

# Optimal Resource Allocation for Survivable Virtual Infrastructures

Gustavo A. de S. Cavalcanti, Rafael R. Obelheiro and Guilherme Koslovski

Graduate Program in Applied Computing

Department of Computer Science

Santa Catarina State University

Email: {gasc@joinville.udesc.br, rro@joinville.udesc.br, guilhermek@joinville.udesc.br}

**Abstract**—Virtual infrastructures (VIs) consolidated the dynamic provisioning of computing and communication resources. A VI is a set of virtual machines interconnected by virtual links and switches/routers. Infrastructure providers (InPs) manage the physical substrates in which virtual resources requested by VIs (such as CPU, disk, memory, bandwidth) are reserved and allocated. Resource allocation is a complex problem that needs to satisfy different goals: users expect to run their applications on survivable VIs, while InPs aim to maximize profits, minimize costs and reduce substrate fragmentation. However, there is a dichotomy between minimizing substrate fragmentation, by co-locating VIs, and maximizing VI survivability, by sparsely allocating resources in order to decrease the impact of substrate failures. In this context, this paper discusses VI survivability and its impact on substrate fragmentation. We propose a mixed integer linear programming model to allocate resources considering the joint coordination of fragmentation and survivability. Experimental results suggest that it is possible to enhance VI survivability without significantly impacting substrate fragmentation.

## I. INTRODUCTION

Cloud computing has introduced dynamic provisioning of virtualized resources and services driven by user requirements. This concept has induced a revolution in IT management by delivering compute, network and storage as services. Users no longer need to make significant up-front investments in IT infrastructure, they can just rent a set of services offered by Infrastructure Providers (InPs) – who own and manage IT infrastructures (or physical substrates) – on a pay-as-you-go basis [1]. Nowadays, InPs can offer completely virtualized infrastructures where users can execute their applications within confined and private sets of resources. A *virtual infrastructure* (VI) can be defined as a set of virtual machines (VMs) interconnected by virtual networking resources (links, switches and routers) [2].

A challenging problem for InPs is allocating physical resources for hosting VIs. An effective management framework must consider InPs’ objectives, usually guided by their financial perspective. InPs want to maximize their revenue by hosting as many VIs as possible using as little infrastructure as feasible. Indeed, previous efforts tried to minimize physical substrate fragmentation (i.e., the amount of physical resources needed) to decrease provisioning costs [3]. This goal may also benefit users, as it usually reduces communication latency among internal VI resources [4]. However, fragmentation is minimized by grouping provisioned resources, which may hamper VI availability and thus put users and providers at odds: a single physical failure can render many VIs and their hosted applications unavailable or inaccessible.

Unplanned data center (DC) outages are fairly common. A recent study [5] indicates that American companies experienced, in the last 24 months, an average of 2.04 complete and 10.16 limited DC outages, each lasting an average of 107 minutes. Another study [6] estimated that each minute of DC outage costs US\$ 5,600 on average. Cloud DCs are also affected by outages, which in some cases can last several hours [7], [8]. Popular cloud providers divulge substrate availability and/or reliability figures (e.g., 99.95% and 99.9%) but, in practice, when a cloud unavailability event occurs, end users just receive credits to re-launch their applications. From the user’s perspective, a single failure can compromise the entire execution of an application. To overcome this situation, some applications can recover from failures but this process usually affects execution time and complicates application development and management. Moreover, a failure can result in an SLA (Service Level Agreement) violation. InPs are aware of this, and can mitigate the impact on applications by considering resource failures (or availability requirements) during physical resource allocation [9]. Usually, this requires allocating spare physical resources [10]–[14], which increases provisioning cost. Indeed, as InPs add redundancy (backup servers, routers, switches, and links), power consumption and management complexity both increase. Although backup resources can be shared among several users (up to some point), the costs associated with reliability are ultimately shifted to end users.

In this context, we propose a novel optimal allocation model to provide survivable virtual infrastructures. The main goal of our model is to map virtual resources onto a physical substrate so as to minimize both substrate fragmentation and the number of virtual resources affected by substrate failures. To achieve this goal we present a mixed integer programming formulation that incorporates the perspective of InPs (reduced fragmentation) and users (increased survivability). Our experimental results discuss the trade-off between fragmentation and survivability. The main contributions of our work are threefold:

- An optimal allocation model that considers both physical substrate fragmentation and VI survivability;
- The introduction of local and global failure metrics, which provide a measure of survivability from the perspective of users and InPs, respectively;
- Experimental results on different physical substrate sizes and sets of VI requests. These results provide an optimal baseline of the fragmentation-survivability trade-off and serve as reference for future analysis.

This paper is organized as follows: Sec. II motivates our problem, while Sec. III presents the mixed integer programming model formulation. Sec. IV discusses the experimental results. Sec. V reviews related work, and Sec. VI concludes the paper indicating future perspectives.

## II. CONCEPTS AND DEFINITIONS

Our model seeks the optimal allocation of physical resources for hosting survivable VIs. This problem involves a set of variables, parameters and functions for representing user's and InP's goals. Fig. 1 illustrates the main concepts. Initially, Fig. 1a defines  $s$  as a *physical substrate* composed of physical machines (PMs)  $m^p$  and switches<sup>1</sup>  $s^p$ , interconnected by unidirectional links  $l^p$ . All physical resources are grouped in racks ( $d^r$ ) and connected to power circuits ( $d^c$ ). Next, Fig. 1b illustrates two *virtual infrastructures*,  $v_a$  and  $v_b$ , both composed of virtual resources: machines ( $m^a$  and  $m^b$ ), switches ( $s^a$  and  $s^b$ ), and links ( $l^a$  and  $l^b$ ). Viewing Figs. 1a and 1b as graphs, edge weights denote link capacities in the substrate, and bandwidth demands in a VI. Similarly, we associate CPU and memory capacity/demands to vertices in the graphs. The idea is that the resource demands of a VI are reserved by the InP in the physical substrate, as exemplified by Figs. 1c and 1d. Particularly, Fig. 1c shows that a virtual link (e.g.,  $l_1^b$ ) can be allocated on a multi-hop physical path ( $l_3^p, l_2^p$ ).

Figures 1c and 1d show how virtual infrastructures  $v_a$  and  $v_b$  can be allocated on substrate  $s$  according to different fragmentation-survivability trade-offs. Allocation  $m^{fr}$  reduces *physical substrate fragmentation*, defined as the ratio of the number of active physical resources (i.e., those hosting virtual resources) to the total number of physical resources. We consider two fragmentation metrics, node fragmentation (NFr) and link fragmentation (LFr). Node fragmentation is given by Eq. 1, where  $|\mathcal{N}_a^p|$  is the number of active nodes (both physical machines  $m^p$  and switches  $s^p$ ) and  $|\mathcal{N}_t^p|$  is the total number of nodes. Similarly, link fragmentation is given by Eq. 2, where  $|\mathcal{L}_a^p|$  is the number of active links and  $|\mathcal{L}_t^p|$  is the total number of links. By decreasing fragmentation (minimizing the sum of Eqs. 1 and 2), InPs can reduce costs by shutting down idle resources. For instance, in allocation  $m^{fr}$  (Fig. 1c), only one rack ( $d_1^r$ ) is hosting both VIs  $v_a$  and  $v_b$ , and so the idle racks ( $d_2^r$  and  $d_3^r$ ) can be deactivated. Table I shows fragmentation values for the allocations in Figs. 1c and 1d. For allocation  $m^{fr}$  we observe 33% of node fragmentation and 22% of link fragmentation, while allocation  $m^f$  – which favors survivability – has 78% of node/link fragmentation, 2.3 and 3.5 times the values for  $m^{fr}$ . However, a slightly different allocation  $m^{f'}$  (the variation in Fig. 1d) results in a smaller increase in node and link fragmentation (1.7 times greater than in  $m^{fr}$ ).

$$\text{NFr} = \frac{|\mathcal{N}_a^p|}{|\mathcal{N}_t^p|} \quad (1) \quad \text{LFr} = \frac{|\mathcal{L}_a^p|}{|\mathcal{L}_t^p|} \quad (2)$$

Physical resources are subject to crash failures; a failure may affect a single resource (e.g., a server) or a group of resources (e.g., an entire rack or data center). Following [9], we define a *fault domain* (FD) as a set of physical resources that share a single point of failure. The same node may belong

<sup>1</sup>The terms *switch* and *router* are used interchangeably in the remainder of the paper.

TABLE I. FRAGMENTATION AND FAILURE METRICS FOR FIG. 1.

$m$	Frag. (%)		Failure (%)							
	NFr	LFr	circuit ( $c$ )		rack ( $r$ )		server ( $s$ )		link ( $l$ )	
			LNF $v_a/v_b$	GNF	LNF $v_a/v_b$	GNF	LNF $v_a/v_b$	GNF	LLF $v_a/v_b$	GLF
$m^{fr}$	33	22	100/100	100	100/100	100	67/50	60	50/50	50
$m^f$	78	78	60/50	57	40/50	43	33/50	20	25/50	33
$m^{f'}$	56	56	60/50	57	40/50	43	33/50	40	25/50	50

to multiple FDs. In a cloud scenario, a VI can be spread across physical resources belonging to different FDs. To quantify the impact of substrate failures, we define a *failure metric* as the fraction of virtual resources (nodes and links) affected by the worst-case physical resource failure (i.e., the physical failure that affects the most virtual resources). This metric is calculated for all *types of fault domains* (or simply *types*) defined by the InP. For example, physical substrate  $s$  (Fig. 1a) has four types, *server*, *rack*, *power circuit*, and *link*. Each FD is denoted by  $d$  and each type is denoted by  $t$ .

Let us exemplify the impact of a rack failure. In Fig. 1, each rack  $d_n^r$  contains two physical machines and one switch. If a rack fails, all virtual resources hosted in a machine or switch within this rack are lost. Considering three FDs for the rack type ( $d_1^r$ ,  $d_2^r$  and  $d_3^r$ ), the worst-case scenario for allocation  $m^{fr}$  is the failure of rack  $d_1^r$ , which induces the loss of all virtual resources. Considering allocations  $m^f$  and  $m^{f'}$ , the worst-case scenario is given by the failure of  $d_3^r$ , which affects 40% of  $v_a$  and 50% of  $v_b$ , as summarized in Table I. *In this context, we observe that users and providers must minimize the failure metrics to consequently achieve an optimal survivability, regardless of financial goals.*

Although our overall *survivability goal* is minimizing the failure metrics of an allocation, users and InPs have different perspectives on survivability. A user typically wants to maximize the survivability of her own VIs without much concern for the survivability of other VIs, whereas an InP will attempt to provide good overall survivability (which may be sub-optimal to specific users). *In fact, InPs want to provide survivable VIs as service.* We model both perspectives as *local failure* and *global failure* metrics, respectively. The local failure metric (LF) embodies the user's perspective, considering both node and link failure metrics (LNF $_{v,t}$  and LLF $_{v,t}$ , respectively). For a given VI  $v$  and a FD type  $t$ , the local node failure metric (LNF $_{v,t}$ ) is given by the ratio of failing nodes ( $|\mathcal{N}_{v,t}^f|$ ) to total nodes ( $|\mathcal{N}_{v,t}^t|$ ) of  $v$  allocated in  $t$ , as shown in Eq. 3; an analogous definition for LLF $_{v,t}$  is given by Eq. 4. By decreasing LNF $_{v,t}$ /LLF $_{v,t}$ , we increase  $v$ 's survivability. Column *Failure* in Table I illustrates this metric for allocations  $m^{fr}$ ,  $m^f$  and  $m^{f'}$ . In  $m^{fr}$ ,  $v_a$  has high LNF, as all nodes are lost when a circuit or rack fails. Even if a physical server fails, only a single virtual node survives. In allocations  $m^f$  and  $m^{f'}$ , LNF $_{v,t}$  for  $v_a$  and  $v_b$  decreased across all types except for server failures affecting  $v_b$ .

The global failure metric (GF) follows the same principle of LF, with the difference that all hosted VIs are considered. Eq. 5 defines the global node failure metric as GNF $_t$ , a ratio of failing resources ( $|\mathcal{N}_t^f|$ ) to total resources ( $|\mathcal{N}_t^t|$ ) allocated in FD type  $t$ ; Eq. 6 provides an analogous definition for the global link failure metric (GLF $_t$ ). As with the local failure metrics,

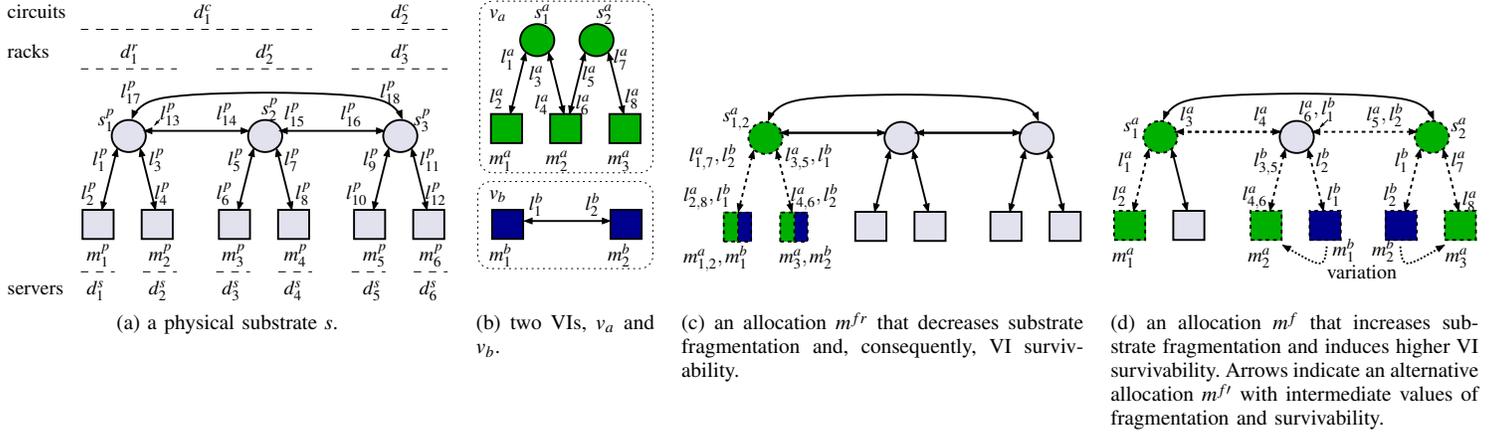


Fig. 1. A mapping example. Physical substrate  $s$  (Fig. 1a) can host two virtual infrastructures  $v_a$  and  $v_b$  (Fig. 1b), as exemplified by the allocations in Figs. 1c and 1d, which show where virtual resources are hosted on the substrate.

InPs want to minimize  $\text{GNF}_t$  and  $\text{GLF}_t$  in order to provide a survivable service. Allocation  $m^{fr}$  illustrates a scenario in which  $\text{GNF}_r = 100\%$  when a circuit or rack failure occurs. On the other hand, it offers a more reasonable value for server failures (at least 60% of the nodes can survive in the worst case). Global failure metrics for allocations  $m^f$  and  $m^{f'}$  are lower, indicating a higher survivability level for circuit and rack failures. Considering the server type, for allocation  $m^f$ ,  $\text{GNF}_s$  is 1/3 of  $\text{GNF}_s$  for  $m^{fr}$ , and the migration of  $v_b$  in  $m^{f'}$  brings  $\text{GNF}_s$  to 2/3 of the value for  $m^{fr}$ . Our model gives the same weight to the global and local metrics, but these weights can be redefined by an InP to prioritize either individual VIs or their ensemble, and to aggregate value to services. The adjustment of the weights and InP pricing strategies to exploit it are subjects for future research.

$$\text{LNF}_{v,t} = \frac{|\mathcal{N}_{v,t}^f|}{|\mathcal{N}_{v,t}^t|} \quad (3) \quad \text{LLF}_{v,t} = \frac{|\mathcal{L}_{v,t}^f|}{|\mathcal{L}_{v,t}^t|} \quad (4)$$

$$\text{GNF}_t = \frac{|\mathcal{N}_t^f|}{|\mathcal{N}_t^t|} \quad (5) \quad \text{GLF}_t = \frac{|\mathcal{L}_t^f|}{|\mathcal{L}_t^t|} \quad (6)$$

Finding a good fragmentation-survivability trade-off is not a trivial task. Compared to  $m^f$ , allocation  $m^{f'}$  in Fig. 1d decreases node fragmentation (NFr) by nearly 29%, without impacting the global and local failure metrics for the circuit and rack types. In fact, the values of local failure metrics for  $m^f$  and  $m^{f'}$  are identical. On the other hand, for the server type, allocation  $m^f$  is significantly more survivable than  $m^{f'}$ , since half as many servers fail in the worst case. Moreover, in  $m^f$ , any single fault will affect at most one VI, while in  $m^{f'}$  both VIs may be affected.

### III. PROPOSED MODEL

In this section we introduce a mixed integer linear programming (MILP) that provides VI survivability without increasing physical substrate fragmentation. The model is presented in four parts: i) a traditional graph embedding<sup>2</sup> problem

<sup>2</sup>The terms *embedding*, *mapping*, and *allocation* are used interchangeably throughout this paper.

(Sec. III-A); ii) a formulation to achieve optimal substrate usage considering fragmentation (Sec. III-B); iii) a formulation to achieve optimal local and global survivability (Sec. III-C); and iv) the combined model to minimize both substrate fragmentation and VI failures (Sec. III-D).

#### A. Graph embedding

Before presenting our fragmentation and survivability models, we have to introduce a traditional formulation for allocating VI resources. This problem aims at allocating resources in a physical substrate (or physical graph) for hosting virtual resources (a VI graph), and is known to be an NP-hard problem [15]. Table II summarizes the notation used in the formulation. The physical substrate is modelled as a set of machines ( $\mathcal{M}^p$ ) which communicate through switches ( $\mathcal{S}^p$ ). A set of nodes  $\mathcal{N}^p$  is used to group physical machines and switches. A set of links  $\mathcal{L}^p$  defines all physical interconnections; a link  $l^p$  between two nodes  $n_1^p$  and  $n_2^p$  is denoted by the tuple  $(n_1^p, n_2^p)$ . Similarly, a VI  $v$  is composed of sets of machines ( $\mathcal{M}^v$ ), switches ( $\mathcal{S}^v$ ), and links ( $\mathcal{L}^v$ ). Set  $\mathcal{N}^v$  denotes the union of  $\mathcal{M}^v$  and  $\mathcal{S}^v$ . A machine  $m$  (from  $\mathcal{M}^v$  or  $\mathcal{M}^p$ ) has capacities  $\text{MC}_{m,r}$ , where  $r \in \mathcal{R}^m$  indexes individual attributes (e.g., CPU, memory, storage) described for that machine. For physical nodes, MC represents the total available capacity, while for virtual ones it indicates the resource demand. Likewise, a switch  $s$  (from  $\mathcal{S}^v$  or  $\mathcal{S}^p$ ) has capacities  $\text{SC}_{s,r}$ , where  $r$  is a resource in  $\mathcal{R}^s$ . Finally, a link  $l$  between two nodes has capacities  $\text{LC}_{l,r}$ , where  $r \in \mathcal{R}^l$  is one of its attributes (e.g, latency, bandwidth). We chose to model CPU capacity for machines, memory capacity for switches, and bandwidth for links. As discussed, our model indicates the presence of sets for modelling these attributes and can be easily extended to cover more functional and/or non-functional capacities.

The goal of VI allocation is to find a combined mapping for allocating virtual nodes on physical nodes ( $nm_{n^v, n^p}$ ) and virtual links on physical paths ( $lm_{l^v, l^p}$ ). A virtual node  $n^v$  is mapped onto a single physical node  $n^p$ , which can host multiple virtual nodes ( $nm: \mathcal{N}^v \uparrow \mathcal{N}^p$ ); if  $n^p$  hosts  $n^v$ , then  $nm_{n^v, n^p}$  assumes 1, else 0. A virtual link  $l^v$  is mapped onto a physical path comprising one or more physical links  $l^p$ . Any

TABLE II. NOTATION FOR THE GRAPH EMBEDDING MODEL.

Notation	Description
<b>Sets</b>	
$m^v \in \mathcal{M}^v$	a virtual machine (VM)
$m^p \in \mathcal{M}^p$	a physical machine (PM)
$r \in \mathcal{R}^m$	a machine capacity (e.g., CPU or memory)
$s^v \in \mathcal{S}^v$	a virtual switch
$s^p \in \mathcal{S}^p$	a physical switch
$r \in \mathcal{R}^s$	a switch capacity (e.g., memory)
$n^v \in \mathcal{N}^v = \mathcal{M}^v \cup \mathcal{S}^v$	a virtual node (machine or switch)
$n^p \in \mathcal{N}^p = \mathcal{M}^p \cup \mathcal{S}^p$	a physical node (machine or switch)
$(n^v, n^p) \in \mathcal{A}^n \subseteq \mathcal{N}^v \times \mathcal{N}^p$	a virtual node preallocated in a physical one
$l^v := (n_1^v, n_2^v) \in \mathcal{L}^v \subseteq \mathcal{N}^v \times \mathcal{N}^v$	an unidirectional virtual link
$l^p := (n_1^p, n_2^p) \in \mathcal{L}^p \subseteq \mathcal{N}^p \times \mathcal{N}^p$	an unidirectional physical link
$r \in \mathcal{R}^l$	a link capacity (e.g., bandwidth)
$(l^v, l^p) \in \mathcal{A}^l \subseteq \mathcal{L}^v \times \mathcal{L}^p$	a virtual link preallocated in a physical one
<b>Parameters</b>	
$\text{MC}\{m \in \mathcal{M}^v \cup \mathcal{M}^p, r \in \mathcal{R}^m\}$ , real	node capacities matrix
$\text{SC}\{s \in \mathcal{S}^v \cup \mathcal{S}^p, r \in \mathcal{R}^s\}$ , real	switch capacities matrix
$\text{LC}\{l \in \mathcal{L}^v \cup \mathcal{L}^p, r \in \mathcal{R}^l\}$ , real	link capacities matrix
<b>Variables</b>	
$nm\{n^v \in \mathcal{N}^v, n^p \in \mathcal{N}^p\}$ , bin.	virtual-physical node mapping matrix
$lm\{l^v \in \mathcal{L}^v, l^p \in \mathcal{L}^p\}$ , bin.	virtual-physical link mapping matrix

physical link can host multiple virtual links ( $lm: \mathcal{L}^v \uparrow \mathcal{L}^p$ ); if  $l^p$  hosts  $l^v$ , then  $lm_{l^v, l^p}$  assumes 1, else 0.

In online VI allocation, users may submit VI requests at different times. In this scenario, when a VI  $v_a$  is already allocated and another VI  $v_b$  is requested, the InP must choose one of two approaches: (i) reallocate all VIs, which may enhance substrate fragmentation and VI survivability metrics but takes more time to solve and introduces VI migration issues; or (ii) keep the preallocated nodes and links on their respective physical ones, which avoids breaking existing SLAs. It is important to keep in mind that in both cases the allocations can be optimal. We opt for strategy (ii), currently adopted by popular cloud providers, which involves the management of only two new sets:  $\mathcal{A}^n$  for a virtual node  $n^v$  preallocated on a physical node  $n^p$ , denoted by the tuple  $(n^v, n^p)$ ; and  $\mathcal{A}^l$  for a link  $l^v$  preallocated on a physical link  $l^p$ , denoted by the tuple  $(l^v, l^p)$ . The constraints for the embedding model are specified by Eqs. 7–15.

Initially, Eq. 7 ensures that a PM will not host more VMs than its capacities allow; similar constraints apply to switches (Eq. 8) and links (Eq. 9). Eqs. 10 and 11 ensure that a virtual node will be allocated in only one physical node, and these nodes will be of the same type (i.e., VMs should be hosted on PMs, not in switches, and vice-versa). To satisfy these conditions, subsets of  $\mathcal{N}^p$  are defined in Eq. 10 as  $\mathcal{N}^{p'} := \forall n^p \in \mathcal{N}^p: (n^v \in \mathcal{M}^v \wedge n^p \in \mathcal{M}^p) \vee (n^v \in \mathcal{S}^v \wedge n^p \in \mathcal{S}^p)$ , and as  $\mathcal{N}^{p''} := \forall n^p \in \mathcal{N}^p: (n^v \in \mathcal{M}^v \wedge n^p \in \mathcal{S}^p) \vee (n^v \in \mathcal{S}^v \wedge n^p \in \mathcal{M}^p)$  in Eq. 11. Thus, if  $n^v$  is a VM, then  $\mathcal{N}^{p'}$  will contain PMs and  $\mathcal{N}^{p''}$  will contain switches (and vice-versa). Following the formulation proposed by [17], Eq. 12 guarantees that a virtual link  $l^v$  between any two virtual nodes,  $n_1^v$  and  $n_2^v$ , will be allocated in one or more physical links that form a connected physical path between the physical nodes  $n_1^p$  and  $n_2^p$  that host  $n_1^v$  and  $n_2^v$ . To satisfy this condition, subsets of  $\mathcal{N}^p$  are defined as  $\mathcal{N}^{p'''} := \forall n_2^p \in \mathcal{N}^p: (n_1^p, n_2^p) \in \mathcal{L}^p$ , in the first sum, and as  $\mathcal{N}^{p''''} := \forall n_2^p \in \mathcal{N}^p: (n_2^p, n_1^p) \in \mathcal{L}^p$ , in the second sum. In both cases,  $\mathcal{N}^{p'''} and  $\mathcal{N}^{p''''}$  will have only physical nodes  $n_2^p$  connected to  $n_1^p$ . Eq. 13 determines that two directly connected virtual nodes cannot be allocated in the same physical node. VI allocation is performed periodically, and each VI remains$

in the substrate during a discrete interval; this information is controlled by an entity outside the allocation model. To guarantee that preallocated VIs will remain on their physical hosts, Eqs. 14 and 15 ensure that virtual nodes and links remain in the same physical nodes and paths, respectively.

$$\sum_{m^v \in \mathcal{M}^v} \text{MC}_{m^v, r} \cdot nm_{m^v, m^p} \leq \text{MC}_{m^p, r}, \quad \forall r \in \mathcal{R}^m, \forall m^p \in \mathcal{M}^p \quad (7)$$

$$\sum_{s^v \in \mathcal{S}^v} \text{SC}_{s^v, r} \cdot nm_{s^v, s^p} \leq \text{SC}_{s^p, r}, \quad \forall r \in \mathcal{R}^s, \forall s^p \in \mathcal{S}^p \quad (8)$$

$$\sum_{l^v \in \mathcal{L}^v} \text{LC}_{l^v, r} \cdot lm_{l^v, l^p} \leq \text{LC}_{l^p, r}, \quad \forall r \in \mathcal{R}^l, \forall l^p \in \mathcal{L}^p \quad (9)$$

$$\sum_{n^{p'} \in \mathcal{N}^{p'}} nm_{n^v, n^{p'}} = 1, \quad \forall n^v \in \mathcal{N}^v \quad (10)$$

$$\sum_{n^{p''} \in \mathcal{N}^{p''}} nm_{n^v, n^{p''}} = 0, \quad \forall n^v \in \mathcal{N}^v \quad (11)$$

$$\sum_{(n_1^{p'''}, n_2^{p'''}) \in \mathcal{N}^{p'''}} lm_{n_1^v, n_2^v, n_1^{p'''}, n_2^{p'''}} + \sum_{(n_2^{p''''}, n_1^{p''''}) \in \mathcal{N}^{p''''}} lm_{n_1^v, n_2^v, n_2^{p''''}, n_1^{p''''}} = nm_{n_1^v, n_1^p} + nm_{n_2^v, n_1^p}, \quad (12)$$

$$\forall (n_1^v, n_2^v) \in \mathcal{L}^v, \forall n_1^p \in \mathcal{N}^p$$

$$nm_{n_1^v, n^p} + nm_{n_2^v, n^p} \leq 1, \quad \forall n^p \in \mathcal{N}^p, \forall (n_1^v, n_2^v) \in \mathcal{L}^v \quad (13)$$

$$nm_{n^v, n^p} = 1, \quad \forall n^v \in \mathcal{N}^v, \forall n^p \in \mathcal{N}^p: (n^v, n^p) \in \mathcal{A}^n \quad (14)$$

$$lm_{l^v, l^p} = 1, \quad \forall l^v \in \mathcal{L}^v, \forall l^p \in \mathcal{L}^p: (l^v, l^p) \in \mathcal{A}^l \quad (15)$$

## B. Fragmentation

Table III summarizes the notation used in the fragmentation model. Fragmentation can be analyzed at different granularity levels, such as resource capacities or physical machines. We consider fragmentation at the level of physical nodes and links, as investigated by previous works [3], [4]. As discussed in Sec. II, if an allocation  $a$  requires less physical nodes/links to host a given VI than an allocation  $b$ , then  $a$  induces less substrate fragmentation than  $b$ . Node fragmentation is formally denoted by  $nfr$ , and link fragmentation by  $lfr$ .

TABLE III. NOTATION FOR THE FRAGMENTATION MODEL.

Notation	Description
$nfr\{n^p \in \mathcal{N}^p\}$ , bin.	the physical <i>node</i> fragmentation vector
$lfr\{l^p \in \mathcal{L}^p\}$ , bin.	the physical <i>link</i> fragmentation vector

Eqs. 16 and 17 specify the fragmentation constraints. Eq. 16 checks if a physical node hosts at least one virtual node. The left-hand side defines the ratio of the number of virtual nodes allocated on a given physical node to the total number of virtual nodes. The right-hand side controls the physical node's fragmentation state: in use, if  $nfr$  assumes 1, or idle, if it assumes 0. It should be noted that, here, the number of hosted virtual nodes is irrelevant, what matters is whether a physical resource is *idle* or *in use*. Analogously, Eq. 17 checks the state of physical links (i.e., if they host at least one virtual link), following the same approach.

TABLE IV. NOTATION FOR THE SURVIVABILITY MODEL.

Notation	Description
<b>Sets</b>	
$v \in \mathcal{V}$	a VI identifier
$d \in \mathcal{D}^n$	a <i>node</i> FD
$t \in \mathcal{T}^n$	a <i>node</i> type
$d \in \mathcal{D}^l$	a <i>link</i> FD
$t \in \mathcal{T}^l$	a <i>link</i> type
<b>Parameters</b>	
$\text{REQ}\{n^v \in \mathcal{N}^v, v \in \mathcal{V}\}$ , bin.	the virtual node-VI mapping matrix
$\text{NDM}\{n^p \in \mathcal{N}^p, d \in \mathcal{D}^n\}$ , bin.	the physical <i>node</i> -FD mapping matrix
$\text{NDTM}\{d \in \mathcal{D}^n, t \in \mathcal{T}^n\}$ , bin.	the <i>node</i> FD-type mapping matrix
$\text{LDM}\{l^p \in \mathcal{L}^p, d \in \mathcal{D}^l\}$ , bin.	the physical <i>link</i> -FD mapping matrix
$\text{LDTM}\{d \in \mathcal{D}^l, t \in \mathcal{T}^l\}$ , bin.	the <i>link</i> FD-type mapping matrix
$\text{LNFE}\{v \in \mathcal{V}, t \in \mathcal{T}^n\}$ , int.	the estimated maximum <i>lnf</i> matrix
$\text{GNFE}\{t \in \mathcal{T}^n\}$ , int.	the estimated maximum <i>gnf</i> vector
$\text{LLFE}\{v \in \mathcal{V}, t \in \mathcal{T}^l\}$ , int.	the estimated maximum <i>llf</i> matrix
$\text{GLFE}\{t \in \mathcal{T}^l\}$ , int.	the estimated maximum <i>glf</i> vector
<b>Variables</b>	
$\text{lnf}\{v \in \mathcal{V}, t \in \mathcal{T}^n\}$ , int.	the matrix of a VI's <i>nodes</i> max. num. in a type's FD
$\text{gnf}\{t \in \mathcal{T}^n\}$ , int.	the vector of <i>all nodes</i> max. num. in a type's FD
$\text{llf}\{v \in \mathcal{V}, t \in \mathcal{T}^l\}$ , int.	the matrix of a VI's <i>links</i> max. num. in a type's FD
$\text{glf}\{t \in \mathcal{T}^l\}$ , int.	the vector of <i>all links</i> max. num. in a type's FD

$$\frac{\sum_{n^v \in \mathcal{N}^v} nm_{n^v, n^p}}{|\mathcal{N}^v|} \leq \lceil nfr_{n^p} \rceil, \quad \forall n^p \in \mathcal{N}^p \quad (16)$$

$$\frac{\sum_{l^v \in \mathcal{L}^v} lm_{l^v, l^p}}{|\mathcal{L}^v|} \leq \lceil lfr_{l^p} \rceil, \quad \forall l^p \in \mathcal{L}^p \quad (17)$$

Eqs. 18 and 19 define the fragmentation metrics used in our objective function (Eq. 28, discussed in Sec. III-D). Both metrics must be minimized in order to decrease the number of physical resources in use. They denote the ratio of the physical resources in use by the total number of resources (Eq. 18 for nodes and Eq. 19 for links).

$$\text{NFr} = \frac{\sum_{n^p \in \mathcal{N}^p} nfr_{n^p}}{|\mathcal{N}^p|} \quad (18) \quad \text{LFr} = \frac{\sum_{l^p \in \mathcal{L}^p} lfr_{l^p}}{|\mathcal{L}^p|} \quad (19)$$

### C. Survivability

Table IV summarizes the notation used in the survivability model. All virtual nodes of a given VI have a unique identifier  $v$ , to be distinguishable from those of other VIs (i.e., in requests set  $\text{REQ}:\mathcal{N}^v \uparrow \mathcal{V}$ ; if  $n^v \in \text{VI } v$ , then  $\text{REQ}_{n^v, v}$  assumes 1, else 0). As discussed in Sec. II, *local failure metrics* measure the impact of a physical failure on a single VI, and *global failure metrics* measure the impact on all nodes and links, regardless of which VI they belong to. To achieve an optimal value for these metrics, virtual resources must be spread across fault domains, taking into account all FD types (i.e., circuits, racks, servers, links). However, spreading a VI's nodes and links across FDs (local failure metric) is not the same as spreading all nodes and links across FDs (global failure metric). For example, the former may spread a VI  $v_a$  over two racks and a VI  $v_b$  over the *same* two racks, even if there is a third rack unused; the worst-case rack failure affects a fraction of each VI, but *always* affects both. The latter can spread all nodes over the three racks, using all of them, but leaving  $v_a$ 's nodes more concentrated on one rack and  $v_b$ 's nodes on another; the worst-case rack failure affects a smaller portion of nodes than before, but is *fatal* to one VI.

A physical node or link belongs to one or more fault domain types. For example, a machine  $m^p$  is a *server* placed into a *rack*, connected to a *power circuit*, so  $m^p$  belongs to three types. In addition, a switch  $s^p$  is placed into a *rack* and also connected to a *circuit*, but is not hosted on a *server* (consequently,  $s^p$  belongs to two types). *FD-type mapping* is denoted as NDTM (Node FD-type Mapping), for nodes (i.e.,  $\text{NDTM}:\mathcal{D}^n \uparrow \mathcal{T}^n$ ), and as LDTM (Link FD-type Mapping), for links (i.e.,  $\text{LDTM}:\mathcal{D}^l \uparrow \mathcal{T}^l$ ). If a FD  $d$  belongs to a type  $t$ , then  $\text{NDTM}_{d,t}$  assumes 1, else 0. The *physical resource-FD mapping* is denoted as NDM (Node-FD Mapping), for nodes (i.e.,  $\text{NDM}:\mathcal{N}^p \uparrow \mathcal{D}^n$ ), and as LDM (Link-FD Mapping), for links (i.e.,  $\text{LDM}:\mathcal{L}^p \uparrow \mathcal{D}^l$ ); if  $n^p$  is in type  $d$ , then  $\text{NDM}_{n^p, d}$  assumes 1, else 0 (the same applies to links).

The survivability objective function aims at minimizing the local and global failure metrics. Eqs. 20–23 specify the

constraints for achieving the optimal objective function guaranteeing local and global survivability. Eqs. 20 and 21 ensure that, for each FD type  $t$ , the number of virtual nodes (or links) from a given VI affected by a failure of this type (in the worst case) is at most *lnf* (or *llf*). Eqs. 22 and 23 provide the analogous constraints for global failure metrics. To simplify the formulation, Eq. 20 defines a subset of  $\mathcal{N}^v$  as  $\mathcal{N}^{v'} := \forall n^v \in \mathcal{N}^v$ :

$\text{REQ}_{n^v, v}$  to consider only nodes of a given VI  $v$ , and Eq. 22 defines a subset of  $\mathcal{N}^p$  as  $\mathcal{N}^{p'} := \forall n^p \in \mathcal{N}^p$ :  $\text{NDM}_{n^p, d}$  to consider only physical nodes in FD  $d$ . The same approach is valid for links: in Eq. 21, a subset of  $\mathcal{L}^v$  is defined as  $\mathcal{L}^{v'} := \forall (n_1^v, n_2^v) \in \mathcal{L}^v$ :  $\text{REQ}_{n_1^v, v}$  to consider only links of a VI  $v$ , and, in Eq. 23, a subset of  $\mathcal{L}^p$  is defined as  $\mathcal{L}^{p'} := \forall l^p \in \mathcal{L}^p$ :  $\text{LDM}_{l^p, d}$  to include only links in FD  $d$ .

$$\sum_{n^{p'} \in \mathcal{N}^{p'}} \sum_{n^{v'} \in \mathcal{N}^{v'}} nm_{n^{v'}, n^{p'}} \leq \text{lnf}_{v,t}, \quad \forall v \in \mathcal{V}, \forall d \in \mathcal{D}^n, \forall t \in \mathcal{T}^n: \text{NDTM}_{d,t} \quad (20)$$

$$\sum_{l^{p'} \in \mathcal{L}^{p'}} \sum_{l^{v'} \in \mathcal{L}^{v'}} lm_{l^{v'}, l^{p'}} \leq \text{llf}_{v,t}, \quad \forall v \in \mathcal{V}, \forall d \in \mathcal{D}^l, \forall t \in \mathcal{T}^l: \text{LDTM}_{d,t} \quad (21)$$

$$\sum_{n^{p'} \in \mathcal{N}^{p'}} \sum_{n^{v'} \in \mathcal{N}^{v'}} nm_{n^{v'}, n^{p'}} \leq \text{gnf}_t, \quad \forall d \in \mathcal{D}^n, \forall t \in \mathcal{T}^n: \text{NDTM}_{d,t} \quad (22)$$

$$\sum_{l^{p'} \in \mathcal{L}^{p'}} \sum_{l^{v'} \in \mathcal{L}^{v'}} lm_{l^{v'}, l^{p'}} \leq \text{glf}_t, \quad \forall d \in \mathcal{D}^l, \forall t \in \mathcal{T}^l: \text{LDTM}_{d,t} \quad (23)$$

Eqs. 24–27 are the failure components of the objective function (Eq. 28). They minimize the number of virtual node and link failures, tending to spread VI resources across the physical substrate. Following the definitions from Sec. II, Eq. 24 (and Eq. 26 for links) denote the ratio of the sum of local failures (e.g., sum of circuit, rack, and server failures for each VI) to the product of the number of types by the number of requested VIs. Similarly, in Eq. 25 (Eq. 27), these values denote the ratio of the global failures to the number of types. These components are normalized by estimated worst-case failure values for local and global node failures: LNFE

(Local Node Failure Estimate), GNFE (Global Node Failure Estimate); and for local and global link failures: LLFE (Local Link Failure Estimate), GLFE (Global Link Failure Estimate). These estimates represent the number of resources in each FD type; estimates for local failure metrics consider each VI separately, while estimates for global failure metrics consider all VIs at once. For example, in Fig. 1,  $v_a$  has three VMs and two switches (five nodes), and  $v_b$  only has two VMs; since both VIs can be allocated in the same rack (as exemplified by allocation  $m^{lr}$ ),  $\text{LNFE}_{v_a,r} = 5$ ,  $\text{LNFE}_{v_b,r} = 2$ , and  $\text{GNFE}_r = 7$ . Each VI has a number of nodes and links compromised by a worst-case failure, denoted as  $lnf$  and  $llf$ , respectively. These values represents the local failure metrics (nodes and links). Similarly, considering all virtual resources as belonging to a single large VI, the global failure metric is noted as  $gnf$  and  $glf$  for nodes and links, respectively. Local and global metrics have a different value for each FD type.

$$\text{LNF} = \frac{\sum_{t \in \mathcal{J}^n} \sum_{v \in \mathcal{V}} \frac{lnf_{v,t}}{\text{LNFE}_{v,t}}}{|\mathcal{J}^n| \cdot |\mathcal{V}|} \quad (24) \quad \text{GNF} = \frac{\sum_{t \in \mathcal{J}^n} \frac{gnf_t}{\text{GNFE}_t}}{|\mathcal{J}^n|} \quad (25)$$

$$\text{LLF} = \frac{\sum_{t \in \mathcal{J}^l} \sum_{v \in \mathcal{V}} \frac{llf_{v,t}}{\text{LLFE}_{v,t}}}{|\mathcal{J}^l| \cdot |\mathcal{V}|} \quad (26) \quad \text{GLF} = \frac{\sum_{t \in \mathcal{J}^l} \frac{glf_t}{\text{GLFE}_t}}{|\mathcal{J}^l|} \quad (27)$$

#### D. Combined Model

The objective function (Eq. 28) for the combined model minimizes both fragmentation (Eqs. 18 and 19) and failure metrics (Eqs. 24–27). Weights  $\alpha$  and  $\beta$  can be adjusted by the InP to achieve the desired balance between fragmentation and survivability. For instance, a provider may want to increase  $\beta$  to offer improved survivability at a premium price. The relationship between  $\alpha$  and  $\beta$  is further explored in Sec. IV.

$$\text{minimize: } \alpha \cdot (\text{NFr} + \text{LFr}) + \beta \cdot (\text{LNF} + \text{GNF} + \text{LLF} + \text{GLF}) \quad (28)$$

## IV. EVALUATION

In this section we evaluate the proposed model, focusing on the fragmentation-survivability trade-off in the combined model (Eq. 28). The model was implemented in CPLEX 12.5.1,<sup>3</sup> and executed on a 4GB-RAM AMD Phenom II X4 computer running Ubuntu Linux 12.04.2. Virtual and physical infrastructure topologies were generated using a simulator written in Python.

TABLE V. VIRTUAL AND PHYSICAL INFRASTRUCTURE SIZES.

Type	Size	Servers	Switch			DC
			Access	Aggregation	Core	
physical	small	24	2	2	1	1
	large	96	8	8	4	2
virtual	variable	1–3		1–2		-

The physical infrastructure topology is fixed, a three-layer DC design that follows Cisco’s reference architecture [16],

which is often implemented in large DCs and has high-resilience purpose. Access links have 1 Gbps, and links in the aggregation and core layers have 10 Gbps. There are two sizes of substrates, *small* and *large*, both with a fixed number of nodes, as shown in Table V. We considered four FD types: power circuit, rack, server, and link.

The number of virtual resources in a VI and their capacities were randomly generated according to a uniform distribution. VIs have between two and five nodes. The resource demands of virtual nodes are sized as a fraction of the physical capacities; thus, a VM demands 25–50% of the CPU capacity of a PM, a virtual switch demands 15–25% of the memory of a physical switch, and a virtual link demands 5–30% of the bandwidth of an access link. These parameters are similar to those used in related work [17], [18].

Table VI presents the evaluation scenarios. VI topologies follow a rule: a VM is always connected to another node, and a switch is always connected to two other nodes. Basically, this rule generates a set of directed graphs that represent real usage scenarios. We generated 150 VIs to be allocated per physical substrate scenario. A new VI arrives at each discrete iteration and is allocated if the substrate is not saturated, remaining active for a defined lifetime. The lifetime of a VI is of a few (for low-usage substrate scenarios) or many (for high-usage substrate scenarios) iterations. The mean allocation time is five seconds. It is important to note that the time to obtain a MILP solution will be exponential and a suboptimal approach will be necessary for a lower time.

TABLE VI. EVALUATION SCENARIOS.

Scenario	Description	Substrate size	VI requests	VI duration
LS	Low-usage Small substrate	small	150	5 iter.
LL	Low-usage Large substrate	large	150	5 iter.
HS	High-usage Small substrate	small	150	25 iter.
HL	High-usage Large substrate	large	150	85 iter.

Fig. 2 shows the results for each scenario using different fragmentation ( $\alpha$ ) and failure ( $\beta$ ) weights in the objective function (Eq. 28). The weights satisfy the condition  $\alpha + \beta = 1$ ; for example, if  $\beta = 0.6$ , then  $\alpha = 0.4$ . Failure weight values below 0.5 are omitted for clarity, since they provided worse survivability without significantly improving fragmentation. For comparison purposes, we also show fragmentation-only ( $\alpha = 1, \beta = 0$ ) and survivability-only ( $\alpha = 0, \beta = 1$ ) curves, which provide fragmentation and failure baselines; the primary aim is to evaluate the scenarios with  $\beta \in [0.5, 0.9]$ .

Figs. 2a–2d show the combined node and link fragmentation (NFr and LFr in Eq. 28) for each scenario. In these scenarios, the failure baseline represents the *upper bound on fragmentation* (i.e., the worst fragmentation values), and the fragmentation baseline represents the *lower bound on fragmentation* (i.e., the optimal fragmentation values). In the LS scenario (Fig. 2a), the curves with  $\beta$  between 0.5 and 0.8 are almost equally closer to the fragmentation baseline than the other values, at 12.3 percentage points (pp) above  $\beta = 0$ , and they provide an improvement of 26.0pp over the failure baseline, on average. In the HS scenario (Fig. 2b),  $\beta = 0.6$  stands alone with a better performance than the fragmentation baseline (1.8pp below), and it shows an improvement of 2.7–26.0pp over other  $\beta$  values. In the LL scenario (Fig. 2c),  $\beta$

<sup>3</sup><http://www.ibm.com/software/products/us/en/ibmilogcpleoptistud/>

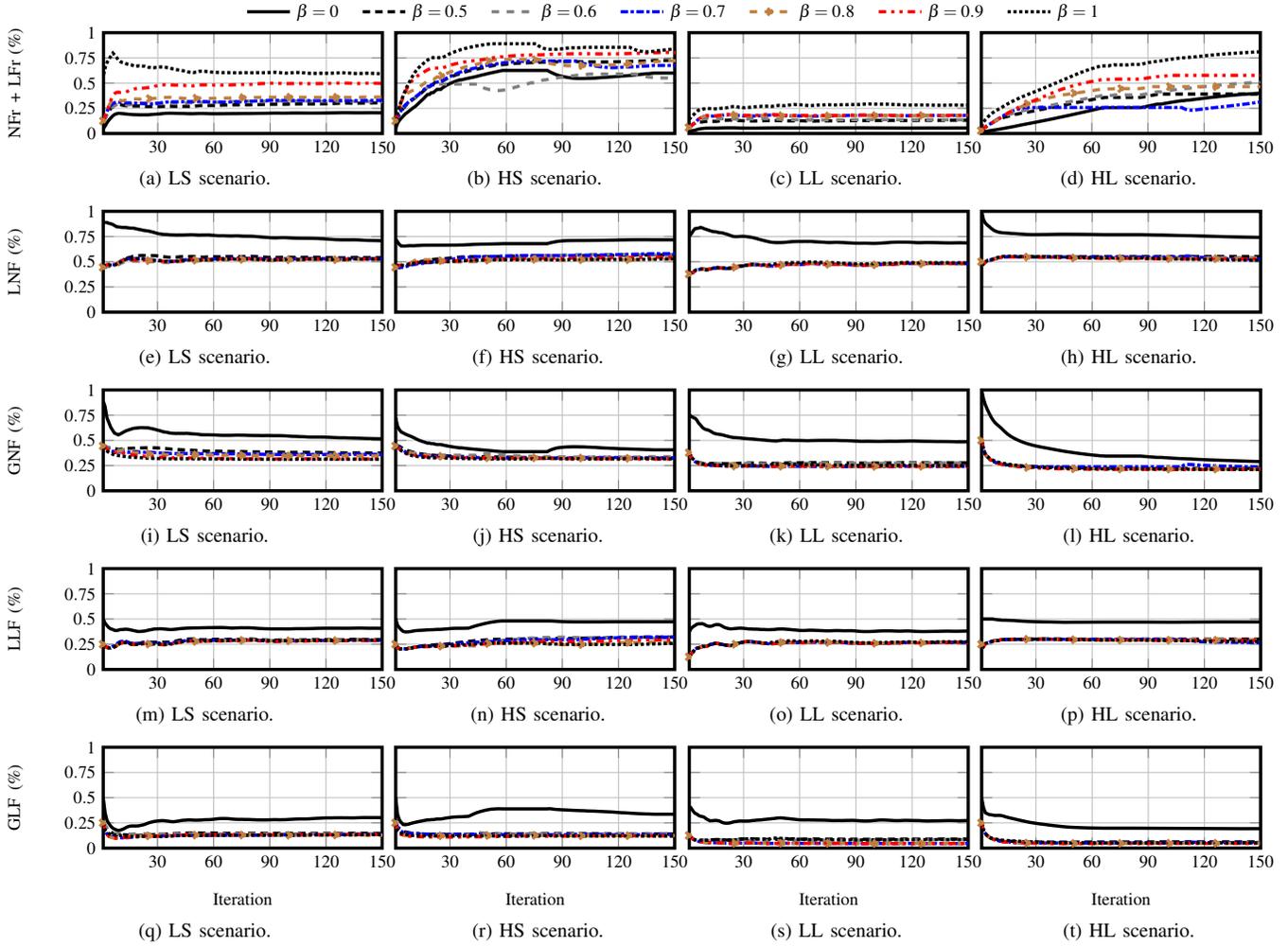


Fig. 2. Experimental results for the scenarios described in Table VI. An efficient trade-off between fragmentation and survivability is obtained with  $\beta \in [0.6, 0.7]$ . Basically, an InP is capable of offering a survivable VI provisioning service without compromising its internal physical substrate fragmentation.

values between 0.5 and 0.9 are almost equally close to the fragmentation baseline (10.5pp above), and show an improvement of 11.8pp over the failure baseline. In the HL scenario (Fig. 2d), despite  $\beta = 0.7$  nearly outperforming the fragmentation baseline,  $\beta$  values in the 0.5–0.8 range are almost equally close to the fragmentation baseline (1.9pp above) and they show an improvement of 39.1 pp over other values, on average. From the fragmentation perspective,  $\beta = 0.9$  proved to be an expensive choice for the InP, as it follows the failure baseline closely. On the other hand,  $\beta = 0.6$  revealed to be the cheapest choice for the InP, as it closely follows the fragmentation baseline; other  $\beta$  values are inconsistent, and they tend to have intermediate fragmentation metrics.

The survivability provided by the different values of  $\beta$  should also be evaluated, in order to enable us to assess which combination(s) of weights improves the fragmentation-survivability trade-off. Figs. 2e–2t show the local and global node and link failures (LNF, GNF, LLF, and GLF in Eq. 28) for each scenario. In these scenarios, the fragmentation baseline represents the *upper bound on failure* (i.e., the worst failure values), and the failure baseline represents the *lower bound on failure* (i.e., the optimal failure values). It can be

seen that, in all scenarios,  $\beta \in [0.5, 0.9]$  provide similar effects on the failure metrics, with the curves following the failure baseline closely. Only the fragmentation baseline is visibly worse in terms of failures.

Therefore, taking into account both perspectives (fragmentation and survivability), we can say that, for the evaluated scenarios,  $\beta = 0.6$  and  $\alpha = 0.4$  provide the better trade-off. An InP performing allocation with our model using these weights will be able to enhance VI survivability at a small increase in cost, balancing these conflicting goals.

While other scenarios may require different values for  $\alpha$  and  $\beta$ , the results evidence that the proposed model can be used to find a good trade-off between fragmentation and survivability in VI allocation. They highlight that InPs can offer survivable VIs without compromising management costs, since it is possible to enhance survivability while keeping fragmentation metrics close to their optimal baseline.

## V. RELATED WORK

Supporting reliability requirements in VI allocation is discussed in [12], [13]. In [12], VI reliability is achieved

by reserving backup resources, which may be shared among several users. In [13] authors proposed a solution for mapping and reallocating survivable VIs in a federated substrate. Their approach starts with an optimal mapping solution refined by a heuristic that adds backup resources considering a sequence of possible failures scenarios and consequently increases provisioning costs.

The body of literature on supporting reliability requirements in virtual network allocation is more extensive. Bodík and colleagues [9] propose a model for mapping VMs to data center resources taking into account availability requirements. Nodes in the substrate are grouped in distinct FDs, and the objective function seeks to maximize node survival in the worst case of failure and minimize bandwidth utilization (e.g., it favors hosting a virtual network in a single rack or under the same aggregation switch). However, as each virtual network is considered in isolation, their approach is analogous to our local failure metric, and can lead to an undesirable situation from the standpoint of global failures. Rahman et al. [10] introduce models for survivable virtual network embedding that minimize penalties for SLA violations and bandwidth utilization by reserving spare substrate resources. Barla et al. [11] propose a model for minimizing latency in virtual networks, which are spread across different data centers to improve reliability. In their approach, mapping virtual nodes to data centers may be controlled either by the user (who assigns one primary and one backup DC to his VN) or by the provider (who determines which DCs should be used). However, the model does not account for virtual resource capacity, and the reliability schemes not only use coarse-grained FDs (e.g., geographic regions), but also rely on the user to perform tasks with conflicting goals (spreading nodes to improve reliability while trying to minimize latency). Recently, this work was extended to provide end-to-end resilience at virtual and physical layers [19]. Similarly with our approach, they consider that an InP can offer a resilient or survivable service to users. They extended the discussion by considering an intermediate layer between end users and InPs, the operators, that can manage end users services. Both approaches can be combined to offer an optimal routing resilience and better quality-of-experience.

Optimal models for embedding virtual networks supporting either link reliability [18] or confidentiality [17] requirements have also been proposed; neither of these works consider resource fragmentation, which may lead to ineffective resource utilization. With the exception of [12], none of the above works incorporates computing and communication resources simultaneously: even when targeting VIs, only a single type of resource (e.g., communication channels) is considered. All proposals that address reliability concerns either reserve spare substrate resources or sparsely allocate virtual resources, which increases fragmentation and, consequently, InP costs.

## VI. CONCLUSION

This paper presented an optimal resource allocation model for survivable VIs. Experimental results demonstrated the model is effective in simultaneously optimizing for fragmentation, minimizing substrate usage and lowering provider costs, and survivability, benefitting users by executing their hosted applications in assuredly survivable VIs and, at the same time, ensuring to the InP that the most number of SLAs will be

fulfilled. This is the first work to jointly consider substrate fragmentation and VI survivability. As future work, we will investigate heuristics for polynomial time solutions. We also intend to augment the model expressiveness, allowing users to specify which components of their VIs must be survivable (and thus allocated in resources that fail in an independent manner).

## ACKNOWLEDGMENT

This research was supported by the PROMOP program of Santa Catarina State University.

## REFERENCES

- [1] P. Mell and T. Grance, "The NIST Definition of Cloud Computing," Tech. Rep., Jul. 2009. [Online]. Available: <http://goo.gl/PsJP5R>
- [2] F. Anhalt, G. Koslovski, and P. V.-B. Primet, "Specifying and Provisioning Virtual Infrastructures with HIPerNET," *Int. J. Network Management*, vol. 20, no. 3, pp. 129–148, 2010.
- [3] A. Greenberg, J. Hamilton, D. A. Maltz, and P. Patel, "The Cost of a Cloud: Research Problems in Datacenter Networks," *ACM SIGCOMM Comp. Comm. Review*, vol. 39, no. 1, pp. 68–73, 2008.
- [4] G. Koslovski, S. Soudan, P. Gonçalves, and P. Vicat-Blanc, "Locating Virtual Infrastructures: Users and InP Perspectives," in *Proc. IFIP/IEEE IM*, 2011, pp. 153–160.
- [5] Ponemon Institute, "2013 Study on Data Center Outages," Emerson Network Power, Tech. Rep., 2013, <http://goo.gl/v5DBJO>.
- [6] Ponemon Institute, "Understanding the Cost of Data Center Downtime: An Analysis of the Financial Impact on Infrastructure Vulnerability," Tech. Rep., 2011, <http://goo.gl/2jM3c>.
- [7] J. McCarthy, "The 10 Biggest Cloud Outages of 2012," *CRN*, 2012, <http://goo.gl/OcivLK>.
- [8] J. R. Raphael, "The Worst Cloud Outages of 2013 (So Far)," *Infoworld*, 2013, <http://goo.gl/6Q7oK7>.
- [9] P. Bodík, I. Menache, M. Chowdhury, P. Mani, D. A. Maltz, and I. Stoica, "Surviving Failures in Bandwidth-Constrained Datacenters," in *Proc. ACM SIGCOMM*, 2012, pp. 431–442.
- [10] M. R. Rahman, I. Aib, and R. Boutaba, "Survivable Virtual Network Embedding," in *Proc. IFIP Networking*, 2010, pp. 40–52.
- [11] I. B. Barla, D. A. Schupke, and G. Carle, "Resilient Virtual Network Design for End-to-End Cloud Services," in *Proc. IFIP Networking*, 2012, pp. 161–174.
- [12] G. Koslovski, W.-L. Yeow, C. Westphal, T. T. Huu, J. Montagnat, and P. Vicat-Blanc, "Reliability Support in Virtual Infrastructures," in *Proc. IEEE CloudCom*, 2010, pp. 49–58.
- [13] H. Yu, C. Qiao, V. Anand, X. Liu, H. Di, and G. Sun, "Survivable Virtual Infrastructure Mapping in a Federated Computing and Networking System under Single Regional Failures," in *Proc. IEEE GLOBECOM*, 2010, pp. 1–6.
- [14] J. Xu, J. Tang, K. Kwiat, W. Zhang, and G. Xue, "Survivable Virtual Infrastructure Mapping in Virtualized Data Centers," in *Proc. IEEE GLOBECOM*, 2012, pp. 196–203.
- [15] N. M. K. Chowdhury, M. R. Rahman, and R. Boutaba, "Virtual Network Embedding With Coordinated Node and Link Mapping," in *Proc. IEEE INFOCOM*, 2009, pp. 783–791.
- [16] "Cisco Data Center Infrastructure: 2.5 Design Guide," 2007.
- [17] L. R. Bays, R. R. Oliveira, L. S. Buriol, M. P. Barcellos, and L. P. Gaspar, "Security-Aware Optimal Resource Allocation for Virtual Network Embedding," in *Proc. ACM/IEEE/IFIP CNSM*, 2012, pp. 378–384.
- [18] R. R. Oliveira, L. R. Bays, D. S. Marcon, M. C. Neves, L. S. Buriol, L. P. Gaspar, and M. P. Barcellos, "DoS-Resilient Virtual Networks through Multipath Embedding and Opportunistic Recovery," in *Proc. ACM SAC*, 2013, pp. 597–602.
- [19] I. Barla, D. Schupke, M. Hoffmann, and G. Carle, "Optimal Design of Virtual Networks for Resilient Cloud Services," in *Proc. DRCN*, 2013, pp. 218–225.