Joint Dimensioning of Server and Network Infrastructure for Resilient Optical Grids/Clouds

Chris Develder, *Member, IEEE,* Jens Buysse, Bart Dhoedt, *Member, IEEE,*
Brigitte Jaumard, *Senior Member, IEEE*

***Abstract*—We address the dimensioning of infrastructure, comprising both network and server resources, for large-scale decentralized distributed systems such as grids or clouds. We design the resulting grid/cloud to be resilient against network link or server failures. To this end, we exploit relocation: under failure conditions, a grid job or cloud virtual machine may be served at an alternate destination (i.e., different from the one under failure-free conditions). We thus consider grid/cloud requests to have a known origin, but assume a degree of freedom as to where they end up being served, which is the case for grid applications of the bag-of-tasks (BoT) type or hosted virtual machines in the cloud case. We present a generic methodology based on integer linear programming (ILP) that (1) chooses a given number of sites in a given network topology where to install server infrastructure, and (2) determines the amount of both network and server capacity to cater for both the failure free scenario and failures of links or nodes. For the latter, we consider either failure independent (FID) or failure dependent (FD) recovery. Case studies on European scale networks show that relocation allows considerable reduction of the total amount of network and server resources, especially in sparse topologies and for higher numbers of server sites. Adopting a failure dependent backup routing strategy does lead to lower resource dimensions, but only when we adopt relocation (especially for a high number of server sites): without exploiting relocation, potential savings of FD versus FID are not meaningful.**

***Index Terms*—Grid computing, Cloud computing, Optical networks, Anycast, Dimensioning, ILP, Column generation, linear programming.**

I. Introduction

The emergence of e-Science applications has been a major driver for Grid computing. Solving scientific and engineering problems increasingly relies on the availability of substantial computing and storage resources, which can only be made available at reasonable cost by sharing infrastructures. Initially those scientific computing needs were addressed by dedicated high performance computing (HPC) infrastructure, such as clusters. During the 1990s, the distributed computing community evolved towards the idea of what became commonly known as a Grid [1]: in analogy with the power grid, this would allow consumers to obtain computing power on demand. The development of this concept gave birth to many world-wide grid infrastructure initiatives [2]–[4]. Scientific experiments run on such grids/clusters are submitted in units called jobs, thus requiring specific interfaces for job submission, and schedulers with associated queuing mechanisms to run them (i.e., making the decision when and where to run what job/task, e.g., [5]). Applications are thus implemented as a bag-of-tasks (BoT) applications, workflows and MPI parallel processes. The complexity of figuring out where to run each constituent of such sets of interdependent tasks clearly increases, and various approaches to tackle the NP-complete problem (i.e., allocating the interdependent tasks to minimize total communication and execution costs) have been identified [6].

Building on the basic concepts of grids, clouds manifest themselves in more commercially oriented applications. A key characteristic that clouds exploit is that of virtualization, as in the case of Infrastructure-as-a-Service (IaaS) where one or more virtual machines (VMs) are deployed on actual physical servers. That virtualization enables migration to other servers, both for performance and resilence against failures, and the problem arises as how to make the choice of provisioning which VMs at which physical servers (as well as the connectivity towards it). To facilitate both grid and cloud applications with efficient communication network infrastructure, optical networks will play an important role (for an overview on optical grids/clouds, see [7]).

In the current paper, we focus on an offline resilient dimensioning problem: how to determine the amount of network and server resources that are needed to meet a certain demand of grid/cloud applications, under failure free conditions as well as under single network or server failures? For the applications, we will focus on those that can be executed at a single location, which is typical for BoT as found in many science and engineering applications (as explained in [8]) or VM provisioning in the IaaS cloud case. We will cater for applications that have non-negligible communication and computation costs (i.e., which are not particularly biased towards either data- or computation-intensive tasks). Thus, we assume the traffic we need to cater for requires a given amount of server resources, as well as a certain bandwidth between the origin and the server location.

Manuscript received July 3, 2012. Revised manuscript received March 3, 2013, and 2nd revision on June 17, 2013. Final version received September 25, 2013. Work described in this paper was partly funded by the European Commission through the 7th ICT-Framework Programme (FP7) project Geysers (grant FP7-ICT-248657) and by Ghent University through GOA Optical Grids (grant 01G01506). Results were obtained using the Stevin Supercomputer Infrastructure at Ghent University. J. Buysse was funded through a Ph.D. grant from the Agency for Innovation by Science and Technology (IWT). B. Jaumard was supported by NSERC (Natural Sciences and Engineering Research Council of Canada) and by a Concordia University Research Chair (Tier I) on the Optimization of Communication Networks.

C. Develder and B. Dhoedt are with Dept. of Information Technology – IBCN, Ghent University – iMinds, Ghent, Belgium. J. Buysse previously also was with IBCN, and as of Sep. 2013 is with Hogeschool Gent, Ghent, Belgium.

B. Jaumard is with Dept. of Computer Science and Software Engineering, Concordia University, Montreal (Qc) H3G 1M8, Canada.

Given the high bandwidth requirements typical for many of the applications that we envisage [7], we assume the underlying network will be an optical circuit-switched one, based on Wavelength Division Multiplexing (WDM). For such optical networks, the offline dimensioning problem has been widely studied, but not in the particular grid/cloud context we are considering here. Optical grid/cloud dimensioning is significantly different, and especially challenging for providers that need to plan and deploy both network and server resources (for both storage and computing). Since users of such grids/clouds typically do not care where exactly their workload is processed ("in the cloud"), freedom arises as to where to install, e.g., data centers. This amounts to the concept of anycast routing [9]: the destination is not a priori given, but can be chosen among a given set of candidate destinations. Consequently, a (source, destination)-based traffic matrix, as assumed in traditional (optical) network dimensioning problems — including many routing and wavelength assignment (RWA) approaches — is not a priori available in the grid/cloud scenario at hand.

To deal with network failures, various resilience strategies for WDM networks have been devised [10]. The well-known classical shared path protection scheme protects against single link failures: a primary path from source to destination is protected by a link-disjoint backup path which is used in case of a failing link (since this link diversity guarantees that the primary and backup paths will never fail simultaneously for any single link failure). In a grid/cloud-like scenario however, we proposed the idea of exploiting relocation [11], which is applicable given the anycast principle: the backup path is allowed to arrive at an alternate destination, possibly different from the primary path's end point under failure free conditions.

In this paper, we expand on the relocation idea to judge the resource requirements to also cater for server site failures — in fact, the dimensioning algorithms based on ILP formulations can cater for any failure that can be modeled as a shared risk link group (SRLG). The remainder of this paper is structured as follows: we start off in the following Section II with an overview of related work, where we also highlight the novel contributions of this paper. Next, in Section III we summarize our approach to resilient grid dimensioning, detailing its two phases and associated model assumptions in the subsequent Sections IV and V. Case studies on three 28-node European network topology variants are discussed in Section VI. Our overall conclusions are outlined in the final Section VII.

II. Related work

In IP networks, the anycast routing problem typically consists of finding a set of paths, one for each source node, such that a particular cost (delay, bandwidth used, etc.) is minimized. For this NP-hard problem, several heuristic algorithms have been proposed (e.g., see [12] and references in [13]).

The current paper addresses anycast in optical circuit-switched (OCS) WDM networks. (For works on optical burst switching (OBS) anycast, we refer to [14], [15].) The anycast routing problem in OCS WDM networks amounts to anycast routing and wavelength assignment (ARWA), finding routes for each anycast request while, e.g., minimizing the total number of wavelengths used, and/or the load on the links [16]. In [17], that *offline* problem for a given set of static traffic is solved in three subsequent phases: (*i*) destination decision, (*ii*) path routing, and (*iii*) wavelength assignment. This phased approach is shown to be outperformed by a heuristic algorithm (based on simulated annealing and genetic algorithms) in [13]. A generalized static *offline* RWA problem, comprising not only anycast, but also unicast and multicast requests, is described in [18], where heuristic algorithms are proposed to solve it. A similar problem is addressed in [19], but the author considers the joint routing of both unicast and anycast connections, and proposes a heuristic solution based on Langrangean relaxation. (Note that [19] also briefly raises the associated *online* routing problem. Heuristic solutions to online anycast routing in WDM networks are also studied in [20], where the authors propose to vary the number of candidate anycast sites over time, according to time-varying load, and highlight the impact of physical layer impairments.)

Whereas the above mentioned works addressed the anycast routing problem in WDM networks to find working paths from source to one of the candidate anycast destinations, the authors of [21] extended the problem to also find backup paths. Also, they considered grooming: traffic granularity is supposed to be sub-wavelength and hence at intermediate nodes traffic flows are re-combined to fill the wavelength channels as much as possible. They solved the *online* routing problem using an algorithm based on an auxiliary graph model, which finds working and backup routes for a single incoming anycast request. The *offline* problem, which we will focus on, is addressed in [22], which considers the optimization of both working and shared backup paths of anycast and unicast demands jointly. The authors consider protection against single link failures and apply shared path protection.

Note that the above works address the network dimensions (i.e., wavelengths) only. However, we are interested in grid/cloud scenarios, and hence also want to size the server resources (for storage and computation). *Online* routing approaches taking into account both network and server constraints for such a scenario are presented in, e.g., [23], [24]. (Note that we consider requests that are entirely served at a single data center; for, e.g., online scheduling of multiple interdependent tasks, we refer to [25], [26].)

In the current paper, we are addressing the *offline* dimensioning problem as first tackled in [27]. In that work, we proposed a phased approach to determining both network and server dimensions for an optical grid scenario, yet did not consider resiliency. A similar problem, but assuming mobile users, was addressed in [28] to find server locations and amount of servers for the case of mobile thin client computing.

The authors of [29] consider a problem setting that is very close to the one studied by us below: given a capacitated network, including servers, they determine the placement of content, as well as primary and backup routing of requests for that content, with a given maximum number of replicas per content item. Thus, the main differences between our work and [29] boil down to the following: (*i*) the focus in [29] is on minimizing used network resources, where server capacity is only indirectly controlled by limiting the number

of replicas per content item rather than minimizing/limiting the server capacity; (*ii*) scalability of the solution approach: [29] considers quite limited-scale case studies, i.e., limited network sizes (11–14 nodes) and small traffic demands (up to 30 requests), likely because their ILP approach does not seem to scale to larger problem instances; (*iii*) the candidate destination of an anycast (content) request in [29] is limited to a subset of all available data center locations, whereas we do not consider such limitation (although our model can be fairly straightforwardly extended by adding constraints); (*iv*) to protect against failures, [29] enforces relocation, whereas in our default model relocation is optional (but can easily be adapted to enforce it). Another difference in our approach is that we assume an uncapacitated network (although constraints can be easily added to cater for capacity limits).

To determine both network and server site dimensions for a grid/cloud infrastructure that is resilient against both server and network failures, we propose a two-step approach:

(S1) find the K most suitable locations to use as server sites where to install data centers, and subsequently

(S2) determine the amount of network and server capacity by finding suitable working and backup routes for all grid/cloud traffic.

For step S1, we will expand upon the initial integer linear programming (ILP) formulation from [27]. For step S2, in contrast to [27], we will now: (*i*) solve the sub-problems of establishing server and network capacity in an integrated way, and (*ii*) additionally provide resilience against both server and network failures. To this end, we will resort to an ILP-based solution using column generation, similar to [30], [31]. Compared to the latter two works, we now extend our recent work [32], [33] and (*i*) protect not only against network, but also server failures (or in general, any failure that can be modeled as an SRLG), (*ii*) simultaneously minimize network and server capacity (instead of only network capacity), (*iii*) do not fix the destination server site (under failure free conditions) a priori, and (*iv*) compare failure-independent (FID) versus failure-dependent (FD) backup path routing strategies.

Our novel contributions in the current paper beyond [32], [33], include

- an extensive comparison of alternative ways of finding K best server sites (beyond the simple approach we previously reused from [27]), and
- an assessment of the influence of both (*i*) the choice of the number of K server locations and (*ii*) the topology (particularly the nodal degree), on the benefits of exploiting relocation as well as the potential benefit of adopting failure-dependent (FD) backup path routing.

III. Dimensioning resilient grids/clouds

A. Problem statement

Stated formally, the dimensioning problem addressed is the following [32]:

Given

- *Topology* comprising the sites where grid/cloud requests originate, as well as the optical network interconnecting them;
- *Demand* stating the amount of requests originating at each of those sites; and
- *Survivability requirements* specifying the failures that should be protected against,

Find

- K *server site locations*, chosen out of a given set of candidate locations, where server infrastructure should be provided;
- *Destination sites and routes* to follow for all grid/cloud requests, originating with given intensity at the various source sites (where each destination should be one of the K server locations);
- *Network and server capacity* to provide on each of the links and server sites;

Such that the total resource capacity (comprising both server and network resources) is minimized.

Thus, the overall optimization objective will be to minimize the infrastructure cost, covering both the (optical) core network and the server capacity at each of the K data centers, while ensuring survivability (e.g., by exploiting relocation, see further). Also, note that we will consider unit requests (i.e., demanding a certain bandwidth and server capacity), where multiple units originating from the same source possibly may be sent to different server site locations.

As pointed out in the introduction, the requests we consider can represent jobs from grid applications, or virtual machines (VMs) to be provisioned in IaaS clouds, that can be met by a single server site (which we assume to house an entire data center, i.e., we consider dimensioning the backbone network interconnecting such centers rather than intra-data center interconnects). The *demand* will be expressed as a request arrival intensity, with which we will associate a certain network bandwidth to reserve between the request source site and a server destination site (to be chosen amongst the K server locations), as well as a certain amount of server capacity. The network bandwidth will be expressed as the sum of the number of wavelengths (aka lambdas) taken over all links, and the server capacity as the number of central processing units (CPUs) summed over all data center locations. Thus, our model is generic and can be used both for data- and computation intensive tasks.

To achieve resource capacity minimization, we will allow sharing of both server and/or network resources for the backup of requests whose resources under working conditions (aka *primary* wavelengths and servers) are disjoint. In particular, we will adopt a shared path protection [34], [35] concept. Similarly, at each server site, we will install the minimum capacity required to cope with each one of the considered failure scenarios (as well as the failure-free case, obviously). Thus, we will allow reclaiming of server and network resources for backup purposes, if they are no longer used as primary resource under failure conditions.

We also want to thoroughly assess the impact of relocation, as first studied in [11]: we will allow the backup destination to be different from the primary one (cf. anycast). For long running services (i.e., grid jobs, or applications communicating with the cloud VMs), one could assume this will involve

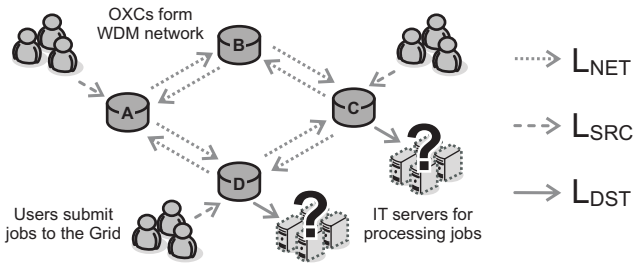


Fig. 1. Input data: (i) network links and nodes (OXCs labeled A-D), (ii) source nodes with job arrival intensity (represented as users), and (iii) candidate server sites.

migration. (Migration was originally developed for load balancing in a server cluster such as MOSIX [36] or Condor [37], and initially only applied to processes without inter-process communication, a drawback that was overcome in [38], [39] or could be circumvented by virtualization [40].) In the current work, we however do not take into account any extra resource requirements that such migration would involve, since we are focussing on a static network dimensioning problem (the request rates are to be seen as steady state traffic estimates, i.e., long-time averages or upper bounds thereof) and not on an online scenario that deals with short-term variations (i.e., we accept potential temporary degradation of the service during the failover procedure). For an assessment of failover techniques such as replication and checkpointing (which are thus out-of-scope in the work at hand), we refer to [41].

We will express the *survivability requirement* through the concept of a shared risk link group (SRLG): a set of resources (links) that may fail jointly, because of shared dependencies (e.g., fibre ducts [42]). Thus, to protect against failure of an SRLG, the backup resources should not include any of the SRLG elements. In our case studies, we will protect against single failures, where the single resource that fails can be either a server or a network link (whereas in earlier work, we only considered network failures [11], [30]). Also, our previous work [32] considered complete server site failures (which would amount to 1+1 protection in terms of number of server resources, if we do not relocate), whereas now our models more generically cater for 1:N server protection. We will adopt a network model (explained in detail in the next subsection) where such a server failure is modeled as a failure of a link to the data center site. Thus, failures of the real-world fiber links as well as servers will be modeled as SRLG failures comprising modeled links. Adopting this generic SRLG model, our ILP formulations will allow to study any failure scenario (e.g., the single link or server failures that we will analyze, but also complete OXC failures) that can be represented as an SRLG.

B. Network model

We will focus on WDM networks interconnecting the data centers providing grid/cloud services and consider the network model illustrated in Fig. 1:

- $G = (V, L)$, directed graph representing an optical grid/cloud, where V is the node set and L is the set of (directed) links, where every link has the same unlimited transport capacity.
- $V = V_{\text{SRC}} \cup V_{\text{NET}} \cup V_{\text{DST}}$, the set of all nodes, indexed by v, comprising pure OXCs (V_{NET}), server sites V_{DST} (with $|V_{\text{DST}}| = K$), and explicitly modeled sources V_{SRC}.
- $L = L_{\text{SRC}} \cup L_{\text{NET}} \cup L_{\text{DST}}$, the set comprising all directed network links, indexed by ℓ, again split into the core network links L_{NET} interconnecting OXCs, and the modeled access links L_{SRC} from request sources and those towards the server sites L_{DST}.
- Δ_v is the number of unit demand requests[1], originating from a source node $v \in V_{\text{SRC}}$. A unit demand will be associated with a single bandwidth unit (i.e., a wavevelength) and a single server. (It is fairly straightforward to introduce a separate server demand Γ_v, to account for decoupled server and network requirements.)
- S represents the set of SRLGs, where an individual $s \in S$ is a set of links that can simultaneously fail (implying that $S \subseteq 2^L$, where 2^L is the powerset of L). Note that the empty set ($s = \emptyset$) will denote the failure free case.

We also will use the following notations:

- $\text{IN}(v)$ represents the set of v's incoming links.
- $\text{OUT}(v)$ represents the set of v's outgoing links.

Note that the server links L_{DST} will be used to count the required server capacity. Thus, they constitute a modeling trick: the link capacity of link $\ell \in L_{\text{DST}}$ will actually represent the number of server CPUs that we need at the data center site it connects to. The link capacity of network links $\ell' \in L_{\text{NET}}$ will be expressed in number of wavelengths. Our model will assume a priori unlimited network and server capacity (thus representative of a greenfield situation), yet can easily be extended to include given capacity upper bounds. Note that we also assume wavelength conversion to be possible in intermediate nodes, i.e., we will not enforce wavelength continuity constraints.

As indicated before (see above, Section II), we will use two steps: (S1) find the best server locations, and (S2) find the amount of servers at each of those locations, as well as routes for the request data traffic to follow towards those servers, from which we derive the amount of wavelengths on each of the network links. The following two sections detail each of those steps. Before discussing them, note that for step (S1) we assume the number of server locations K is given a priori, and we thus do not optimize that number (but we do discuss the impact of different K values in Section VI-B). Clearly, increasing K would allow shorter paths for demands and hence lower network capacity requirements, as well as better opportunities to spread the risk of server failures (assuming at most one failing data center, and perfect load balancing, we need in the order of $1/K$ backup capacity). Yet, having many data center locations incurs additional capital and operational expenses not incorporated in the model below.

[1]Note that our model can easily be extended to multiple traffic types, that each can have different (number of) destination sites to serve it.

IV. CHOOSING K SERVER LOCATIONS (STEP S1)

We assume that the number of server sites, K, is given a priori, but not their locations. To solve this problem of finding the K best locations, given the topology and the demand, in [27] we originally proposed a clustering heuristic[2] and an ILP approach. Below, we present and extend the ILP approach to account for both working and backup resources (as [27] did not consider protection). In terms of aforementioned network model, the current step (S1) will fix the data center locations, i.e., determine V_{DST}. The ILP model below will therefore consider only the network of OXCs (i.e., V_{NET}) as given. Hence, we will associate demand $\Delta_{v'}$ with the OXC v' connected to the source node v ($\Delta_{v'} = \Delta_v$).

Our approach is based on some simplifying assumptions: (*i*) each source site v will send all its requests to a single destination D_v, and (*ii*) shortest path routing is used. Hence, given a choice of K locations, a site v will send its requests to server site v' if the routing distance $h_{vv'}$ is the minimum over all $h_{vv''}$ values for $v'' = 1..K$. This can be formulated easily as an ILP, by introducing the following variables and parameters:

- t_v is a binary variable, equalling 1 if and only if site v is chosen as one of the K server sites, i.e., v is a potential target for anycast traffic.
- $f_{vv'}$ is a binary variable, equalling 1 if and only if source node v sends its requests to server site v'.
- $h_{vv'}$ is a given parameter, accounting for the cost (e.g., hop count) of sending a unit request's data traffic from v to v'.

The original ILP (rephrased from [27]), thus becomes:

$$\min \quad \sum_{v \in V_{\text{NET}}} \sum_{v' \in V_{\text{NET}}} \Delta_v \cdot h_{vv'} \cdot f_{vv'}, \tag{1}$$

subject to:

$$\sum_{v \in V_{\text{NET}}} t_v = K, \tag{2}$$

$$\sum_{v' \in V_{\text{NET}}} f_{vv'} = 1 \quad \forall v \in V_{\text{NET}}, \tag{3}$$

$$f_{vv'} \leq t_{v'} \quad \forall v, v' \in V_{\text{NET}}. \tag{4}$$

The objective (1) thus is to minimize the total number of primary wavelengths (assuming each unit demand Δ_v calls for a single wavelength) that would need to be foreseen, if $h_{vv'}$ stands for the length of the shortest path from v to v'. We will refer to this location choosing approach as SW (from *shortest working* path).

As indicated before, we will be dimensioning resilient grids/clouds using a path protection approach. The SW choosing approach however only accounts for working paths. Hence, we devised a SC approach (from *shortest cycle*), where we use the same ILP (1)–(4), but now set $h_{vv'}$ to the length of the shortest combination of two link-disjoint paths between nodes v and v' (e.g., using Suurballe's algorithm [44]).

[2]Basically we rephrased the well-known k-means clustering algorithm [43] as a k-medoids algorithm using shortest path lengths as distance metric rather than euclidian distance.

The SC chooser thus accounts for both a working and a backup path to the same destination. Yet, if we allow relocation, the backup path can end in a different site. Therefore, we also will consider an SRO chooser (from *shortest relocation optional*), and introduce the following variable and parameter:

- $f_{vv'v''}$ is a binary variable, equalling 1 if and only if source node v sends its requests to primary server site v' under normal, and to backup site v'' under failure conditions.
- $h_{vv'v''}$ is a given parameter, accounting for the cost of sending a unit request demand from v to v' under normal, and to v'' under failure conditions.

The SRO chooser ILP thus becomes (5)–(9), with $h_{vv'v''}$ defined as the sum of the lengths of the shortest combination of 2 disjoint paths from v to v' and v''. The latter can be easily computed using any well-known disjoint path (or shortest cycle) algorithm (e.g., [44]), extending the topology for this purpose with an additional virtual node σ connected to the primary (v) and backup (v') server sites (similarly to the approach of [23] to solve the anycast routing problem; see also Section V-A2).

$$\min \quad \sum_{v \in V_{\text{NET}}} \sum_{v' \in V_{\text{NET}}} \sum_{v'' \in V_{\text{NET}}} \Delta_v \cdot h_{vv'v''} \cdot f_{vv'v''} \tag{5}$$

subject to:

$$\sum_{v \in V_{\text{NET}}} t_v = K \tag{6}$$

$$\sum_{v' \in V_{\text{NET}}} \sum_{v'' \in V_{\text{NET}}} f_{vv'v''} = 1 \quad \forall v \in V_{\text{NET}} \tag{7}$$

$$f_{vv'v''} \leq t_{v'} \quad \forall v, v', v'' \in V_{\text{NET}} \tag{8}$$

$$f_{vv'v''} \leq t_{v''} \quad \forall v, v', v'' \in V_{\text{NET}}. \tag{9}$$

Table I summarizes the choosers we will evaluate later in Section VI-A. For the chosen data center locations $v \in V_{\text{NET}}$ (i.e., those for which $t_v = 1$), we will expand the network by adding server nodes $v' \in V_{\text{DST}}$, connected via links $\ell = (v, v') \in L_{\text{DST}}$ (as in Fig. 1) and continue with step S2.

V. DIMENSIONING THE NETWORK AND SERVERS (STEP S2)

For a chosen set V_{DST} comprising K server locations, in step S2 we determine for each request which primary and backup server sites to use, as well as via which route to connect to them, in order to minimize the total network (i.e., wavelengths) and server capacity. As indicated before, we aim to ensure that we can meet the demand for network and server capacity also under failure scenarios.

Those failure scenarios will be generically represented as SRLGs. In our case studies, we will in particular consider single failures of either bidirectional links, or servers. A bidirectional network link failure will be modeled as an SRLG comprising the two opposite directed links ($\ell, \ell' \in L_{\text{NET}}$) between two network nodes. Since we model servers as the links between network and server nodes, a server failure will be represented as a failing link $\ell \in L_{\text{DST}}$. As indicated, we are interested in providing 1:N server protection. One way to model this, is to provide $N+1$ parallel links between a single

Table I
OVERVIEW OF THE VARIOUS LOCATION CHOOSER STRATEGIES.

Chooser	Explanation	ILP model
SW	Accounts for a single shortest path to a server site.	(1)–(4), with $h_{vv'}$ = length of shortest path from v to v'
SC	Accounts for disjoint primary and backup paths to the same server site (i.e., using shortest cycle).	(1)–(4), with $h_{vv'}$ = length of shortest cycle joining v and v'
SRO	Accounts for disjoint primary and backup paths to possibly different server sites, i.e., optional relocation.	(5)–(9), with $h_{vv'v''}$ = sum of lengths of the shortest combination of two disjoint paths from v to v' and v''

pair of network and server nodes, and let only a single one of them fail. If we do not use relocation (i.e., the NR case), we however know that the same destination will be chosen, and we can simply calculate the total number of servers from the maximum number of operational servers, say x, that is required to meet the demand. Indeed, since 1:N protection implies that for N servers, 1 additional backup server will be provided to cater for the case that at most one of the N primary servers fails, we then need to install a total of $\lceil(1+1/N)\cdot x\rceil$ servers. Hence, in the NR case, we do not need to explicitly model server link failures, but can accommodate 1:N protection with an overprovisioning factor $\rho_\ell = 1+1/N$ for the capacity of links $\ell \in L_{\text{DST}}$.

We will consider two protection strategies of coping with the failures. The first considers a single backup path for each unit request, i.e., we adopt a shared path protection concept. Thus, for a given request unit the alternate path (possibly to a different destination) under any failure condition affecting the primary path is always the same. This is generally known as failure-independent (FID) restoration [42], [45], [46], which previously also has been described as state-independent restoration [47], [48]. The second protection strategy is that of failure-dependent (FD), aka state-dependent, backup routing: the alternate path (and possibly alternate destination) can be different for each individual failure scenario. Both FID and FD strategies are described in detail below.

A. Failure-independent (FID) backup path routing with relocation

1) Methodology: Given the scalability issues of a single ILP problem formulation addressing the FID case (see [31]), we use a column generation (CG) approach to find so-called configurations and the number of times to use them. A configuration $c \in C$ will be associated with a particular source-site $v \in V_{\text{SRC}}$, and will consist of a pair of working and backup paths, both originating from v and terminating in one of the server sites in V_{DST} (possibly different in case of relocation). This involves solving what are called the Restricted Master Problem (RMP) and a Pricing Problem (PP) iteratively. The next subsections will detail the constituent phases of a CG scheme, that can be summarized as follows:

1) Find a set of initial configurations and assign it to C;
2) Solve the linear program (LP) relaxation of the RMP, minimizing required network and server resources;
3) Solve the PP to try and find a new configuration c for a source node $v \in V_{\text{SRC}}$, that could lead to a cost reduction of the RMP objective function (i.e., that has a negative reduced cost). If successful, add c to the set of configurations C.
4) Repeat steps 2–3 until no new configurations (with negative reduced cost) can be found.
5) Solve the final resulting RMP as ILP, to find an integer solution, determining the number of times z_c to use each configuration $c \in C$.

In each iteration of Step 3, source nodes V_{SRC} are considered in a round-robin fashion. Step 2 is performed every time a new configuration was added in Step 3. (For the CG methodology, see also [31].) Note that the gap between the resulting objective function value of the ILP and LP solutions of the RMP is very small (for the case study results, the relative ILP vs. LP gap on average amounted to below 0.50%, with an observed maximum of at most a few %).

2) Finding initial configurations: To find initial configurations, we use a heuristic inspired by [49], and detailed in Algorithm 1. We introduce the set of candidate server locations V_{LOC}. For the case without relocation, $V_{\text{LOC}} = V_{\text{DST}}$. Yet, for the case with relocation, we add a (virtual) node σ to the node set V of the graph, and introduce additional links $(v,\sigma), \forall v \in V_{\text{DST}}$ and set $V_{\text{LOC}} = \{\sigma\}$. Then, for each source site $v \in V_{\text{SRC}}$, we find initial configurations by finding the shortest pair of disjoint paths to each candidate server site in V_{LOC}. For this, we use the algorithm originally developed by Suurballe and Tarjan [44]. In a subsequent step, we find additional configurations by trying to find alternate backup paths that share links with other configuration's backup.

3) Restricted Master Problem (RMP): The parameters and variables of our column generation ILP are:

c	A configuration, defined for a given source node $v \in V_{\text{SRC}}$.
C_v	The set of configurations associated with a source node $v \in V_{\text{SRC}}$.
C	$= \bigcup_{v \in V_{\text{SRC}}} C_v$
S	The set of SRLGs, indexed by s.
z_c	Integer decision variable, counting the number of times configuration c is used.
$p^W_{c\ell}$	Binary parameter, equaling 1 if and only if link ℓ is used in the *working* path in configuration c.
$p^B_{c\ell}$	Binary parameter, equaling 1 if and only if link ℓ is used in the *backup* path in configuration c.
w_ℓ	Auxiliary integer variable, counting the number of wavelengths used on link ℓ.

Algorithm 1 Finding an initial set of configurations C

// Find shortest paths from each source to each possible destination
$C_0 \leftarrow \emptyset$
for all $v \in V_{\text{SRC}}$, and $s \in V_{\text{LOC}}$ **do**
  $c \leftarrow$ DisjointPathPair(v, s)
  $c' \leftarrow c$ with working and backup swapped
  Add c and c' to C_0.
end for
// Find new configurations that share (part of) backup path with others
$C \leftarrow C_0$
for $c \in C_0$ **do**
  Construct a copy G' of the graph G.
  Remove links of working path of c from G'.
  for all $c'' \in C$ with $c'' \neq c$ **do**
    if working paths of c'' and c are disjoint **then**
      In G', set weights of backup path links of c'' to 0.
    end if
  end for
  Construct a new configuration c'.
  Set working path of c' to that of c.
  Set backup path of c' to shortest path in G' between source and backup destination of c.
  if backup path c' shorter than that of c **then**
    Add c' to C.
  end if
end for

The master problem will determine which configurations to use, using decision variables z_c. The objective function is given in (10): we minimize the amount of network resources (wavelengths w_ℓ for $\ell \in L_{\text{NET}}$) and the amount of server resources, which is modeled as the capacity of the links towards server nodes (w_ℓ for $\ell \in L_{\text{DST}}$). We introduce a factor α that expresses the cost ratio of a single unit of server capacity (i.e., a single server CPU), compared to the cost of a single unit of network bandwidth (i.e., a wavelength) on a single link. (Recall that we assume 1 unit request asks for 1 network capacity unit, and 1 server unit.)

$$\min \quad \left(\sum_{\ell \in L_{\text{NET}}} w_\ell + \alpha \cdot \sum_{\ell \in L_{\text{DST}}} w_\ell \right) \tag{10}$$

Note that our formulation implies that the number of server resources required for a unit demand of jobs is assumed to be linearly proportional to the bandwidth (i.e., wavelengths) they need. Yet, by introducing another set of parameters stating the amount of server resources required for jobs originating at source site v (e.g., define Γ_v), it is fairly straightforward to rewrite the model (i.e., add a factor Γ_v/Δ_v to the z_c in equations (12)-(13) below). For ease of notation, in the following we stick to the assumption that each unit request needs a single wavelength and single server CPU.

In the case with no relocation (NR), we will calculate the number of servers (whose amount is expressed as w_ℓ for $\ell \in L_{\text{DST}}$), by introducing a factor to account for 1:N server protection:

ρ_ℓ An overprovisioning factor that we will use in the (NR) case, when we use 1:N server protection (see before; for $\ell \in L_{\text{DST}}$ it will be $1 + 1/N$ and 1 for the network links $\ell \in L_{\text{NET}}$). For any other scenario (no server protection, or the relocation case (RO)), it will be 1 for all links.

The first set of constraints (11) are obviously to meet the requested demands. Next, in constraints (12) we enforce the number of wavelengths to be sufficient to carry all selected configurations under failure-free conditions. For each considered failure case, represented as an SRLG $s \in S$, we have constraints (13), of which the right hand side comprises as first term a summation covering all unaffected configurations and secondly the affected ones. Therefore, we define two auxiliary parameters (whose values in this RMP are constants, depending on the configuration at hand; they will be variables in the PP):

$\pi^W_{c\ell s}$ Binary, equaling 1 if and only if the working path of configuration c crosses link ℓ, which remains unaffected by failure of SRLG s (thus, $\ell \in$ *workingPath*(c) and *workingPath*$(c) \cap s = \emptyset$).

$\pi^B_{c\ell s}$ Binary, equaling 1 if and only if link ℓ is part of the backup path of configuration c, whose working path is affected by SRLG s (that is, $\ell \in$ *backupPath*(c) and *workingPath*$(c) \cap s \neq \emptyset$).

(Note that according to (13), we only need to define $\pi^W_{c\ell s}$ and $\pi^B_{c\ell s}$ for $\ell \notin s$.) Observe that this model does not limit the maximal capacity of either links or server nodes (i.e., we assume an uncapacitated network), yet capacity constraints can be trivially imposed through upper bounds for w_ℓ.

$$\sum_{c \in C_v} z_c \geq \Delta_v \quad \forall v \in V_{\text{SRC}} \tag{11}$$

$$w_\ell \geq \rho_\ell \cdot \sum_{c \in C} p^W_{c\ell} \cdot z_c \quad \forall \ell \in L \tag{12}$$

$$w_\ell \geq \rho_\ell \cdot \left(\sum_{c \in C} \pi^W_{c\ell s} \cdot z_c + \sum_{c \in C} \pi^B_{c\ell s} \cdot z_c \right) \quad \forall s \in S, \forall \ell \notin s. \tag{13}$$

4) Pricing Problem (PP): Solving the master problem with the set of all possible configurations is not scalable. Yet, in order to answer the dimensioning question, it suffices that the master problem includes the possible configurations associated with a non-zero basis variable to reach the overall optimum (of the linear relaxation). Thus, in the column generation approach, we start from an initial limited set of promising configurations and solve the master only for a subset of all possible configurations: this is the Restricted Master Problem (RMP). Based on the solution of the RMP, we subsequently add new configurations c by solving the pricing problem (PP): it finds such c that is able to reduce the RMP objective value. In our case, a PP is associated with a given source node $v_{\text{SRC}} \in V_{\text{SRC}}$. The PP uses the values (as found by the RMP, relaxed as linear program) of dual variables corresponding to constraints of the RMP:

u_v^1 value of RMP dual variable corresponding to (11).
u_ℓ^2 value of RMP dual variable corresponding to (12).
$u_{\ell s}^3$ value of RMP dual variable corresponding to (13).

(Note that u_v^1 will be positive, while u_ℓ^2 and $u_{\ell s}^3$ will be negative, given the different position of z_c with respect to the inequality sign in (11) versus (12) and (13).)

The objective function (14) of the PP will be to minimize the reduced cost. (The first explicit 0 term is the coefficient of z_c in the RMP objective.) The PP's decision variables p and its auxiliary variables π have the same definitions as before, but we drop the c index.

$$\min \quad \overline{\text{COST}}(p,\pi) = 0 - u_{v_{\text{SRC}}}^1 + \sum_{\ell \in L} u_\ell^2 \cdot \rho_\ell \cdot p_\ell^W + \sum_{s \in S} \sum_{\ell \notin s} u_{\ell s}^3 \cdot \rho_\ell \cdot \left(\pi_{\ell s}^W + \pi_{\ell s}^B\right). \tag{14}$$

The first set of equations (15) represent the flow conservation equations, expressing that the net flow going into a node should be either -1 (for the source node), $+1$ (for a destination node) or 0 otherwise.

$$\sum_{\ell \in \text{IN}(v)} p_\ell^\star - \sum_{\ell \in \text{OUT}(v)} p_\ell^\star = \begin{cases} -1 & \text{if } v = v_{\text{SRC}} \\ \sum\limits_{\ell \in \text{IN}(v)} p_\ell^\star & \text{if } v \in V_{\text{DST}} \\ 0 & \text{otherwise} \end{cases} \quad \forall v \in V \text{ and } \star = W, B. \tag{15}$$

Next, constraints (16) assure that there will be no loops, and exactly 1 working and backup path will be constructed. Additionally (17) enforces that a single working and backup destination will be chosen. Finally, working and backup paths obviously need to be disjoint (18) with respect to an SRLG $s \in S$ (note that we protect against single SRLG failures only).

$$\sum_{\ell \in \text{IN}(v)} p_\ell^\star \leq 1, \sum_{\ell \in \text{OUT}(v)} p_\ell^\star \leq 1, \quad \forall v \in V, \star = W, B \tag{16}$$

$$\sum_{\ell \in L_{\text{DST}}} p_\ell^\star = 1, \quad \text{for } \star = W, B \tag{17}$$

$$p_\ell^W + p_{\ell'}^B \leq 1, \quad \forall s \in S, \forall \ell, \ell' \in s. \tag{18}$$

It remains to define constraints so that the definitions of π^W and π^B apply as before. For this, we define additional auxiliary variables a_s^W, each associated with an SRLG $s \in S$:

a_s^W Binary variable, equaling 1 if any of the links $\ell' \in s$ is used as working link (i.e., if $\exists \ell' \in s : p_{\ell'}^W = 1$), hence if the chosen working path is *affected* by the SRLG.

Constraints (19) enforce that $\pi_{\ell s}^W \equiv p_\ell^W \wedge \neg a_s^W$, and constraints (20) ensure the logical relation $\pi_{\ell s}^B \equiv p_\ell^B \wedge a_s^W$, to express the definitions of $\pi^\cdot$ given before. The definition of a_s^W amounts to $a_s^W \equiv \bigvee\limits_{\ell' \in s} p_{\ell'}^W$, or thus (21).

$$\left.\begin{aligned} \pi_{\ell s}^W &\geq p_\ell^W - a_s^W \\ \pi_{\ell s}^W &\leq p_\ell^W \\ \pi_{\ell s}^W &\leq 1 - a_s^W \end{aligned}\right\} \forall s \in S, \forall \ell \notin s \tag{19}$$

$$\left.\begin{aligned} \pi_{\ell s}^B &\geq p_\ell^B + a_s^W - 1 \\ \pi_{\ell s}^B &\leq p_\ell^B \\ \pi_{\ell s}^B &\leq a_s^W \end{aligned}\right\} \forall s \in S, \forall \ell \notin s \tag{20}$$

$$\left.\begin{aligned} M \cdot a_s^W &\geq \sum_{\ell' \in s} p_{\ell'}^W \\ a_s^W &\leq \sum_{\ell' \in s} p_{\ell'}^W \end{aligned}\right\} \forall s \in S, \text{with } M = |s|. \tag{21}$$

The above constraints all apply regardless whether we consider relocation or not. Yet, if we do not want to relocate, we need to enforce one additional constraint (22), stating that working and backup destination need to be the same:

$$\sum_{\ell \in \text{IN}(v)} p_\ell^W = \sum_{\ell \in \text{IN}(v)} p_\ell^B, \quad \forall v \in V_{\text{DST}}. \tag{22}$$

Note that the summations are there to account for parallel links between an OXC and a server site (links L_{DST} in Fig. 1). Such parallel links can be used to model protection against server failures (see above).

Alternatively, if we want to enforce relocation, i.e., we do not allow working and backup server site to be the same, then we can include the following constraint:

$$\sum_{\ell \in \text{IN}(v)} p_\ell^W + \sum_{\ell \in \text{IN}(v)} p_\ell^B \leq 1, \quad \forall v \in V_{\text{DST}}. \tag{23}$$

One final remark: the above model can easily accommodate the analysis of cases where not all traffic is anycast, i.e., a mixture of unicast and anycast (as in [22]), by enforcing $p_\ell^\star = 1$ for the particular unicast destination's link $\ell \in L_{\text{DST}}$ (and fixing $p_{\ell'}^\star = 0$ for the other server links $\ell' \in L_{\text{DST}} \setminus \ell$). Our case studies discussed next, however, will focus on anycast traffic only.

B. Failure-dependent (FD) backup path routing with relocation

To study the FD case, we make use of a reasonably straightforward ILP as sketched below. Since we did not observe scalability issues in studying fairly large problems (as exemplified in Section VI), we did not resort to column generation as in the FID case.

In addition to the constants ρ_ℓ as defined before in Section V-B, we define the following ILP variables:

$p_{v\ell s}$ The number of unit demands originating from $v \in V_{\text{SRC}}$ that are crossing link $\ell \in L$ under failure of $s \in S$ ($s = \emptyset$ represents the failure free case).
w_ℓ The capacity to provide on link $\ell \in L$. For network links $\ell \in L_{\text{NET}}$, it amounts to the number of wavelengths. For data center links $\ell \in L_{\text{DST}}$, it represents the number of servers to install at that site.

The objective is given in (24): we minimize the amount of network resources (wavelengths w_ℓ) and the amount of server

resources, which, in our model, is conveniently represented as the capacity on server links. We introduce a factor α that expresses the cost ratio of the server capacity corresponding to a workload filling a single wavelength with data, compared to the cost of a single wavelength on a single link (as before in the FID case).

$$\min \quad \left(\sum_{\ell \in L_{\text{NET}}} w_\ell + \alpha \cdot \sum_{\ell \in L_{\text{DST}}} w_\ell \right). \tag{24}$$

The first set of constraints constitute the demand constraints which dictate where traffic originates (25)–(26) and ends (27), as well as the traditional flow conservation constraint in intermediate network nodes (28). These constraints have to hold $\forall v \in V_{\text{SRC}}, \forall s \in S$.

$$p_{v\ell' s} = \Delta_v \quad \text{where } \{\ell'\} = \text{OUT}(v) \tag{25}$$

$$p_{v\ell s} = 0 \quad \forall \ell \in L_{\text{SRC}} \setminus \text{OUT}(v) \tag{26}$$

$$\sum_{\ell \in L_{\text{DST}}} p_{v\ell s} = \Delta_v \tag{27}$$

$$\sum_{\ell \in \text{IN}(v')} p_{v\ell s} = \sum_{\ell \in \text{OUT}(v')} p_{v\ell s} \quad \forall v' \in V_{\text{NET}}. \tag{28}$$

The next constraint (29) expresses that traffic cannot cross affected links for each respective failure scenario:

$$p_{v\ell s} = 0 \quad \forall v \in V_{\text{SRC}}, \forall s \in S, \forall \ell \in s. \tag{29}$$

The final constraint amounts to counting the capacity required for each link (or data center), where ρ_ℓ is defined as in the FID case:

$$w_\ell \geq \rho_\ell \cdot \sum_{v \in V_{\text{SRC}}} p_{v\ell s} \quad \forall v \in V_{\text{SRC}}, \forall s \in S, \forall \ell \in L_{\text{NET}} \cup L_{\text{DST}}. \tag{30}$$

The equations (24)–(30) cover the case of failure-dependent backup routing, exploiting relocation (if it is beneficial). To obtain resource dimensions for the non-relocation case, we need to enforce that for all failure cases the same data center is chosen as in the failure free case. This implies $\forall v \in V_{\text{SRC}}, \forall s \in S \setminus \{s'\}$, $s' = \emptyset$, $\forall v' \in V_{\text{DST}}$:

$$\sum_{\ell \in \text{IN}(v')} p_{v\ell s} = \sum_{\ell \in \text{IN}(v')} p_{v\ell s'}. \tag{31}$$

(As before, the summation is only there for the case where we have parallel server links in our network model.)

As a final remark, note that the model defined by (24)–(30) amounts to optional relocation, i.e., an alternate destination site will only be chosen if that leads to lower overall costs as per the objective. In case we want to enforce relocation under failure conditions (as in [29]), we can impose that if v' is used as destination under working conditions, then it cannot serve under failure scenarios. This amounts to the following constraint[3], $\forall v \in V_{\text{SRC}}$, $\forall v' \in V_{\text{DST}}$, $s' = \emptyset$:

$$\left(\sum_{\ell \in \text{IN}(v')} p_{v\ell s'} \neq 0 \right) \Rightarrow \left(\sum_{\ell \in \text{IN}(v')} \sum_{s \in S \setminus \{s'\}} p_{v\ell s} = 0 \right). \tag{32}$$

[3]Note that a constraint of the form $(a \neq 0) \Rightarrow (b = 0)$ is logically equivalent to $(a = 0) \vee (b = 0)$, which can be easily modeled as linear constraints. Let A, B be binary variables, so that $A \equiv (a = 0)$ and $B \equiv (b = 0)$. Then the $\vee$ constraint becomes $A + B \geq 1$.

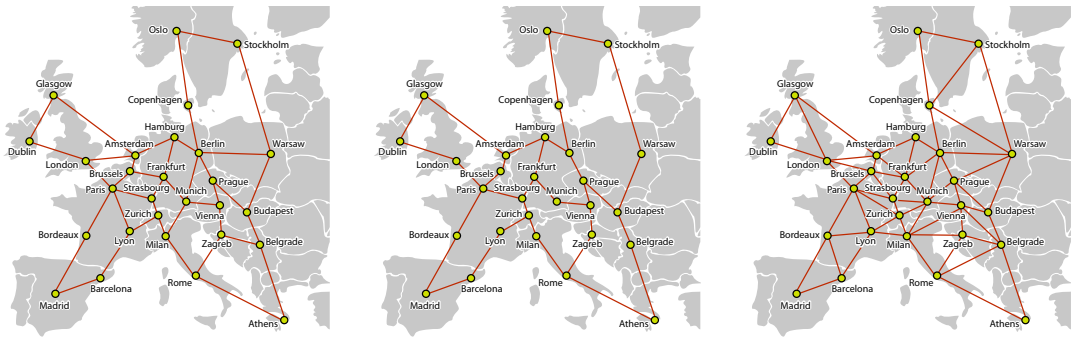
(Again, summation over $\text{IN}(v')$ is necessary in the case of parallel server links in the network model.)

VI. Case studies

We evaluated the above methodology on European network topologies taken from [50], as illustrated in Fig. 2: (*a*) *EU-basic* comprising 28 nodes and 41 bidirectional links (avg. nodal degree of 2.93), (*b*) *EU-sparse* with 28 nodes and 34 bidirectional links (avg. node degree of 2.42), and (*c*) *EU-dense* with 28 nodes and 60 bidirectional links (avg. node degree of 4.29). We will consider demand instances comprising varying number unit demands of jobs, where each unit demand will be assumed to require a single full wavelength and one server. (As indicated before, the (CG-)ILP models can be easily adapted to uncorrelated server and wavelength requirements.) The total number of unit demands (i.e., $\sum_{v \in V} \Delta_v$) will vary between 10 and 350, to demonstrate the scalability of our approach. For each given number of unit demands, we have randomly generated 10 instances, drawing the sources v uniformly from the set V of 28 network nodes. Hence, each data point for a given number of unit demands in the graphs that will be presented will constitute the average over the respective 10 random instances.

We will assess the benefit of exploiting relocation when protecting against failures, in two scenarios: (*i*) single link failures only (1L), or (*ii*) single failures of either a link or a server (1LS). The modeling approach is the following, as summarized formally in Table II:

- *NR*: No relocation, i.e., primary and backup server sites have to be the same:
  - *1L*: For the single network link failure case, we will consider that a single bidirectional link will completely fail. In our network model, this corresponds to an SRLG comprising the two opposite directed links.
  - *1LSN*: The single failure a bidirectional network link will be modeled similarly as for 1L. In addition, we need to cater for single server failures. Yet, we will not model them as additional SRLGs, but we will rather (as explained before) account for backup capacity at a particular data center site through an overprovisioning factor ρ_ℓ for the single link $\ell \in L_{\text{DST}}$ connecting it to a network node (recall that that link's capacity represents the number of servers to install).
- *RO*: Relocation is optional, thus primary and backup server sites can (but not necessarily will) differ, if this is beneficial in terms of cost (i.e., leads to lower total network and server resource dimensions).
  - *1L*: Single link failures are modeled as for the NR case.
  - *1LS*: Again, single link failures will be modeled through SRLGs. To model server failures, we will use $1+N$ parallel server links connecting each data center to its corresponding network node, of which only at most one will fail (each modeled as a singleton SRLG). The total capacity over the $1 + N$ links together will reflect the required number of servers to achieve $1{:}N$ protection.



(a) EU-basic (b) EU-sparse (c) EU-dense

Fig. 2. EU network topologies used for the case studies, which all comprise 28 nodes: (a) Basic topology, with 41 bidirectional links (the *base* topology from [50]), (b) Sparse topology with only 34 bidirectional links (the *ring* topology from [50]), and (c) Dense topology with 60 bidirectional links (the *triangular* topology from [50]).

Table II
MODEL SETTINGS FOR THE CONSIDERED FAILURE SCENARIOS.

Case	ILP model settings
1L	$S = \{\{\ell, \ell'\} : \ell, \ell' \in L_{\text{NET}},$ ℓ and ℓ' are each other's reverse$\} \triangleq S_{1L}$ $\rho_\ell = 1, \forall \ell \in L$ $\lvert L_{\text{DST}}\rvert = K$ (single server link per data center site)
1LS	$S = S_{1L} \cup \{\{\ell\} : \ell \in L_{\text{DST}}\}$ $\rho_\ell = 1, \forall \ell \in L$ $\lvert L_{\text{DST}}\rvert = (1+N) \cdot K$ (parallel server links)
1LSN	$S = S_{1L}$ $\rho_\ell = 1 + 1/N$ if $\ell \in L_{\text{DST}}$, else 1 $\lvert L_{\text{DST}}\rvert = K$ (single server link per data center site)

Note that in the model for the latter 1LS case, we indeed achieve truly *optional* relocation: we allow the choice between adding extra backup server resources and relocating. In this RO case, a failure of a server is modeled as a single failing $\ell \in L_{\text{DST}}$ and can be resolved by either (*i*) adding extra server capacity locally (modeled as extra capacity on a parallel link $\ell' \in L_{\text{DST}}$ to the same $v \in V_{\text{DST}}$ destination as in the failure free case), or by (*ii*) relocating to another server site $v' \in V_{\text{DST}}$ while accounting also for possibly extra network capacity on the path towards it.

A. Finding the best K server locations

Our first set of studies aimed to evaluate the most suitable chooser to use. For this, we considered case studies on the EU-basic and EU-sparse topologies. We used the various location chooser strategies formally presented in Section IV for step S1, and subsequently used either one of the (CG-)ILP approaches for failure-dependent (FD) or failure-independent (FID) backup routing for step S2, as detailed in Section V. In particular, the chooser strategies we considered are:

- *SW, SC, SRO*: See Table I, using the randomly generated demands Δ_v (ranging between 10 and 350 unit requests).
- *Random*: This is a benchmark case, where we randomly select K server sites amongst all network nodes V.

To compare the various chooser strategies, we will obviously look at the total cost, in terms of server and network resources that are required for the resulting dimensioned grid/cloud. We expect that the difference will mainly pertain to the network dimensions, i.e., wavelengths, since the location of servers will most likely not have a significant influence on the number of servers that will be required to match the demand. Looking at the rationale of our various choosers (see Table I), we expect that (*i*) SC will lead to lowest resource requirements when we do not relocate (i.e., the NR cases); (*ii*) SRO will be the best choice when considering relocation, at least when considering server failures (1LS, RO), but likely also in case of just link failures (1L, RO).

The comparison of the various choosers is summarized in Fig. 3, Fig. 4 and Table III. We note that for NR our intuitive expectation is met and SC indeed leads to overall minimal cost (i.e., comparing the minimum cost over the various location choices). Similarly, for RO with 1LS we find SRO to be best. However, for the RO with 1L case we find that SC more often than SRO leads to lower cost. The reason could be that to protect against single link failures only (1L), relocation does not lead to a cost reduction that is as substantial as in the 1LS case (see further, Section VI-B). Yet, note that the difference among all intelligent choosers (i.e., SW, SC, SRO) is limited: e.g., in the EU-basic topology, the total costs they achieve differ only by a few % (for EU-basic, FID, maximally 3% in the NR case; in the RO case less than 1.6% for 1L, and less than 1% for 1LS). Since these differences fall within the 95% confidence intervals, they are not significant. Thus, what this comparison between these choosers seems to learn is that any "good enough" choice of servers achieve almost the same overall cost (in terms of network and server capacity) — which however is significantly lower than a purely random, non-intelligent choice. In the remainder of this paper, we will stick to the SC chooser.

B. Exploiting relocation to ensure resilience

In the case of protecting against *single link failures (1L)*, we note the clear advantage of exploiting relocation on network capacity (see Fig. 5): for $K = 3$ we observe a reduction of the required number of wavelengths in the order of around 8.9% (average over the larger demands $\Delta_v \in [100, 350]$ cases). The

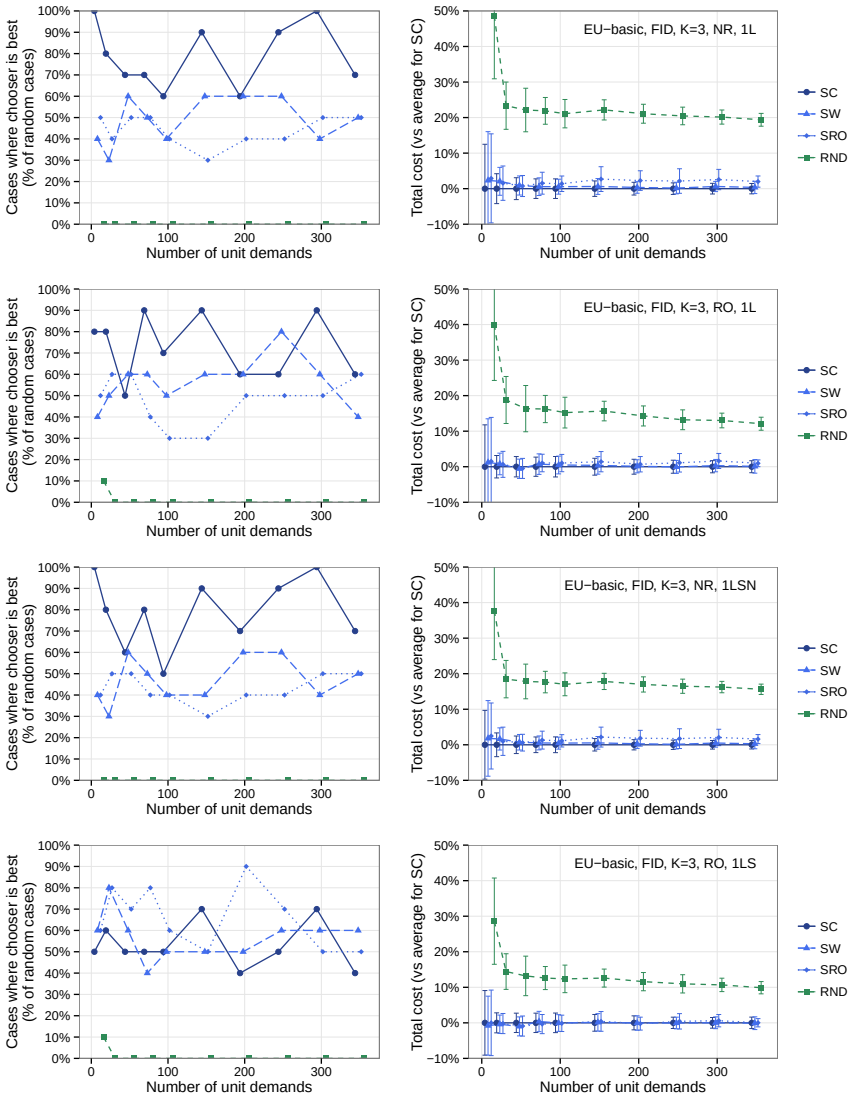


Fig. 3. The relative difference in total cost between the intelligent choosers SW, SC, SRO is limited, but that total cost is substantially higher for the random data center location chooser baseline (RND). The graphs show (i) for which fraction of the 10 random cases per demand size each chooser performed best, (ii) the relative total cost difference compared that for SC, i.e., $cost(x)/average\_cost(SC) - 1$, where error bars indicate the 95% confidence interval. (RO: optional relocation; NR: no relocation; 1L: single link failure protection; 1LS: protection against single failures of either a link or a server; 1LSN: single link failure and *1:N* server protection.)

Table III
COMPARISON OF THE VARIOUS CHOOSERS, IN TERMS OF FOR HOW MANY OF THE 100 RANDOMLY GENERATED DEMAND INSTANCES (I.E., 10 INSTANCES FOR EACH OF THE DEMAND SIZES IN [10,300]) THEY LEAD TO THE LOWEST OVERALL COST.

	Chooser	1L, NR	1L, RO	1LSN, NR	1LS, RO
EU-basic, FID	SC	**79%**	**64%**	**80%**	47%
	SW	48%	53%	45%	55%
	SRO	43%	49%	42%	**63%**
EU-basic, FD	SC	**78%**	**61%**	**76%**	47%
	SW	52%	52%	52%	45%
	SRO	45%	50%	43%	**70%**
EU-sparse, FID	SC	46%	59%	46%	52%
	SW	53%	31%	54%	41%
	SRO	31%	41%	31%	47%

RO: optional relocation; NR: no relocation; 1L: single link failure protection; 1LS: protection against single failures of either a link or a server; 1LSN: single link failure and 1:N server protection.

price paid is a modest increase in the number of required servers, still resulting in a net cost benefit.

When we want to protect against *both single link and server failures (1LS)* and relocate, clearly we need more resources than for the 1L case, and especially extra servers[4]. That backup server capacity can, however, be quite optimally shared among all failure scenarios, so that (for the assumed uniform traffic) we need about $1/K$ extra server capacity (versus $1/N$ for 1:N server protection without relocation; recall that we used $N = 1$ in the results presented). To exploit that shared server capacity maximally, we need some extra wavelengths to reroute towards an alternate location. Thus, the RO, 1LS case needs more

[4]Observe that the total amount of server resources in the 1LS case cannot be trivially calculated exactly, since we allow sharing of that backup server capacity for protection against different failure cases: total backup server capacity may depend on the chosen locations, and certainly their number.

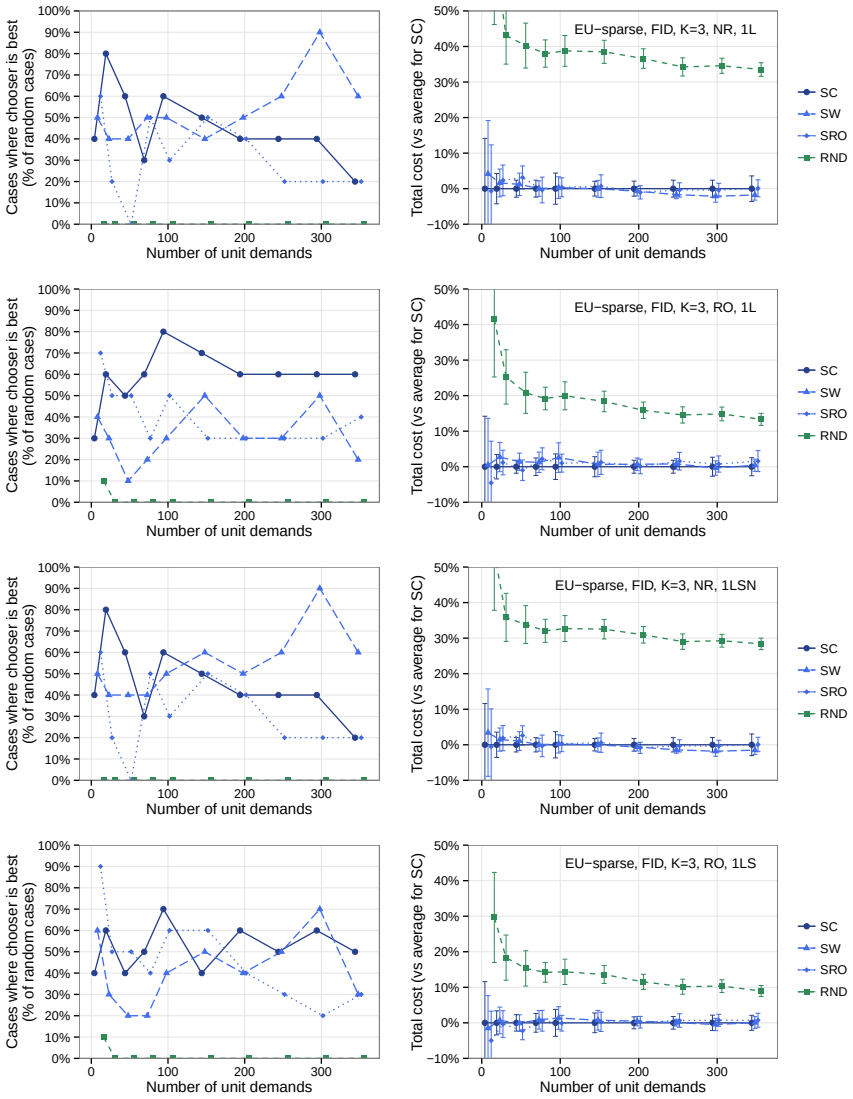


Fig. 4. For the EU-sparse topology the relative difference in total cost between the intelligent choosers SW, SC, SRO again is limited. The relative cost difference in the right hand side graphs is defined as $cost(x)/average\_cost(SC) - 1$, where error bars indicate the 95% confidence interval. (RO: optional relocation; NR: no relocation; 1L: single link failure protection; 1LS: protection against single failures of either a link or a server; 1LSN: single link failure and *1:N* server protection.)

network capacity than the RO, 1L case. Yet, we note that the overall total cost (accounting for both server and wavelength capacity) increase of RO, 1LS compared to NR, 1L is quite limited (and clearly much smaller than the NR, 1LSN case).

Regarding the *influence of the number of server sites* K, we note in Table IV that for increasing K, (*i*) the relative advantage of exploiting relocation in terms of reduced network capacity becomes more significant, while (*ii*) the price paid in terms of increased server capacity diminishes for the 1LS case. This is intuitively clear: when we increase the number of server locations, a backup path to an alternate destination becomes shorter, thus the network capacity drops further. In the 1LS case, it is beneficial to relocate to protect against server failures, and we can reuse its associated extra network capacity to recover also from link failures. (As indicated above, the additional server capacity can be limited to around $1/K$ by exploiting relocation.) For sufficiently high number of server locations, we thus observe that for large demand instances exploiting relocation even allows to protect against both single link and single server failures (1LS, RO) at a lower cost than merely covering single link failures without relocating (1L, NR) — at least in the considered case of uniform traffic.

In Fig. 6, we show the *influence of topology* on relocation benefits. As in our earlier work [31] (where we only considered network capacity), we find that relocation is especially beneficial in sparse topologies. Intuitively, it is clear that in a sparser network it is more likely that a backup path (disjoint from the working path) towards a different destination is shorter than one to the original end point (e.g., think of a simple ring network). Hence, we expect more network savings (i.e., lower total number of wavelengths) can be reached in such a sparse topology by exploiting relocation. Our results confirm this. If the network is very dense (recall that EU-dense has an average

Table IV
THE PROS AND CONS OF EXPLOITING RELOCATION: RO INCURS A SIGNIFICANT REDUCTION IN TOTAL WAVELENGTH CAPACITY COMPARED TO NR, WHILE A PRICE IS PAID IN TERMS OF INCREASED SERVER CAPACITY. RESULTS LISTED ARE RELATIVE DIFFERENCES COMPARED TO THE 1L, NR, AVERAGED OVER THE DEMAND CASES $\Delta_v \in [100, 350]$, FOR EU-BASIC, FID AND SERVER COST FACTOR $\alpha = 1$.

Case	K	Total wavelengths	Total servers	Total cost
1L, RO	3	-8.9%	$+7.5\%$	-5.0%
	5	-14.3%	$+6.1\%$	-8.6%
	7	-18.3%	$+6.4\%$	-10.5%
1LS, RO	3	-3.9%	$+29.9\%$	$+4.3\%$
	5	-8.9%	$+20.5\%$	-0.5%
	7	-11.8%	$+14.3\%$	-3.2%

node degree of 4.29), the net advantage of RO disappears if we only protect against single network link failures (1L). Nevertheless, if we need to protect against both server and link failures (1LS), relocation still may offer an advantage. However, this advantage stems quasi uniquely from reduced server resources: the difference there boils down to an increase of the total number of servers with a factor $1 + 1/K$ for RO, versus a factor $1 + 1/N$ for NR with $1{:}N$ protection (recall that plotted results assume $N = 1$).

C. Providing failure-dependent backup path routing

Coming now to the difference between failure dependent (FD) and failure independent (FID) backup path routing, we first of all note that the discussion above (on the advantages of exploiting relocation, and the impact of the number of server sites K therein) continues to hold qualitatively. The main interest of our current discussion pertains to the possible advantage of adopting backup paths that may be adapted to the failure at hand. Intuitively, we do expect possibly lower resource requirements (especially in terms of wavelengths) of such a FD approach compared to FID. Yet, earlier work on simple unicast routing problems (thus without the opportunity to exploit relocation and hence potentially increasing capacity sharing) reported limited advantages in terms of network capacity [42], [48].

Our results presented in Fig. 7, comparing the respective cases in terms of exploiting relocation and server/link protection scenarios, confirm the expectation that FID never outperforms FD. Also, for a small number of server sites ($K = 3$) we note that the deviations, i.e., *cost*(FD)/*cost*(FID) $- 1$, are quite limited: below 1% for the NR cases, and mostly below 2% for the RO cases (in line with known results for unicast traffic). However, for larger K, the advantage of adopting FD becomes more substantial when exploiting relocation: e.g., for $K = 7$ network capacity reduces with around 6% when protecting against both link and server failures (RO, 1LS).

While FD thus offers advantages in terms of capacity savings, we remark that it implies higher operational complexity: more state to maintain (i.e., multiple pre-computed routes to be stored as routing state), and conditional switching to one of the possibly many backup paths based on proper identification of the observed failure (versus unconditional switching to the single backup for any failure affecting the primary for FID).

Table V
RUNNING TIMES ON THE EU-BASIC TOPOLOGY, USING THE SC CHOOSER: MINIMUM, AVERAGE AND MAXIMUM OVER ALL 100 EXPERIMENTS (I.E., 10 RUNS FOR EACH OF THE 10 DEMAND INSTANCES) PER SCENARIO. TIMES ARE FORMATTED AS *days*:*hours*:*min*:*s*.

		NR, 1L	NR, 1LSN	RO, 1L	RO, 1LS
FID	min	6:51:51	10:46:54	9:30:07	7:26:12
	avg	22:41:57	1:01:38:07	2:01:31:41	21:59:53
	max	1:20:11:21	2:10:34:43	2:23:31:38	2:00:45:16
FD	min	4	3	3	5
	avg	5:47	6:13	2:32	3:05
	max	1:55:24	48:02	43:24	1:13:11

D. Runtime comparison

As stated previously, we found that the FD problem was solvable in its single ILP form, as opposed to the FID case where we had to resort to column generation. In Table V, this higher complexity of FID is also demonstrated by the running times we recorded for our case studies reported above. Our implementation in Java, using IBM ILOG CPLEX 12.5, was run on cluster infrastructure composed of IBM HS 21 XM blades, where each such blade has 8 cores (dual-socket quad-core Intel Xeon L5420 at 2.5GHz, with 6MB L2 cache per quad-core chip) and 16GB RAM. We observe that solving the column generation FID case on average requires in the order of one full day, whereas the FD case can be solved on average in less than 10 minutes. Intuitively, this can be explained by the greater degree of flexibility in choosing routes in the failure-dependent (FD) case: in each of the failure scenarios under consideration, we can adapt the backup routes more or less independently of other failure cases, and all we aim to optimize is the maximal capacity needed over all these failure scenarios. This indeed is less complex than finding the single(!) backup route configuration that should be used under any failure, which the failure-independent (FID) case solves.

VII. CONCLUSION

In grid/cloud scenarios, users are typically not concerned with the exact location their applications end up being run. This leads to the additional complexity in network dimensioning (as in online routing/scheduling of the requests) of choosing an appropriate location for each demand, cf. the anycast routing principle. Yet, it also offers optimization opportunities: upon failures we can choose to use different data center locations to serve the corresponding requests. Thus, we can exploit this relocation to minimize the amount of network and server resources required to fulfill a given demand. In this paper, we quantitatively assess the net benefit that relocation may bring in an optical grid/cloud scenario, in terms of total cost comprising both servers and network capacity (i.e., wavelengths). To this end, we developed ILP-based solutions to decide, in a two-step approach, on (S1) the K best locations to install servers (i.e., the data center locations), and (S2) the required network and server capacity, as well as routing of the requests towards the data centers. We considered both (*i*) failure-independent (FID) backup routing, where each working path is protected by a single backup path to cover for

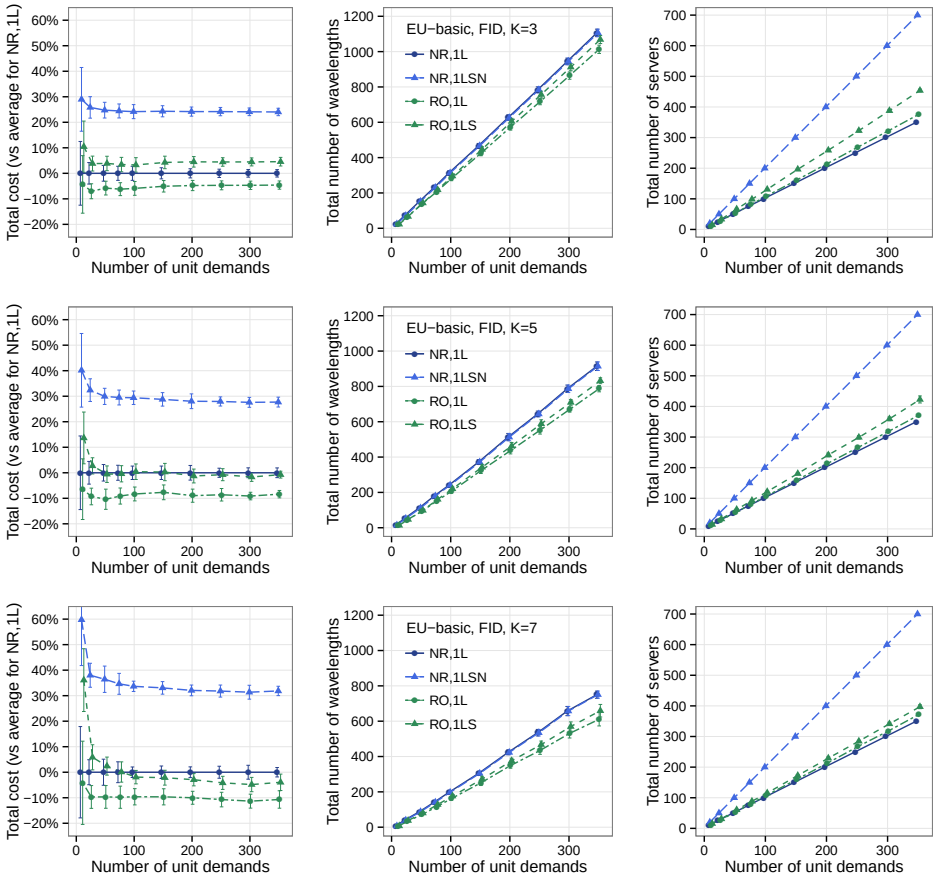


Fig. 5. When the number of server sites K increases, the advantage of exploiting relocation (RO) becomes more pronounced. The relative decrease in network capacity compared to no relocation (NR) rises, and the penalty of additional server capacity diminishes. Eventually ($K = 7$), we can provide protection by relocation against single link or server failures (RO, 1LS)) at lower overall cost than single link failure protection without relocating (NR, 1L). Error bars indicate the 95% confidence intervals.

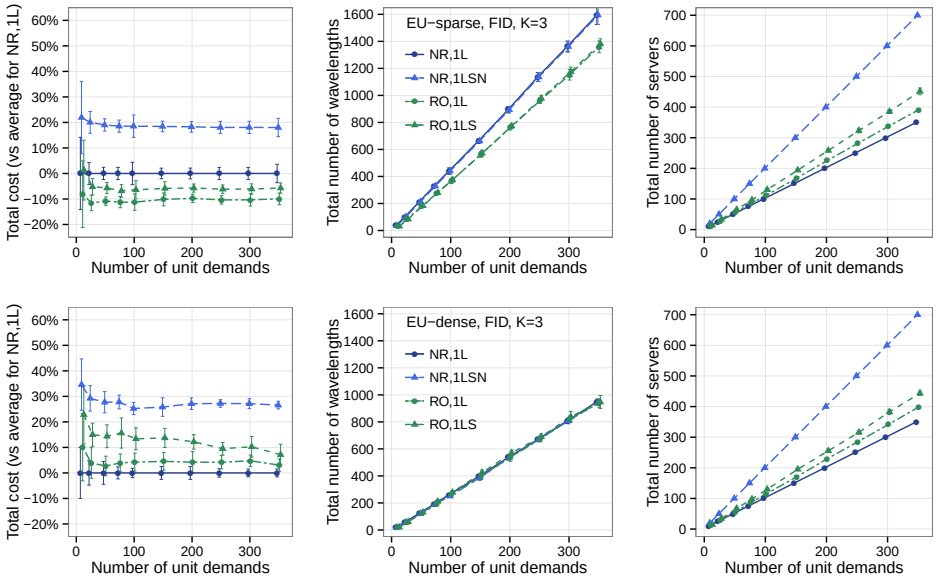


Fig. 6. When the topology becomes more dense, the advantage of exploiting relocation (RO) diminishes. When the topology becomes sufficiently dense (EU-dense), opportunities to find paths to an alternate destination that are shorter than a backup path to the same primary destination are limited: potential network savings for RO eventually disappear. 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.

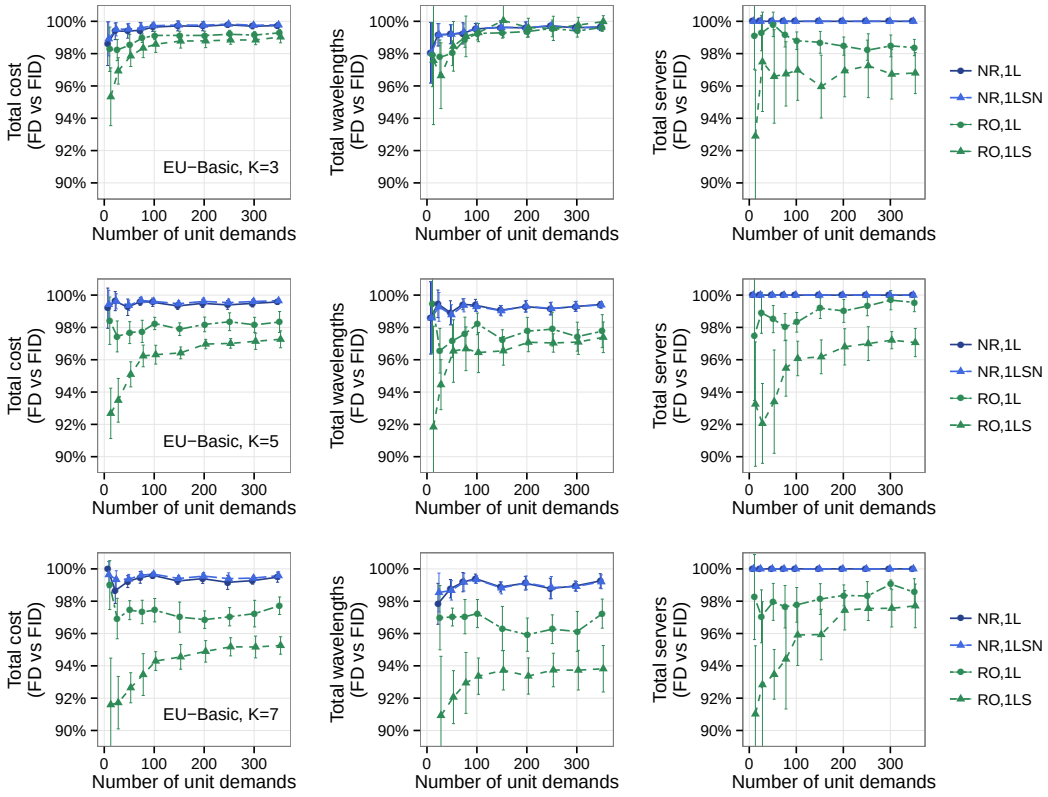


Fig. 7. The advantage of failure-dependent (FD) backup paths only appears if the number of sever sites K increases ($K = 3, 5, 7$ from top to bottom). This can be seen from the relative cost values *cost*(FD)/*cost*(FID), with *cost* being, from left to right: (i) total cost, (ii) total number of wavelengths, (iii) total number of servers. Error bars indicate 95% confidence intervals derived from the 10 random instances per data point.

all failure scenarios affecting the working one, and (*ii*) failure-dependent (FD) backup routing where the protecting path can be varied depending on the failure scenario. Our methodology is generic with respect to the possible failure scenarios, as long as the failure can be modeled as a joint failure of multiple links in the network model (i.e., as a shared risk link group, SRLG).

In our case studies, we find that for protection against single network link failures (1L), or against single failures of either a network link or a server (1LS), exploiting relocation (RO) can be beneficial in terms of total cost, accounting for both network and server capacity. Adopting RO to protect against single link failures incurs extra server capacity (compared to sticking to the same destination data center as under failure free conditions, i.e., NR), but that cost increase is more than outweighed by a reduction in network capacity. Particularly for the case where we protect also against single server failures (1LS), the net advantage of exploiting relocation (RO) can be more substantial, since even also in terms of server resources the cost can be lowered. Yet, note that the latter conclusion may differ for larger values of N in the considered 1:N server protection scheme. Also, the benefits of exploiting relocation depend on both topology and number of data center locations. For sparser topologies, the benefits of RO are more pronounced (while they can disappear for very dense topologies). When the number of data center sites K increases, relocation advantages become more significant.

Similar to studies considering unicast traffic, we find that the net advantage, in terms of total network and server cost, of adopting a failure-dependent (FD) backup routing strategy is fairly limited. Yet, this advantage increases for larger K, and compared to failure-independent (FID) backup routing further reduces the total server and network resources when we exploit relocation, and especially when covering for both single network link and single server failures: in those cases FD seems to be valuable as to reduce overall resource capacity requirements (despite increasing control plane level complexity).

REFERENCES

[1] I. Foster and C. Kesselman, *The Grid: Blueprint for a New Computing Infrastructure (2nd ed.)*. Elsevier, 2004.

[2] R. Pordes, D. Petravick, B. Kramer, D. Olson, M. Livny, A. Roy, P. Avery, K. Blackburn, T. Wenaus, F. Würthwein, I. Foster, R. Gardner, M. Wilde, A. Blatecky, J. McGee, and R. Quick, "The open science grid," *J. Physics Conf. Series*, vol. 78, no. 1, pp. 12–57, 2007.

[3] F. Gagliardi, B. Jones, F. Grey, M.-E. Bégin, and M. Heikkurinen, "Building an infrastructure for scientific grid computing: status and goals of the EGEE project," *Philos. Transact. A Math Phys. Eng. Sci.*, vol. 363, no. 1833, pp. 1729–1742, 15 Aug. 2005.

[4] D. Reed, "Grids, the TeraGrid and beyond," *IEEE Computer*, vol. 36, no. 1, pp. 62–68, Jan. 2003.

[5] V. Hamscher, U. Schwiegelshohn, A. Streit, and R. Yahyapour, "Evaluation of job-scheduling strategies for grid computing," in *Proc. 1st IEEE/ACM Int. Workshop Grid Comput. (GRID 2000)*, ser. Lecture Notes in Computer Science, R. Buyya and M. Baker, Eds., vol. 1971, Bangalore, India, 17–20 Dec. 2000, pp. 191–202.

[6] J. Yu and R. Buyya, "A taxonomy of scientific workflow systems for grid computing," *ACM SIGMOD Rec.*, vol. 34, no. 3, pp. 44–49, Sep. 2005.

[7] 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.

[8] Y. C. Lee and A. Z. Zomaya, "Rescheduling for reliable job completion with the support of clouds," *Futur. Gener. Comp. Syst.*, vol. 26, no. 8, pp. 1192–1199, Oct. 2010.

[9] C. Partridge, T. Mendez, and W. Milliken, "Host anycasting service," Internet Engineering Task Force, United States, RFC 1546, Nov. 1993.

[10] S. Ramamurthy, L. Sahasrabuddhe, and B. Mukherjee, "Survivable WDM mesh networks," *IEEE J. Lightwave Technol.*, vol. 21, no. 4, p. 870, Apr. 2003.

[11] 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, D.C., USA, 25–28 Oct. 2009, pp. 100–106.

[12] T. Stevens, M. De Leenheer, C. Develder, F. De Turck, B. Dhoedt, and P. Demeester, "ASTAS: Architecture for scalable and transparent anycast services," *J. Commun. Netw.*, vol. 9, no. 4, pp. 1229–2370, 2007.

[13] D.-R. Din, "A hybrid method for solving ARWA problem on WDM networks," *Comput. Commun.*, vol. 30, pp. 385–395, Jan. 2007.

[14] M. De Leenheer, F. Farahmand, K. Lu, T. Zhang, P. Thysebaert, F. De Turck, B. Dhoedt, P. Demeester, and J. P. Jue, "Anycast algorithms supporting optical burst switched grid networks," in *Proc. 2nd Int. Conf. Networking and Services (ICNS 2006)*, Santa Clara, CA, USA, 16–19 Jul. 2006.

[15] B. G. Bathula and J. M. Elmirghani, "Constraint-based anycasting over optical burst switched networks," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 1, no. 2, pp. A35–A43, Jul. 2009.

[16] M. Tang, W. Jia, H. Wang, and J. Wang, "Routing and wavelength assignment for anycast in WDM networks," in *Proc. 3rd Int. Conf. Wireless and Optical Commun. (WOC 2003)*, Banff, Canada, 14–16 Jul. 2003, pp. 301–306.

[17] D.-R. Din, "Anycast routing and wavelength assignment problem on WDM network," *IEICE Trans. Commun.*, vol. EE88-B, no. 10, pp. 3941–3951, Oct. 2005.

[18] E. Hyytiä, "Heuristic algorithms for the generalized routing and wavelength assignment problem," in *Proc. 17th Nordic Teletraffic Seminar (NTS-17)*, Fornebu, Norway, 25–27 Aug. 2004, pp. 373–386.

[19] K. Walkowiak, "Anycasting in connection-oriented computer networks: Models, algorithms and results," *J. Appl. Math. Comput. Sci.*, vol. 20, no. 1, pp. 207–220, Mar. 2010.

[20] K. Bhaskaran, J. Triay, and V. M. Vokkarane, "Dynamic anycast routing and wavelength assignment in WDM networks using ant colony optimization," in *Proc. IEEE Int. Conf. Commun. (ICC 2011)*, Kyoto, Japan, 5–9 Jun. 2011.

[21] Q. She, X. Huang, Q. Zhang, Y. Zhu, and J. Jue, "Survivable traffic grooming for anycasting in WDM mesh networks," in *Proc. IEEE Global Telecommun. Conf. (Globecom 2007)*, Washington D.C., USA, 26–30 Nov. 2007, pp. 2253–2257.

[22] K. Walkowiak and J. Rak, "Shared backup path protection for anycast and unicast flows using the node-link notation," in *Proc. IEEE Int. Conf. Commun. (ICC 2011)*, Kyoto, Japan, 5–9 Jun. 2011.

[23] T. Stevens, M. De Leenheer, C. Develder, B. Dhoedt, K. Christodoulopoulos, P. Kokkinos, and E. Varvarigos, "Multi-cost job routing and scheduling in grid networks," *Futur. Gener. Comp. Syst.*, vol. 25, no. 8, pp. 912–925, Sep. 2009.

[24] S. Demeyer, M. De Leenheer, J. Baert, M. Pickavet, and P. Demeester, "Ant colony optimization for the routing of jobs in optical grid networks," *J. Opt. Netw.*, vol. 7, no. 2, pp. 160–172, Feb. 2008.

[25] X. Liu, C. Qiao, W. Wei, X. Yu, T. Wang, W. Hu, W. Guo, and M.-Y. Wu, "Task scheduling and lightpath establishment in optical grids," *IEEE J. Lightwave Technol.*, vol. 27, no. 12, pp. 1796–1805, 15 Jun. 2009.

[26] X. Liu, C. Qiao, D. Yu, and T. Jiang, "Application-specific resource provisioning for wide-area distributed computing," *IEEE Netw.*, vol. 24, no. 4, pp. 25–34, Jul.-Aug. 2010.

[27] 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.

[28] L. Deboosere, P. Simoens, J. De Wachter, B. Vankeirsbilck, F. De Turck, B. Dhoedt, and P. Demeester, "Grid design for mobile thin client computing," *Futur. Gener. Comp. Syst.*, vol. 27, no. 6, pp. 681–693, Jun. 2011.

[29] M. Habib, M. Tornatore, M. De Leenheer, F. Dikbiyik, and B. Mukherjee, "A disaster-resilient multi-content optical datacenter network architecture," in *Proc. 13th Int. Conf. Transparent Optical Netw. (ICTON 2011)*, Stockholm, Sweden, 26–30 Jun. 2011, pp. 1–4.

[30] B. Jaumard, J. Buysse, A. Shaikh, M. De Leenheer, and C. Develder, "Column generation for dimensioning resilient optical grid networks with relocation," in *Proc. IEEE Global Telecommun. Conf. (Globecom 2010)*, Miami, FL, USA, 6–10 Dec. 2010.

[31] A. Shaikh, J. Buysse, B. Jaumard, and C. Develder, "Anycast routing for survivable optical grids: scalable solution methods and the impact of relocation," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 3, no. 9, pp. 767–779, Sep. 2011.

[32] C. Develder, J. Buysse, A. Shaikh, B. Jaumard, M. De Leenheer, and B. Dhoedt, "Survivable optical grid dimensioning: anycast routing with server and network failure protection," in *Proc. IEEE Int. Conf. Commun. (ICC 2011)*, Kyoto, Japan, 5–9 Jun. 2011.

[33] C. Develder, J. Buysse, M. De Leenheer, B. Jaumard, and B. Dhoedt, "Resilient network dimensioning for optical grid/clouds using relocation (invited paper)," in *Proc. Workshop on New Trends in Optical Networks Survivability, at IEEE Int. Conf. on Commun. (ICC 2012)*, Ottawa, Ontario, Canada, 11 Jun. 2012.

[34] A. M. Koster, A. Zymolka, M. Jäger, and R. Hülsermann, "Demand-wise shared protection for meshed optical networks," *J. Netw. Syst. Manag.*, vol. 13, no. 1, pp. 35–55, Mar. 2005.

[35] T. Stidsen, B. Petersen, S. Spoorendonk, M. Zachariasen, and K. B. Rasmussen, "Optimal routing with failure-independent path protection," *Netw.*, vol. 55, no. 2, pp. 125–137, Mar. 2010.

[36] A. Barak and R. Wheeler, *Mobility: processes, computers, and agents*. New York, NY, USA: ACM Press/Addison-Wesley Publishing Co., 1999, ch. MOSIX: an integrated multiprocessor UNIX, pp. 41–53.

[37] D. Thain, T. Tannenbaum, and M. Livny, "Distributed computing in practice: the Condor experience," *Concurr. Comput.: Pract. Exper.*, vol. 17, no. 2–4, pp. 323–356, Feb.–Mar. 2005.

[38] S. Osman, D. Subhraveti, G. Su, and J. Nieh, "The design and implementation of Zap: a system for migrating computing environments," *SIGOPS Oper. Syst. Rev.*, vol. 36, no. SI, pp. 361–376, Dec. 2002.

[39] G. J. Janakiraman, J. R. Santos, and D. Subhraveti, "Cruz: Application-transparent distributed checkpoint-restart on standard operating systems," in *Proc. 2005 Int. Conf. Dependable Systems and Networks (DSN 2005)*, Yokohama, Japan, 28 Jun.–1 Jul. 2005, pp. 260–269.

[40] S. L. Scott, G. Vallée, T. Naughton, A. Tikotekar, C. Engelmann, and H. Ong, "System-level virtualization research at Oak Ridge National Laboratory," *Futur. Gener. Comput. Syst.*, vol. 26, no. 3, pp. 304–307, Mar. 2010.

[41] M. Chtepen, F. Claeys, B. Dhoedt, F. De Turck, P. Demeester, and P. Vanrolleghem, "Adaptive task checkpointing and replication: Toward efficient fault-tolerant grids," *IEEE Trans. Parallel Distrib. Syst.*, vol. 20, no. 2, pp. 180–190, Feb. 2009.

[42] H. Zang, C. Ou, and B. Mukherjee, "Path-protection routing and wavelength assignment (RWA) in WDM mesh networks under duct-layer constraints," *IEEE/ACM Trans. Netw.*, vol. 11, no. 2, pp. 248–258, Apr. 2003.

[43] J. B. MacQueen, "Some methods for classification and analysis of multivariate observations," in *Proc. 5th Berkeley Symp. Mathematical Statistics and Probability*, vol. 1, Berkeley, CA, USA, 18–21 Jul. 1967, pp. 281–297.

[44] J. W. Suurballe and R. E. Tarjan, "A quick method for finding shortest pairs of disjoint paths," *Networks*, vol. 14, no. 2, pp. 325–336, 1984.

[45] Y. Liu, D. Tipper, and P. Siripongwutikorn, "Approximating optimal spare capacity allocation by successive survivable routing," *IEEE/ACM Trans. Netw.*, vol. 13, no. 1, pp. 198–211, Feb. 2005.

[46] S. Orlowski and M. Pióro, "Complexity of column generation in network design with path-based survivability mechanisms," *Networks*, vol. 59, no. 1, pp. 132–147, Jan. 2012.

[47] Y. Xiong and L. G. Mason, "Restoration strategies and spare capacity requirements in self-healing ATM networks," *IEEE/ACM Trans. Netw.*, vol. 7, no. 1, pp. 98–110, Feb. 1999.

[48] Y. Xiong and L. Mason, "Comparison of two path restoration schemes in self-healing networks," *Comput. Netw.*, vol. 38, no. 5, pp. 663–674, Apr. 2002.

[49] J. Buysse, M. De Leenheer, B. Dhoedt, and C. Develder, "On the impact of relocation on network dimensions in resilient optical grids," in *Proc. 14th Int. Conf. on Optical Network Design and Modelling (ONDM 2010)*, Kyoto, Japan, 31 Jan.–3 Feb. 2010, travel Grant Award.

[50] S. De Maesschalck, D. Colle, I. Lievens, M. Pickavet, P. Demeester, C. Mauz, M. Jaeger, R. Inkret, B. Mikac, and J. Derkacz, "Pan-european optical transport networks: An availability-based comparison," *Photonic Netw. Commun.*, vol. 5, no. 3, pp. 203–225, May 2003.

Chris Develder currently is associate professor with the research group IBCN of the Dept. of Information Technology (INTEC) at Ghent University - iMinds, Ghent, Belgium. He received the M.Sc. degree in computer science engineering and a Ph.D. in electrical engineering from Ghent University (Ghent, Belgium), resp. in Jul 1999 and Dec. 2003. In Sep. 2005, he re-joined INTEC as a post-doctoral researcher, and as a post-doctoral fellow of the FWO since Oct. 2006. In Oct. 2007 he obtained a part-time, and since Feb. 2010 a fulltime professorship at Ghent University. His research interests include dimensioning, modeling and optimizing optical (grid/cloud) networks and their control and management, smart grids, information retrieval and extraction. He has (co)authored over 130 papers in international journals or conference proceedings.

Jens Buysse received his Licentiate degree in computer science in 2007 from Ghent University, after which in 2013 he obtained a Ph.D. in computer science. His main interests lie in the field of photonic networks, distributed computing, virtualization techniques, energy efficient design principles, software development and mathematical modeling. He has participated in a Belgian project GEISHA and in the European projects PHOSPORUS and GEYSERS. He is currently working at the Hogeschool Gent as a lector in software development and analysis.

Bart Dhoedt received the M.Sc. degree in electrotechnical engineering from Ghent University, Ghent, Belgium, in 1990. His research, addressing the use of micro-optics to realize parallel free-space optical interconnects, resulted in the Ph.D. degree from Ghent University, in 1995. After a two-year postdoc in optoelectronics, he became Professor at the Department of Information Technology, Ghent University. He is responsible for various courses on algorithms, advanced programming, software development, and distributed systems. His research interests include software engineering, distributed systems, mobile and ubiquitous computing, smart clients, middleware, cloud computing, and autonomic systems. He is author or coauthor of more than 300 publications in international journals or conference proceedings.

Brigitte Jaumard holds a Concordia University Research Chair, Tier 1, on the Optimization of Communication Networks in the CIISE - Concordia Institute for Information Systems and Engineering - Institute at Concordia University. She was previously awarded a Canada Research Chair - Tier 1 - in the Department of Computer Science and Operations Research at Université de Montréal. She is an active researcher in combinatorial optimization and mathematical programming, with a focus on applications in telecommunications and artificial intelligence. Recent contributions include the development of efficient methods for solving large-scale mathematical programs, and their applications to the design and the management of optical and wireless, access and core networks. In Artificial Intelligence, contributions include the development of efficient optimization algorithms for probabilistic logic (reasoning under uncertainty) and for automated mechanical design. B. Jaumard has published over 150 papers in international journals in Operations Research and in Telecommunications.