Joint Mapping and Routing of Virtual Network Functions for Improved Disaster Recovery Support

D. Oliveira[1], J. Crichigno[2], N. Siasi[1], E. Bou-Harb[3], N. Ghani[1]
[1]*University of South Florida,* [2]*University of South Carolina,* [3]*Florida Atlantic University*

***Abstract*—Network function virtualization (NFV) and software-defined networking (SDN) are two recent networking paradigms that strive to increase manageability, scalability, programmability and dynamism. The former decouples network functions and hosting devices, while the latter decouples the data and control planes. However, as more operators adopt these new paradigms, there is a growing need to address multi-failure conditions, particularly those arising from large-scale disaster events. Hence this study proposes enhanced "risk-aware" virtual function mapping and traffic routing schemes to improve the reliability of user services as well as reduce deployment and routing costs.**

***Index Terms*—network function virtualization (NFV), disaster recovery, survivability**

I. INTRODUCTION

Carriers and service providers have traditionally used vendor-proprietary systems to build their network infrastructures and offer client services. These systems feature a high degree of hardware-software integration/coupling and offer carrier-grade availability, widely termed as "five 9's" or 99.999% (25 seconds of annual downtime) [5]. Nevertheless, these legacy systems are very expensive and constrained by complex upgrade cycles. These limitations present many challenges in today's fast-paced services markets, i.e., in terms of vendor dependency, cost and lack of service flexibility.

In light of the above, network designers have introduced new paradigms to achieve more rapid and flexible service provisioning capabilities. The key innovations here include *software defined networking* (SDN) and *network function virtualization* (NFV). Namely, SDN decouples the data and control planes, thereby facilitating streamlined data forwarding and programmable services. Meanwhile NFV virtualizes and replicates data plane functionality in software over dispersed commoditized servers, i.e., *virtual network functions* (VNF). This allows operators to rapidly (re)deploy network services by efficiently placing virtual functions and routing data flows between sites supporting these desired functions. As a result, capital and operational expenses (capex, opex) can be greatly reduced along with time to market. Overall, SDN and NFV technologies are seeing strong traction in the data-center and enterprise markets and even making strong inroads into larger metro-regional and core networks.

As NFV paradigms gain traction, researchers have started to address a range of provisioning concerns. Foremost, network operators have to design/adopt an orchestration solution capable of interfacing the virtual functions with the underlying virtual infrastructure, i.e., *virtual network function infrastructure* (NFVI). Next, efficient VNF placement schemes are also needed to efficiently place VNFs, or simply *virtual functions*, to reduce costs and consequently increase revenue. Along these lines, various studies have looked at NFV *management and orchestration* (MANO) and the VNF location/placement problem. In addition, VNF redeployment schemes have also been studied in order to investigate redeployment times and control procedures [3].

However, as NFV adoption grows, service reliability is becoming a major concern [5]. Indeed many clients will still expect a high degree of reliability despite their services being provisioned over commodity-based systems. Nevertheless, survivable NFV provisioning has not received much attention to date. For example, there are only a handful of related studies on single node and link failures [1]-[7]. In general, these methods are largely inadequate for handling large disaster-type events which can yield multiple system failures with high levels of temporal and spatial correlation [8]. Hence the distribution of NFV-based services across dispersed infrastructures offers much promise for improving service reliability here.

Overall, disaster recovery has been well-studied within the context of regular connection routing [9],[10] and also *virutal network embedding* (VNE) [11]-[13]. Most of these efforts leverage detailed a-priori stochastic multi-failure models to characterize large-scale disasters and then develop a range of provisioning and backup methodologies to minimize failure risk. Nevertheless there are no known studies on NFV provisioning for disaster scenarios. Hence this effort leverages existing a-priori multi-failure risk models to develop more resilient NFV provisioning strategies. In particular, a novel *integer linear programming* (ILP) optimization scheme is developed to improve resiliency and also achieve a proper tradeoff with underlying costs, i.e., including methods based upon resource usage minimization, routing cost minimization, failure risk minimization, and hybrid schemes. Additionally, a greedy heuristic scheme is also presented as a complementary method for large-scale networks.

This paper is organized as follows. First, a background review of existing work in VNF placement and routing is presented in Section II, including a look at the latest NFV survivability schemes. Subsequently, Section III presents the overall notation and stochastic multi-failure disaster model. A detailed optimization formulation for "risk-aware" VNF provisioning is then presented in Section IV, followed by the greedy heuristic scheme. Finally, Section V presents some detailed performance results as well as future directions.

978-1-5386-6133-8/18/$31.00

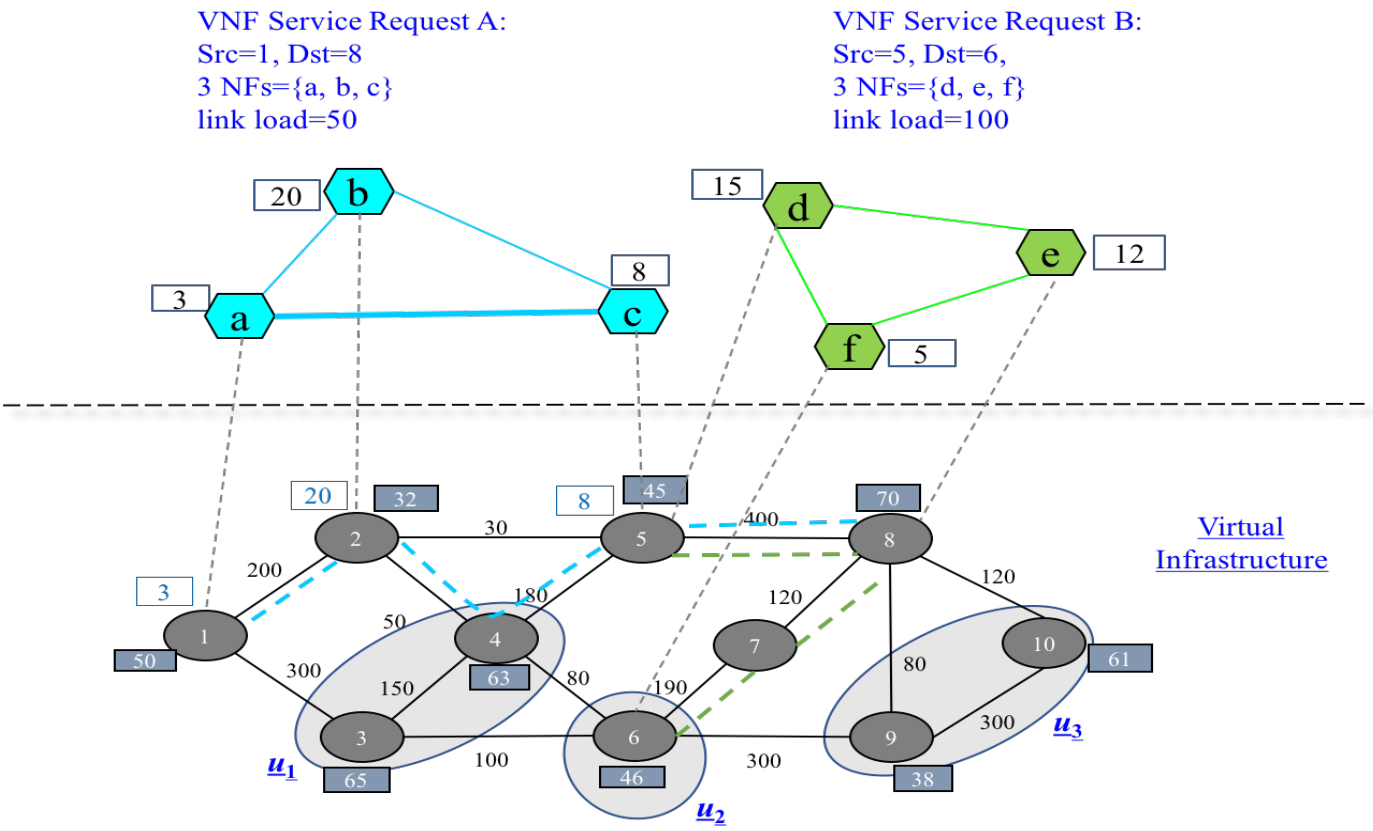


Fig. 1: Overview of VNF mapping

II. BACKGROUND SURVEY

The overall topic of NFV provisioining/design has received notable attention in recent years. In particular, a wide range of studies have already been done, including [2], [14], [15], [16], [18] and [19]. For example, [15] presents an ILP scheme to minimize VNF deployment costs using multiple approximation techniques. However, routing costs and bandwidth capacity are not taken into consideration. Meanwhile, the authors in [19] also study the joint placement of functions and routing of services with *traffic engineering* (TE) considerations. In particular, a detailed optimization model is presented along with a greedy heuristic strategy to maximize the number of satisfied VNFs (while minimizing VNF setup and routing costs). Load balancing is also implemented here. Overall results show that both methods increase the number of satisfied VNFs without overloading link capacity.

Recent studies have also looked at survivability concerns for VNF mapping and routing, albeit for isolated single node and link failures only. For example, in [1] the authors consider the interdependency between VNFs, i.e., *service function chaining* (SFC), and the fact that the failure of a single datacenter can affect the whole end-to-end service. Hence a novel orchestrator is proposed to dynamically redefine flows, steering traffic to establish new paths and reduce downtime. Although an extended orchestration architecture is proposed, the authors do not consider pre-provisioning metrics, i.e., probabilistic failures, resources and bandwidth requirements, etc. Furthermore, the authors only consider rerouting capabilities for disaster recovery. Meanwhile [2] proposes a resilient allocation scheme for SFCs. First the authors implement a greedy heuristic to successfully map the *virtual network functions forwarding graph* (VNF-FG), i.e., by considering resource allocation constraints and VNFs interdependence. Next, a link-disjoint backup path is set up between neighboring datacenters as well as backup nodes for each VNF host. However the VNF resource allocation backtracking scheme is very time consuming.

Meanwhile, [3] focuses on heuristic-based schemes for *joint* placement/combination of functions (topology design) and routing of demands (service chains) to minimize bandwidth usage. The authors also consider reliability concerns by assuming the availability of node (cloud) and link availability metrics. Namely, several heuristics are developed to improve overall service chain reliability for single failures, i.e., by computing node and link disjoint backup service chains. In particular, two methods are proposed here, dedicated and shared, with the latter allowing backup resource multiplexing between backups for service chains without any overlaps, i.e., as done for regular shared connection protection strategies [9]. Results show a clear tradeoff between reliability and efficiency, with dedicated protection giving higher (lower) reliability (efficiency) than shared protection.

Meanwhile, [4] proposes a greedy VNF provisioning heuristic that evaluates the probabilistic availability of resources (VNFs instances, servers, links and clusters). This

value is computed as the product of the availability of all resources satisfying the request. Although a-priori probabilities are considered here, the authors assume that enough physical resources are available. Additionally, routing and load balancing concerns are not addressed, which plays a critical role in VNF placement, see [19]. Similarly, [6] proposes a greedy heuristic to compute the actual temporal availability of physical resources, which is then used to define the probabilistic availability of a path. Furthermore, [5] presents a high-level look at VNF resiliency and outlines a range of requirements, i.e., failure management, state management/synchronization, VNF migration, and even handling of larger correlated failures. Finally, [7] introduces a heuristic scheme to efficiently place primary and redundant/backup VNFs since most other efforts do not analyze the availability of backup deployments. Here the authors assign a reliable SFC by computing the cost of all physical resources within that SFC.

Researchers have also looked at network survivability for large-scale disasters with multiple near-simultaneous and correlated failures. For example, [9] introduces a realistic stochastic failure model, i.e., *probabilistic shared risk link group* (p-SRLG), to specify a-priori failure risk regions with conditional node and link failure probabilities. Novel optimization and heuristic algorithms are then designed and evaluated to minimize joint primary/backup connection route failures. However these schemes do not account for resource efficiency concerns and can give longer detoured risk-averse routes. Hence the authors in [10] propose a further heuristic to incorporate additional load-balancing concerns with reliability. Recent efforts have also looked at disaster recovery for *virtual network* (VN) services using a-priori failure risk region models. For example, [11] and [12] proposes two heuristics for backup virtual node/link provisioning, i.e., termed as supplemental and incremental. The former computes mappings for each risk region and then combines them by using resource sharing on common nodes and links. Meanwhile the latter adds backup virtual nodes/links as needed to a base mapping in order to protect against each risk region. However these schemes can yield high resource inefficiency, and hence [13] proposes improved strategies to group the failure regions.

Despite these contributions, there are still no known studies on disaster-aware provisioning for multiple failures in NFV-capable infrastructures. This is a major concern since these events can cause widespread service disruption due to the high levels of (virtual function) multiplexing in NFV-based setups. In light of the above, there is a pressing need to study NFV provisioning within the context of large-scale multi-failure disasters, i.e., network function placement and routing. These solutions should incorporate known risk (vulnerability) regions as well as resource efficiency concerns. Some new solutions in this problems space are now detailed.

III. NOTATION OVERVIEW & FAILURE MODEL

The overall notation is first presented to specify the physical network, NFV-related demand models, and multi-failure outage model (see also Fig. 1).

A. Physical Network

The physical network is denoted by graph G=(V, E),where V is the set of nodes and E the set of links. Each link $(i, j) \in E$ also has an associated cost c^{ij} and bandwidth capacity b^{ij}, which quantifies the cost of routing through that link and the link capacity, respectively. Also the subset $D \subseteq V$ represents the set of datacenters where NFs are implemented and can be provisioned, and the set of all possible NFs is also denoted by F. Hence a given datacenter $d \in D$ implements a subset of functions $F_d \subseteq F$. The customizable number of resource types is also denoted by the integer m. For example, m=3 can refer to processor, storage and memory resources. It is also assumed that a datacenter $d \in D$ has a finite amount of resources $W_d = \{w_{d,1}, w_{d,2}, ..., w_{d,m}\}$. Hence in order to implement a function $i \in F_d$, datacenter $d \in D$ uses $w^i_{d,1}, w^i_{d,2}, ..., w^i_{d,m}$ resources. This resource requirement is datacenter-dependent, which mirrors the fact that some datacenters may specialize in implementing certain functions. Also, the setup cost of locating an instance of a function $i \in F_d$ at datacenter d is c^i_d, and an instance of function i at datacenter d can serve λ^i_d requests. In order to accommodate more requests (defined in Section III-B), multiple instances of function i can also be deployed at datacenter d. However each instance here will consume additional resources and entails added setup cost.

B. NFV Demand Model

Now let R be the set of requests arriving from clients. Here each request $r \in R$ is characterized by a 4-tuple format (src_r, dst_r, F_r, b_r), which denotes the source and destination nodes of the flow, the set of requested functions $F_r \subseteq F$, and the minimum required bandwidth capacity. Fig. 1 shows a sample infrastructure with two requests, Request A and Request B. Here, Request A defines datacenter d_1 as its source and datacenter d_8 as its destination, and requests 3 NFs, i.e., a, b, c. Additionally, a minimum bandwidth availability of b_r=50 is also requested. Similarly, Request B has the following parameters: $src_r = d_5$, $dst_r = d_6$, $F_r = d, e, f$ and b_r=100. The VNF placement scheme then focuses on defining the best placement locations according to the objective function.

Now the overall cost of provisioning all functions F_r associated with request r along a path P_r is given by:

$$d(\mathbf{P_r}) = \sum_{d \in D} \sum_{i \in F_r} c^i_d y^i_d \tag{1}$$

Similarly, the total routing cost for this path is also given by:

$$c(\mathbf{P_r}) = \sum_{r \in R} \sum_{(i,j) \in E} c^{ij} l^{ij}_r \tag{2}$$

C. Multi-Failure Outage Model

A realistic *probabilistic* model is used to specify large-scale disaster events with multiple highly-correlated spatial and temporal link failures. Namely, the set $\mathbf{U}$ defines an a-priori set of outage events, $\mathbf{U}$=$\{u_1, u_2, ..., u_N\}$, where each event u_n has an associated occurrence probability, $p(u_n)$. As per [12],[13], it is also assumed that all events are

sufficiently rare and therefore can be treated as independent and mutually-exclusive, $\sum_{\forall u_n \in \mathbf{U}} p(u_n)$=1. Without loss of generality, it is assumed that all outages are non-overlapping in the geographic domain. Now each event has an associated set of vulnerable links, termed as the *shared risk link group* (SRLG). A non-conditional failure probability $\omega(i,j)$ is also defined for each physical link $(i,j) \in \mathbf{E}$ in the region of event u_n (with respect to the occurrence of event u_n). Note that this framework can also be extended for conditional failure probabilities with overlapping risk regions.

IV. RELIABLE PROVISIONING STRATEGIES

As noted in Section II, to the best of the authors' knowledge, there are no known studies on VNF placement under multiple correlated failures, i.e., to minimize provisioning costs, increase survivability, achieve load balancing, etc. Accordingly, a pre-fault "risk-aware" ILP optimization model is proposed here to reduce failure downtime and maximize the number of satisfied requests, termed as the *risk-aware ILP* (RA-ILP) scheme. These events can include natural disasters, *weapon of mass destruction* (WMD) attacks or cascading power outages. In addition, a greedy heuristic scheme is also presented to overcome some of the scalability limitations of the optimization model, termed as *risk-aware greedy heuristic* (RA-GH) scheme. Furthermore, a "non-risk-aware" ILP with all link failure probabilities set to zero is also presented for comparison reasons, termed as *joint routing and placement ILP* (JRP-ILP). Finally, a "non-risk-aware" greedy heuristic scheme is also implemented, termed as *joint routing and placement greedy heuristic* (JRP-GH).

A. Optimization Model

A novel disaster-aware ILP optimization scheme (RA-ILP) is proposed here. Foremost, consider some additional variable definitions:

$$x^i_{r,d} \quad \in \quad \{0,1\} \quad r \in R, i \in F_r, d \in D | i \in F_d \tag{3}$$

$$y^i_d \quad \in \quad \mathbb{Z}^+ \quad d \in D, i \in F_d \tag{4}$$

$$l^{i,j}_r \quad \in \quad \{0,1\} \quad r \in R, (i,j) \in E \tag{5}$$

$$0 \quad \leq \quad \alpha \quad \leq 1 \tag{6}$$

Namely, $x^i_{r,d}$ is a binary value which indicates whether function $i \in F_r$ requested by request $r \in R$ is implemented at datacenter $d \in D$ or not. Meanwhile y^i represents the number of instances of function $i \in F$ at node $d \in D$. Also $l^{i,j}_r$ is also binary and indicates whether link $(i,j) \in E$ is used to route the traffic flow for request $r \in R$. Finally, α represents the highest link usage ratio, i.e., sum of all b_r using a link $(i,j) \in E$ divided by link capacity $b_{i,j}$. The overall objective function is given by:

$$\begin{aligned} \text{max F} \quad = \quad & w_1 \sum_{r \in R} \sum_{i \in F_r} \sum_{d \in D | i \in F_d} x^i_{r,d} - w_2 \sum_{d \in D} \sum_{i \in F_d} c^i_d y^i_d \\ & - w_3 \sum_{r \in R} \sum_{(i,j) \in E} c^{ij} l^{ij}_r * (1 + \omega(i,j)) - \alpha w_4 \end{aligned} \tag{7}$$

The above model tries to maximize the total number of satisfied NFs and is composed of 4 terms multiplied by respective weight factors, w_1, w_2, w_3 and w_4 (for establishing a tradeoff between deployment and routing costs). Namely, the first term above represents the total number of satisfied NFs, and the main goal is to maximize this value. Meanwhile the total cost of deploying NFs instances and accepting requests at the various datacenters is given by the second term. Similarly, the third term is the total routing cost. However note that the link failure probabilities $\omega(i,j)$ are also incorporated here. Namely, the routing cost can be increased proportional to the link failure probability. Meanwhile the fourth term represents the maximum overall link load. Note that the second, third and fourth terms are all negative since maximizing a negative term is equivalent to minimizing it [19].

Next, Eqs. 8 to 15 specify the required constraints for the above optimization model:

$$0 \quad \geq \quad \omega(i,j) \quad \leq 100 \tag{8}$$

$$\sum_{d \in D} x^i_{r,d} \quad \leq \quad 1 \quad r \in R, i \in F_r \tag{9}$$

$$x^i_{r,d} \quad \leq \quad y^i_d \quad r \in R, i \in F_r, d \in D | i \in F_d \tag{10}$$

$$\sum_{i \in F_d} w^i_{d,j} y^i_d \quad \leq \quad w_{d,j} \quad d \in D, r \in R, j \in \{1,2,...,m\} \tag{11}$$

$$\sum_{r \in R} x^i_{r,d} \quad \leq \quad \lambda^i_d y^i_d \quad d \in D, i \in F_d \tag{12}$$

$$\sum_{j:(i,j) \in E} l^{ij}_r - \sum_{j:(j,i) \in E} l^{ji}_r = \begin{cases} -1; i = dst_r, src_r \neq dst_r \\ 1; i = src_r, src_r \neq dst_r \\ 0; \text{ otherwise. } i \in V, r \in R \end{cases} \tag{13}$$

$$\sum_{(d,j) \in E} l^{dj}_r \quad \geq \quad x^i_{r,d} \quad r \in R, i \in F_r, d \in D | i \in F_d \tag{14}$$

$$\sum_{r \in R} l^{i,j}_r b_r \quad \leq \quad \alpha b_{i,j} \quad \{i,j\} \in E \tag{15}$$

In particular, Constraint 8 ensures that the percentage failure probability assigned to a specific link ranges between 0-100, while Constraint 9 ensures that a function i requested by request r is serviced by at most one datacenter d. Since the objective of the ILP is to maximize the sum of all variables

$x^i_{r,d}$, i.e., first term of Eq. 7, the optimal solution will drive Constraint 9 to equality. Meanwhile Constraint 10 assures that function i is located at datacenter d in case request r is assigned to function i at datacenter d. Constraint 11 also states that the overall amount of type j resources used by all functions instantiated at datacenter d is bounded by the total amount of resources $w_{d,j} \in W_d, j \in \{1, 2, ..., m\}$. Similarly, Constraint 12 indicates that the total number of requests for function i served by datacenter d is at most the number of instances of i at d times the capacity λ^i_d of an instance i. Also, Constraint 13 represents flow conservation, and Constraint 14 guarantees that if a function i requested by request r is placed at datacenter d, then traffic flows will be routed through that datacenter. Finally, Constraint 15 guarantees that the sum of all links (i, j) used by a traffic flow associated with request r times the load b_r demanded by request r is bounded by the product of the link capacity $b_{i,j}$ times α. As a result, traffic flows are mapped to links that have little/no load profile (higher available capacity), i.e., load balancing. This is critical for the fourth term of the objective function, which also avoids/minimizes link overload by introducing a link load minimization function variable, α.

B. ILP Complexity Analysis

Since the objective function and constraints are linear, and the variables are all integral, the proposed algorithm is NP-hard [15]. Now the complexity of both ILP schemes can be determined by the number of variables that they utilize. Namely, Constraints 3, 9 and 11 dominate the upper-bound for the total number of variables, defined by the product $|R||F||D|$, i.e., $O(|R||F||D|)$. Now it is generally difficult to specify limits on the number of requests (or number of functions) beforehand to ensure ILP convergence. However for small-to-medium network topologies the proposed ILP can still be solved in a reasonable amount of time, i.e., less than a few hundred nodes.

C. Heuristic Strategies

The exponential time complexity introduced by the ILP optimization is a concern for complex scenarios with a large number of variables, i.e., many datacenters, links, requests, functions. As a result, polynomial-time heuristic schemes can provide timely but sub-optimal solutions. Now many heuristic schemes are available, i.e., genetic algorithms, particle swarm optimization, simulated annealing, greedy heuristics, etc. However, the latter approach is chosen here due to its low complexity and its usage in most current efforts, see [2], [4], [6]. Accordingly, a new greedy heuristic scheme that solves the risk-aware NF placement problem for larger networks is shown in Fig. 2.

Overall, the proposed algorithm implements a two-stage solution. Namely, given a graph and input parameters (akin to RA-ILP scheme; source, destination, set of NFs and minimum link capacity), the algorithm returns values for the variables x^i, r, d, y^i_d, $l^{i,j}_r$ and α. Namely the First Stage (lines 4-13) in Fig. 2 shows how the algorithm places the NFs at datacenters to reduce deployment cost. However since only SRLG (link failure probabilities) are considered here, this stage does not

1: INPUT: G(V, E), $src_r, dst_r, F_r, b_r, \forall r \in R$
2: OUTPUT: $x^i_{r,d}, y^i_d, l^{ij}_r$ values
3: set $x^i_{r,d} = 0, y^i_d = 0, l^{ij}_r = 0$ for all $r \in R, i \in F_r, d \in D, (i,j) \in E$
{BEGIN FIRST STAGE}
4: **for** all $r \in R$
5: $D(r) = \{\}$
6: $k = 1$
7: **for** all $i \in F_r$
8: $dk =$ datacenter that implements i at minimum cost and has enough resources to serve an additional request
9: Update resources of d_k
10: Update $y^i_{d_k}$
11: Set $x^i_{r,d_k} = 1$
12: $D(r) = D(r) \cup d_k$
13: $k = k + 1$
{END FIRST STAGE}
{BEGIN SECOND STAGE}
14: **for** all $r \in R$
15: $src = src_r$
16: $C(r) = \{src\}$
17: **for** $k = 1$ to $|D(r)|$
18: $dst = d_k$
19: **if** $d_k \notin C(r)$
20: $SP = RA\_Dijkstra(src, dst)$
21: set $l^{ij}_r = 1$ for all link $(i,j) \in SP$
22: $C(r) = C(r) \cup d_k$
23: $C(r) \cup j$, for all datacenter $j \in SP, j \in D(r)$
24: $src = dst$
25: $dst = dst_r$
26: $SP = RA\_Dijkstra(src, dst)$
27: set $l^{ij}_r = 1$ for all $(i,j) \in SP$
{END SECOND STAGE}
28: **return** $x^i_{r,d}, y^i_d, l^{ij}_r$

Fig. 2: RA-GH Algorithm

consider failure probabilities. For each function i requested by $r \in R$, the algorithm also selects the datacenter d_k that implements i with the lowest setup cost, and at the same time has sufficient resources to host upcoming requests (line 8). Once a NF is placed, the resources for d_k are updated accordingly as follows. Namely if there is an instance of NF i at datacenter d_k with enough instance capacity to satisfy request r, the remaining capacity is decremented by 1. Otherwise, if there is no instance of i to serve request r, another instance is created at cost $c^i_{d_k}$. Either way, the available resources $w_{d_k,j}$ at d_k are reduced by $w^i_{d_k,j}$ (line 9, $j = \{1, 2, ..., m\}$ where m is the number of resources). Each newly-created instance can serve $(\lambda^i_{d_k} - 1)$ requests. Note here that serving a request does not incur any cost, only new instantiations do. Additionally, for each new instance of i, the total number of instances $y^i_{d_k}$ at d_k are incremented by 1 (line 10), and for each deployed NF, x^i_{r,d_k} is set to 1 (line 11). To conclude NF placement (first stage), datacenter d_k is also added to the subset of datacenters that serve request r, i.e., set $D(r)$.

Meanwhile, the Second Stage in Fig. 2 (lines 14-27) computes the shortest path connection between the source and destination, that passes through all datacenters $d_k \in D(r)$ by considering the link failure probabilities $\omega(i,j)$. Initially, the algorithm connects the source node src_r to the first

datacenter d_1 of $D(r)$. Specifically, this computation uses a *risk-aware* (RA) Dijkstra's shortest path algorithm (line 20). This approach verifies whether each link has enough capacity to support the requested b_r. All variables $l_r^{i,j}$ along the path are then set to 1 (line 21), and datacenter d_1 is also added (line 22) to the set $C(r)$. This subset defines the datacenters that serve request r and are already connected to their respective neighbors within the src_r and dst_r path. If there is another datacenter $j \in D$ in the path between src and dst, it is also added to $C(r)$ (line 23). This avoids duplicated path computation for datacenters yet to be analyzed in upcoming iterations. Before returning to the beginning of the loop, the src variable is replaced by the destination datacenter dst (line 25), which was initially set to d_k (line 18). At the top of the loop, dst is set to the next datacenter $d_k \in D(r)$. Therefore, a risk-aware shortest path is computed between the previous destination and the following datacenter, i.e., in the second iteration a path is computed between d_1 and d_2, and so on and so forth. In the two final steps, a risk-aware shortest path is computed between the last datacenter and the destination dst (line 26), and all links $l_r^{i,j}$ within that traffic flow are set to 1 (line 27). To achieve this, the routing cost is defined as the sum of the setup cost c^{ij} of all links associated with a traffic flow, and each respective link request-capacity ratio (b_r/b_{ij}) multiplied by its correlated failure probability $\omega(i,j)$, as:

$$\sum_{(i,j)\in E} (c^{ij} + b_r/b_{ij}) * (1 + \omega(i,j)) \tag{16}$$

Overall, the routing cost is proportionally increased by the link failure probabilities. Akin to the JRP-ILP scheme, the greedy heuristic scheme also sets all $\omega(i,j) \in E$ to zero.

D. Greedy Heuristic Complexity

The greedy algorithm basically finds the first acceptable solution. Namely, the First Stage selects a datacenter d that can implement function $i \in F_r$ at the lowest cost c_d^i (line 8). Note that NF placement here is done in a serialized manner, i.e., each NF is placed independently of its subsequent NF. Moreover, akin to the ILP optimization scheme, the computational time complexity in the First Stage is $O(|R||F||D|)$.

However, the Second Stage has much higher complexity. Namely, a "risk-aware" link-weighted shortest path is computed between the datacenters and endpoints. Clearly, the complexity of this algorithm is defined by the number of requests times the Dijkstra's shortest path complexity, i.e., $O(|E|log|V|)$ [20]. Hence the total run-time complexity for the RA-GH heuristic is bounded by $O(|R||D||E|log|D|)$.

V. PERFORMANCE EVALUATION

The proposed "risk-aware" NF placement and routing schemes are evaluated using the infrastructure topology shown in Fig. 3 with three potential risk failure regions. This configuration has 16 datacenter nodes, each of which is capable of instantiating all NFs. Now the number shown next to a link represents its corresponding failure probability $\omega(i,j) \in E$, as defined according to the u_n SRLG region it falls within, where n=$\{1,2,3\}$. Note that these values are used by the risk-aware RA-ILP and RA-GH schemes only.

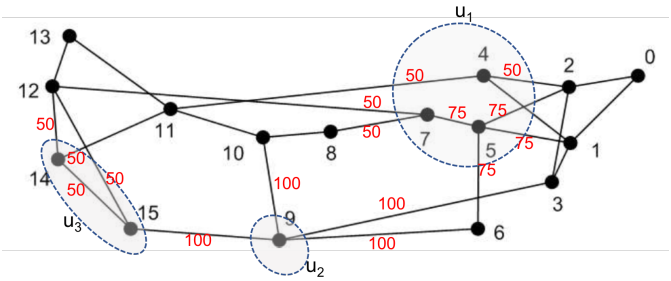


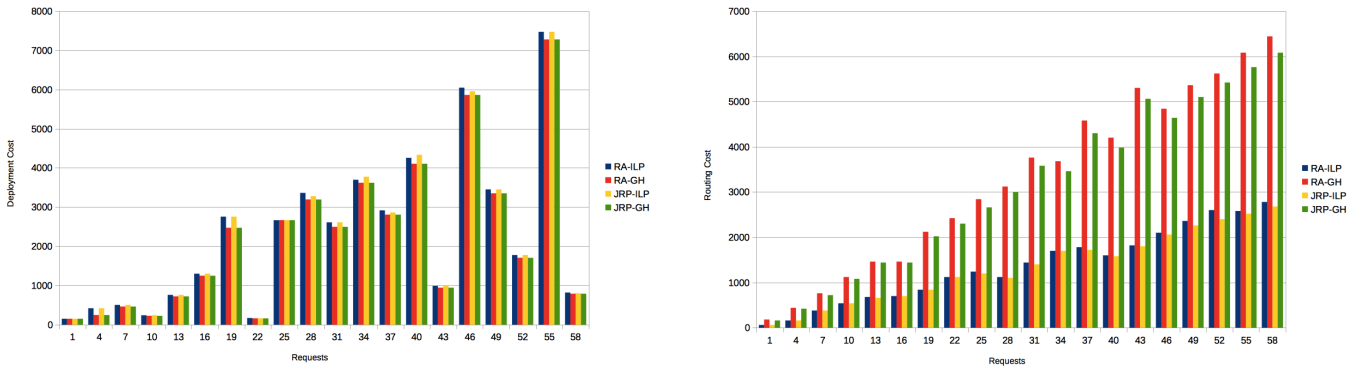
Fig. 3: NSF network topology.

Three types of data resources are also defined here, i.e., W_d=$\{w_{d,1}, w_{d,2}, w_{d,3}\}$, i.e., representing processor, memory and storage respectively. Without loss of generality, it is also assumed that $w_{d,i} = 5,000$ units, i=1, 2, 3. All link capacities are set to $10,000$ units for all $l_d^{i,j}$, where $(i,j) \in E$. Additionally, the set of functions F has five types, i.e., $F = \{f_0, f_1, ..., f_4\}$. Meanwhile, the amount of resources $w_{d,j}^i$ required to implement a function $i \in F_d$ is uniformly distributed between $30 \leq w_{d,j}^i \leq 70$ units. Also, the setup cost of placing an instance of function $i \in F_d$ at datacenter $d \in D$ is set to $c_d^i = 50$. The instance capacity λ_d^i of a function i at datacenter d is also 2 units. It is also assumed that the set of functions F_d implemented by a datacenter d is composed of all NFs $i \in F$ ($F_d = F$), and each request r needs 4 NFs, i.e., F_r is randomly selected from F ($F_r \subseteq F$). Finally, the objective function weighting factors w_1, w_2, w_3, and w_4 are set to 1,000, 1, 1 and 1,000, respectively. These values are chosen based upon the sensitivity analysis conducted in [19].

Initially all four schemes (RA-ILP, RA-GH, JRP-ILP and JRP-GH) are compared in terms of the number of satisfied NFs, mapping/deployment cost and routing cost (under working non-failure conditions). Overall, 60 rounds of r requests are processed, where the round number, rn, identifies the amount of incoming requests for that specific batch, i.e., rn=$\{1, 2, ..., 60\}$, r_{rn}=rn. Furthermore, each request requires 4 NFs, therefore the total number of NFs is $rn * 4$. Overall, the first performance test analyzes the ability to place and route the requests. Here all four schemes manage to satisfy all request sets, indicating accuracy.

Next, further analysis is done to compare performance with regards to cost and survivability. In particular, the deployment and routing costs are compared for all schemes in Fig. 4. Namely, Fig. 4a shows that deployment costs are mostly similar. However the results in Fig. 4b indicate that the routing costs for the greedy heuristic (RA-GH and JRP-GH schemes) are much higher. Specifically, each greedy heuristic yields over twice the cost of its ILP optimization counterpart, i.e., the RA-GH scheme is about 2.3 times higher than the RA-ILP scheme and the JRP-GH scheme is about 2.3 times higher than the JRP-ILP scheme. Furthermore, the same plot also shows that the routing costs for the ILP optimization schemes are very similar. In fact, the costs of the RA-ILP scheme range between 0-4% higher than the JRP-ILP scheme.

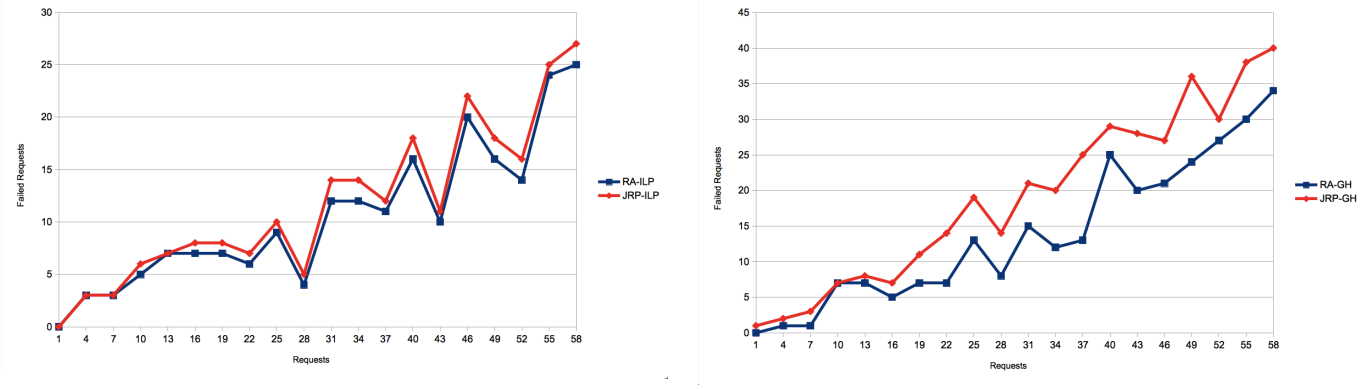
Once the location problem has been solved, further tests are done for multi-failure conditions. Specifically, the first



(a) Deployment costs

(b) Routing costs

Fig. 4: Costs associated with each scheme



(a) Requests assignment failures for RA-ILP and JRP-ILP

(b) Requests assignment failures for RA-GH and JRP-GH

Fig. 5: Requests failures

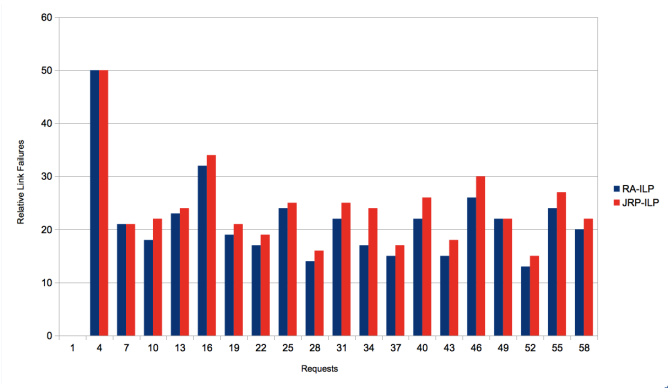
failure region, u_1, is chosen and its related links are randomly failed. In particular, the the links are failed according to the failure probabilities $\omega(i,j)$, as defined in Fig. 3. Specifically, the following 4 physical $l(i,j)$ links are disrupted according to their probabilities: $l(1,5)$, $l(2,4)$, $l(2,5)$ and $l(5,7)$. Note that this number represents 16% of the total network links, i.e., large multi-failure scenario. Now due to computational scalability limitations, the optimization and greedy heuristic schemes differ. Namely, the former can only handle small-to-medium networks, whereas the latter is more suitable for large-scale topologies. As a result their respective survivability performances are studied separately. Overall, Fig. 5a compares the ILP optimization schemes in terms of the number of requests that can be satisfied, i.e., due to the tradeoff between routing and link failures. These results show that the "risk-aware" RA-ILP scheme reduces overall failures by 10-20%. Similarly, Fig. 5b compares the number of failed requests for both of the greedy heuristic schemes and confirms that the risk-aware RA-GH scheme gives notably better survivability, i.e., 50% less failures.

Now the results in Fig. 5 assume that a single link failure disrupts an entire request path. However, the accuracy of the proposed risk-aware models can also be gauged further by computing a link failure ratio:

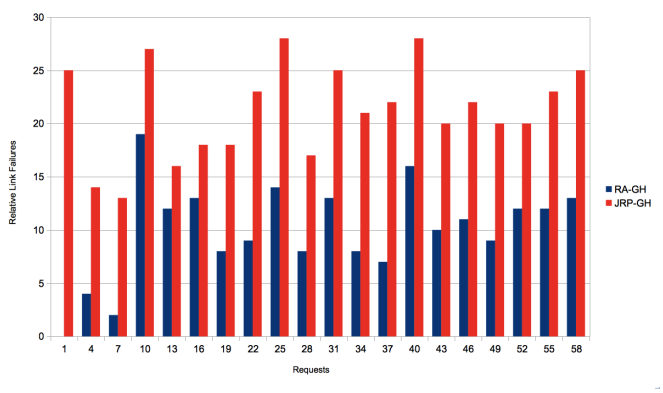
$$fr_{rn} = fl_{rn} * 100/total_{rn} \tag{17}$$

where fl_{rn} is the number of failed links, and $total_{rn}$ is the total number of links in a request path.

Along these lines, Fig. 6a compares this ratio for the optimization schemes and indicates that the "risk-aware" RA-ILP scheme reduces link failures by about 23%. Similarly, the same model is used to compare the RA-GH and JRP-GH schemes. Again, the "risk-aware" scheme reduces link failures by up to 84%, as shown in Fig. 6b. Clearly, introducing failure risk information into the VNF placement and routing process is very beneficial here, since deployment and routing costs have no considerable variation when this model is adopted. However, the number of discontinued services and links (in multi-failure scenario) are considerably reduced.



(a) Link failure rates for RA-ILP and JRP-ILP



(b) Link failure rates for RA-GH and JRP-GH

Fig. 6: Link failures

VI. CONCLUSIONS & FUTURE WORK

This paper presents a first look at disaster recover support with the context of NFV-based infrastructures. In particular, a stochastic failure model is used for a-priori characterization of large-scale disaster events. Novel optimization and heuristic-based solutions are then presented for network function (NF) placement and routing, with a focus in minimizing failure (mapping, routing) risk as well as maximizing the number of satisfied demands. Detailed performance analysis results show that the proposed methods can significantly reduce virtual service disruption and link failure rates. Future efforts will focus on developing further "risk-aware" pre-provisioned protection strategies.

REFERENCES

[1] A. Medhat, G. Carella, M. Pauls, M. Monachesi, M. Corici, T. Magedanz, "Resilient Orchestration of Service Functions Chains in a NFV Environment," *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) 2016*, Palo Alto, CA, Nov. 2016.

[2] M. Beck, J. Botero, K. Samelin, "Resilient Allocation of Service Function Chains," *IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN) 2016*, Palo Alto, CA, Nov. 2016.

[3] Z. Ye, *et al.,* "Joint Topology Design and Mapping of Service Function Chains for Efficient, Scalable, and Reliable Network Functions Virtualization," *IEEE Network*, Vo. 30, No. 3, May/June 2016, pp. 81-87.

[4] M. Casazza, P. Fouilhoux, M. Bouet, "Securing Virtual Network Function Placement with High Availability Guarantees," *IFIP Networking Conference (IFIP Networking) and Workshops 2017*, Stockholm, Sweden, June 2017.

[5] B. Ha, *et al.,* "On the Resiliency of Virtual Network Functions," *IEEE Communications Magazine*, Vol. 55, No. 7, July 2017, pp. 152-157

[6] J. Fan, C. Guan, Y. Zhao, C. Qiao, "Availability-aware Mapping of Service Function Chains," *IEEE Conference on Computer Communications (INFOCOM) 2017*, Atlanta, Georgia, Oct. 2017.

[7] W. Ding, H. Yu, S. Luo, "Enhancing the Reliability of Services in NFV with the Cost-Efficient Redundancy Scheme," *IEEE International Conference on Communications (ICC) 2017*, Paris, France, May 2017.

[8] M. Rahnamay-Naeni, et al, "Modeling Stochastic Correlated Failures and Their Effects on Network Reliability," *International Conference on Computer Communications Networks (ICCN) 2011*, Maui, HI, Aug. 2011.

[9] H. Lee, E. Modiano, K. Lee, "Diverse Routing in Networks with Probabilistic Failures," *IEEE/ACM Transactions on Networking*, Vol. 18, No. 6, Dec. 2010, pp. 1895-1907

[10] O. Diaz, F. Xu, N. Min-Allah, M. Khodeir, M. Peing, S. Khan, N. Ghani, "Network Survivability for Multiple Probabilistic Failures," *IEEE Communications Letters*, Vol. 16, No. 8, Aug. 2012, pp. 1320-1323

[11] H. Yu, *et al*, "Survivable Virtual Infrastructure Mapping in a Federated Computing and Networking System under Single Regional Failures," *IEEE GLOBECOM 2010*, Miami, USA, December 2010.

[12] G. Sun, *et al*, "Efficient Algorithms for Survivable Virtual Network Embedding," *Asia Communications and Photonics Conference and Exhibition (ACP) 2010*, China, December 2010.

[13] F. Gu, *et al*, "Survivable Cloud Network Mapping for Disaster Recovery Support," *IEEE Transactions on Computers*, Vol. 64, No. 8, September 2014, pp. 2353-2366.

[14] B. Addis, D. Belabed, M. Bouet, S. Secci, "Virtual Network Functions Placement and Routing Optimization," *IEEE International Conference on Cloud Networking*, Niagara Falls, Canada, October 2015.

[15] R. Cohen, L. Lewin-Eytan, J. Naor, D. Raz, "Near Optimal Placement of Virtual Network Functions," *IEEE International Conference on Computer Communications*, Hong Kong, April 2015.

[16] M. Xia, M. Shirazipour, Y. Zhang, H. Green, A. Takacs, "Network Function Placement for NFV Chaining in Packet/Optimal Datacenters," *IEEE/OSA Journal of Lightwave Technology*, Vol. 33, Issue 8, April 2015, pp. 1565-1570.

[17] M. Bouet, J. Leguay, V. Conan, "Cost-Based Placement of Virtualized Deep Packet Inspection Functions in SDN," *IEEE Military Communications Conference (MILCOM)*, San Diego, CA, November 2013.

[18] J. Crichigno, D. Oliveira, N. Ghani, "Joint Routing and Placement of Virtual Network Funcations," *IEEE Telecommunications and Signal Processing (TSP) 2017*, Barcelona, Spain, July 2017.

[19] D. Oliveira, J. Crichigno, N. Ghani, "On Sensitive and Weighted Routing and Placement Schemes for Network Function Virtualization," *Infocommunications*, Vol. IX, Issue 4, Dec. 2017.

[20] M. Barbehenn, "A Note On the Complexity of Dijkstra's Algorithm for Graphs with Weighted Vertices," *IEEE Transactions on Computers*, Vol. 47, Issue 2, Feb. 1998, pp. 263.