there we assumed an iterative approach: we formulated the optimization problem to find the best routing going from one period to the next. Our current work presents a model addressing joint optimization over all time periods together.

The paper is organized as follows. In Section 2, we give an overview of resilience in cloud computing settings, where the physical infrastructure is usually shared by multiple virtual network operators (VNOs). We highlight the scenarios we consider for changing (or not) configurations that last multiple periods in time-varying traffic settings. In Section 3, we present our new model for finding the routes (for each of the multiple time periods) that minimize the bandwidth requirements to serve time-varying cloud traffic. We apply it in a case study to quantitatively compare the various scenarios of changing primary and/or backup routes in Section 4. We summarize our conclusions in Section 5.

2. Problem Statement

2.1. Virtualization and Resilience in Cloud Computing

The recent evolution towards grid and cloud computing illustrates the crucial role played by (optical) networks in supporting today’s applications [6]. Cloud computing relies quite heavily on the concept of virtualization (cf. virtual machines): physical infrastructure is partitioned logically into distinct entities, such that applications can be run in virtually isolated environments while maximizing the use of physical resources by sharing them. This idea has also been adopted in networking: multiple virtual network operators (VNOs) share the same underlying physical infrastructure (i.e., fibers and optical cross-connects, OXCs, ROADMs), while VNOs only have access to (and full control over) their own resources.

In this paper, we focus on resilient virtual topology mapping: how to decide on what routes to follow in the physical network to map the virtual connections from source nodes to data centers where the applications are being served? The cloud services’ requests are offered by a virtual network operator (VNO), which runs its virtual network (VNet) on top of the physical network resources offered by a physical infrastructure provider (PIP). The problem we address is how to determine a resilient VNet topology that minimizes the bandwidth resources that are requested by the VNO to the PIP, assuming time-varying traffic. We assume a VNO-resilience scheme, i.e., rerouting in the virtual network under the VNO control (see below, Section 2.2, or, e.g., [7]). We design the VNet such that requests can survive single failures, which can each affect either the physical network or data center infrastructure.

2.2. VNO-resilience

The VNO-resilience model we adopt is exactly the same as in our earlier work [5, 7] and illustrated in Fig. 1: it provides 1:1 protection routing in the VNet for network failures, where the working and protection paths of a service have to be physically link/node disjoint. The working path (π^W) routes the services from their source node (v_s) towards the primary DC (d_1), the protection path (π^B) towards the backup DC (d_2), while π^W and π^B are disjoint in their physical layer mapping. In addition, a synchronization path (π^S) is established in order to handle migration and failure routing requirements when a DC failure occurs: services then need to be rerouted from the primary d_1 to backup d_2 . Thus, the resulting VNet for the request from source v_s comprises three virtual paths, mapped to resp. the physical π^W, π^B and π^S paths. Note that both π^W and π^B need to carry the overall traffic (but π^B only when π^W or d_1 are affected by a failure), but π^S possibly only

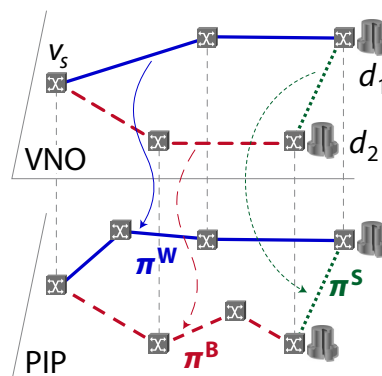


Fig. 1: The VNO-resilience scheme.

a fraction thereof, since it is used just to keep the state at the backup location d_2 synchronized with that of d_1 (or vice versa) to allow smooth migration upon d_1 failure (or recovery).

Further, we assume that there is an automatic switch-back to the original network path and DC once a fault is repaired, and therefore we allow reusing the same network/DC capacity to protect against other failures: backup capacity is shared. Under the assumptions that (A1) the backup DC has a different location than the primary DC, (A2) π^W and π^B are link disjoint and, (A3) π^W and π^S are link disjoint, protection is guaranteed against any single link failure and any single DC failure. We now qualitatively discuss the various failure cases we protect against:

- (i) **Failure of link** $\ell \in \pi^W$: the request is rerouted to the backup data center d_2 , using the backup path π^B (which is link disjoint from π^W , thus $\ell \notin \pi^B$). If $\ell \in \pi^S \cap \pi^W$, then as long as the failure is not restored, the primary data center d_1 cannot be kept in sync with the now operational d_2 . Thus, right after the repair of ℓ , the primary d_1 is in stale state, and hence switching back to d_1 either suffers from this stale state or needs to wait some extra time to handle the requests again. The remedy is of course to enforce $\pi^W \cap \pi^S = \emptyset$. (Yet, note that the same issue of a non-synchronized primary d_1 clearly also occurs after the repair of d_1 that failed itself.)
- (ii) **Failure of link** $\ell \in \pi^S \setminus \pi^W$: there is no immediate issue. Yet, if shortly after ℓ 's repair, working path π^W fails, the switchover to the backup d_2 (via path π^B) suffers from stale state since the failing π^S interrupted the synchronization between primary and backup DCs. This can only be remedied by providing a second synchronization path $\pi^{S'}$ that is link disjoint with π^S .
- (iii) **Failure of link** $\ell \in \pi^B$: again no immediate problem arises (since this means that π^W is operational, given $\pi^W \cap \pi^B = \emptyset$). However, if $\ell \in \pi^S \cap \pi^B$ and shortly after ℓ 's repair the primary path π^W (or d_1) fails—meaning that now π^B is followed towards d_2 —the secondary data center d_2 might not be fully synchronized yet. Clearly, this can be remedied by choosing $\pi^B \cap \pi^S = \emptyset$. Yet, the issue is similar to the one of case (ii), which obviously remains, even if we take $\pi^S \cap \pi^B = \emptyset$.
- (iv) **Failure of primary DC** d_1 : requests are rerouted to backup d_2 via the π^B path. Clearly, the failing d_1 cannot be kept in sync with the now operational backup d_2 . Thus, we might need to wait some time after d_1 's repair to switch back requests via π^W . Any failure that would occur shortly after d_1 's repair and which would prevent services to remain being served at d_2 clearly could imply service degradation because of the unsynchronized d_1 : (a) failure of π^S , (b) failure of π^B , or (c) failure of d_2 . However, protection against such a failure event requires extra DC resources or extra paths.

2.3. Time-varying Traffic

As in [5], the current paper investigates whether it is worth reconfiguring the primary and the backup paths in order to save bandwidth when the communication traffic pattern changes. Note that this change is not necessarily limited to a scaling of the volume, but also its geographical pattern: when considering large backbone networks (as the ones that we are designing VNet over), they might comprise different time zones where activities are shifted in time, and hence the resulting volume of cloud requests fluctuates differently.

As changing the VNet mapping operations clearly may have an impact on the real-time performance of the cloud requests they are servicing, we propose to investigate three scenarios:

- In *Scenario I* (very conservative), we do not allow reconfiguring already established paths;
- In *Scenario II* we only allow reconfiguring backup and/or synchronisation routes (π^B and/or π^S) for traffic that continues from one period to the next;
- In *Scenario III* we assume complete freedom and thus also allow to change the primary paths (π^W). Yet, we always look for the optimal solution (in terms of minimal link bandwidth consumed on the PIP layer) with the lowest number of configuration changes.

Whereas in [5] we developed a model to decide the transition from a single period to the next, we now present a model to jointly and globally decide on the routing in all time periods together.

3. Optimization Model

3.1. Notations

The cloud network is modeled by an undirected graph $G = (V, L)$ where V is the node set (indexed by v) and L is the link set (indexed by ℓ), for which $\omega(v)$ denotes the set of links adjacent to v .

We consider some multi-period traffic, such that for each time period $t \in T$, the traffic is defined by the number of service requests (demands), originating from a set of source/service nodes $V_S \subseteq V$, with generic index v_S . To simplify the notation, we assume $V_S = V$ from now on. We assume that requests originating from the same source node are aggregated, so that each request originating from source node v can be indexed by v and characterized by its bandwidth requirement $\Delta_{v,t}$ at time period t and δ_v (with $0 \leq \delta_v \leq 1$), representing the fraction of its demand that is required for synchronization between the primary and the backup data center. Note that the routing of a request may be distributed over several DCs. We denote by T^* the set of time periods except for the first one.

3.2. Configurations

The mathematical model we propose relies on the notion of configurations, where a configuration is associated with a set of service requests originating at a given source node. Let C be the overall set of configurations: $C = \bigcup_{v \in V_S} C_v$, where C_v is the set of configurations associated with source node $v \in V_S$.

We define a configuration $c \in C_v$ by a set of 3 paths, one primary path π^w originating at v towards a primary data center d^w , one backup path π^b originating at v_s towards a primary data center d^b , and one synchronization path π^s between the primary and the backup data center. We protect against single link failures as well as single data center failures.

More formally, in our mathematical model, a configuration is characterized by the given parameters:

- $p_{c,\ell}^w$ (resp. $p_{c,\ell}^b, p_{c,\ell}^s$) = 1 if link ℓ is used by the working (resp. backup, synchronization) path of configuration c , 0 otherwise;
 - $a_{c,v}^w$ (resp. $a_{c,v}^b$) = 1 if node $v \in V_D$ is selected as the primary (resp. backup) data center, 0 otherwise;
- For each link ℓ , let β_ℓ^w be the working bandwidth on ℓ , β_ℓ^b the backup bandwidth on ℓ , and β_ℓ^s the synchronization bandwidth on ℓ .

We will consider a two-phase model, following the column generation strategy. The so-called (restricted) master problem (RMP) assumes a set of candidate configurations to be given and chooses which ones to use as to satisfy the requested demands (over all periods) with minimal bandwidth requirements. Given an (initial) solution, the second phase, a so-called pricing problem (PP), will generate a new configuration (if possible) that would allow to reduce the overall cost in the RMP if added to the set of possible configurations. A PP will be associated with a given source node: it will find a new set of working, backup and synchronization paths for the given source node and some chosen working and backup data centers. The new configuration is then added to the candidate configuration set for the RMP. Note that to derive the PP, the RMP is solved as the LP relaxation of the ILP model. Once the PP(s) do not find no new configurations, we solve the RMP with the final set of configurations as an ILP.

Note that PPs for different source nodes can be solved simultaneously in parallel (based on the same RMP solution). By solving multiple PPs in parallel, the number of RMP iterations (with growing number of configurations) can potentially be reduced. We however leave the study of such parallelization strategies (e.g., how many PPs to solve in parallel, and how to cycle through the source nodes) out of scope for this paper. Suffice to note that for the experiments presented in Section 4, we solved all PPs (i.e., one PP for each source node) in parallel, adding potentially one extra configuration for each source node before solving the RMP with the updated configuration set.

The master problem is described in detail next. The pricing problem (PP) follows quite straightforwardly from the RMP: the objective follows automatically (see the general column generation strategy, e.g., [8]), and the constraints amount to the classical flow constraints for each of the working, backup and synchronization paths, extended with the appropriate disjointness constraints (as discussed qualitatively in Section 2.2).

3.3. Master problem: Objective

We first define the variables:

- $z_{c,t}$: number of bandwidth units of the demand originating from v that is supported by configuration c at time period t .
- $x_{v,t}^W$ (resp. $x_{v,t}^B, x_{v,t}^S \in \{0, 1\}$): amount of rerouted working (resp. backup/synchronization) bandwidth at time period $t \in T^*$ for the requests originating from v .
- $\beta_{\ell,t}^W$ (resp. $\beta_{\ell,t}^B, \beta_{\ell,t}^S$): required amount of bandwidth on link ℓ at time period $t \in T$, for the working (resp. synchronization, backup) provisioning.

The objective function should take care of minimizing the overall (working + backup + synchronization) bandwidth requirements, and, in case of ties, should encourage not to disturb the working paths as a first priority, and, as a second priority, not to disturb the backup/synchronization paths in case the same routing paths can be used for the previously routed requests:

$$\min \sum_{\ell \in L} \underbrace{(\beta_{\ell}^W + \beta_{\ell}^B + \beta_{\ell}^S) \cdot \|\ell\|}_{\text{BW}_{\ell}} + \text{PENAL}^{\text{DISRUPT-BS}} \sum_{v \in V} \sum_{t \in T^*} x_{v,t}^{\text{BS},t} + \text{PENAL}^{\text{DISRUPT-W}} \sum_{v \in V} \sum_{t \in T^*} x_{v,t}^{\text{W},t} \quad (1)$$

where $\|\ell\|$ represents the length of link ℓ , $\text{PENAL}^{\text{DISRUPT-B}}$ and $\text{PENAL}^{\text{DISRUPT-W}}$ are weight penalty factors such that $\text{PENAL}^{\text{DISRUPT-B}} \geq \text{PENAL}^{\text{DISRUPT-W}}$, and BW_{ℓ} is total bandwidth cost for a given link ℓ . Note that in Scenario I, only the first term of (1) is minimized, while the second plays a role only in Scenarios II and III, and the third term only for Scenario III.

3.4. Master Problem: Constraints

$$\sum_{c \in C_v} z_{c,t} \geq \Delta_{v,t} \quad v \in V, t \in T \quad (2)$$

$$\sum_{c \in C} p_{c,\ell}^W z_{c,t} = \beta_{\ell,t}^W \quad ; \quad \sum_{v \in V} \sum_{c \in C_v} \delta_v p_{c,\ell}^S z_{c,t} = \beta_{\ell,t}^S \quad \ell \in L, t \in T \quad (3)$$

$$\sum_{c \in C} p_{c,\ell'}^W p_{c,\ell}^B z_{c,t} \leq \beta_{\ell,t}^B \quad \ell' \in L, \ell \in L \setminus \{\ell'\}, t \in T \quad (4)$$

$$\sum_{c \in C} a_{c,v}^W p_{c,\ell}^B z_{c,t} \leq \beta_{\ell,t}^B \quad v \in V_D, \ell \in L, t \in T \quad (5)$$

$$\sum_{\pi \in \Pi_v} \left| \sum_{c \in C_{\pi}^W} (z_{c,t} - z_{c,t-1}) \right| - \left| \Delta_{v,t} - \Delta_{v,t-1} \right| \leq 2 x_{v,t}^W \quad v \in V, t \in T^* \quad (6)$$

$$\sum_{\pi \in \Pi_v} \left| \sum_{c \in C_{\pi}^B} (z_{c,t} - z_{c,t-1}) \right| - \left| \Delta_{v,t} - \Delta_{v,t-1} \right| \leq 2 x_{v,t}^B \quad v \in V, t \in T^* \quad (7)$$

$$\sum_{\pi \in \Pi_v} \left| \sum_{c \in C_{\pi}^S} (z_{c,t} - z_{c,t-1}) \right| - \left| \Delta_{v,t} - \Delta_{v,t-1} \right| \leq 2 x_{v,t}^S \quad v \in V, t \in T^* \quad (8)$$

$$x_{v,t}^{\text{BS}} = x_{v,t}^B + x_{v,t}^S \quad v \in V, T \quad (9)$$

$$z_{c,t} \in \mathbb{R} \quad c \in C, t \in T \quad (10)$$

$$\beta_{\ell,t}^W, \beta_{\ell,t}^B, \beta_{\ell,t}^S \in \mathbb{R} \quad \ell \in L, t \in T \quad (11)$$

$$x_{v,t}^W, x_{v,t}^B, x_{v,t}^S \in \mathbb{R} \quad v \in V, t \in T \quad (12)$$

In Scenario I, requests cannot be reconfigured, so we add following constraints:

$$x_{v,t}^W = x_{v,t}^B = x_{v,t}^S = 0 \quad v \in V, t \in T^* \quad (13)$$

In Scenario II, only backup and synchronization paths can be modified:

$$x_{v,t}^W = 0 \quad \text{and} \quad x_{v,t}^B, x_{v,t}^S \in \mathbb{R} \quad v \in V, t \in T^* \quad (14)$$

In Scenario III, all paths can be modified without any restriction:

$$x_{v,t}^W, x_{v,t}^B, x_{v,t}^S \in \mathbb{R} \quad v \in V, t \in T^* \quad (15)$$

Constraints (6)-(8) allow the estimation of the amount of rerouted bandwidth at time period $t \in T^*$. The change in bandwidth requested from source node v is equal to $|\Delta_{v,t} - \Delta_{v,t-1}|$. For a given working path π , the bandwidth difference going from period $t-1$ to t is equal to $|\sum_{c \in C_\pi^w} (z_{c,t} - z_{c,t-1})|$. Since moving a bandwidth unit from one path π to π' means an increase on π' and a reduction on π , we obtain the overall amount of rerouted bandwidth by summing over all paths and dividing by 2. Also, we obviously cannot avoid the change caused by the overall bandwidth in/decrease (i.e., $|\Delta_{v,t} - \Delta_{v,t-1}|$). Hence, the amount of bandwidth change that we should account for in the reconfiguration cost at time t (i.e., $x_{v,t}^\bullet$) for traffic originating from source node v is given by: $\frac{1}{2} \left(\sum_{\pi \in \Pi_v} \left| \sum_{c \in C_\pi^w} (z_{c,t} - z_{c,t-1}) \right| - |\Delta_{v,t} - \Delta_{v,t-1}| \right)$, if we consider the working path. We introduce $\beta_{\pi,t}^\bullet$ to linearize $\left| \sum_{c \in C_\pi^w} (z_{c,t} - z_{c,t-1}) \right|$, and thus get:

$$\sum_{c \in C_\pi^w} z_{c,t} = \beta_{\pi,t}^w \quad \pi \in \Pi, t \in T \quad (16)$$

$$\sum_{c \in C_\pi^b} z_{c,t} = \beta_{\pi,t}^b \quad \pi \in \Pi, t \in T \quad (17)$$

$$\sum_{c \in C_\pi^s} z_{c,t} = \beta_{\pi,t}^s \quad \pi \in \Pi, t \in T \quad (18)$$

$$\sum_{\pi \in \Pi_v} (\beta_{\pi,t}^w - \beta_{\pi,t-1}^w) - |\Delta_{v,t} - \Delta_{v,t-1}| \leq 2 x_{v,t}^w \quad v \in V, t \in T^* \quad (19)$$

$$\sum_{\pi \in \Pi_v} (\beta_{\pi,t-1}^b - \beta_{\pi,t}^w) - |\Delta_{v,t} - \Delta_{v,t-1}| \leq 2 x_{v,t}^w \quad v \in V, t \in T^* \quad (20)$$

$$\sum_{\pi \in \Pi_v} (\beta_{\pi,t}^b - \beta_{\pi,t-1}^w) - |\Delta_{v,t} - \Delta_{v,t-1}| \leq 2 x_{v,t}^b \quad v \in V, t \in T^* \quad (21)$$

$$\sum_{\pi \in \Pi_v} (\beta_{\pi,t-1}^w - \beta_{\pi,t}^b) - |\Delta_{v,t} - \Delta_{v,t-1}| \leq 2 x_{v,t}^b \quad v \in V, t \in T^* \quad (22)$$

$$\sum_{\pi \in \Pi_v} (\beta_{\pi,t}^s - \beta_{\pi,t-1}^s) - |\Delta_{v,t} - \Delta_{v,t-1}| \leq 2 x_{v,t}^s \quad v \in V, t \in T^* \quad (23)$$

$$\sum_{\pi \in \Pi_v} (\beta_{\pi,t-1}^s - \beta_{\pi,t}^s) - |\Delta_{v,t} - \Delta_{v,t-1}| \leq 2 x_{v,t}^s \quad v \in V, t \in T^* \quad (24)$$

4. Numerical Results

4.1. Traffic and Topology

For our case study, we consider 3 different traffic volumes (A, B, and C) per time slot, that cyclically repeat: of all traffic requests that start in any of the three time slot types, 13% start in time period A, 38% in the second time period B, 49% in third time period C. We further assume three time zones (Regions), each one time slot shifted in time compared to the neighboring one: while the most western Region 1 goes through (A, B, C), simultaneously Region 2 goes through (B, C, A) while the most eastern Region 3 goes through (C, A, B). We distribute the total traffic volume, i.e., the total set of requests (over the whole day), over the three Regions proportionally to the number of network nodes in each region. As topology, we use the

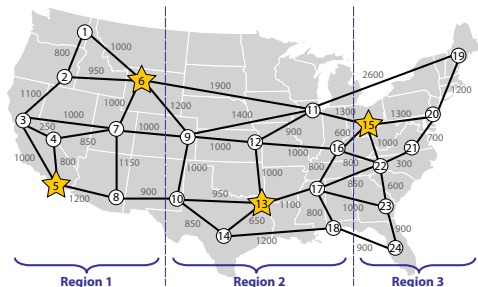


Fig. 2: USA network with our 3 regions. Yellow stars indicate the data centers, labels of links l their length $\|l\|$ in km.

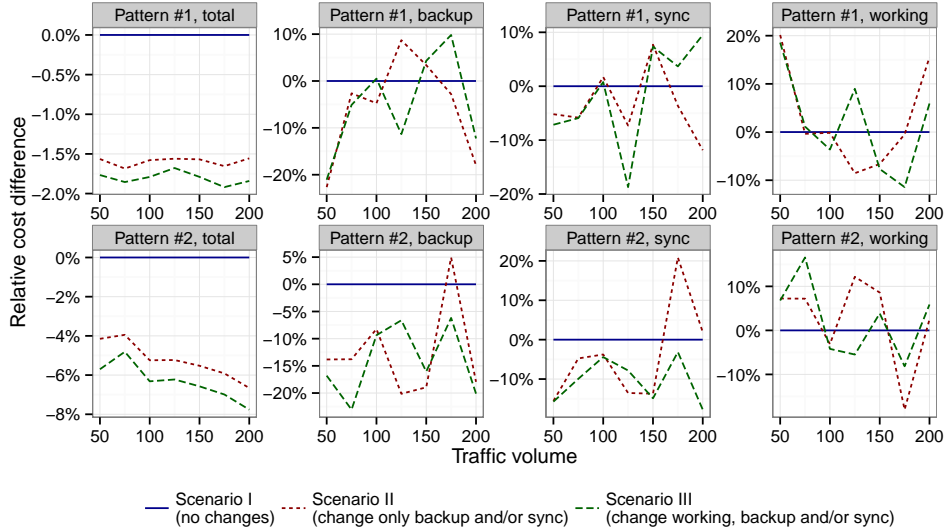


Fig. 3: Relative cost difference compared to not changing any routing from one period to the next (i.e., relative cost change compared to the corresponding Scenario I solution).

USA network illustrated in Fig. 2, where 33.33% of traffic requests originate in Region 1, 37.50% in Region 2 and 29.17% in Region 3. We will consider two cases:

- *Pattern #1*: 20% of requests in each time zone and each time slot just last two slots, while the other 80% last just for the single time slot where they start.
- *Pattern #2*: 80% of requests in each time zone and each time slot last two slots, 20% last just one.

4.2. Bandwidth Requirements

The relative change in bandwidth cost (i.e., the first summation of the optimization objective (1)) for the various scenarios is shown in Fig. 3. From these numerical results, we learn that the total bandwidth cost is reduced with on average 1.6% (resp. 1.8%) for Scenario II (resp. Scenario III) with traffic Pattern 1, and by 5.2% (resp. 6.3%) with Pattern 2 (where the average is taken over all traffic instances). This net saving mainly stems from a reduction of bandwidth for the backup paths, due to increased sharing: we noted an average reduction of the backup bandwidth cost with on average 5.5% (resp. 5.0%) for Pattern 1 and 12.6% (resp. 14.0%) for Pattern 2, for Scenario II (resp. Scenario III). We verified that these savings do not require all 2-period traffic requests to change their routing when going from one period to the next, but only about half of them. Further, these preliminary results suggest that the cost advantage can be achieved by only changing the backup/synchronization paths (Scenario II): there is only a limited advantage of allowing also the working path to be changed (Scenario III).

5. Conclusion

We studied the interest of re-provisioning the working and the backup paths in the context of resilient anycast routing traffic in cloud computing, assuming time-varying traffic, where the path provisioning can be updated periodically, assuming a time-slotted routing approach. We proposed a global optimization model, considering optimization of the routing over a set of multiple subsequent time slots in a single step. We propose a column generation model, using a path formulation in the master problem and pricing problems to find new (combinations of) paths. In an initial (small) case study, we note that the bandwidth cost savings may be reduced with up to almost 8% of the total cost, without needing to change all requests that survive multiple time slots (i.e., only about half of them need routing changes to realize these cost savings). Also, our small example suggests that the added value of also allowing the working routes (as opposed to only backup/synchronization routes) seems limited. Further experiments

should confirm these hypotheses. Also, we suggest to evaluate the impact of the server locations (e.g., scattered vs. paired as in [9]), and investigate a broader range of time-varying traffic patterns.

ACKNOWLEDGMENT

B. Jaumard has been supported by a Concordia University Research Chair (Tier I) and by an NSERC (Natural Sciences and Engineering Research Council of Canada) grant.

REFERENCES

- [1] C. Develder, J. Buysse, B. Dhoedt, and B. Jaumard, “Joint dimensioning of server and network infrastructure for resilient optical grids/clouds,” *IEEE/ACM Trans. Netw.*, vol. 22, no. 5, pp. 1591–1606, Oct. 2014.
- [2] X. He and G.-S. Poo, “Sub-reconfiguration of backup paths based on shared path protection for WDM networks with dynamic traffic pattern,” in *Proc. 9th Int. Conf. Commun. Systems (ICCS 2004)*, Singapore, China, 7 Sep. 2004, pp. 391–395.
- [3] S. Srivastava, S. R. Thirumalasetty, and D. Medhi, “Network traffic engineering with varied levels of protection in the next generation internet,” in *Performance evaluations and planning methods for the next generation internet*, A. Girard, B. Sansò, and F. Vazquez-Abad, Eds. Springer Verlag, 2005, pp. 99–124.
- [4] S. Sebbah and B. Jaumard, “Differentiated quality-of-protection in survivable WDM mesh networks using p -structures,” *Computer Commun.*, vol. 36, pp. 621–629, Mar. 2013.
- [5] M. Bui, T. Wang, B. Jaumard, D. Medhi, and C. Develder, “Time-varying resilient virtual network mapping for multi-location cloud data centers,” in *Proc. 16th Int. Conf. Transparent Optical Netw. (ICTON 2014)*, Graz, Austria, 6–10 Jul. 2014, pp. 1–8.
- [6] C. Develder, M. De Leenheer, B. Dhoedt, M. Pickavet, D. Colle, F. De Turck, and P. Demeester, “Optical networks for grid and cloud computing applications,” *Proc. IEEE*, vol. 100, no. 5, pp. 1149–1167, May 2012.
- [7] M. Bui, B. Jaumard, and C. Develder, “Anycast end-to-end resilience for cloud services over virtual optical networks (invited),” in *Proc. 15th Int. Conf. Transparent Optical Netw. (ICTON 2013)*, Cartagena, Spain, 23–27 Jun. 2013.
- [8] V. Chvatal, *Linear Programming*. Freeman, 1983.
- [9] M. Bui, B. Jaumard, and C. Develder, “Resilience options for provisioning anycast cloud services with virtual optical networks,” in *Proc. IEEE Int. Conf. on Commun. (ICC 2014)*, Sydney, Australia, 10–14 Jun. 2014.