8-1-2015

Survivable Cloud Network Mapping for Disaster Recovery Support

Feng Gu

Khaled B. Shaban

Nasir Ghani

Samee Khan

Mahshid R. Naeini

See next page for additional authors

Authors

Feng Gu, Khaled B. Shaban, Nasir Ghani, Samee Khan, Mahshid R. Naeini, Majeed M. Hayat, and Chadi Assi

IEEE Transactions on Computers, Vol. 64, No. 8 (August 1, 2015): 2353-2366. DOI.

Survivable Cloud Network Mapping for Disaster Recovery Support

Feng Gu
VMware Inc, Palo Alto, CA
Khaled Shaban
Qatar University, Doha, Qatar
Nasir Ghani
University of South Florida, Tampa, FL
Samee Khan
North Dakota State University, Fargo, ND
Mahshid R. Naeini
Texas Tech University, Lubbock, TX
Majeed M. Hayat
University of New Mexico, Albuquerque, NM
Chadi Assi

Concordia University, Montreal, QC, Canada

Abstract
Network virtualization is a key provision for improving the scalability and reliability of cloud computing services. In recent years, various mapping schemes have been developed to reserve VN resources over substrate networks. However, many cloud providers are very concerned about improving service reliability under catastrophic disaster conditions yielding multiple system failures. To address this challenge, this work presents a novel failure region-disjoint VN mapping scheme to improve VN mapping survivability. The problem is first formulated as a mixed integer linear programming problem and then two heuristic solutions are proposed to compute a pair of failure region-disjoint VN mappings. The solution also takes into account mapping costs and load balancing concerns to help improve resource efficiencies. The schemes are then analyzed in detail for a variety of networks and their overall performances compared to some existing survivable VN mapping schemes.

SECTION 1 Introduction

Cloud computing services are being rapidly deployed across a range of sectors to improve the scalability and cost-effectiveness of *information technology* (IT) services. These services allow clients to adjust their computing and storage resources in an "on-demand" manner in order to meet their needs and preclude capital-expensive data-center deployments [1]. However, as these paradigms take hold, there is a growing need to distribute operation and data across multiple physical data-center sites, i.e., for improved speed, responsiveness, load-balancing, etc. It is here that *network virtualization* can be very beneficial. Namely, this paradigm will allows users (i.e., cloud service providers) to build customized *virtual network* (VN) overlays to interconnect resource pools across underlying network substrates. Since multiple users can share these substrates, very high cost efficiency can be achieved.

Now a typical client network virtualization setup consists of a set of VN nodes (e.g., processing power and storage resources) interconnected by a set of VN bandwidth links [2]. In order to provision such demands, each VN node has to be mapped to a distinct substrate node and each VN link has to be mapped to a unique bandwidth connection between the mapped VN node endpoints, i.e., such that client demands for storage, processing power, and interconnection are met. Along these lines, a range of VN mapping algorithms have been developed in recent years, see [3], [4], [5], [6], [7]. Some common provisioning objectives here include revenue maximization, cost reduction, and others, see [8].

However, as more IT services migrate to the cloud, VN survivability is becoming a major concern. For example, failures in underlying network substrate nodes or links can easily disrupt higher-layer VN mappings, causing service degradation and even client *service level agreements* (SLA) violations. This issue is particularly acute for "mission-critical" applications running in the cloud, as they may require very high availability levels (even on par with dedicated data-centers). Perhaps most challenging here are large-scale stressor events that can cause widespread service outages, e.g., as resulting from natural disasters, malicious *weapons of mass destruction* (WMD) attacks, cascading power outages, etc. Many of these occurrences can yield highly-correlated (time and space) node and link failures [9], [10].

Despite these concerns, only a handful of studies have looked at survivable VN mapping design. For example, protection schemes have been studied for single link failures in [11], [12], [13], [14], [15], and single node failures in [16], [17], [18]. However, due the highly-disruptive, i.e., multi-failure, nature of disaster events, there is a growing need to develop robust VN mapping schemes to handle such cases. To the best of our knowledge, the only related work for VN mapping under regional failures is proposed in[19], [20]. Namely, here the authors compute backup resources for each potential (multi-failure) stressor and then use resource sharing to reduce

overall costs. Although this work presents a good baseline, pre-provisioning against all possible stressors is very resource-intensive and yields high blocking (reduced revenues).

To address the VN survivability issue, this paper proposes a failure region-disjoint VN mapping scheme. The work pursues a similar philosophy as some existing *shared risk link group* (SRLG) protection schemes for regular *point-to-point* (P2P) connections [21]. Namely, working and protection VN mappings are computed for each request to ensure that they are failure region disjoint, i.e., to guarantee recovery from a single regional failure event. To achieve this, a *mixed integer linear programming* (MILP) formulation is proposed, similar to [5], which adds new constraints to prevent the working/protection mappings from traversing the same failure region. However, owing to high MILP computational complexity, two heuristic algorithms are also developed for survivable VN mapping. Namely, the *failure region group based mapping* (FRGBM) scheme divides all failure regions into two fixed groups and only allows a working mapping to use resources in one of these groups. Meanwhile, the *dynamic failure region disjoint mapping* (DFRDM) scheme computes mappings without prior separation of failure regions. Since both these solutions only provision two VN mappings, they achieve lower blocking (higher revenues) versus other survivable VN strategies.

The overall paper is organized as follows. Section 2 first presents a background survey of existing work in VN mapping and survivability design. Section 3 then details the survivable VN mapping model and its objectives. Subsequently, Section 4 presents a MILP model, followed by heuristic solutions in Section 5 . Performance evaluation results are then presented in Section 6 for two network topologies. Finally, conclusions and directions for future work are presented in Section 7.

SECTION 2 Background Review

The VN mapping problem has seen much focus in recent years and has been shown to be NP-hard [8]. As a result, researchers have studied a variety of optimization and heuristic schemes subject to various constraints [3], [4], [5], [6], [7]. Now most optimization schemes use MILP formulations to minimize substrate network resource usages or maximize revenues. For example, a well-known method is proposed in [5], where each VN node is treated as a meta-node connected to all substrate nodes by meta-links with infinite capacity. Additional constraints are then introduced to ensure that each meta-node only routes its flow to a single (adjacent) meta-link, i.e., as a VN node can only be mapped to a unique substrate node. Other MILP formulations are also presented in [16], [20].

However, since MILP formulations pose high complexity, researchers have also developed VN mapping heuristics. In general, these strategies can be classified into two types, i.e., separate node/link mapping (two-stage) and joint node/link mapping (single-stage). The former types first map VN nodes to substrate network nodes without mapping VN links using various methods. For example, [3] computes a cluster center by considering node stress and adjacent link stresses. After the center is determined, the remaining nodes are selected based upon node stress and distance to all other assigned nodes. Meanwhile [4] selects a set of candidate substrate nodes to satisfy certain restrictions, and then maps each VN node to a substrate node with the maximum resources in the candidate set. VN links are then routed after the VN node mappings are done, i.e., connection routing. In general, shortest path [3] or k-shortest path [4]methods are commonly used here, although multi-commodity flow schemes can also be applied if path-splitting is supported at the substrate level. By contrast, single-stage VN mapping schemes immediately route all VN links for a mapped VN node after it is mapped, i.e., if the other end of the VN link has been mapped [6], [7]. This joint approach can improve resource efficiency and also reduce blocking.

Now many studies have looked at P2P connection recovery in data routing and optical DWDM networks. These schemes can be classified into two categories, *protection* and post-fault *restoration*, and mostly focus on single

link failures. However, some recent efforts have also looked at more complex multi-failure scenarios. For example, [22], [23] studies network vulnerability (topology robustness) under multi-failure conditions. Reliable connection routing for generalized multi-failure scenarios is also treated in [24]. Furthermore, recent efforts have also looked at survivable VN mapping, i.e., for single link failures in [11], [12], [13], [14], [15] and single node failures in [16] , [17], [18]. Consider the details.

The schemes in [12], [13], [15] address single substrate link failures and compute a pair of link disjoint paths for each mapped VN link. Similarly, [10] computes a set of backup detour routes for each substrate link. These studies use both *integer linear programming* (ILP) and heuristic methods, and some also implement resource sharing [15]. Meanwhile, [16] and [17] study single node failures and propose 1-redundant and k-redundant solutions. Namely, the former approach uses one backup substrate node to protect all VN nodes. However this approach can lead to nodal congestion when trying to route backup VN links to all the neighbors of a failed node. Hence the k-redundant scheme provides k backup nodes to help balance protection distribution. The problem is formulated as an MILP and then solved using a two-state simplification. Meanwhile, [18] details another protection scheme based upon a single backup node (akin to the 1-redundant scheme). This strategy basically remaps all VN nodes after a failure event even if one node is affected. Although this flexibility lowers backup resource costs and request blocking versus [16], [17], it entails more operational complexity. Also, the solution does not solve the MILP and instead decomposes VN topologies into small star-topologies and uses resource bipartite graphs to assign backup resources.

Finally, the more challenging case of multiple substrate node/link failures is studied in[19], [20], i.e., for large-scale regional stressors/disaster events. In particular multiple backup VN mappings are computed for each possible failure region. However, since it is assumed that only one regional stressor can occur at a given time, further resource sharing is done across all backup VN mappings (and the working VN mapping). This effort also details a MILP formulation and then presents two relaxation-based solutions using Lagrangian and decomposition-based rounding. In addition, a heuristic solution, termed *separate optimization with unconstrained mapping* (SOUM), is also outlined based upon a single-stage VN mapping approach to compute the working and backup VN mappings. Another *incremental optimization with constrained mapping* (IOCM) heuristic is also proposed by incrementally adding backup resources for each failure region, i.e., backup resources are only added for nodes affected by a given failure. Overall the multi-failure VN mapping solutions in [19], [20] will likely consume excessive resources if there are many failure regions. Hence an alternate failure region-disjoint strategy is proposed next.

SECTION 3 Network Model And Problem Description

The overall network model for the VN mapping problem is now presented along with the requisite notation. For reference sake, detailed descriptions of the key terms and their associated variables are also given in Table 1.

TABLE 1 Notation Description and Overview

Term	Description	Variable
Substrate network	Physical network topology with data-center resources (computing, storage), network switches, and network links	$G_s = (V_s, E_s)$
Virtual network request	Virtual network overlay requested by client, VN nodes to be mapped to substrate data-center resources, VN links to be mapped to substrate network connections	$G_v = (V_v, E_v)$
Failure/ stressor	Large disaster affecting multiple data-centers, network nodes, and network links. Represented as a connected (failure) graph of affected substrate nodes/links, can cause failure of multiple overlying client VN mappings	$G_u = (V_u, E_u)$
Augmented graph	Intermediate working graph used to compute VN node mappings. Defines additional meta-nodes/meta-edges over substrate network	$G_a = (V_a, E_a)$

3.1 Substrate Network

A substrate network is modeled as an undirected graph Gs=(Vs,Es) , where Vs is the set of substrate nodes and Es is the set of substrate links. Now each substrate node vs∈Vs can have different resource types, e.g., computing, storage. However (without loss of generality) to simplify the discussions, the capacity of nodal resources in vs is given by R(vs) and its unit cost as C(vs) . The available bandwidth of substrate link es∈Es is also given by B(es) and the unit cost of bandwidth by C(es). Additionally, substrate link es is also denoted as (vs,v's), where vs and v's are the two link end-points. Fig. 1a shows a sample 10-node substrate network where the numbers over the links (nodes) represent the available bandwidth (nodal resources) levels.

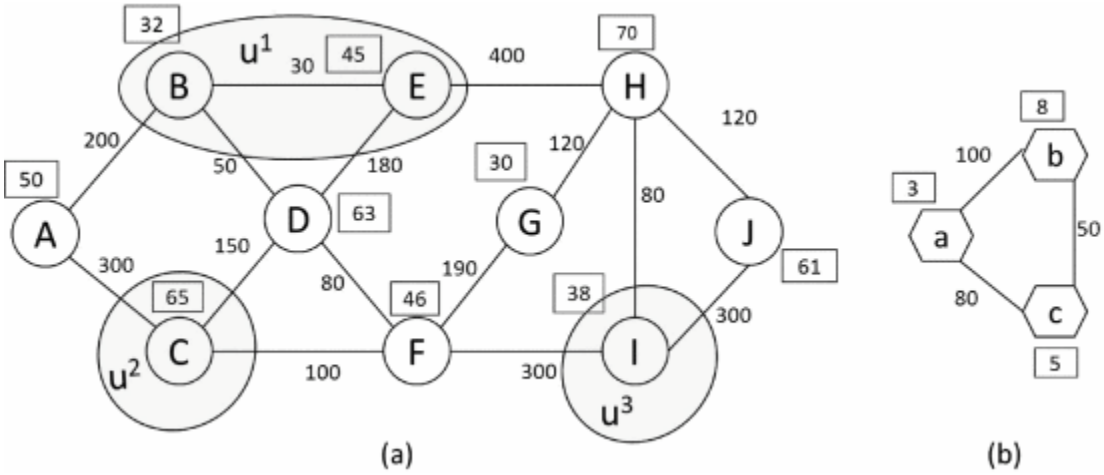

Fig. 1.

(a) substrate network, (b) sample VN request.

3.2 VN Request

A VN request is given by an undirected graph Gv=(Vv,Ev), where Vv is the set of VN nodes and Ev is the set of VN links. Here each VN node vv∈Vv requires a certain amount of nodal resources denoted as r(vv) and each VN link ev∈Ev also needs b(ev)bandwidth. Similar to a substrate link, a VN link ev is also denoted as (vv,v'v). Fig. 1b shows a sample VN request with its associated nodal resources and bandwidth requirements.

3.3 Regional Failure

In general it is safe to assume that there will be a finite number of failure regions within a network [9]. It is also realistic to assume that these regions are disjoint, i.e., only one failure stressor will occur at a given time. Hence these regions can be represented by the set U, with each failure region u∈U covering one or more substrate nodes, which in turn are denoted by Gu=(Vu,Eu), Vu⊆Vs and Eu⊆Es. Note that a substrate node failure will cause all adjacent substrate links of the node to fail. As an example, Fig. 2shows five failure regions in a 24-node network.

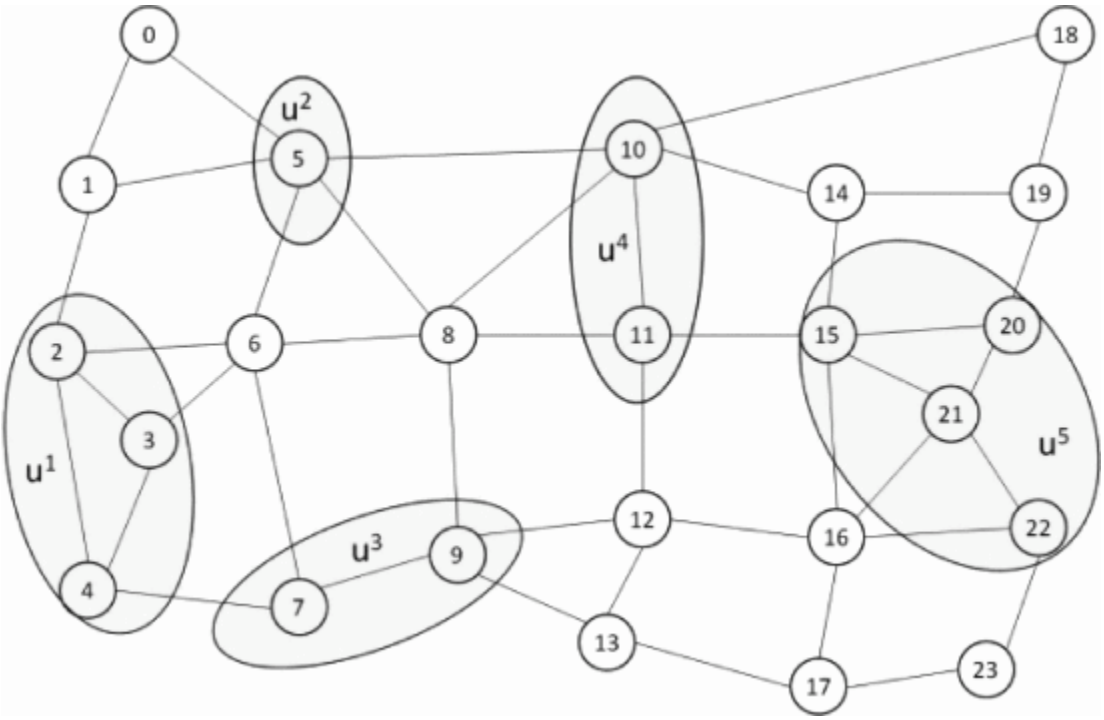

Fig. 2.

24-node substrate network with five failure regions, |U| = 5.

3.4 Survivable VN Mapping

In general, survivable VN designs must allocate nodal and link resources for both working and protection mapping(s) to handle failures. Now first consider the working mapping, where a VN request has to be placed over the substrate network by computing and allocating nodal resources for each VN node and bandwidth capacity for each VN link. In particular, VN nodes need to be mapped to different substrate nodes and VN links need to be mapped to connection paths over substrate network links (after the VN end-point nodes are mapped) to help improve robustness/survivability. Hence the VN node-to-substrate node mapping is denoted as <vv,vs> , i.e., VN node vv to substrate node vs.

Next consider protection mappings, which require backup resources to be allocated in advance for VN nodes and VN links. For example, if a VN node is mapped (in the working mapping) to a substrate node within a failure region, then another substrate node outside this failure region must be allocated to this VN node in the protection mapping. Since a regional stressor can cause multiple substrate node failures, multiple VN nodes can also fail. Hence a sufficient set of backup substrate nodes must be provisioned for disaster recovery. These backup nodes also require added VN links to connect to neighboring VN nodes in the protection mapping.

However, as neighboring VN nodes may also fall in the same failure region, backup VN links may end up connecting two backup VN nodes instead of a backup VN node and its neighboring VN node.

Since the above method poses many complexities, an alternate approach is proposed here by computing multiple VN mappings, with each avoiding a different failure region [19]. Furthermore, resource sharing can also be implemented between these mappings to help lower resource usages. This is shown more clearly in Fig. 3, where VN request G1vis assigned a working mapping { <a,A>,<b,B>,<c,C>} and a protection mapping { <a',C>,<b',D>,<c',F> }. Hence since the substrate link (C,D) is used by both the working and protection mappings, bandwidth sharing can be done. In addition, substrate node C is allocated to VN node c in the working mapping and to VN node a in the protection mapping. Again, nodal resource sharing can also be done here. Another sharing case is also seen for request G2v where a VN node/link is mapped to the same substrate node/path in both the working mapping and protection mappings.

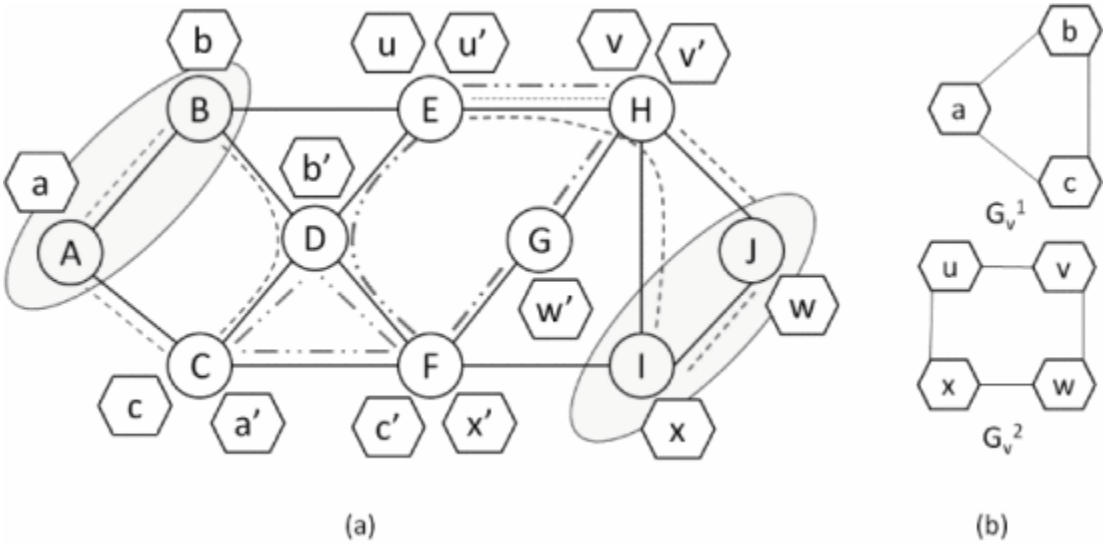

Fig. 3.

(a) substrate w. two failure regions. (b) two VN requests.

3.5 Objectives

Now most operators provisioning VN services will want to achieve a high level of resource efficiency over their underlying substrate network. In addition, associated revenue generation (cost reduction) concerns will also be important here [2], [7]. Along these lines various metrics are now detailed. First, the revenue associated with provisioning a VN request is formulated similar to [7] as:

$$REV(Gv)=\sum_{ev\in Evb}(ev)*I(ev)+\rho\sum_{vv\in Vvr}(vv)*I(vv),(1)$$

View Source where ρ is the fraction of nodal resource revenue, I(ev) is the revenue for a unit of bandwidth, and I(vv) is the revenue for a unit of nodal resource. Hence the cost of accepting a VN is:where π is the fraction of nodal resource cost, FGves is the total amount of bandwidth allocated on substrate link es for mapping the VN, and NGvvs is the total amount of nodal resources allocated on the substrate node vs for mapping the VN (andC(es) and C(vs) are introduced in Section 3.1). Note that this cost formulation is different from the non-survivable VN mapping in [7] since the bandwidth/nodal resource cost for a substrate link/node in Eq. (2) is not directly related to a VN link/node. Specifically, survivable VN mappings also require protection link bandwidth/nodal resources and resource sharing is also implemented. Hence the amount of link

bandwidth/nodal resources allocated on substrate links/nodes is determined only after the survivable mapping is computed.

Now from an operator's point of view, it is very desirable to increase long-term revenue, and this can be defined as:

$$\sum_i REV(Giv)T, \forall Giv \in A, \quad (2)$$

View Source where Giv is the *ith* VN request, A is the set of all accepted VN requests, and T is the total running time. Similarly, the long-term average cost is defined as:

$$\sum_i COST(Giv)T, \forall Giv \in A. \quad (3)$$

View Source However, to precisely describe operator profit, the net revenue is also computed as follows:

$$\sum_i (REV(Giv) - COST(Giv))T, \forall Giv \in A. \quad (4)$$

View Source Note that VN request blocking also provides a measure of lost revenue, and hence many operators will want to minimize this value to improve performance. Finally, protection switching times after failures (to backup VN nodes and links) will also be of concern to help minimize user service disruptions.

SECTION 4 Milp Formulation

A detailed optimization formulation for survivable VN mapping is now presented based upon the above notation.

4.1 Augmented Substrate Graph Construction

In order to formulate the survivable VN mapping problem, an augmented substrate graph $Ga=(Va,Ea)$ is first constructed, akin to [5], [19]. Namely, a meta-node vm is created for each VN node vv and connected to all underlying substrate nodes vs via meta-edges em with infinite bandwidth. Hence the augmented substrate graph is a combination of the original substrate graph and the meta-nodes/meta-edges, i.e., $Va=Vs \cup Vm$, $Ea=Es \cup Em=Es \cup \{(vm,vs) | vm \in Vm, vs \in Vs\}$. It is also assumed that regional failures do not affect any of the meta-nodes. However, meta-edges can fail if their underlying connecting substrate nodes fail. An example of an augmented graph is also shown in Fig. 4.

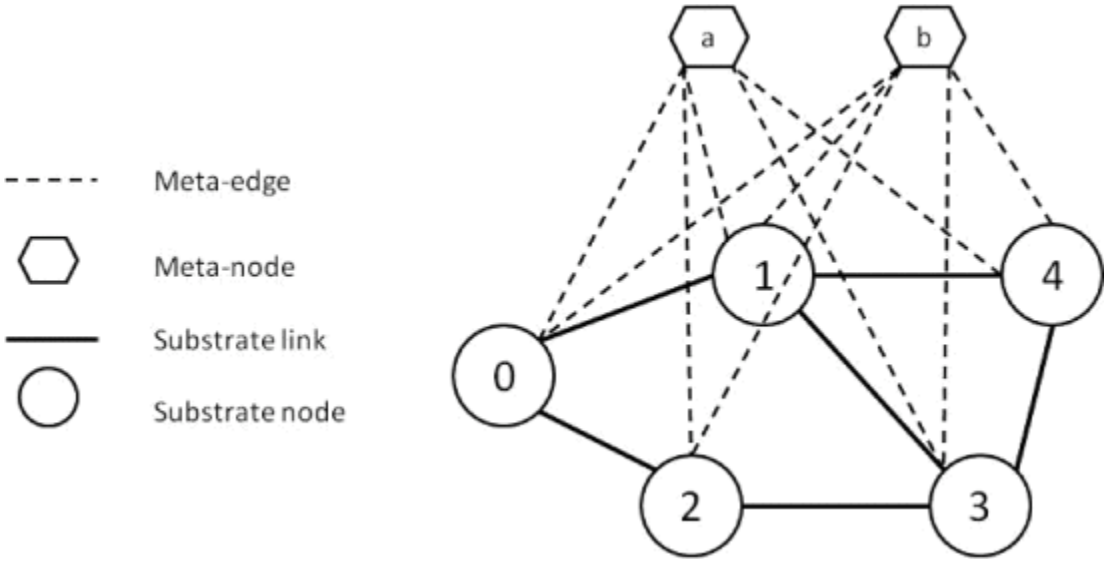

Fig. 4.

Example of augmented graph.

4.2 Failure Region Disjoint MILP Formulation

The failure region-disjoint survivable VN mapping is formulated as a mixed integer multi-commodity flow problem. Namely, each VN link ev is treated as a commodity with source and destination nodes s∈Vm and t∈Vm, i.e., flows start and end at distinct meta-nodes. By further restricting the meta-edges, each meta-node can be forced to choose a single meta-edge as its only connection to the substrate network. This effectively selects a substrate node for each meta-node, corresponding to its mapping. At the same time VN links must also be mapped over substrate network links. Since the goal is to generate failure region-disjoint working and protection mappings, added restrictions are needed to ensure that only one mapping is assigned to a failure regionu∈U. This is denoted by the index variable z∈Z={1,2}, i.e., z=1 for working and z=2 for protection. The detailed MILP formulation is now presented, i.e., consider the following:

Variables:

- $f_{q,zmn}$. Flow variable denoting total amount of flow from node m to node n on the substrate edge (m,n)∈Ea for the VN link q∈Ev of mapping z∈Z.
- α_{zmn}. Binary variable which is 1 if the flow in any of the VN links in mapping zuses the substrate edge (m,n), i.e., $\sum_{q\in E_v}(f_{q,zmn}+f_{q,znm})>0$; otherwise it is 0.
- ε_{zu}. Binary variable which is 1 if the mapping z can be mapped in failure regionu∈U; otherwise it is 0.
- be. Variable denoting the maximal amount of flow on the substrate link e∈Es.
- rn. Variable denoting the maximal amount of nodal resource allocated from the substrate node n∈Vs.

Objective:

$$\min \sum_{e\in E_s} b_e * C(e) + \sum_{n\in V_s} r_n * C(n). \quad (5)$$

Constraints:

$$\sum_{z\in Z} \varepsilon_{zu} \le 1, \forall u \in U \quad (6)$$

$$\alpha_{zmn} \le \varepsilon_{zu}, \forall (m,n) \in E_u, \forall z \in Z, \forall u \in U \quad (7)$$

$$\sum_{n\in V_a} \alpha_{zmn} \le A * \varepsilon_{zu}, \forall m \in V_u, \forall z \in Z, \forall u \in U \quad (8)$$

$$\sum_{q\in E_v} (f_{q,zmn}+f_{q,znm}) \le B(m,n) * \alpha_{zmn}, \forall (m,n) \in E_a, \forall z \in Z \quad (9)$$

$$r(m) * \alpha_{zmn} \le R(n), \forall m \in V_m, \forall n \in V_s, \forall z \in Z \quad (10)$$

$$\sum_{n\in V_a} f_{q,zmn} - \sum_{n\in V_a} f_{q,znm} = 0, \forall q \in E_v, \forall m \in V_a \setminus \{s_q, t_q\}, \forall z \in Z \quad (11)$$

$$\sum_{n\in V_a} f_{q,zs_qn} - \sum_{n\in V_a} f_{q,zns_q} = b(q), \forall q \in E_v, \forall z \in Z \quad (12)$$

$$\sum_{n\in V_a} f_{q,zt_qn} - \sum_{n\in V_a} f_{q,znt_q} = -b(q), \forall q \in E_v, \forall z \in Z \quad (13)$$

$$\sum_{n\in V_s} \alpha_{zmn} = 1, \forall m \in V_m, \forall z \in Z \quad (14)$$

$$\sum_{m\in V_m} \alpha_{zmn} \le 1, \forall n \in V_s, \forall z \in Z \quad (15)$$

$$\alpha_{zmn} = \alpha_{zmn}, \forall m,n \in V_a, \forall z \in Z \quad (16)$$

$$\sum_{q \in E_v}(f_{q,zmn}+f_{q,znm}) \leq b_e, \forall e \equiv (m,n) \in E_s, \forall z \in Z \quad (17)$$

$$\sum_{m \in V_m} \alpha_{zmn} * r(m) \leq r_n, \forall n \in V_s, \forall z \in Z \quad (18)$$

$$f_{q,zmn} \geq 0, \forall m,n \in V_a, \forall q \in E_v, \forall z \in Z \quad (19)$$

$$\alpha_{zmn} \in \{0,1\}, \forall m,n \in V_a, \forall z \in Z \quad (20)$$

$$\varepsilon_{zu} \in \{0,1\}, \forall z \in Z, \forall u \in U. \quad (21)$$

The objective function in Eq. (6) tries to minimize the cost of mapping a VN request. Meanwhile, Eq. (7) also constrains each failure region to be covered by at most one mapping. Meanwhile Eq. (8) pertains to link failures and ensures that if a link (m,n) is located in a failure region u, then αzmn can only be set to 1 if this mapping z can be mapped to failure region u. Similarly, Eq. (9) deals with node failures, and A is large constant (set greater than the maximal node degree in Ga). This constraint ensures that if a node m is located in a failure region u, then its adjacent links can only carry flows if the whole mapping z can be placed in failure region u. Next, Eqs. (10) and (11) bound link capacity and nodal resources, respectively, i.e., summing fq,zmn and fq,znm ensures that the total flow in both directions of an undirected link (m,n) is less than the available bandwidth. Meanwhile, Eqs. (12), (13), and (14) implement flow conservation. Furthermore, Eq. (15) ensures that only one substrate node is selected for a meta-node, whereas Eq. (16) ensures that a substrate node can only be allocated to at most one meta-node. Next, Eq. (17) ensures that αzmn is the same in both link directions. Also, Eqs. (18) and (19) compute the maximum resource allocations for each substrate link/node for each mapping z∈Z. Finally, Eqs. (20), (21), and (22) denote the non-negativity and binary constraints on fq,zmn , αzmn, εzu.

Overall, the MILP formulation has a total variable complexity of $4(|E_v|+1)(|E_s|+|V_v||V_s|)+2|U|+|E_s|+|V_s|$. This is a difficult problem to solve and is only tractable for smaller network sizes and/or failure region counts. For example, mapping a VN request with five nodes and 10 links over a substrate network with 20 nodes, 40 links, and five failure regions requires a total of 6,230 variables.

SECTION 5 Failure Region Disjoint Vn Mapping

Owing to the high computational complexity of the MILP model, some efficient heuristics algorithms are also proposed here. These schemes basically compute two separate failure region-disjoint VN mappings for each incoming request, i.e., working and protection VN mappings. Now since the survivable VN mapping algorithm is decomposed into two *non-survivable* mappings, any regular VN mapping scheme can be applied with this solution. As a result, for the purposes of this study, the *non-survivable virtual infrastructure mapping* (NSVIM) scheme from [19], [20] is chosen as a base VN mapping solution. In particular, this scheme uses a single-stage mapping approach to map a VN node and its attached VN links and has been shown to outperform the well-studied *Randomized Rounding Based Virtual Network Embedding* (R-ViNE) algorithm [5]. A brief overview of this scheme is now presented followed by complete details on the novel region-disjoint solution.

5.1 Overview of NSVIM Scheme

The NSVIM scheme [19], [20] basically maps a VN request over a substrate network without regard to failure regions. This algorithm uses a single-stage approach which maps each VN node to a substrate node and part of its attached VN links (to substrate paths) in the same step. The key objective here is to minimize the total mapping cost, and the NSVIM psuedocode description is shown in Fig. 5. Consider the details.

1: Set $MVN = \emptyset$, $ASN = \emptyset$, $UMVN = V_v$, $UASN = V_s$
2: Sort VN nodes in *UMVN* according to their node degree
3: Choose v_v with highest degree
4: Find candidate substrate nodal list $\mathbb{L}$ in *UASN* according to nodal resource and bandwidth restrictions described below. If $\mathbb{L} == \emptyset$, return FAIL
5: Assign v_v to v_s in $\mathbb{L}$ with min. cost computed by Eq. (23)
6: Reserve $r(v_v)$ nodal resource on substrate node v_s
7: For every VN link between v_v and $v'_v \in M - adj(v_v)$, compute minimum-cost path $\mathbb{P}$, where each link in $\mathbb{P}$ has available bandwidth greater than or equal to the requested $b((v_v, v'_v))$. If no such path is found, return FAIL, else reserve bandwidth along $\mathbb{P}$
8: Move v_v from *UMVN* to *MVN* and v_s from *UASN* to *ASN*
9: If $UMVN = \emptyset$, return SUCCESS, else go to step 2

Fig. 5.

NSVIM algorithm.

First, two sets are defined to track mapped and unmapped VN nodes, i.e., *MVN* and *UMVN* , respectively. In addition, two sets are also defined to track the allocated and unallocated substrate nodes, i.e., *ASN* and *UASN*, respectively. The NSVIM algorithm starts by initializing *MVN* and *ASN* as empty and setting *UMVN* equal to Vv and *UASN*is equal to Vs. Now in order to map a VN node vv∈ *UMVN*, a candidate substrate node list L is first built by selecting substrate nodes vs from *UASN* based upon two criteria. First, the available nodal resources R(vs) at a candidate node vs must be greater than or equal to the requested amount r(vv). Next, to ensure a feasible mapping, the maximum available bandwidth among all adjacent substrate links of vs must not be less than bmax(vv), i.e., where bmax(vv) is the maximum bandwidth requirement between vvand the set of all VN nodes that are adjacent to vv in Gv, further defined as the setadj(vv). Hence after L is constructed, a substrate node with minimum cost is selected from L as follows:

C(<vv,vs>)=C(vs)+AC(vs)+UAC(vs),(22)

View Source where C(vs) is the cost of nodal resource used in substrate node vs and AC(vs) is the communication cost from vs to a subset of substrate nodes in *ASN*, i.e., which are allocated for VN nodes belonging to the set M−adj(vv)=adj(vv)∩MVN. Next consider the details for computing the various terms in Eq. (23).

First, the nodal resource cost C(vs) is set to C(vs)∗r(vv) for mapping <vv,vs>. Next, in order to compute AC(vs), a set of substrate nodes allocated to M−adj(vv) is defined, i.e., *SCMA* . The minimum-cost path is then computed for each node pair (vs,v′s∈SCMA) using a shortest-path algorithm by setting link weights to their link costs, C(es),

which equals C(es)∗b(ev). The sum of link costs along a path is then assigned as the path cost. Overall, a total of |M−adj(vv)| paths are then computed and AC(vs) is determined as total sum of costs over all |M−adj(vv)| minimum-cost paths. Meanwhile, the third term in Eq. (23), UAC(vs), is defined as the estimated communication cost from vs to a subset of substrate nodes in *UASN*, which might be allocated to VN nodes in the set UM−adj(vv)=adj(vv)∩UMVN. This value can be computed as follows. First, for every VN link (vv,v'v∈UM−adj(vv)) , a related minimum-cost path is computed from v_s to as many v's∈UASN as possible. The average cost over these paths is then used as the estimated cost to map the VN link (vv,v'v). If no such path can be found, the cost is set as infinite and vs is not deemed a feasible mapping. Based upon this, UAC(vs) is computed as the sum of the average costs for all VN links between vv and v'v∈UM−adj(vv). Overall UAC(vs) is added to the formulation to provide a lookahead capability and prevent premature mappings to low cost substrate nodes/links, i.e., since NSVIM is a greedy scheme and earlier choices may result in higher overall cost.

5.2 Failure Region Group Based Mapping (FRGBM)

The proposed survivable FRGBM heuristic works by separating failure regions in U into two groups, G1 and G2, with the goal of clustering regions that are closer to each other. Although this sub-problem can itself be formulated as an optimization, a heuristic is used to simplify matters. The psuedocode for this solution is shown in Fig. 8 and details are now presented.

To obtain distances between each failure region, a failure region distance matrix is first computed, i.e., D={dij} where dij is the distance between regions ui and uj. Now to compute dij, the distance from node vs∈ui to failure region uj is defined as the average distance from vs to each substrate node in uj. The dij values are then calculated as the average of the distances between each node vs∈ui to failure region uj, Fig. 6. After the distance matrix D is computed, the next step divides the failure regions into two groups, G1 and G2, see Fig. 7. Initially, the two failure regions with the largest distance are determined and added to G1 and G2, respectively. The distances for all other non-grouped failure regions to G1 and G2 are then computed by averaging their distances to the existing grouped failure regions in the respective groups. These regions are then placed in the group with the shorter average distance.

```
1: for i=0 to |U|
2:     for j=0 to |U|
3:         dist = 0
4:         for each v_s^i ∈ u^i
5:             dist_i = 0
6:             for each v_s^j ∈ u^j
7:                 dist_ij=shortest path length from v_s^i to v_s^j
8:                 dist_i=dist_i+dist_ij
9:             dist_i=dist_i / (number of nodes in u^j)
10:            dist=dist+dist_i
11:    d_ij=dist / (number of nodes in u^i)
```

Fig. 6.

Algorithm of computing matrix D.

```
1: 𝔾¹ = ∅, 𝔾² = ∅
2: Find maximum d_ij in D
3: Add u^i into 𝔾¹ and u^j into 𝔾²
4: for i=0 to |U|
5:    if u^i ∉ 𝔾¹ and u^i ∉ 𝔾²
6:       dist1=0, dist2=0, num1=0, num2=0
7:       for each u^j ∈ 𝔾¹
8:          dist1=dist1+d_ij
9:          num1++
10:      dist1=dist1 / num1
11:      for each u^j ∈ 𝔾²
12:         dist2=dist2+d_ij
13:         num2++
14:      dist2=dist2 / num2
15:      if dist1<dist2
16:         add u^i to 𝔾¹
17:      else
18:         add u^i to 𝔾²
```

Fig. 7.

Failure region grouping algorithm.

The FRGBM scheme leverages the above information and is presented in Fig. 8 . Namely, this algorithm first prunes all failure regions in G1 and introduces a variation of the NSVIM algorithm in Section 5.1 to map the working VN over the substrate network, termed as *connectivity-aware NSVIM* (C-NSVIM). The G1 regions are then restored and those in G2 are pruned, and the C-NSVIM algorithm re-run to compute the protection VN mapping. Now the key difference between NSVIM and the modified C-NSVIM is an added constraint for computing the node list, L. Namely, the maximum connectivity of a candidate node is also checked in addition to checks for nodal resource and adjacent link bandwidth constraints, i.e., to avoid mapping VN nodes to areas isolated by pruned failure regions. For example consider the network in Fig. 2. If the failure regions u1 and u2 belong to the same group (G1 or G2) and are pruned during computation, then substrate nodes v0s and v1s will be not feasible for mapping any VN requests with three or more nodes, i.e., since v0s and v1s will become isolated after pruning. Hence if the maximum connectivity number, τ, of a substrate node is less than the number of VN nodes in the request, this node is not considered as a mapping candidate. Note that the connectivity of each substrate node after pruning G1 or G2 is fixed since these two sets are computed at startup. Additionally, substrate nodal and link bandwidth resource allocations are also shared

between the working and protection VN mappings since only one failure is assumed to occur at a given time. Hence in order to implement this feature, an added data structure is introduced to record substrate node resource and link bandwidth allocations for each mapping, i.e., Zi in Step 6,Fig. 8. Resource sharing is then done by selecting the maximum resource usage in the respective substrate node (link) amongst the two mappings, i.e., Step 8, Fig. 8.

1: Define temp. substrate graph $G_t(V_t, E_t)$ with V_t and E_t
2: **for** i=1 to 2
3: Copy current nodal/link resource from G_s to G_t
4: Prune each failure region $u^j \in \mathbb{G}^i$ in G_t
5: Run C-NSVIM on G_t to map Z_i, if failed return FAIL
6: Record substrate nodal/link resource allocation for Z_i
7: Restore each failure region $u^j \in \mathbb{G}^i$ in G_t
8: Assign nodal/link resource sharing between Z_1 and Z_2 according to records generated in Step 6 and compute final resource usage
9: Reserve nodal/link resource in G_s according to final resource usage computed in Step 8
10: Return SUCCESS

Fig. 8.

FRGBM algorithm.

Now consider the computational complexity of the FRGBM scheme. Foremost, the grouping algorithm in Fig. 7 has complexity O(|U|2) , i.e., as it loops over all failure regions. Similarly, the distance matrix computation algorithm in Fig. 6 has complexity of O(|Vs|2|Es|log|Vs|). Finally, the VN mapping procedure uses a modified C-NSVIM algorithm with complexity O(|Ev|(2+|Vs|)|Vs||Es|log|Vs|). Since the failure region grouping is static, the related algorithm only needs to be run once before start up, bounding the total *run-time* complexity by O(|Ev|(2+|Vs|)|Vs||Es|log|Vs|). By comparison, the SOUM and IOCM schemes have computation complexities of O(|U||Ev||Es||Vs|2log|Vs|) and O((|Ev||Vs|+|U||Vs|+|Ev|)|Vs||Es|log|Vs|), respectively. Hence the FRGBM scheme is notably faster than the SOUM scheme, i.e., by O(|U|). In addition, its complexity is as good as or lower than the IOCM scheme, i.e., by O(1+|U||Ev|).

5.3 Dynamic Failure Region Disjoint Mapping (DFRDM)

As noted above, the FRGBM scheme uses fixed failure region groupings, determined at startup. However since network substrate loads are dynamically-varying quantities, these groupings may lead to resource inefficiency. To address this concern a modified DFRDM scheme is proposed, Fig. 9.

1: Define temp. substrate graph $G_t(V_t, E_t)$ with V_t and E_t
2: Copy current nodal/link resource from G_s to G_t
3: Run NSVIM with penalty cost as described above on G_t, if failed return FAIL
4: Record substrate nodal/link resource allocation for Z_1
5: Copy current nodal/link resource from G_s to G_t
6: Prune failure regions accessed by Z_1 in G_t
7: Run C-NSVIM on G_t to map Z_2, if failed return FAIL
8: Record substrate nodal/link resource allocation for Z_2
9: Restore failure regions accessed by Z_1 in G_t
10: Assign nodal/link resource sharing between Z_1 and Z_2 according to records generated in Steps 4 and 8, compute final resource usage
11: Reserve nodal/link resource in G_s according to final resource usage computed in Step 10
12: Return SUCCESS

Fig. 9.

DFRDM algorithm.

The basic idea behind the dynamic approach is also to compute a working VN mapping and then prune all the failure regions used by this mapping before computing the protection mapping. However, since the working VN mapping may span a large number of failure regions, this approach will limit the resources available for the protection mapping. Hence to solve this problem, additional penalty costs are introduced to prevent the working VN mapping from covering too many failure regions. Specifically, a penalty cost P(vs) is added to C(vs) in Eq. (23) if a candidate node vs is located in a new failure region (from the regions already covered by the mapping). Similarly, a penalty cost P(es) is also added to AC(vs) in Eq. (23) if the path mapping for (vs,v's∈SCMA) traverses any new failure regions. Note that an added data structure is needed to record all accessed failure regions for a VN request, Fig. 9, and also a penalty is not applied to UAC(vs) since its value will depend upon all possible unmapped VN nodes (which may span all failure regions).

Next, the protection mapping is computed by running the C-NSVIM algorithm. Akin to the FRGBM scheme, nodal connectivity is also used to help select candidate nodes and avoid mapping nodes to isolated areas. Namely, a static connectivity status is computed for each substrate node for all possible failure region pruning scenarios in the initial stage. In particular, let the number 1 (0) denote a pruned (non-pruned) failure region. Using this representation, a failure region prune scenario can be expressed as a $|U|$ -dimensional vector binary $<au1,au2,\ldots,au|U|>$ and the total number of potential scenarios is given by $2|U|$. Now for each scenario, the number of substrate nodes that can be reached after pruning the corresponding failure regions, i.e., maximum connectivity number τ, can be computed for every substrate node via a breadth-first search. Hence by searching the pre-computed connectivity status, substrate nodes having a smaller τ value than the number of

VN nodes in a VN request will not be considered as candidates. Also note that connectivity information is only computed at initialization and hence does not affect run-time complexity, i.e., like FRGBM scheme (Section 5.2). As this step does not affect *run-time* complexity, this scheme also has the same complexity as the FRGBM scheme, i.e., O(|Ev|(2+|Vs|)|Vs||Es|log|Vs|).

5.4 Load Balancing

Overall, the FRGBM and DFRDM algorithms compute substrate link and node costs according to a pure *"mapping cost"* (MC). However, since these schemes ignore real-time nodal and link bandwidth resource loads at the substrate level, they will likely yield higher congestion at specific congested nodes/links. Hence, *load balancing* (LB) strategies can also be applied to alleviate such concerns. Namely, substrate link costs (weights) can be defined as inversely-proportional to load:

C(es)=BcB(es)+σ,(23)

View Source where Bc is the full capacity of a substrate link and σ is a small value (to avoid division errors). Nodal costs can also be defined as follows:

C(vs)=RcR(vs)+σ,(24)

View Source where Rc is the full resource capacity of a substrate node.

SECTION 6 Performance Evaluation

The performance of the survivable VN mapping schemes is tested using customized OPNETModelerTM models. Two substrate topologies are used here, including the small 10-node network with three failure regions in Fig. 1a (with modified nodal resource/link capacities) and the large 24-node network with five failure regions in Fig. 2 . Namely, all substrate nodes have 100 units of resource capacity and all substrate links have 10,000 units of bandwidth. VN requests are varied between 3-5 nodes and 4-7 nodes each for the 10- and 24-node topologies, respectively. Average VN topology node degrees are also set to 2.3 and 2.6 for these two network sizes. Meanwhile, requested VN node capacities are uniformly distributed between 1-10 units and requested VN link capacities are uniformly distributed between 50-1,000 units. Overall, these VN parameters are set to fractions of the capacity values in the physical substrate network topology, as most cloud services will likely cluster operation at a few key sites.

Next, all VN requests holding and inter-arrival times are generated using randomly-distributed exponential variables with means μ and λ, respectively. In particular, μ is fixed to 600 time units and λ is varied to achieve different input loads. Meanwhile, the load is measured using a modified Erlang metric that accounts for VN request sizes, i.e., by taking the product of average number of VN links and μ/λ. Finally, the *CPLEX 12.4*tool is also incorporated with OPNETModelerTM to solve the MILP formulation.

For comparison purposes, the SOUM and IOCM survivability schemes in [19] are also implemented. In addition, all schemes are tested using both the MC and LB cost assignment strategies (Section 5). Finally, all heuristics are evaluated for both topologies using 10,000 VN requests for the 10-node topology and 100,000 VN requests for the 24-node topology. However, the MILP scheme is only tested for the 10-node topology owing to high computational complexity. Failure events are also randomly triggered after an average of 1,000 incoming VN requests. Namely, a failure region is selected in random uniform manner and its nodes/links failed.

6.1 Blocking Rates

Initial tests are done to measure request blocking rates in Fig. 10. The findings for the 10-node topology show that the MC-based schemes yield very similar results, with the IOCM-MC and FRGBM-MC schemes giving slightly

lower blocking. However, both FRGBM-MC and DFRDM-MC also achieve significantly lower blocking in the larger 24-node topology, e.g., about 94 percent lower blocking with FRGBM-MC versus SOUM-MC at low-medium loads, Fig. 10b. Overall, the 24-node topology provides more substrate nodes and links and allows the failure region-disjoint schemes to use network resources more efficiently, i.e., versus SOUM and IOCM which exhaustively provision for every possible failure region. FRGBM-MC also gives lower blocking than DFRDM-MC except at heavy loads. The reason here is that the pure cost computation strategies do not use real-time network substrate load information (and hence cannot efficiently generate two disjoint failure regions).

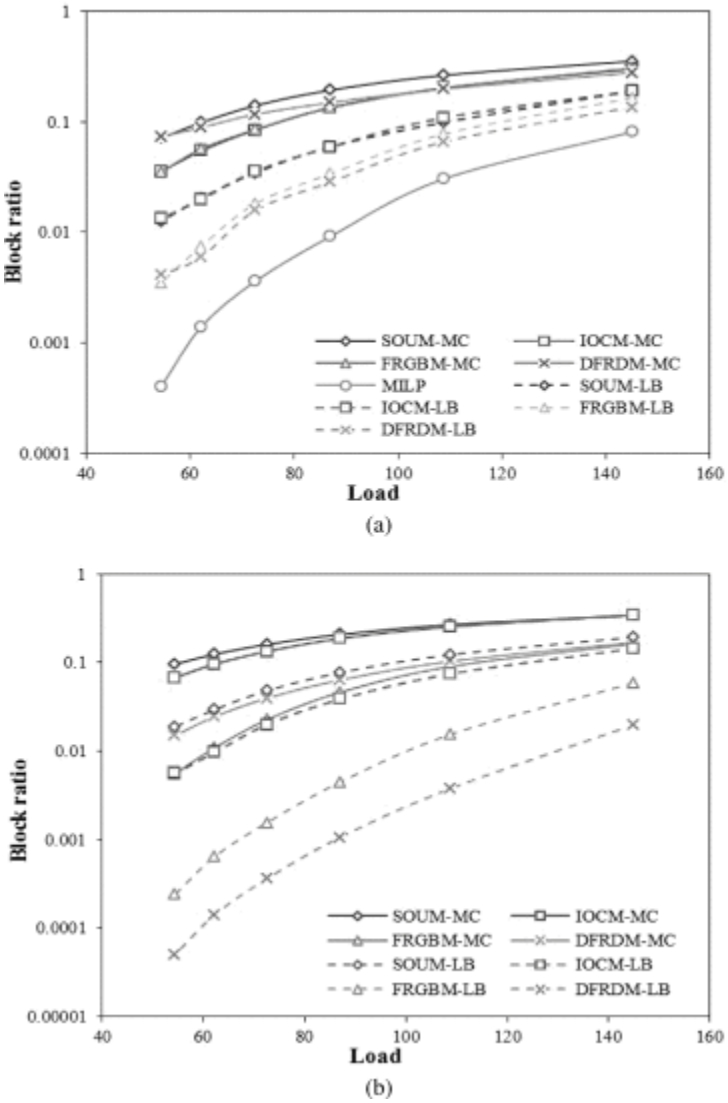

Fig. 10.

Blocking rate: a) 10-node topology. b) 24-node topology.

Additionally, the LB schemes yield notably lower blocking versus the MC variants for both topologies. These findings confirm that balancing VN loads across network substrate nodes/links can generally provide much better performance. However, the FRGBM-LB and DFRDM-LB schemes also outperform SOUM-LB and IOCM-LB, akin to the MC findings. DFRDM-LB is also seen to give lower blocking than FRGBM-LB (especially for the 24-node topology) since it performs dynamic mapping to disjoint failure regions (and can further leverage real-time

substrate load information). Finally, the MILP solution gives the lowest blocking results in the 10-node network, and this is expected since this is an optimal strategy. Moreover, since this solution also yields the best results for nearly all the other metrics evaluated (presented in the following sub-sections), the remainder of this paper only considers the heuristic VN mapping strategies.

6.2 Long-Term Revenue

The long term revenue results are also plotted in Fig. 11 and show that the FRGBM and DFRDM schemes achieve much higher revenue than SOUM and IOCM, i.e., for both the MC and LB variants. For example FRGBM-MC gives almost 44 percent more revenue than SOUM-MC in the 24-node topology. In addition the LB schemes provide a significant increase in revenue as well. Carefully note that revenue discrepancies also decrease with load since request blocking rates are reduced.

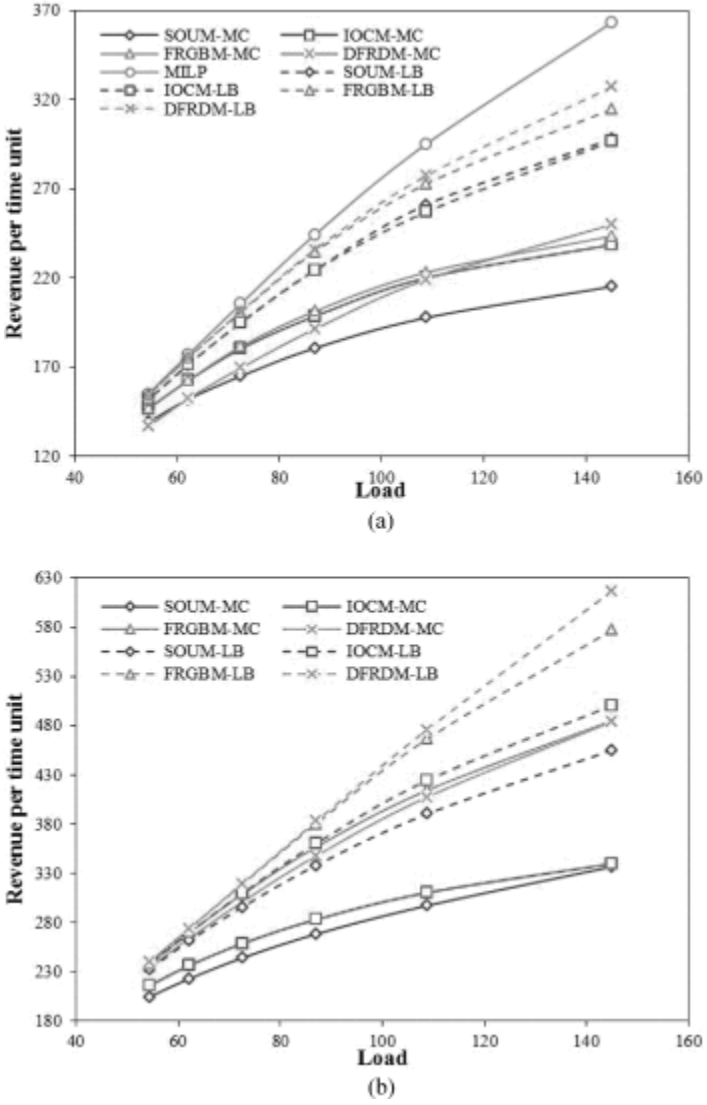

Fig. 11.

Long-term revenue: a) 10-node topology. b) 24-node topology.

6.3 Long-Term Cost

Long term cost is further plotted in Fig. 12. In comparison to the SOUM and IOCM strategies, the FRGBM and DFRDM schemes show slightly higher costs with the MC approach, but slightly lower costs with the LB approach. Also, all LB variants yield higher costs than their MC counterparts. This increase is due to the fact that the LB-based schemes accept more VN requests (lower blocking), which in turn leads to additional resource consumption and increased cost. In light of this, long term cost may not be a proper metric for evaluating VN mappings, i.e., lower costs can result from lower acceptance ratios. Hence the average cost per VN is evaluated next.

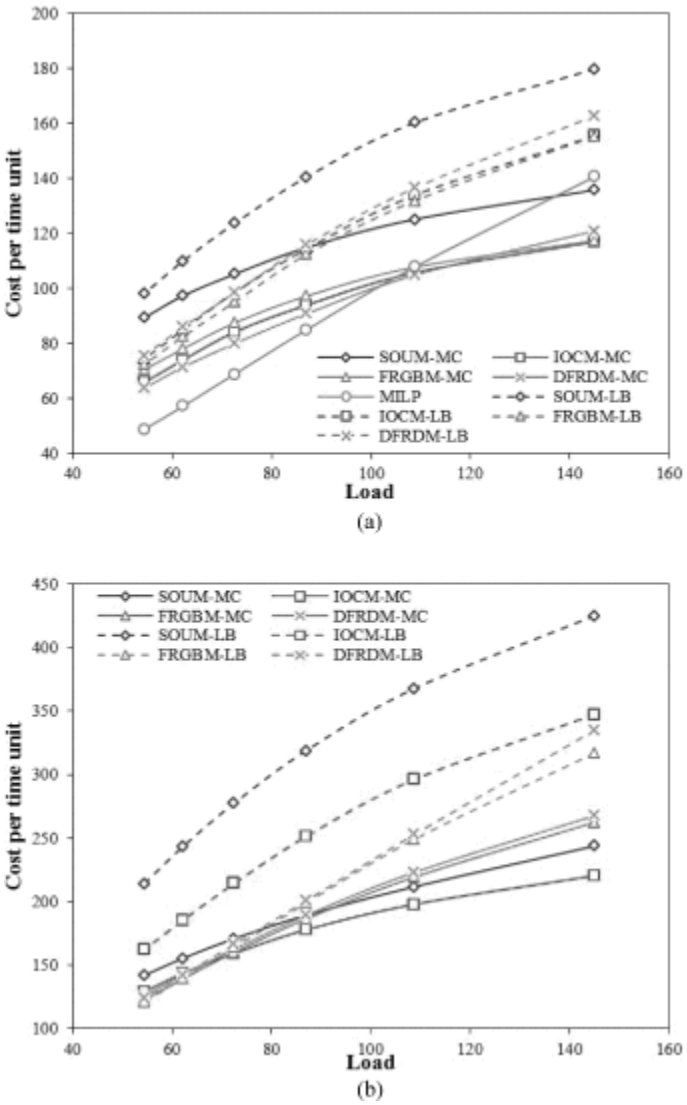

Fig. 12.

Long-term cost: a) 10-node topology. b) 24-node topology.

6.4 Average Cost

The average cost per VN is plotted in Fig. 13, and results show that both the FRGBM and DFRDM schemes achieve lower average cost than SOUM and IOCM (except for MC variants in the 10-node topology). The average cost under light traffic is also slightly higher than the cost under heavy traffic (except for MILP solution). The

reason here is that VN requests with large numbers of VN nodes/links are more likely to be accepted under lighter loads. Hence these larger VN sizes will use more resources and increase the average per VN cost. To show this more clearly, the average revenue per VN is plotted in Fig. 14, and results indicate that revenue is high when traffic is light and is low when traffic is heavy. Since the revenue for each VN is only related to the VN topology, this shows that larger VN requests are accepted more easily when traffic is low. These results also show that FRGBM and DFRDM can support larger VN sizes.

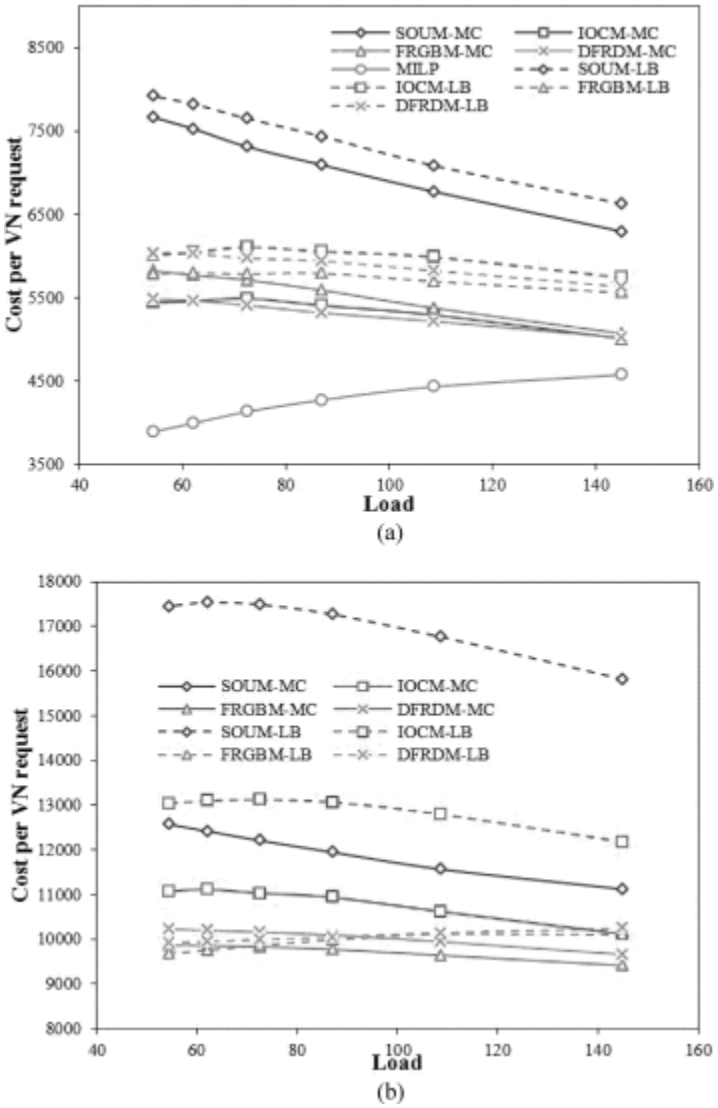

Fig. 13.

Avg. cost per VN: a) 10-node topology. b) 24-node topology.

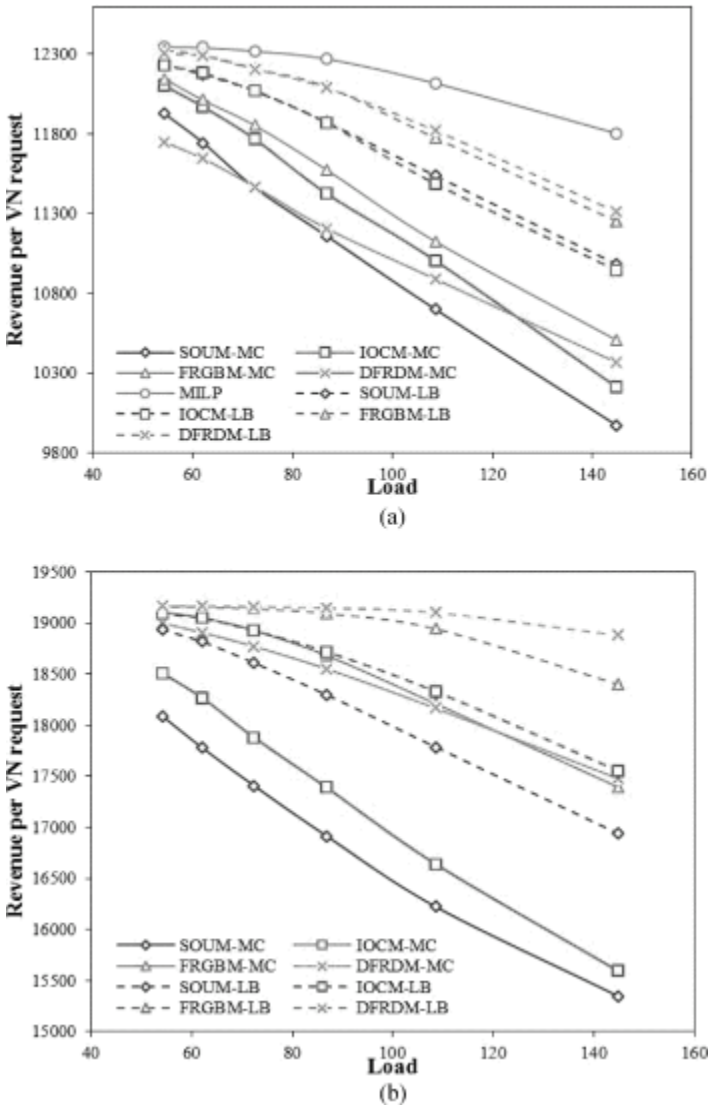

Fig. 14.

Revenue per VN: a) 10-node topology. b) 24-node topology.

6.5 Net Revenue

The net revenues from Eq. (5) are now plotted in Fig. 15 and clearly show that both the FRGBM and DFRDM schemes generate significantly-higher revenues than SOUM and IOCM (for both MC and LB variants). These results validate the gains of using failure region-disjoint mapping strategies. In addition, the LB variants generally give higher net revenues (than the MC-based variants) owing to their lower request blocking rates. The only exception here is the SOUM-LB scheme which gives rather low revenues, i.e., due to the fact that LB strategies tend to yield longer path routes for mapped VN links. Compounding this increase is the fact the SOUM scheme exhaustively computes multiple protection mappings for all failure regions, further increasing costs for larger failure regions. As noted earlier, net revenues also decrease with traffic load, i.e., akin toFig. 11.

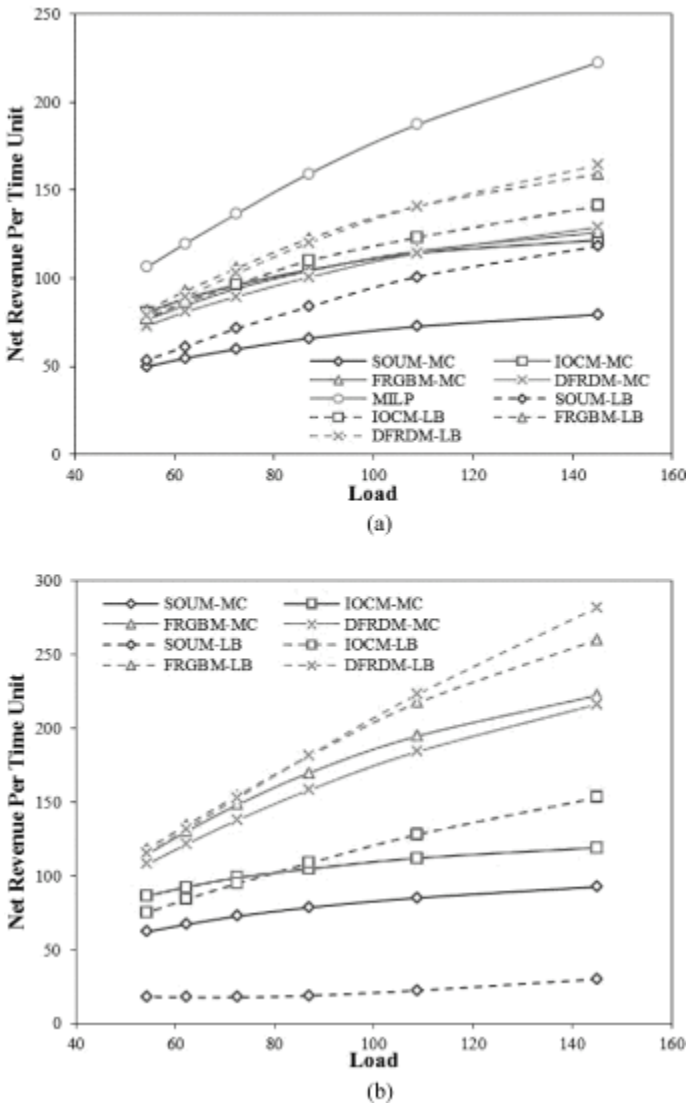

Fig. 15.

Net revenue: a) 10-node topology. b) 24-node topology.

6.6 Number of Failed VN and Total Number of Migrated Nodes

Finally, the number of failed VN requests is plotted in Fig. 16. The results here show that the FRGBM scheme is the most robust of all, as it groups failure regions and prevents the working mapping from spanning across too many regions. In addition, the total number of VN node migrations after failures are plotted in Fig. 17. Again, the FRGBM (and IOCM) scheme gives the least number of migrations since it yields the lowest number of VN failures.

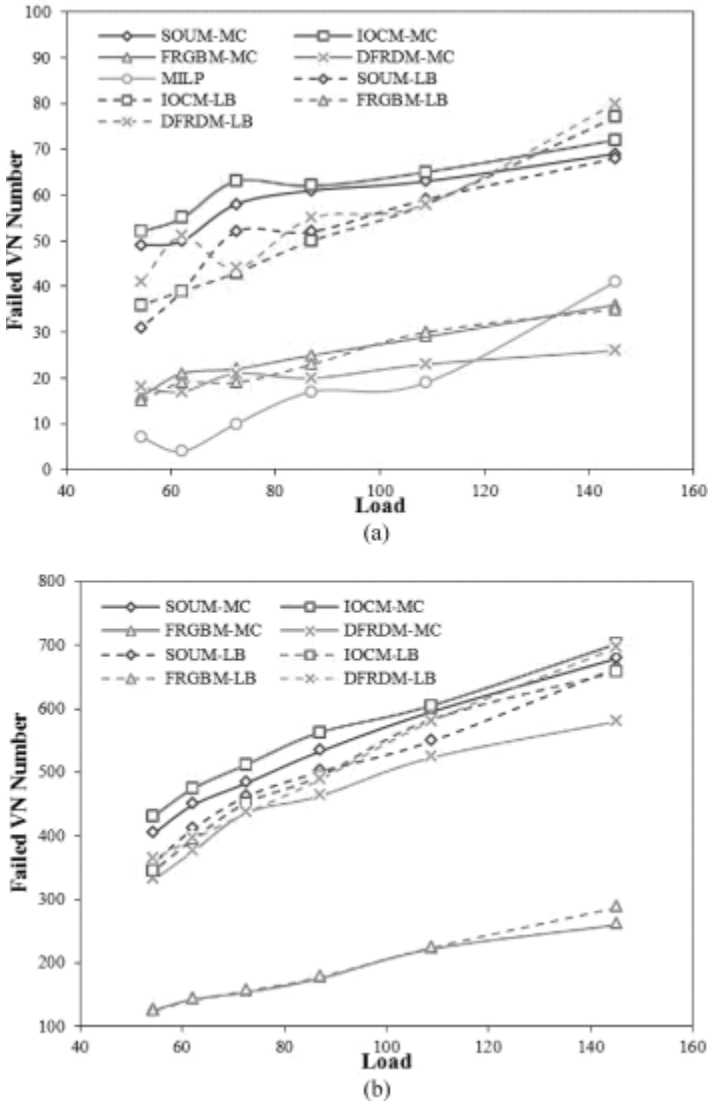

Fig. 16.

Num. failed VN: a) 10-node topology. b) 24-node topology.

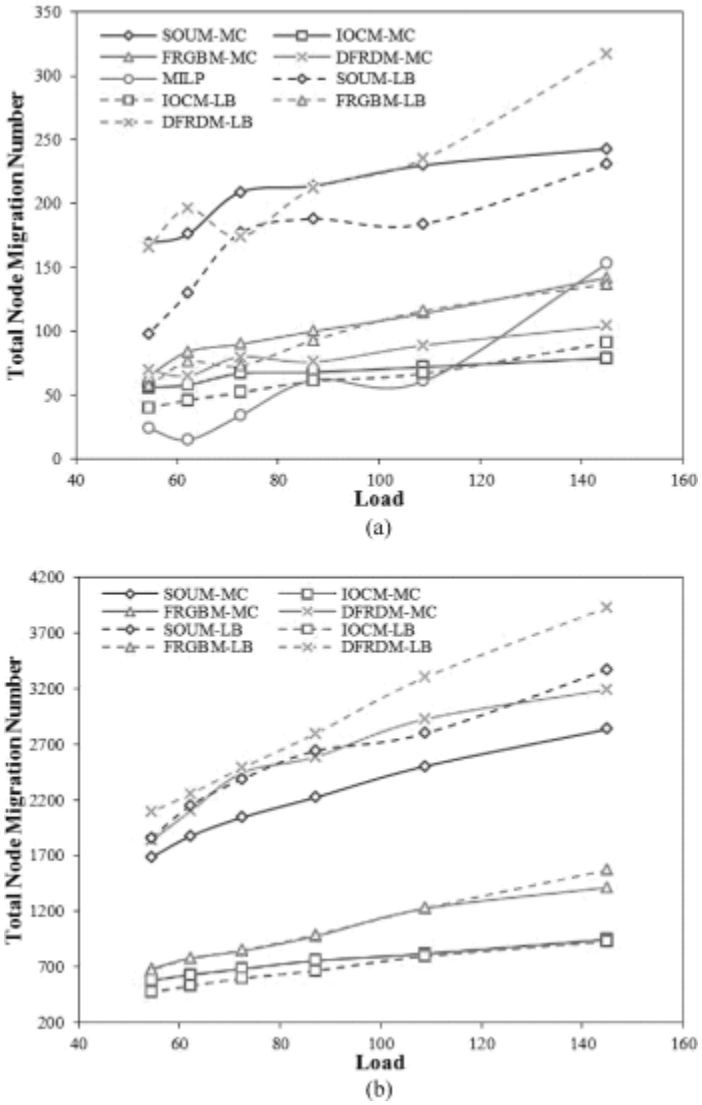

Fig. 17.

Num. migrated VN nodes: a) 10-node topology. b) 24-node topology.

SECTION 7 Conclusions And Future Work

This paper studies VN survivability for regional multi-failure events and proposes some novel failure region-disjoint solutions. An optimization formulation is first developed and two heuristic strategies are then proposed to compute efficient working and protection VN mappings. These latter strategies are also tailored to use both pure mapping cost and load-balancing information in the mapping process. Simulations show that the new failure region-disjoint schemes give lower blocking and higher (net) revenues as compared to existing survivable VN schemes, albeit with slightly higher long term costs. These gains are more pronounced with increased numbers of failure regions. In addition, the proposed heuristics also yield lower migration overheads during failure recovery. Finally, the results show substantially-higher revenues for all schemes when using load-balancing information, especially with the proposed failure region-disjoint heuristics.

Overall, this work provides a strong base from with to expand in additional, detailed efforts. Foremost, the formulation presented herein assumes that all failure events are equiprobable and does not consider their

probabilistic nature. Indeed, different regions will likely have varying levels of risk exposure, and hence it is important to develop further probabilistic VN mapping/survivability schemes to minimize failure rates. Furthermore, many regional disasters will likely induce further cascading (time-delayed) cyber-infrastructure faults, e.g., as arising from rolling power outages/blackouts. As a result, it is critical to incorporate the temporal nature of regional faults and develop multi-stage mappings for maintaining VN connectivity during such periods. This is an open problem area today.

ACKNOWLEDGMENTS

This work was made possible by the NPRP 5-137-2-045 grant from the Qatar National Research Fund (a member of Qatar Foundation). The statements made herein are solely the responsibility of the authors.

References

1. B. Rimal, "A taxonomy and survey of cloud computing systems", *Int. Joint Conf. INC IMS and IDC (NCM) 2009*, Aug. 2009.

2. G. Sun, "Optimal provisioning for virtual network request in cloud-based data centers", *Photonic Netw. Commun.*, Feb. 2012.

3. Y. Zhu, "Algorithms for assigning substrate network resources to virtual network components", *Proc. IEEE INFOCOM*, pp. 1-12, Apr. 2006.

4. M. Yu, "Rethinking virtual network embedding: Substrate support for path splitting and migration", *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 17-29, Apr. 2008.

5. N. Chowdhury, Apr. 2009.

6. J. Lischka, "A virtual network mapping algorithm based on subgraph isomorphism detection", *Proc. ACM SIGCOMM Workshop Virtualized Infrastructure Syst. Arch.*, pp. 81-88, Aug. 2009.

7. X. Cheng, "Virtual network embedding through topology-aware node ranking", *ACM SIGCOMM Comput. Commun. Rev.*, vol. 41, no. 2, pp. 38-47, Apr. 2011.

8. M. Chowdhury, R. Boutaba, "Network virtualization: State of the art and research challenges", *IEEE Commun. Mag.*, vol. 47, no. 7, pp. 20-26, Jul. 2009.

9. H. Lee, E. Modiano, and K. Lee, "Diverse routing in networks with probabilistic failures", *IEEE/ACM Trans. Netw.*, vol. 18, no. 6, pp. 1895-1907, Dec. 2010.

10. M. Rahnamay-Naeini, "Modeling stochastic correlated failures and their effects on network reliability", 2011.

11. H. Yu, "Virtual infrastructure design for surviving physical link failures", *Comput. J.*, Apr. 2012.

12. X. Zhang, C. Phillips, and X. Chen, “An overlay mapping model for achieving enhanced QoS and resilience performance,” presented at the Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops, Budapest, Hungary, Oct., 2011.

13. Y. Chen, Dec. 2010.

14. M. Rahman, May 2010.

15. T. Guo, Jun. 2011.

16. H. Yu, "Cost efficient design of survivable virtual infrastructure to recover from facility node failures", *IEEE ICC 2011*, Jun. 2011.

17. H. Yu, "Enhancing virtual infrastructure to survive facility node failures", *OFC 2011*, Mar. 2011.

18. C. Qiao, "A novel two-step approach to surviving facility failures", *OFC 2011*, Mar. 2011.

19. H. Yu, "Survivable virtual infrastructure mapping in a federated computing and networking system under single regional failures", *IEEE GLOBECOM 2010*, Dec. 2010.

20. G. Sun, "Efficient algorithms for survivable virtual network embedding", *Asia Commun. Photonics Conf. and Exhib. (ACP) 2010*, Dec. 2010.

21. E. Oki, "A disjoint path selection scheme with shared risk link groups in GMPLS networks", *IEEE Commun. Letters*, vol. 6, no. 9, pp. 406-408, Sep. 2002.

22. S. Neumayer, "Assessing the vulnerability of the fiber infrastructure to disasters", *IEEE INFOCOM 2009*, Apr. 2009.
23. P. Agarwal, "The resilience of WDM networks to probabilistic geographical failures", *IEEE INFOCOM 2011*, Apr. 2011.
24. S. Stefanakos, "Reliable routings in networks with generalized link failure events", *IEEE Trans. Net.*, vol. 16, no. 6, pp. 1331-1339, Dec. 2008.