Dimensioning backbone networks for multi-site data centers: exploiting anycast routing for resilience

(Invited Paper)

Chris Develder
INTEC – IBCN
Ghent University – iMinds
Ghent, Belgium
Email: chris.develder@intec.ugent.be

Brigitte Jaumard
Computer Science and Software Engineering
Concordia University
Montreal (QC) H3G 1M8 Canada
Email: bjaumard@cse.concordia.ca

***Abstract*—In the current era of big data, applications increasingly rely on powerful computing infrastructure residing in large data centers (DCs), often adopting cloud computing technology. Clearly, this necessitates efficient and resilient networking infrastructure to connect the users of these applications with the data centers hosting them. In this paper, we focus on backbone network infrastructure on large geographical scales (i.e., the so-called wide area networks), which typically adopts optical network technology. In particular, we study the problem of dimensioning such backbone networks: what bandwidth should each of the links provide for the traffic, originating at known sources, to reach the data centers? And possibly even: how many such DCs should we deploy, and at what locations? More concretely, we summarize our recent work that essentially addresses the following fundamental research questions: (1) Does the anycast routing strategy influence the amount of required network resources? (2) Can we exploit anycast routing for resilience purposes, i.e., relocate to a different DC under failure conditions, to reduce resource capacity requirements? (3) Is it advantageous to change anycast request destinations from one DC location to the other, from one time period to the next, if service requests vary over time?**

I. Introduction

Over the past few decades, the networking and computing industry has gradually moved towards the realization of the old idea of "computation provided as a public utility" as suggested back in 1961 by John McCarthy. Optical network technology, providing high capacities and low latencies form the crucial foundation of that evolution [1]. In the 1990s, the *grid computing* idea came to fruition: mainly originating in academic circles, the grid concept was conceived to offer access to powerful computing facilities to remote users. Pushing that idea further to essentially provide computing power on demand, the *cloud computing* concept originated in the 2000s. The development of various tools to not only efficiently partition and safely share resources among different users, but also to distribute massive workloads over multiple physical machines, more recently led to the era of *big data*. These applications, which are challenging in terms of volume, velocity, variety and/or veracity[1], can be cost-efficiently supported by adopting optical technology [2].

[1]Dimensions of big data known as the four Vs, see http://www.ibmbigdatahub.com/infographic/four-vs-big-data.

In this paper, we discuss our recent work on dimensioning backbone (optical) networks in light of these novel cloud-enabled applications. With dimensioning, we mean determining the capacity required in a given network topology (e.g., amount of wavelengths occupied on each of the optical links), to support a given amount of service requests (or "traffic" in more general terms). In particular, we will focus on designing resilient networks: we will determine the required capacity that will suffice not only to satisfy all requests in a fully operational network, but also in case when some failure occurs (in which case the affected traffic will follow an alternate route, circumventing the failure(s) at hand). While traditionally a substantial body of work in research literature has intensively studied several variants of optical network dimensioning problems (e.g., the classical routing and wavelength assignment (RWA) problems), those solutions typically cannot be directly applied in the cloud scenario. Cloud computing essentially is characterized by two core ideas that are not accounted for in more traditional optical networking literature: (i) anycast routing, and (ii) virtualization.

The principle of *anycast routing* is that for a given service request, originating from a known node in the network topology, the destination is not fixed a priori, but rather can be chosen out of a set of candidate destinations. For example, a service request could imply the allocation of virtual machines (see below) for a particular customer at a data center (DC) that can be freely chosen among a set of candidate ones. Indeed, users in general do not care where their applications end up being served. Looking back to classical (optical) network dimensioning as studied in research literature, this anycast concept fundamentally changes the input to the dimensioning problem. Indeed, the service requests can no longer be modeled as a so-called traffic matrix that specifies the bandwidth exchanged by every source node pair. Instead, we only know the source a priori (since the destination needs to be chosen out of a set of DCs).

Virtualization refers to the general concept of logically partitioning physical resource(s) in order to share the physical infrastructure among different users, while also providing isolation between them. To the users, it is as if they are using their own physical resource. Yet, if users are using the resource

only part of the time, the physical capacity can be shared among multiple users, thus leading to potential cost savings compared to effectively granting each user exclusive access to a physical resource. This virtualization concept is not only common in computing (cf. the notion of virtual machines), but it is also adopted in networking: a physical network may be managed by a so-called Physical Infrastructure Provider (PIP), that provides services to instantiate a (virtual) network topology on demand to so-called Virtual Network Operator(s) (VNOs). VNOs in turn may then interact with their own customers, running actual applications that require network services over the VNO's (virtual) network. This amounts to a multi-layer network architecture (e.g., see Fig. 4).

These fundamental characteristics of cloud networks substantially change the assumptions in the classical research literature on optical network dimensioning. This gives rise to some essential research questions on dimensioning optical networks to support cloud applications, which we will discuss in more detail in the next sections:

1) What is the impact of the anycast routing strategy on the amount of required network resources? (Section II)
2) Does it make sense to exploit anycast routing for resilience purposes, i.e., relocate to a different DC under failure conditions and thus reduce resource capacity requirements? (Section III)
3) Is there an advantage of rerouting anycast requests to different DC locations if the service requests vary over time? (Section IV)

After discussing each of these questions in turn, we will conclude the paper and outline possible future work in Section V.

II. Impact of anycast routing on network load

Before considering resilience, the first problem we study is the dimensioning of cloud infrastructure with a given topology for a given set of anycast requests. Intuitively, at least two factors will influence how much traffic will cross each link: (i) the strategy to choose a particular DC for a given request, and (ii) the location and capacity of data centers (DCs).

We studied these questions in our original paper [3] (which was phrased in a grid computing context, but is generic in terms of its methodology and analysis). In particular, we proposed a solution for the following problem:

Given:

- The network topology, represented as graph (with nodes V representing switches and DCs, links L the optical fibers connecting them),
- The demand, formalized as the arrival rate of requests at each source node (i.e., a subset $V_S \subset V$),
- The processing capacity of a single server in a DC,

Find:

- The location of k data center sites, to be chosen among (a subset of) all nodes V,
- The number of servers to install in each chosen DC,
- The link capacity, i.e., bandwidth, to provide on each link $\ell \in L$,

Such that we minimize the total network capacity (i.e., link bandwidths summed over all $\ell \in L$), and meet the service quality criteria[2].

Our proposed solution to determine both data center and network capacities was a phased approach, comprising the following sequential steps:

(S1) Find the k best DC locations,
(S2) Determine the DC capacities,
(S3) Determine the network link capacities.

For step S1, we use a simple ILP formulation (which solves more quickly than the more naive heuristic of using a k-means clustering algorithm), where we make the simplifying assumption that each source node will send all of its traffic to the closest DC. The latter stems from our intuition that in order to minimize the total network capacity (cf. the objective of the above problem statement), the best strategy to pick a particular destination DC for a given source node is to choose the nearest one. In step S2, we first determine the total number of servers (over all DCs together) based on the service quality criteria.[3] Then, we distribute the servers over the individual k DC locations heuristically: we compared (i) a naive *uniform* distribution (i.e., the same number of servers for each site) with (ii) one that distributed the number of servers *proportionally to the traffic* that would arrive to that location (assuming shortest path routing to the closest site). Finally, to find network capacities in step S3, we used simulations[4]. For this we considered the following scheduling strategy to solve the anycast choice between server sites: (1) always first try the closest DC, and only in case no servers are available there, (2) try another (further) DC. For the latter step (2), we compared three alternatives for picking the alternate DC:

- *Random:* a naive baseline of randomly picking any DC that still has a server available,
- *Mostfree:* pick the DC with the highest number of free servers, or
- *SP:* pick the nearest DC that still a free server, thus assuming shortest path (SP) routing to drive the DC scheduling choice.

Results of applying this method to a European network topology (see [3] for detailed case study assumptions) are shown in Fig. 1. From these results, we observe that

- There is an optimal value of the number of DC locations, i.e., k, which tends to be reasonably low (around 5);
- The total number of servers are best distributed non-uniformly over all k locations, i.e., proportionally to the traffic originating closest to each location; and

[2]For example, in [3] we considered a maximal blocking probability of arriving requests.

[3]In [3], we assumed Poisson arrivals and exponentially distributed service times, so we used the well-known Erlang-B formula to find the number of requested bandwidth units.

[4]To analytically calculate the amount of traffic arriving at each DC surprisingly is not easy for the general case, although (rather complex) derivations are possible for special cases such as a large number of source sites that can be partitioned into a limited number of classes [4].

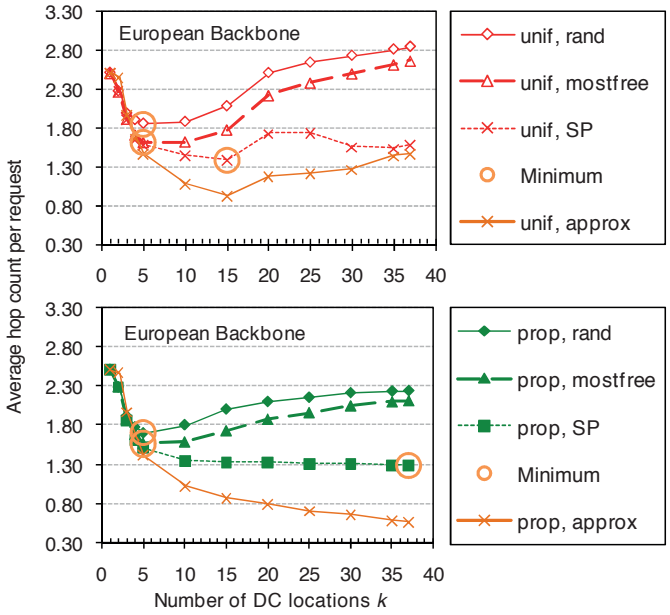

Fig. 1. The required network capacity, which is proportional to the average hop count per request, is minimized by adopting shortest path routing and scheduling (SP), intelligently positioning server capacity (*prop*: proportional to the traffic originating closest to that DC), and deploying a reasonable number of server locations. The circled points indicate minimal hop count values per data series. The *approx* series is the result of analytical approximations and typically underestimates resource requirements.

- The anycast routing strategy is best set to pick the closest DC with free servers ("SP") if we want to minimize bandwidth requirements.

To answer the 1st high level question as stated at the end of Section I, we thus find that we can indeed influence (i.e., minimize) bandwidth requirements by exploiting the freedom of anycast routing to control the choice of DC destination.

III. Anycast routing for resilience

Given the high bandwidth that crosses a typical optical link, resilience against failures is of prime importance, especially since an increasing amount of (critical) applications is being pushed to the cloud. For the optical backbone networks that support those applications with long distance connections, varying resilience strategies have been devised [5], [6]. A well-known and well-studied scheme is that of shared backup paths: e.g., to protect against single link failures, a primary path from source to destination is protected by a link-disjoint backup path. The "sharing" refers to non-exclusive use of (in this case link) resources: capacity on a link of the backup path A' for primary path A can be reused for a backup path B' for primary path B, on the condition that no failure simultaneously affects both primaries A and B. Now, for the case where the anycast principle applies, such as in typical cloud computing scenarios, we proposed the idea of relocation [7]: allow the backup path to lead to a DC that is different from the end point of the corresponding primary path. The question that we now want to answer is whether this relocation makes sense (compared to sticking to the same DC destination), in terms of potential reduction of the amount of resources required to resiliently provision a given set of cloud service requests. We first look into this question for the case of static traffic. (For time-varying traffic, see the next Section IV.)

It is worthwhile to point out that "resources" in the cloud scenarios of interest comprise not only the network resources (e.g., number of used wavelengths summed over all links in the WDM network), but also the server resources (e.g., number of servers to install summed over all DCs): we will formulate and solve a dimensioning problem to jointly optimize both network and server resources. The formal problem statement is the following:

Given:

- The network topology, comprising the sites where cloud service requests originate, as well as the optical network interconnecting them,
- The demand, stating the volume of cloud service requests that originates at each of the source sites, and
- The survivability requirements, in the form of a set of failures to protect against,[5]

Find:

- The location of k data center sites, to be chosen among (a subset of) all nodes V, where to provide servers,
- The routes to follow for each request, i.e., the primary and backup paths (thus including the primary and backup DC location to use, chosen among the k DCs),
- The network and server capacity, i.e., link bandwidth (or number of wavelengths) and number of servers,

Such that the total resource capacity, comprising both the (optical) network and the server infrastructure in the DCs, is minimized.

In [8], we solve this problem in two consecutive steps:

(S1') Find the k best DC locations,

(S2') Determine the primary and backup DC locations and the paths towards them, for each request, and calculate the amount of network and server resources (which follows from the routing choices).

For step S1', we can fairly easily formulate an adaption of the ILP from step S1. For step S2', we now determine server and network capacities in a single integrated optimization problem. If we consider failure-independent rerouting, i.e., we chose a single backup path (and DC location) that is used regardless of the exact failure that affects the primary path, we use a column generation formulation to find the optimal routing choices. Basically, column generation is a decomposition technique to solve (I)LP instances that otherwise are not solvable in reasonable time. It is based on the notion of so-called *configurations*, where in our case a configuration is associated with a source node (where cloud requests originate), and comprises a particular primary path and a corresponding

[5]The model as detailed in [8] formalizes failures as so-called shared risk groups (SRGs), i.e., sets of resources that can fail simultaneously. Furthermore, the graph model $G = (V, L)$ adopted there allows to represent also DC failures as failing edges in the graph, such that a failure f is represented as a subset of L.

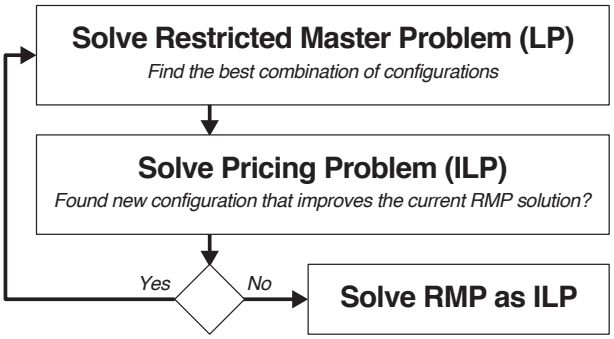

Fig. 2. The general column generation approach.

backup path. The optimization problem is then split into a so-called *(restricted) master problem (RMP)* and a corresponding *pricing problem (PP)*. The RMP in our case then determines for a given (limited) set of configurations C determines what configuration choices lead to the lowest resource requirements. The PP subsequently is solved to find a new configuration c that could potentially further reduce these requirements (when added to the set C in the RMP). The RMP and PP are then solved iteratively, following the general scheme sketched in Fig. 2. The full optimization model solving the above problem is detailed in [8].

We now use the mentioned model to answer the question: is it beneficial to exploit anycast routing for resilience purposes, and adopt relocation to a backup DC site that is different from the primary? Intuitively, it is clear that a path to a different DC (i.e., a relocated backup path) may be found that is shorter than a path to the original primary DC that is also disjoint from the original path. How likely that possibility is will depend on the sparsity of the network topology: we expect that how sparser the network, the likelier a relocated backup path will be shorter than a non-relocated backup to the original primary destination. Further, we also speculate that for higher number of DCs (i.e., higher k), the benefit of relocation will increase, since we expect a higher chance that we may find another DC on (or close to) an alternate path to the original DC (or that the path to the second closest DC is shorter than the second shortest disjoint path to the original DC).

A sample result of the case study on a 28-node network topology — see [8] for a full list of the case study settings — is given in Fig. 3. (Note that for the total cost values, the single link bandwidth cost for a single unit request is assumed to be the same as the server capacity cost incurred at a DC.) Examining these results leads to the following observations:

- For protection against single link failures (1L), relocation (RO) leads to a reduction of the network cost for the backup paths, while the server cost slightly rises (given that extra resources are required at different DCs) compared to the case of no relocation (NR). There is still a net cost advantage (under the aforementioned assumed cost ratios of link bandwidth vs. server cost), that ranges up to 10% in the case study at hand;
- For protection against failures of either network links or servers (1LS), relocation (RO) allows a potentially significant reduction of server resources[6], while the network cost savings are slighly lower than in the 1L case;
- As expected, the cost advantage of exploiting relocation (RO vs. NR) increases when the number of DC sites (k) is higher.

Thus, the 2nd high level research question stated at the end of Section I again can be answered positively: it *can* make sense to relocate to alternate DCs as to minimize the amount of resources required to satisfy a given set of cloud service requests. Note that we say "*can*", since additional results (not shown here) for topologies of varying sparsity suggest that when the network topology becomes dense (i.e., node degrees increase), the cost advantage of exploiting relocation (RO) eventually disappears [8].

IV. Anycast (re)routing for multi-period traffic

So far, we have shown that how exactly we choose to solve the anycast routing decision (i.e., how we choose the single destination DC to allocate traffic to) may have a significant impact on the required amount of overall network bandwidth, and that we can beneficially exploit the destination DC choice also to provide resilience by adopting relocation under failure conditions. Both questions considered static network traffic: the amount of requests originating from a given source site was assumed to be stable, i.e., did not change over time. Answering a third question, we now investigate whether or not it makes sense to relocate traffic originating at a given source from one destination DC to another, when traffic conditions vary over time. We still answer this question when considering resilience: apart from a primary DC and a route to get there, we also assume that we need to pre-calculate a backup path to an alternate DC. Furthermore, we will now also account for network capacity to connect the primary and backup DCs, as to keep them synchronized: we assume that traffic needs to be exchanged to maintain the backup DC in sync with that of the primary, as to guarantee fast switch-over to the backup (with minimal service impact) if needed. This amounts to the general routing setup sketched in Fig. 4.

The problem statement that we address now is:[7]

Given:

- The (physical) network topology, represented as graph $G = (V, L)$ (with nodes V representing switches and DCs, links L the optical fibers connecting them);
- The *time-varying* demand, formalized as a volume of requests at each source node $v \in V_s (\subset V)$, for every timeslot $t \in T$, thus denoted as $\Delta_{v,t}$, and we furthermore assume that to keep primary and backup DC in sync, a fraction δ_v of the full traffic bandwidth is required;

[6]The case study assumes 1:1 server protection in the NR case, whereas relocation with shared backup resources basically amounts to 1:k shared protection when we have k data centers.

[7]Remark that we do not consider the server resources in this case, although accounting for them in is in principle fairly easy: adopt the same modeling approach as for the static traffic case as summarized in Section III [8].

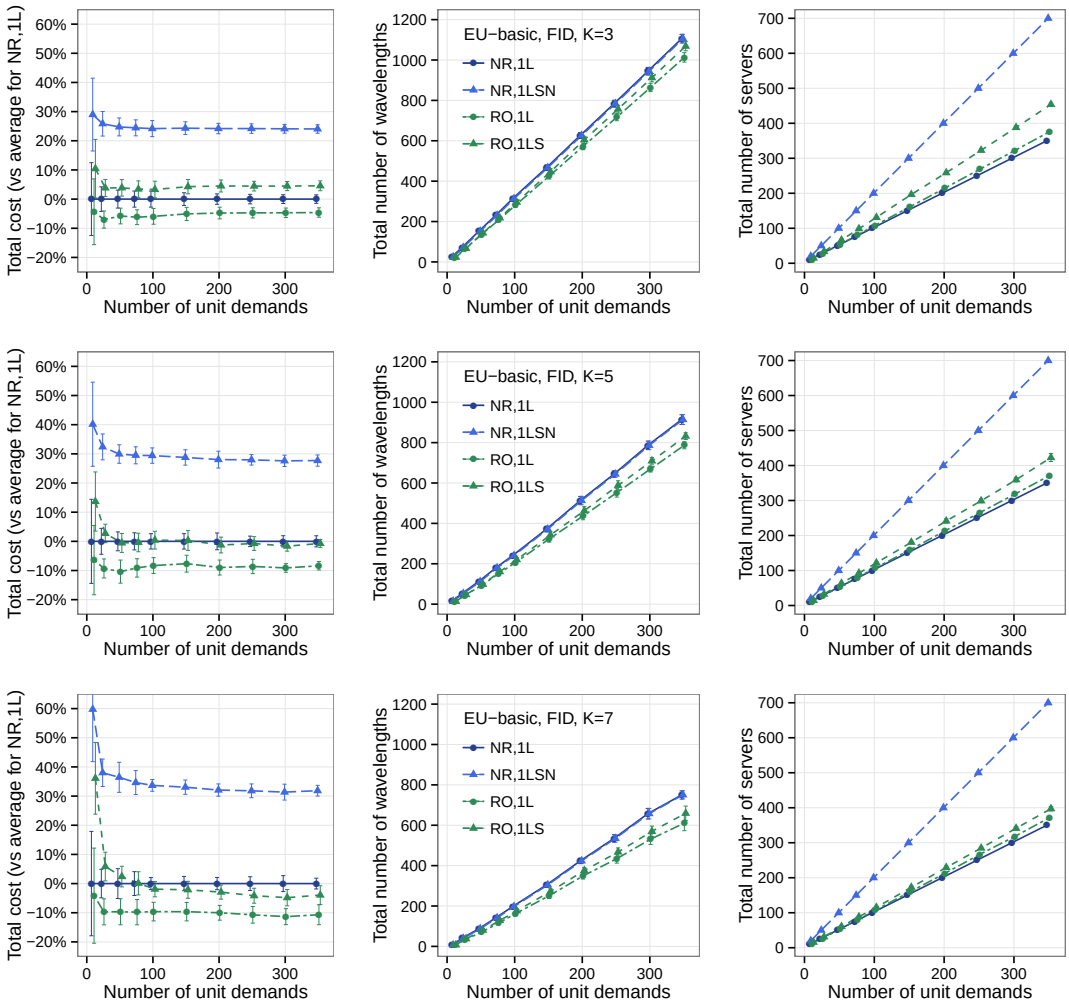

Fig. 3. Cost comparison of exploiting relocation (RO) vs. no relocation (NR), for the cases of single link failures (1L) as well as single failures that are either a network link (1L) or a server (1LS)). Graphs show, from left to right: (i) relative total cost compared to the NR, 1L case, (ii) total number of wavelengths, (iii) total number of servers. Error bars indicate 95% confidence intervals derived from the 10 instances per data point. *(Graph taken from [8].)*

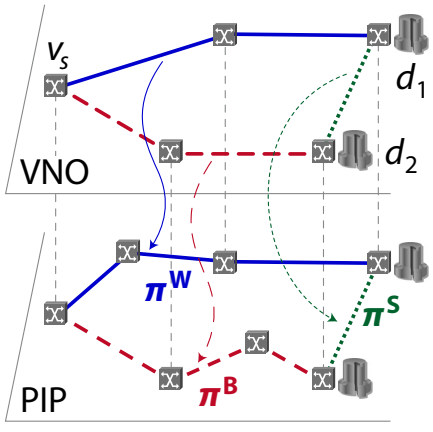

Fig. 4. The VNO-resilience scheme.

- The location of k data center (DC) sites;
- The failures to protect against (e.g., single link failures and complete DC failures);

Find:

- The link capacity, i.e., bandwidth, to provide on each link $\ell \in L$;
- The working path to the primary DC, the backup path from source to secondary DC, and the synchronization path between primary and backup DC;

Such that we minimize the total network capacity (i.e., link bandwidths summed over all $\ell \in L$).

We will solve this problem for varying degrees of rerouting flexibility:

- In *Scenario I* (the conservative baseline), we do not allow reconfiguring already established paths;
- In *Scenario II* we only allow reconfiguring backup and/or synchronization routes (π^B and/or π^S) for traffic that continues from one period to the next;

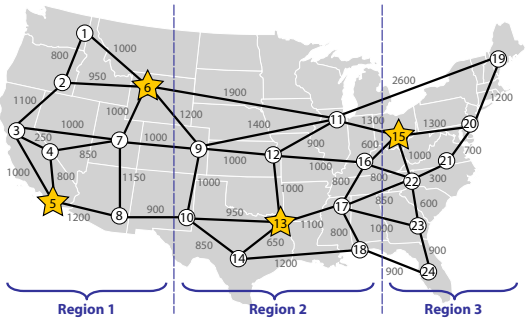

Fig. 5. USA network with 3 artificial time zone regions. Yellow stars indicate the data centers, labels of links l their length $\|l\|$ in km.

- In *Scenario III* we assume complete freedom and thus also allow to change the primary paths (π^{W}).

To solve this problem, we again developed a column generation model, detailed in [9]. Here, we summarize the results of a first case study on a nation-wide US network topology, with an synthetic 3-period traffic profile spanning 3 geographical time zones [9]. Note that this is an artificial setup and thus results are to be seen just as a proof-of-principle: we will focus on drawing qualitative conclusions (rather than, e.g., taking quantitative benefits of rerouting as indicative for real-world scenarios).

A. Case study setup

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 shifted one time slot 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 USA network illustrated in Fig. 5, 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 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 time slot last two slots, 20% last just one.

B. Results

As main performance metric of interest, we define the *total network cost* as the sum over all links $\ell \in L$ of the bandwidth allocated on the link ℓ multiplied with its length $\|\ell\|$. The relative change in bandwidth cost for the various scenarios is shown in Fig. 6 (using the baseline Scenario I as reference). From these numerical results, we learn that the total bandwidth cost is reduced with on average 5.1% (resp. 6.4%) for Scenario II (resp. Scenario III) with traffic Pattern #1, and by 6.9% (resp. 8.2%) 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 11.5% (resp. 13.4%) for Pattern #1 and 14.2% (resp. 16.3%) for Pattern #2, for Scenario II (resp. Scenario III). 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).

In summary, this study suggests that the answer to the 3rd question raised at the end of Section I again is positive: we can save on the amount of required network resources (i.e., bandwidth) by rerouting anycast requests to different DC locations if the service requests vary over time. Yet, it seems that keeping the working paths to the primary DC (i.e., the π^{W} path in Fig. 4) fixed — thus not affecting traffic routing, and thus the perceived network service — and only (potentially) changing the backup DC and routes thereto (π^{B}, π^{S}) suffices to obtain the bulk of the possible bandwidth reduction.

V. Conclusion and Outlook

We have presented an overview of our work in the domain of dimensioning backbone networks to support cloud-like applications that adhere to the anycast routing principle: to serve a request, we have the freedom of choosing a destination data center (DC) out of a given candidate set. We answered three fundamental research questions on resiliently dimensioning backbone networks, where we adopt the shared protection idea, thus sharing backup resources to protect against different failure scenarios (rather than to exclusively reserve resources for each failure scenario independently). We qualitatively conclude that:

1) The exact choice of anycast routing strategy, i.e., how exactly the destination DC is picked, may have a substantial effect on the bandwidth requirements. The latter depend also on the number (and location) of DC sites, but we suggest that wisely picking a relatively low number of DCs (with optimized routing strategies to pick a request's chosen destination) achieves minimal network resource cost [3].
2) It does make sense to exploit anycast for resilience, and thus relocate traffic to a different DC under failure conditions, as to reduce resource capacity requirements [8].
3) It also is advantageous to reroute anycast requests to different DC locations if traffic patterns change from one period to the next. Our preliminary results suggest that it even suffices to only adapt the backup routes (and thus secondary DC choices) to obtain (close to) minimal bandwidth cost [9].

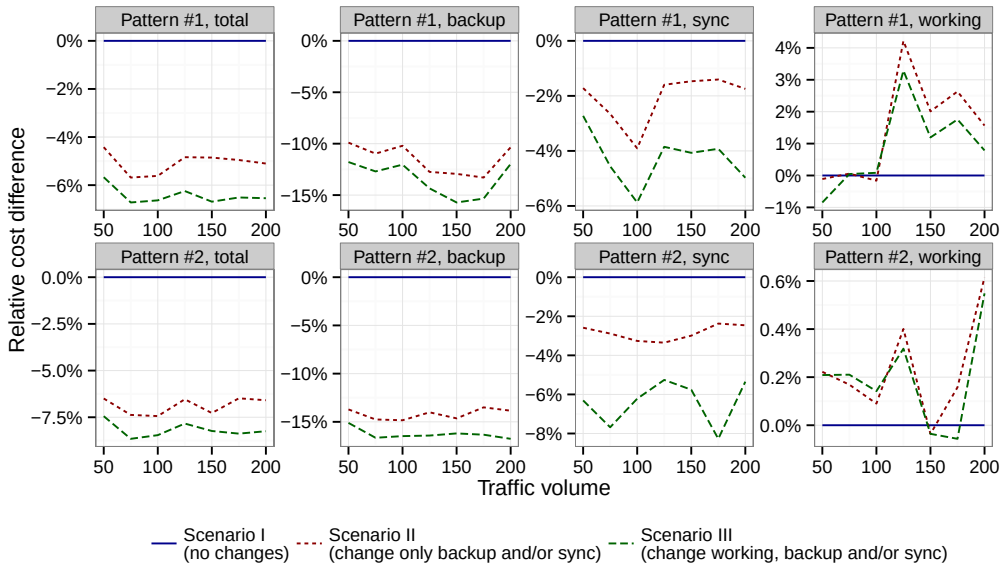

Fig. 6. Relative cost difference compared to not changing any routing from one period to the next (i.e., relative cost change compared to the corresponding Scenario I solution). *(Graph taken from [9].)*

Even though the three major studies summarized above do answer fundamental research questions, further work is required to further solve the problem of resilient network dimensioning for cloud(-like) traffic that allows anycast (re)routing. We see the following (partially) open problems that the research community can further address:

- A single-step dimensioning strategy, that optimizes the choice of server locations, integrated with the routing decisions (as opposed to the phased approaches summarized in the current paper);
- In-depth study and enhancement of scalability of models to real-world, large, network topologies and realistic multi-period traffic conditions;
- Development and analysis of online routing strategies, e.g., and then study the impact of the frequency of rescheduling/rerouting of requests (e.g., the impact of the granularity of "stable" traffic periods);
- Determine whether our qualitative results change in light of elastic, and/or transparent optical networking: there might be constraints for changing bandwidth on an existing path, etc.
- Deal with non-uniform failures: our current models assume that all failures have the same probability, and we need to equally protect against all of them. But what if different failures have different probabilities? Can we translate the model to minimize failure probability, or keep it below a given bound (rather than to minimize total cost).
- Can we incorporate QoS diversity of the services requested, and thus potentially have different resilience requirements: e.g., does that impact the potential overall bandwidth saving stemming from relocation for resilience?

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, 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.

[2] K. Walkowiak, M. Woźniak, M. Klinkowski, and W. Kmiecik, "Optical networks for cost-efficient and scalable provisioning of big data traffic," *Int. J. Parallel Emerg. Distrib. Syst.*, vol. 30, no. 1, pp. 15–28, Jan. 2015.

[3] C. Develder, B. Mukherjee, B. Dhoedt, and P. Demeester, "On dimensioning optical grids and the impact of scheduling," *Photonic Netw. Commun.*, vol. 17, no. 3, pp. 255–265, Jun. 2009.

[4] B. Van Houdt, C. Develder, J. F. Pérez, M. Pickavet, and B. Dhoedt, "Mean field calculation for optical grid dimensioning," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 2, no. 6, pp. 355–367, Jun. 2010.

[5] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *IEEE/OSA J. Lightwave Techn.*, vol. 21, no. 4, pp. 870–883, Apr. 2003.

[6] J.-P. Vasseur, M. Pickavet, and P. Demeester, *Network recovery: Protection and Restoration of Optical, SONET-SDH, IP, and MPLS*. Elsevier, 2004.

[7] J. Buysse, M. De Leenheer, B. Dhoedt, and C. Develder, "Exploiting relocation to reduce network dimensions of resilient optical grids," in *Proc. 7th Int. Workshop Design of Reliable Commun. Netw. (DRCN 2009)*, Washington, DC, USA, 25–28 Oct. 2009, pp. 100–106.

[8] 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.

[9] T. Wang, B. Jaumard, and C. Develder, "Network mapping for resilient multi-siteite data centers," in *Proc. IEEE Int. Conf. Advanced Netw. and Telecommun. Sys. (ANTS 2015)*, Kolkata, India, 15–18 Dec. 2015.