IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received May 10, 2020, accepted June 2, 2020, date of publication June 26, 2020, date of current version July 9, 2020.

Digital Object Identifier 10.1109/ACCESS.2020.3005287

VNF Availability and SFC Sizing Model for Service Provider Networks

SIDHARTH SHARMA[1], ANNA ENGELMANN[2], (Graduate Student Member, IEEE), ADMELA JUKAN[2], AND ASHWIN GUMASTE[1], (Senior Member, IEEE)

[1]Department of Computer Science and Engineering, Indian Institute of Technology (IIT) Bombay, Mumbai 400076, India
[2]Department of Electrical Engineering, Information Technology, Physics, Technische Universität Braunschweig, 38106 Braunschweig, Germany

Corresponding author: Sidharth Sharma (sidharth@cse.iitb.ac.in)

ABSTRACT This paper considers service provider network design with a view of meeting application availability. Our primary goal is to design a service provider network using disparate network components and low-availability Virtual Network Functions (VNFs) while achieving high-availability Service Function Chains (SFCs). To this end, we attempt to answer the questions as to how much redundancy, and what type of redundancy should be incorporated in the network. Initially, we model a state-of-the-art provider architecture that is spread across the access, metro-core and backbone networks with associated discrete components such as switches, routers, optical equipment and data-centers. Our network design model leads to computing the amount of over-the-top (OTT) services that can be provisioned over a given network while achieving a particular availability. To this end, we formulate a constrained optimization model whose objective is profit maximization subject to availability measures that OTT services demand. We provide constraints of robustness that facilitate traffic churn. Four heuristics are also proposed with objectives that consider the breadth of the three key impacting parameters: VNF licensing cost, server utilization and delay optimization and the case for dynamic traffic. A simulation model presents comparative data for efficiency, latency and server utilization as well as validates our optimization model. The results stress the importance of an efficient optimization model in planning the network, as well as planning VNF placement ahead in time.

INDEX TERMS Availability, service function chaining, VNF.

I. INTRODUCTION

Application availability is of utmost importance to over-the-top (OTT) services that are offered by the behemoth OTT operators like Apple, Amazon, Facebook, Microsoft, etc. The issue of application availability is further exacerbated, given the vast geographies over which these OTT applications are provisioned. Much of the OTT providers often lease bandwidth from existing telecom service providers (TSPs) and application space from third-party data-centers (DCs) or set up DCs in customer domains (local data-centers, edge computing, edge central offices). With the advent of Network Function Virtualization (NFV) paradigm, some of these applications could be virtual network functions (VNFs). Multiple VNFs can be aptly concatenated to form service function chains (SFCs) [21]. The desire is to create a network, where despite the use of so many disparate components (VMs, servers, switches, etc.), we want to offer high-availability SFCs. This high availability SFC problem is somewhat loosely related to the VNF placement problem [2]. However, there is significant complexity added, whereby the availability of various individual components involved in an end-to-end SFC, now have different values of availability. These independent components of the network and the various VNFs together contribute towards availability of the SFC. We hence want to create a method that would aid in designing a network that can offer SFC based service level objectives (SLOs) despite these various discrete elements involved in the network.

When an OTT player seeks SFCs from a country-wide TSP (e.g. AT&T or Deutsche Telekom), there is a sizable challenge in meeting availability guarantees at low price-points for the TSP. Despite this challenge, there is a mutual benefit: By selling SFCs, a TSP can monetarily gain from the OTT player, while the OTT player can do away with managing complex telecom equipment. To a TSP, the placement and provisioning question involves how much resources (equipment) should be installed for meeting the SLOs for

The associate editor coordinating the review of this manuscript and approving it for publication was Yulei Wu.

availability? In order to meet availability requirements for SFCs, the provider's network is also impacted – sizing core routers, deploying load-balancers, session controllers, optical equipment, all of which have impact within and outside of the data-center.

Solving the SFC availability problem requires an understanding of the current provider network. In particular, we are interested in designing a network, whereby we are given heterogeneous availability values of the various components. These components include software on VMs, VMs on servers, servers in pods of a data-center, and then the network itself. While the network components traditionally have high availability, the IT-components like VMs, servers are built using COTS (commercial off-the-shelf) gear [10]. This makes the design problem hard, especially over a large topology. The challenge in this problem is that the various components have different availability values, whereas an SFC is specified by a single value. This means different numbers of disparate components are arranged in such a way that their combination meets the SFC availability requirement.

We build in this paper, a case for network sizing and design that would enable a provider to offer demarcated services to an OTT operator. We provide constraints of robustness to model uncertainty in the bandwidth requirements of the traffic requests. Four heuristics are proposed with different objectives: 1) optimizing server utilization, 2) minimizing VNF licensing cost, 3) minimizing delay, and 4) for handling dynamic traffic.

Contributions of this work: This work has the following contributions:

1) SFC-level availability is non-linear in terms of the product of individual VNF availability due to the parallel formation of VNFs. To attain SFC-level availability without a smart computation method could prove to be cost-wise expensive. We solve this problem with the use of a unique data-structure in an interesting combinatorial optimization approach. One of the key contributions of our optimization formulation is to make the non-linear problem of availability computation linear, by some minimum pre-processing and the use of a one-hot vector towards obtaining SFC-level and VNF-level availability, despite the basic computation requiring parallelism (implying non-linearity). To the best of our knowledge, we, for the first time, solve this non-linear problem and convert the entire optimization exercise to a linear form for all the constraints. Even for a large regional network, such a problem can then be solved in minutes on a modest machine.
2) Our optimization model is extremely effective in cost-saving by using COTS equipment efficiently in conjunction with telecommunication gear.
3) With the inclusion of robustness, we provide a way to incorporate traffic uncertainty. Robust optimization [3] is a better way to deal with uncertainty than any of the four heuristics that we propose. The only challenge is that the uncertainty in traffic must be within some known bounds, and for those bounds, there is a minor excess price to pay, but the performance is superior (compared to the heuristics).

The rest of the paper is organized as follows: Section II showcases the reference network architecture and the problem statement. In Section III, we formulate this problem as a static problem, whereby we want to dimension a provider network to facilitate the provider to understand how much traffic and of what type can be provisioned by making strategic investments. Section IV proposes tractable heuristic algorithms. Section V describes the numerical results. Section VI discusses related work. Finally, Section VII concludes the paper.

II. NETWORK ARCHITECTURE AND PROBLEM STATEMENT

A. REFERENCE NETWORK ARCHITECTURE

A country-wide TSP is shown in Fig. 1. Such an TSP has a last-mile access, a metro-core and a backbone network. In this TSP, we will assume residential users, enterprises and wireless users. All users are assumed to be part of the access network. Typically, the access is a hub-and-spoke set-up, with the central office (CO) at the hub and users at the spokes. The COs are backhauled to a provider edge (PE) router. Typically access networks have 8-20 COs connected per PE router and usually, this connection is done in a dual-homing method. The metro-core network consists of IP and optical equipment, such as a PE router and ROADMs at every node. Cost reduction in transport networks is done by the use of packet-optical integrated technology, an example of which is Carrier Ethernet (CE) [20]. With a CE device, the data need not have to go to an IP router at an intermediate node thus saving OpEx. A typical metropolitan core has less than 8-14 nodes and spans across a city. A couple of metro core nodes provide a gateway to connect to the nation-wide backbone network. These connecting nodes are special: they have a large layer-3 P-router, which is generally a multi-chassis implementation along with a 3 or 4-degree ROADM, which is instrumental in creating a fiber-based backbone mesh. At times, the optical transport network (OTN) is used for providing signal extension using forward error correction (FEC), as well as to enable layer 0/1 50ms restoration. Tunable transponders are also used in routers, switches, etc. for providing access to wavelengths of the WDM spectrum.

The TSP also has data-centers (DCs). We consider two types of DCs – one connected to the backbone, called Backbone DC (bDC), and a DC at the edge (access), which is the CO along with some caching facility, which we term as mini-DC (mDC). [39].

B. AVAILABILITY MODELS

To design a network for high-available OTT services, we need to consider various placement strategies of VMs and servers. To this end, we consider backup protection and define two concepts that are intrinsic towards VNF availability

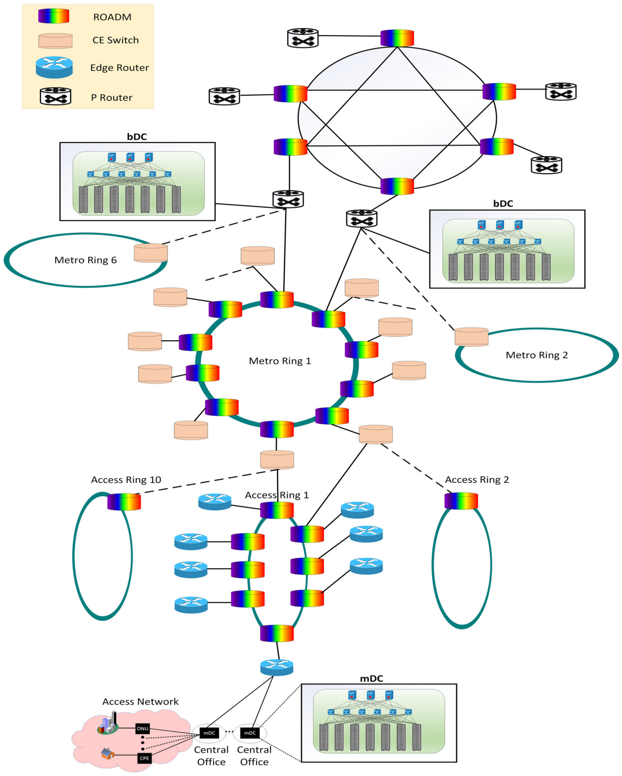


FIGURE 1. **Service provider network.**

computation from the network architecture perspective: over-provisioning and geo-redundancy.

Over-provisioning from the VNF perspective implies that additional capacity, i.e., backup resources, are provisioned in the DC for tolerating VNF outages. In the simplest case, a backup VM can be provisioned in a server that is adjacent to the primary VM's server. Fig. 2a illustrates the placement in which a SFC is to be provisioned with VNFs A1, A2 and A4. For redundancy, a backup VNF corresponds to each of the primary VNFs (A1B, A2B, A4B) is placed in a different adjacent server. Therefore, now the SFC consists of the sequence A1, A1B, A2, A2B, A4, A4B (SFC path is shown by the dotted line). The problem with this case is that both the servers are connected to the same switch, which may or may not suffice the availability requirements. In a slightly more elaborate design (Fig. 2b), the primary VM/server and backup VM/server are located in different corners of the DC. Primary and secondary VMs/servers may be in different DC pods. In this case, more links need to be provisioned, as compared to Fig. 2a.

Geo-redundancy works to keep multiple copies of the same VNF in different locations to achieve good fault tolerance. In the most exhaustive geo-redundant configuration, the primary and backup VMs/servers are located in different DCs. Fig. 2c demonstrates the example of geo-redundancy in which backup VNFs of the SFC are placed in another DC. In the example, the SFC needs to traverse from DC1 to DC2 for accessing backups corresponding to each of the primary VNFs. This approach consumes more links as compared to over-provisioning, as it also consumes bandwidth for inter-DC communication for each VNF in the SFC. This approach implies haphazard growth in the required resources. Moreover, geo-redundancy has an impact on SFC latency and software license pricing. VNF licenses may or may not be linear additive and a strategy of VNF placement must consider license costs.

C. PROBLEM STATEMENT

Given the complexity of the network and the intricacies of VNF placement to achieve high-availability, the design problem is quite complex. This is especially so when we consider a large provider network. For designing a high-availability network, especially one that considers geo-redundancy and over-provisioning, we must take into consideration the intricacies that present with a parallelized placement of VMs on servers, and servers in pods of a DC. In a large network, such as the one described above, there would be numerous possibilities of VNF placements, implying a convex optimization problem. Such problems are known to be hard [18]. However, creating a high-available architecture within the domain of a provider network with a large number of protocols, disparate devices and technologies is an even more complex problem. Our specific issue while designing an optimization problem is to consider two or more parallel VMs or servers and then take their join availability values (in parallel) as a part of the optimization process. This is especially complex due to two reasons: (1) the parallel arrangement implies a non-linear operator to be used to obtain the joint availability and (2) the number of parallel elements required, itself is not known ahead in time. Together, this implies that we need a way to linearize parallel implementations so that we can solve an integer linear program in a quick time.

III. NETWORK DESIGN FOR AVAILABILITY: OPTIMIZATION MODEL

In this section, the network design problem is formulated as a constrained optimization exercise. As compared to approaches in [40] and [17], our approach to network design for a NFV-centric network with availability as a major constraint includes modeling a state-of-the-art provider network and taking into consideration the heterogeneous availability values of disparate telecom and IT equipment.

Experiences from the telecom world indicate that equipment (such as routers, switches and optical gear) have high-availability in the vicinity of 99.999% [10]. However, servers/VMs have lower availability values, typically 99.9% or, in some circumstances, as low as 96%. Our approach designs a network that has enough redundancy to achieve an availability threshold. We assume that an application/VM is always shared by multiple traffic requests, whereby the targeted availability is guaranteed at all times for each assigned traffic request.

We define three sets of variables: network-centric input variables described in Table 1, that defines the choices to be fed to the optimization formulation; binary decision

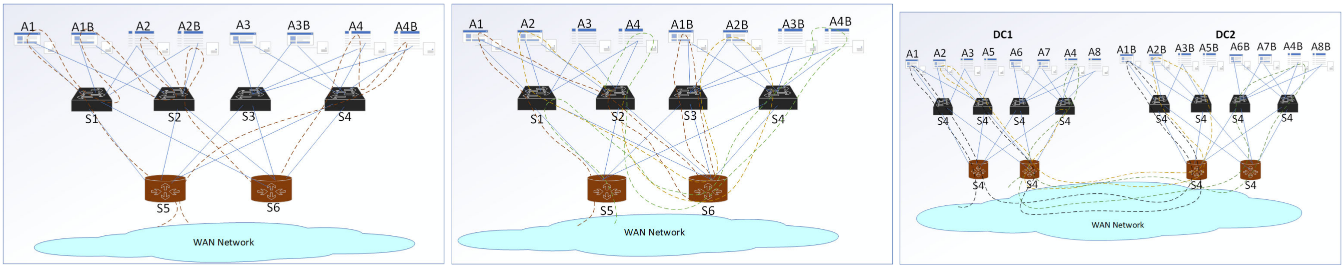


FIGURE 2. a) Backups of VMs reside in the same DC and same pod (left), b) Backups of VMs reside in the same DC and different pod (middle), c) Backups of VMs in different DC (right).

TABLE 1. Input parameters.

$G(V, E)$	Network graph with V nodes and E edges
T^n_{abkm}	m^{th} traffic request on k^{th} path using n^{th} service function chain between node V_a and V_b
$\lvert T^n_{abkm}\rvert$	bandwidth requirement of T^n_{abkm}
$\lvert T^n_{abkm}\rvert(peak)$	peak bandwidth requirement of T^n_{abkm}
$\lvert T^n_{abkm}\rvert(avg)$	average bandwidth requirement of T^n_{abkm}
T	Traffic set
IP^i_c	IP Router size at node V_i
E^i_c	Ethernet Switch size at node V_i
R^i_c	ROADM size at node V_i
SC_n	Represents n^{th} SFC
$\{SC_n\}$	VNFs part of SFC n
C_{bDC}	Size of a bDC
C_{mDC}	Size of a mDC
PM^k_{ab}	k^{th} path between nodes V_a and V_b
B_{qi}	Capacity of the server q in DC_i
B_{si}	Capacity of VM s in DC_i
Th_q	Target availability for server q
Th_s	Target availability for VM s
Th_{sc_n}	Target availability for SFC n
r	Unit revenue for every 100Mbps of traffic
c_{IP}	100Mbps processing IP Router cost
c_E	100Mbps processing Ethernet switch cost
c_R	100Mbps processing ROADM cost
c_{VNF}	Cost of VNF license for processing 100Mbps of traffic
c_q	Cost of server for processing 100 Mbps of traffic
$\triangle^n_{abkm}$	Maximum allowable delay for T^n_{abkm}
δ_i	Delay introduced by node V_i

TABLE 2. Binary decision variables.

α^i_{IP}	$=1$ if IP router is present at the node V_i; 0 otherwise
α^i_E	$=1$ if Ethernet switch is present at the node V_i; 0 otherwise.
α^i_R	$=1$ if ROADM is present at the node V_i; 0 otherwise.
α^i_D	$=1$ if DC is present at the node V_i; 0 otherwise.
α^i_q	$=1$ if server is present at the node V_i; 0 otherwise.
α^i_s	$=1$ if VM is present at the node V_i; 0 otherwise.

variables in Table 2, that are chosen by the formulation towards reaching the global maximization; and auxiliary variables in Table 3, that are used for cushioning the solution. Our formulation is unique in such a way that it not only considers the availability of network resources but also includes the intricacies of designing a highly-available telecom network.

TABLE 3. Binary auxiliary variables.

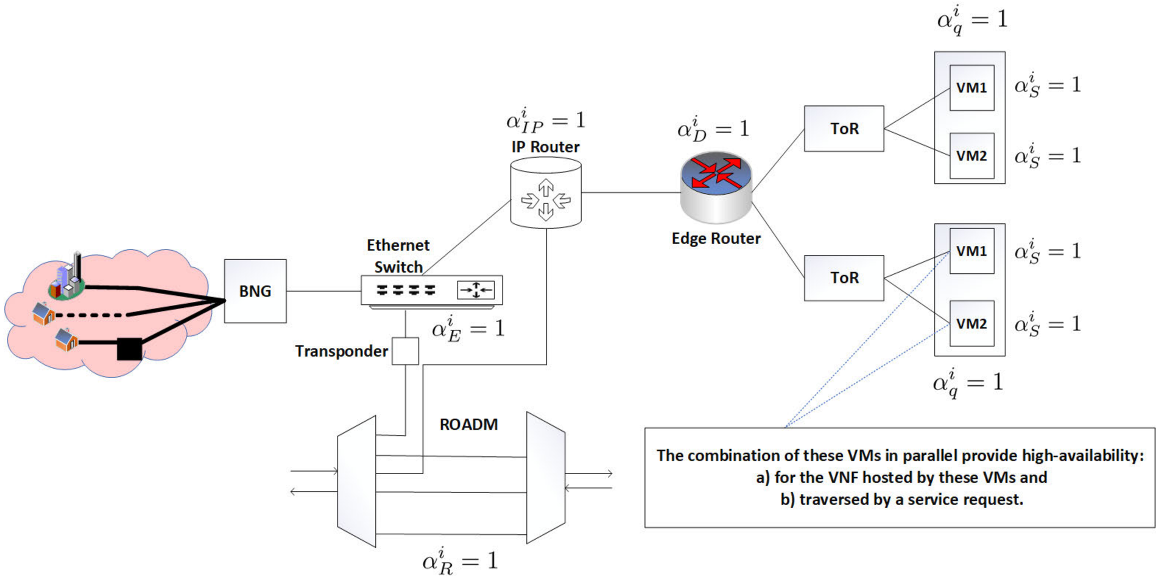
θ^n_{abkm}	$=1$ if T^n_{abkm} is provisioned; 0, otherwise
ϵ^{qi}_{abkmn}	$=1$ if T^n_{abkm} is provisioned on server q at node V_i; 0, otherwise
γ^{si}_{abkmn}	$=1$ if T^n_{abkm} is provisioned on VM s at node V_i; 0, otherwise
W^i_{abkmn}	$=1$ if $V_i \in PM^k_{ab}$ for each T^n_{abkm} provisioned; 0, otherwise
Z^i_{abkmn}	$=1$ if the switch at node V_i is used to provision the backup path for T^n_{abkm} ; 0, otherwise

A. OBJECTIVE FUNCTION

We define the objective function as the maximization of profit. Profit is estimated as the difference between revenue and expense. A provider typically has Capital Expenditure (CapEx) and Operational Expenditure (OpEx), whereby, for this formulation, we assume all CapEx to be amortized over a 7-year period. Our expense includes equipment cost, power consumption cost, salaries and marketing as well as bandwidth and other costs. Hence, the objective function becomes:

$$max \sum\nolimits_{\forall a,b,k,m,n} r.|T^n_{abkm}|.\theta^n_{abkm} - [(\sum\nolimits_{\forall i} \alpha^i_{IP}.IP^i_c.c_{IP} + \sum\nolimits_{\forall i} \alpha^i_E.E^i_c.c_E + \sum\nolimits_{\forall i} \alpha^i_R.R^i_c.c_R + \sum\nolimits_{\forall i} \alpha^i_q.B_{qi}.c_q + \sum\nolimits_{\forall i} \alpha^i_s.B_{si}.c_{VNF})],$$

where, r is a revenue measure of some SLO parameter and, for simplicity, is made proportional to the bandwidth. In Fig. 3 is the intuition behind designing a highly-available network through our optimization formulation. Fig. 3 showcases how disparate network equipment (such as ROADM, Ethernet switch, IP Router and Edge router) and IT equipment (such as VMs and servers) are instantiated through a binary variable α. The end-users are connected through a Broadband Network Gateway (BNG), which is further connected to an Ethernet switch. The Ethernet switch is connected to an IP Router. The IP Router is also connected to a ROADM for optical transport in a metro-core network. In a DC, there are multiple VMs and servers. In order to provide a highly-available service, VMs (hosting VNFs required for a service request) and servers are replicated.



The combination of these VMs in parallel provide high-availability:
a) for the VNF hosted by these VMs and
b) traversed by a service request.

FIGURE 3. **Designing the provider network through our optimization formulation.**

1) EQUIPMENT CAPACITY CONSTRAINTS

We now state the following constraints at each node i. We say, $\forall \theta^n_{abkm} > 0$:

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.|T^n_{abkm}| \leq IP^i_c, \quad \forall i : \alpha^i_{IP} = 1 \tag{1}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.|T^n_{abkm}| \leq E^i_c, \quad \forall i : \alpha^i_E = 1 \tag{2}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.|T^n_{abkm}| \leq R^i_c, \quad \forall i : \alpha^i_R = 1 \tag{3}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.|T^n_{abkm}| \leq B_{qi}, \quad \forall i : \alpha^i_q = 1 \tag{4}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.|T^n_{abkm}| \leq C_{bDC}, \quad \forall i : \alpha^i_D = 1 \tag{5}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.|T^n_{abkm}| \leq C_{mDC}, \quad \forall i : \alpha^i_D = 1 \tag{6}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.|T^n_{abkm}| \leq B_{si}, \quad \forall i : \alpha^i_S = 1 \tag{7}$$

Eq. (1) ensures that the summation of the bandwidth of all the traffic requests at V_i should not exceed the capacity of the IP router. Eq. (2) states that the provisioned traffic on an Ethernet switch should not exceed its capacity. Similarly, eqs. (3)-(7) specify that traffic passing through ROADM, server, bDC, mDC and VM should not exceed their respective capacities.

We say, for a service request T^n_{abkm} to be provisioned, $\theta^n_{abkm} = 1$ implying that the particular request is provisioned, if the k^{th} path is chosen and a server q is on this path such that T^n_{abkm} is allocated to the server. On this server must be a VM s such that it runs a VNF application of type $\phi \in SC_n$. In such a situation the following conditions hold:

$$\exists \epsilon^{qi}_{abkmn} = 1 : \underset{i}{\arg}[\epsilon^{qi}_{abkmn}] \in PM^k_{ab} \tag{8}$$

$$\exists \gamma^{si}_{abkmn} = 1 : \underset{i}{\arg}[\gamma^{si}_{abkmn}] \in PM^k_{ab} \tag{9}$$

Eqs. (8) and (9) state that for each VNF in the SFC, there must be some VMs and servers (provisioning the VNF) on the chosen path for a service request. The constraints indicate that the value of binary variables ϵ^{qi}_{abkmn} and γ^{si}_{abkmn} will be 1 only when the path PM^k_{ab} contains node i. Note, however, that these constraints do not specify the exact numbers of VMs and servers hosting the required VNFs. The quantity of VMs/servers will be handled by the VNF and SFC availability constraints.

2) BANDWIDTH CONSTRAINTS

Eqs. (10) and (11) further tightens the convex polytope by appropriate choice of VM and server. The total traffic provisioned on server q is less than its bandwidth B_{qi}. We say, $\forall \theta^n_{abkm} > 0$:

$$\sum |T^n_{abkm}|.\epsilon^{qi}_{abkmn} \leq B_{qi}, \quad \forall a, b, k, m, n, i \tag{10}$$

Similarly, for the VM s, we say:

$$\sum |T^n_{abkm}|.\gamma^{si}_{abkmn} \leq B_{si}, \quad \forall a, b, k, m, n, i \tag{11}$$

The above constraints imply that if either a server q or VM s are used to provision traffic T^n_{abkm}, then the aggregate bandwidth of all the traffic requests provisioned on the VM or server should be less than the server/VM capacity.

3) VNF AVAILABILITY CONSTRAINTS

We want to design a network such that the combined availability of VNFs of certain type exceeds the stipulated requirement, i.e., the availability threshold Th. We define a vector ρ_r : $\rho_r[w]$ is the availability when w units (servers/VM) are in parallel. ρ_r is of size $1 \times Y$: Y is the maximum number of servers/VMs in parallel that is practically possible.

The maximum bound is chosen based on practical availability values as vector Λ_C. We define Λ_C as a *one-hot vector* (whose only one element is 1, while all others are zero). Therefore, Λ_C,where $C = 4$ implies $\Lambda_C = [0, 0, 0, 1, 0]$. Then we say, $\forall \epsilon^{qi}_{abkmn} > 0$ and $\forall \gamma^{si}_{abkmn} > 0$:

$$\Lambda_{\sum \epsilon^{qi}_{abkmn}} . [\rho_r]^T \geq Th_q$$
$$\Lambda_{\sum \gamma^{si}_{abkmn}} . [\rho_r]^T \geq Th_s \tag{12}$$

The above constraint implies that for each VNF/application type required for a traffic request, there must be enough instances of the VNF/application type (and hosting servers) in the infrastructure to provide the desired availability for that VNF/application.

For example, suppose the availability of a server is 0.9 and the availability threshold (Th_q) is 0.999. In this case, ρ_r will be $[0.9, 0.99, 0.999, 0.9999, 0.99999]^T$. Note that the first value of the vector i.e., 0.9 represents availability when only one server is present. Likewise, the last value, i.e., 0.99999, calculated as $1-(1-0.9)^5$ represents the availability obtained by having 5 servers in parallel. This implies that each element of the vector represents a resulting availability value obtained by placing i servers/VMs in parallel. Where, i denotes the index of an element in the vector (starting from 1). Based on the value of threshold (Th_q, Th_s), the formulation selects the exact number of servers/VMs to be placed in parallel to obtain the desired availability of each VNF in the SFC. In this example, the one-hot vector selected is: $\Lambda_C = [0, 0, 1, 0, 0]$ leading to $\sum \epsilon^{qi}_{abkmn} = 3$. Which means T^n_{abkm} should be provisioned at exactly three servers in parallel for traversing each VNF in the SFC.

4) SFC AVAILABILITY CONSTRAINTS

VNF availability constraints ensure the required availability at the VNF-level. However, it is also desirable to provide high-availability for complete SFCs. We define X^j_{abkmn} to denote the availability value for j^{th} VNF in SC_n obtained by provisioning traffic request T^n_{abkm} on the redundant servers (hosting VNF j). Similarly, we define Y^j_{abkmn} for the VM case.

$$X^j_{abkmn} = \Lambda_{\sum \epsilon^{qi}_{abkmn}} . [\rho_r]^T \tag{13}$$
$$Y^j_{abkmn} = \Lambda_{\sum \gamma^{si}_{abkmn}} . [\rho_r]^T \tag{14}$$
$$\sum_{\forall j \in SC_n} X^j_{abkmn} . \epsilon^{qi}_{abkmn} \geq Th_{sc_n}/|SC_n| \tag{15}$$
$$\sum_{\forall j \in SC_n} Y^j_{abkmn} . \gamma^{si}_{abkmn} \geq Th_{sc_n}/|SC_n| \tag{16}$$
$$X^j_{abkmn} . \epsilon^{qi}_{abkmn} \geq Th_{sc_n}, \quad \forall j \in SC_n, \ \forall a, b, k, m, n, i \tag{17}$$
$$Y^j_{abkmn} . \gamma^{si}_{abkmn} \geq Th_{sc_n}, \quad \forall j \in SC_n, \ \forall a, b, k, m, n, i \tag{18}$$

Eqs. (13) and (14) assign the availability values obtained as a result of VNF availability constraints to X and Y which correspond to server and VM availability. Constraints in eqs. (15)-(18) ensure that the overall SFC availability should be above a threshold by taking into account the availability values of the constituent VMs and servers. Since the multiplication operation leads to non-linearity, we break the multiplication into summation in eqs. (15) and (16). Finally, eqs. (17) and (18) (our main SFC availability constraints) ensure the overall SFC availability threshold should be met. Note that X^j_{abkmn} and Y^j_{abkmn} are not variables but preassigned fixed numbers obtained from eqs. (13) and (14) and hence eqs. (17) and (18) are linear.

5) GEO-REDUNDANCY CONSTRAINTS

We want the primary and backup servers/VMs of a SFC to be in a different DC or different node. To incorporate this constraint, we say that $\forall \theta^n_{abkm} > 0$ and $\forall \sum \epsilon^{qi}_{abkmn} > 1$

$$\arg_i[\epsilon^{qi}_{abkmn}] \neq \arg_j[\epsilon^{qj}_{abkmn}] \tag{19}$$

Eq. (19) states that two or more redundant servers are forced to be in two different DCs thus satisfying our geo-redundancy constraint. Similarly, for VMs, $\forall \sum \gamma^{si}_{abkmn} > 1$

$$\arg_i[\gamma^{si}_{abkmn}] \neq \arg_j[\gamma^{sj}_{abkmn}] \tag{20}$$

Eq. (20) implies that the VMs for traffic connection T^n_{abkm} are at different nodes (and hence achieve geo-redundancy).

6) VM-SERVER MAPPING CONSTRAINT

Since the provisioned VM must be assigned to a server, we state that $\forall \theta^n_{abkm} > 0$: $\forall \sum \epsilon^{qi}_{abkmn} > 0$ and $\forall \sum \gamma^{sj}_{abkmn} > 0$

$$\arg_i[\epsilon^{qi}_{abkmn}] = \arg_j[\gamma^{sj}_{abkmn}] \tag{21}$$

The above constraint guarantees that there is a one-to-one placement of VM on a server.

7) DELAY CONSTRAINT

We want the end-to-end delay for a traffic request should be bounded. We say $\forall T^n_{abkm} : \theta^n_{abkm} > 0$

$$\sum_{V_i \in PM^k_{ab}} \delta_i \leq \Delta^n_{abkm} \tag{22}$$

The next constraint in eq. (23) guarantees that there is bandwidth preserved between the DC switch and each branch of the path towards a VM that is used for creating a backup combination. Hence the constraint for ensuring that we have enough bandwidth at the switch that subtends the backup path (for additional availability) is stated as:

$$\sum_{\forall a,b,k,m,n,i} |T^n_{abkm}| . (W^i_{abkmn} + Z^i_{abkmn}) \leq E^i_c, \quad \forall i \in V_i \tag{23}$$

8) WAVELENGTH CONTINUITY CONSTRAINT

If T^n_{abkm} is provisioned (θ^n_{abkm}) on the path PM^k_{ab}, then there must exist a wavelength $\lambda_w : T^n_{abkm}$ is mapped to $\lambda_w \ \forall \alpha^i_R = 1$.

B. TAKING UNCERTAINTY OF TRAFFIC INTO CONSIDERATION

The case of uncertainty that we consider is for users of a particular service swelling their demand. An example of such a service may be a sports event that is being video-streamed after its content goes through a few VNFs (some processing, adding special content, codec, etc.). Such a service has its value increase only at select times. Further, we assume that of the N possible service types, only K would swell at a given time. Naturally, the amount to which they would swell would be known ahead in time due to the type of service, however, which of these services would swell is not known ahead in time.

To solve such a provisioning problem subject to availability requirement, factoring in geo-redundancy and over-provisioning, we make use of robust optimization. We replace the traffic matrix T, with T_{peak} and T_{avg} that denote peak and average traffic values of each connection and then further decompose the service function chain set into SC_{swell} and SC_{others}. The set SC_{swell} consists of K SFCs that have the largest values, while SC_{others} are the remaining SFCs. The set SC_{swell} hence has all SFCs (and hence traffic requests) that are at peak value. We define a binary variable O^n_{abkm} that is 1, if the corresponding traffic request belongs to the SC_{swell} value, or is 0, if otherwise. Further, we assume that VNFs are elastic, i.e., they can be grown in size based on service requirement – naturally subject to the server capacity. The capacity constraints (1)-(7) now change as follows:

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.(|T^n_{abkm}|(peak) + |T^n_{abkm}|(avg)) \leq IP^i_c, \quad \forall i : \alpha^i_{IP} = 1 \ \forall O^n_{abkm} \tag{24}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.(|T^n_{abkm}|(peak) + |T^n_{abkm}|(avg)) \leq E^i_c, \quad \forall i : \alpha^i_E = 1 \ \forall O^n_{abkm} \tag{25}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.(|T^n_{abkm}|(peak) + |T^n_{abkm}|(avg)) \leq R^i_c, \quad \forall i : \alpha^i_R = 1 \ \forall O^n_{abkm} \tag{26}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.(|T^n_{abkm}|(peak) + |T^n_{abkm}|(avg)) \leq B_{qi}, \quad \forall i : \alpha^i_q = 1 \tag{27}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.(|T^n_{abkm}|(peak) + |T^n_{abkm}|(avg)) \leq C_{bDC}, \quad \forall i : \alpha^i_D = 1 \ \forall O^n_{abkm} \tag{28}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.(|T^n_{abkm}|(peak) + |T^n_{abkm}|(avg)) \leq C_{mDC}, \quad \forall i : \alpha^i_D = 1 \ \forall O^n_{abkm} \tag{29}$$

$$\sum_{\forall a,b,k,m,n} W^i_{abkmn}.(|T^n_{abkm}|(peak) + |T^n_{abkm}|(avg)) \leq B_{si}, \quad \forall i : \alpha^i_S = 1 \ \forall O^n_{abkm} \tag{30}$$

Eq. (24) ensures that the aggregate provisioned traffic (that consists of some traffic requests at their peak values while others are at their average values) through any IP router should not exceed the capacity of the IP router. Similarly, eqs. (25)-(30) specify that the provisioned traffic on an Ethernet switch, ROADM, server, bDC, mDC and VM should not exceed their respective capacities. Similarly, server and VM bandwidth capacity constraints will also change. Therefore, eqs. (10) and (11) will now transform into following equations:

$$\sum((|T^n_{abkm}|(peak) + |T^n_{abkm}|(avg))).\epsilon^{qi}_{abkmn} \leq B_{qi}, \quad \forall a, b, k, m, n, i \ \forall O^n_{abkm} \tag{31}$$

Similarly, for the VM s, we say:

$$\sum((|T^n_{abkm}|(peak) + |T^n_{abkm}|(avg))).\gamma^{si}_{abkmn} \leq B_{si}, \quad \forall a, b, k, m, n, i \ \forall O^n_{abkm} \tag{32}$$

Eq. (31) ensures that all the traffic requests (either at their peak or at average) passing through the server q at node V_i should not exceed the bandwidth capacity of the server. Similarly, eq. (32) specifies that combined bandwidth values of the traffic request passing through VM s should not exceed VM's capacity.

IV. FAST ALGORITHMS FOR DESIGNING HIGH-AVAILABILITY

In practice, a provider is expected to run the optimization model every few months and re-arrange network equipment if required. For operational purposes, a requirement is to have a series of heuristic algorithms that take the current form of network and the traffic demands as input and create a VNF assignment, while guaranteeing high-availability.

In this section, we propose four heuristic algorithms. The first three algorithms focus on three critical components of an NFV-enabled network – maximizing server utilization, minimizing VNF license cost and SFC latency minimization. These fast algorithms serve as a way to quickly compute network design as well as facilitates comparison with our optimization model. Given these three parameters are of utmost importance in NFV, these three algorithms enable us to see which of these three parameters give as close as possible performance to the optimization model.

The optimization model, as well as the first three algorithms, considers a static TSP network, whereby traffic requests are assumed to be known ahead in time. Hence we also propose a fourth algorithm that considers dynamic traffic requests. We now present four heuristic techniques.

Problem: Given a network graph $G(V, E)$, along with a set of IP routers, Ethernet switches, ROADMs as well as DCs

Algorithm 1 For Optimizing Server Utilization

```
1 $\forall T^n_{abkm}$, we create a temporary set of VNFs in $\{SC_n\}$ and
  denote this as $\phi_{abkmn} = \{SC_n\}$
2 Denote $< T^n_{abkm} >$: sorted set of traffic requests in
  descending order of granularity $|T^n_{abkm}|$.
3 while $< T^n_{abkm} > \neq \varnothing$ do
4     max $[< T^n_{abkm} >] + \sum_{\exists < T^n_{abkm} >} T^n_{abkm} + \mu = |S|$,
      where $S$ is the capacity of the server.
5     Choose $V_i \in [PM^k_{ab}]$ to place group of requests
      obtained from step 4.
6     Provision the group of requests selected in step 4.
7     For the assignment above repeat such that each
      $\phi \in \phi_{abkmn}$ is assigned in a way such that:
      $\phi_{abkmn} \rightarrow V_i \neq V_j$ and for every $< T^n_{abkm} >$ that is
      provisioned and the availability value exceeds a
      threshold.
8     Delete group of requests obtained in step 4 from
      $< T^n_{abkm} >$.
```

Algorithm 2 For Optimizing VNF License Cost

```
1 $R^{(n)}_{SC}$ = Rank of the $n^{th}$ service function chain in terms of
  the number of traffic requests attached to the SFC.
2 $\forall T^n_{abkm}$, $Y_n = \sum_{a,b,k,m} T^n_{abkm}$ is the total traffic assigned
  to a particular $n^{th}$ SFC.
3 From $Y_n$, we create the set of all $Y_n$ such that
  $Y = \{Y_1, Y_2, \ldots, Y_p\}$, where $p$ is the number of possible
  SFCs.
4 $Y = sort[Y]$ in descending order.
5 For every SFC in $Y$, $R^{pop(n)}_{SC} = arg[Y]$, where $R^{pop(n)}_{SC}$ =
  Rank of $n^{th}$ SFC based on the granularity of assigned
  traffic to it.
6 $F_n = R^{pop(n)}_{SC}.R^{(n)}_{SC}$ is the final rank of the SFC.
7 $temp = \bigcup_{n \in (1,p)} SC_n$
                                        // Provision the traffic
8 for $SC_n \in temp$ : do
9     Compute $DC_i$ with max $T^n_{abkm} \rightarrow SC_n$
10    For this DC, provision $SC_n$: $F_n \rightarrow DC_i$
11    Provision each request from the set to different
      servers in the DC (or across DCs in case of
      geo-redundancy) for providing the desired
      availability.
12    Delete $SC_n$ from $temp$
13    Repeat till $temp \neq \varnothing$
14 For any new traffic request, we add the request to $T$ and
   recompute the rank as in step 6.
```

Algorithm 3 For Optimizing Delay

```
1 Sort requests in ascending order of target latency
  (request with tightest latency comes first)
2 Select request $T^i_{abkm}$ from the sorted set:
3 For each VNF type of this $SC_i$, choose VMs considering
  the redundancy.
4 if VMs within the same servers are available then
5     provision the request.
6 else if VMs within the same DC pod are present then
7     provision the request.
8 else if VMs within the same DC are available then
9     provision the request.
10 else
11     provision the request in VMs at geo-redundant DCs.
12 Repackage these by moving the SFCs on VMs within
   the same server or DC pod.
13 Repeat till all the request are provisioned and
   availability is achieved at the desired level.
```

at the core and in COs, we want to provision a given traffic matrix T, whose element T^n_{abkm} is as defined in Table 1.

Algorithm 1 is designed to maximize server utilization and assumes that server replication is sufficient for availability planning. The latter assumption is most prevalent in contemporary data-centers. Algorithm 1 sorts traffic requests into a descending order based on traffic granularity (bandwidth). Subsequently, the algorithm provisions those traffic requests that will occupy a full server and then a group of requests such that the group will nearly occupy a full server and so on. Algorithm 1 then provisions this group while adhering to the availability requirements of the group by incorporating sufficient redundancy. The grouping, provisioning and redundant provisioning for availability continue till all the requests are provisioned.

In step 1, for each service request, we list the set of VNFs required to be traversed by the associated SFC (i.e., SC_n). We then assign this set of VNFs to ϕ_{abkmn}. In step 2, we sort traffic requests for each SFC type in the descending order. The provisioning procedure starts from step 3. We select the largest request in the sorted set along with one or more requests with lesser bandwidth requirements such that the combination "fits" into a server. Though there may be some space (i.e., μ) remaining. We then choose a server such that it belongs to a DC along the path of the chosen traffic requests. In step 6, the selected group is provisioned at the chosen server. The provisioning procedure is repeated for each VNF type in the set ϕ_{abkmn}. Additionally, traffic requests are assigned to the replicated VNFs ($\forall \phi \in \phi_{abkmn}$) such that the availability threshold is met. Finally, the selected requests (in step 4) are deleted from the set. The procedure is repeated until all the requests are provisioned. The complexity of this algorithm is of the order $O(N^2 logN)$, where N is the number of traffic requests.

This algorithm works well under the following assumptions: i) VNF replication cost is linear and additive; ii) DC space is designed with few constraints to scalability; and iii) the hop count for a service function chain n, $h(SC_n)$, if larger than the shortest path, does not severely impact latency.

Dynamic Addition and Deletion of Traffic: For the case of dynamic traffic, we propose the following sub-module to Algorithm 1. For every new traffic request T^n_{abkm}, our goal is to find the group that can just about take T^n_{abkm} as its newest member i.e., the group which after taking T^n_{abkm} has the least residual capacity remaining. For example, if there are two groups of granularity 8.5 Gbps and 9.5 Gbps occupying two servers, each of capacity 10 Gbps and the new request is of granularity of 0.4 Gbps. Our algorithm pairs the request with the second group, i.e., 9.5 Gbps.

When a suitable match is not found, then a new server is chosen for provisioning the request. To justify the presence of the new server, *temp* time is allotted to find a few other requests for pairing. If the server is unable to find new requests, then we delete the provisioned request and it is assumed that the request cannot be provisioned. It is critical to choose the correct value of *temp*. The deletion of a server does not impact CapEx but does optimize energy and management effort (i.e., OpEx). Algorithm 1 is fast and is optimized for conserving server space.

We now propose Algorithm 2 that is optimized for VNF license cost. By this, we imply that VNF vendors tend to charge slab based pricing to a provider whether or not a VNF is utilized. Hence it is in the service provider's interest to maximize VNF utilization. To this end, we begin by ranking the SFCs in two ways: number and granularity of traffic requests for each SFC type. Next we provision high ranked SFCs along with enough redundancy to meet their availability values. Typically, this is analogous to creating H cuts of equal measure in H sub-graphs, where H cuts together satisfy the availability requirement. Without loss of generality, we assume that these H cuts shall be in geo-redundant DCs. If geo-redundancy is not required, then H cuts can be in the same DC. Further, if over-provisioning is required the value of each cut is determined by an over-provisioning factor. Steps 1–6 in Algorithm 2 discuss the ranking of service function chains based on the number of traffic requests and granularity of the traffic assigned to a SFC. We subsequently obtain the rank of a service function chain F_n and then provision traffic based on this rank with backup VNFs fulfilling availability and geo-redundancy requirements. Due to fractional server utilization, the complexity of Algorithm 2 is of the $O(N^2)$, where N is the number of traffic requests.

Algorithm 3 achieves latency minimization while maximizing server utilization by placing the VNFs of the SFC in close proximity. If the VMs in the same group are deployed in nearby hosts, then latency can be reduced [56]. We sort traffic requests based on the targeted latency. We then start provisioning traffic requests with tightest latency bounds, followed by others from the sorted set. The algorithm initially checks whether VNFs of a traffic request can be placed in the same server and, if not, checks for VMs available in the same pod. If the algorithm does not find such VMs, VNFs can be placed anywhere in the same DC. Once we find the placement of VNFs, we repackage some of the VMs to attain good server utilization, whereby we also adhere to availability values by placing service requests on redundant VNFs. The work in [27] presents a method to measure the affinity of network functions. This affinity information can be utilized for Algorithm 3.

Algorithm 4 For Dynamic Traffic Provisioning

```
1  for $T^n_{abkm}$ do
2      $prevX = Edge\_Switch$
3      for $vnf\_type$ in $\{SC_n\}$: do
4          $i = 0$
5          while $i < redundancyCount$ do
6              $X$ = choose from set of VNF instances of
               $vnf\_type$ with available capacity $> |T^n_{abkm}|$.
7              if $X = \varnothing$ then
8                  $X$ = Create a new VNF instance of
                   $vnf\_type$ at a vacant server.
9              Provision $T^n_{abkm}$ on shortest path connecting
               $X$ and $prevX$ of the $SC_n$.
10             $prevX = X$
11             $i = i + 1$
```

In algorithm 1–3, we consider a static set of demands. However, requests often arrive dynamically. For the case of dynamic traffic, we choose Algorithm 1 as the basis to derive Algorithm 4 (our dynamic traffic algorithm). The reason for this being that Algorithm 1 is optimized for server utilization, which is a key constraint for dynamic traffic provisioning. The idea behind Algorithm 4 is that we create a new VNF instance only if no other VNF instance of the same VNF type is able to accommodate the new request. Once a new request arrives, the algorithm checks for all available VNF instances of each VNF type associated with the SFC of the traffic request. The availability aspect is also addressed by Algorithm 4 – provisioning a request to enough redundant VNFs of each VNF type in the SFC till the availability measure is achieved (*redundancyCount* in step 4 represents the number of redundant instances for each VNF type in SFC). Each SFC is assumed to be initiated from an Edge Switch as in Step 2 (entry point of NFV infrastructure). The time complexity of algorithm 3 and 4 is $O(N \cdot S)$, where S is the number of servers in the NFV infrastructure.

V. EVALUATION

We built a discrete event simulator in Python for the four heuristics and also modeled the constrained optimization in Gurobi (on an Intel Core i7 7^{th} generation machine with 16GB RAM). The reason behind using a customized python-based simulator is that the results of the optimization formulation are fed as an input to design the network in the simulator.

We model an TSP network (as described in Section II), with 6 core nodes and each core node is connected to a metro ring of 8 nodes. The P-Router at the core node consists of a cross-connect with IO cards of 4×100Gbps configuration. The ROADM at a core node is 6-degree colorless, directionless ROADM and is connected to the P-router. Each metro node consists of an Ethernet switch of the capacity 1Tbps, with up to 20 cards, each of varying IOs. Each metro ring

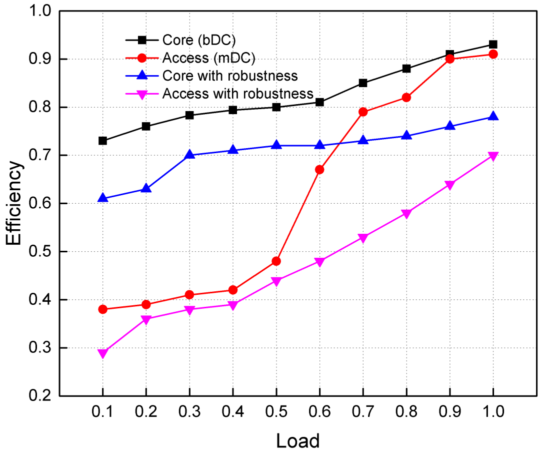


FIGURE 4. **Comparison of efficiency in the core and access as a result of the optimization.**

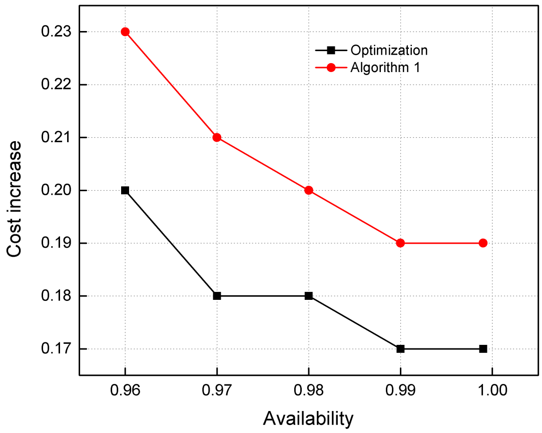


FIGURE 5. **Impact of server's availability on cost.**

is connected to a P-Router in a dual-homing configuration. We also model 3 DCs connected to the core nodes. DCs are modeled as a leaf-and-spine architecture. Servers (10 Gbps capacity) in the DC are connected to a ToR switch with 48-ports. ToR switches are connected to 64-port aggregation switches (10 × 100Gbps ports for uplink). Each metro-node is connected to two PE routers. PE routers are connected to a distribution network that contains multiple COs. Each CO is connected to 1000 residences along with the edge routers of the enterprises. The COs are modeled as an mDC and can have up to 100 servers. We assume 20 VNF types and 40 SCs. The average SFC length is 4.5 VNFs.

Load is calculated as the ratio of the traffic sent by all the users (residences and enterprises) in the network, to the total input capacity of the network. The input capacity is computed by adding all the maximum bandwidths across all the input ports in the network. Load is hence in interval [0,1]. For cost computation in USD we assume server price of 1K, VM price of 0.5K, P-router 200K, PE router 40K, ROADM 150K (6-degree with 100Gbps transponders) and Metro Ethernet switch 24K.

Next, we investigate the impact of the difference between the availability of telecom gear and IT equipment on network design. We assume that telecom equipment has five to six nines availability, while IT-equipment has availability in the range of 0.96 to 0.999.

Fig. 4 shows a comparison of the efficiency computed as VMs' occupancy as a function of load. To generate this plot, we ran the optimization module multiple times and averaged the result. The lowest-cost solution was developed by the optimization module. In particular, we are interested in the efficiency in two parts of the network – the core and the access. The efficiency at these two places will determine the impact of approaches like CORD vis-a-vis core data-centers. We also computed efficiency with uncertainty in load. To measure uncertainty we considered robustness of 30%, i.e. 30% of the traffic values could swell to their peak, while the remaining 70% were stationary at some known average value. However, we do not know ahead in time which 30% traffic requests can swell to their peak values. For generating this graph, we assumed that the difference between peak and average values was 25%. As shown in Fig. 4, the efficiency of VM utilization in the core is significantly better than in the access when no robustness (no uncertainty) is considered. Even with robustness, the drop in efficiency is about 13% despite almost 25% granularity variation in 30% of the connections. A similar result is seen for the access part of the network, where initially there is a 19% difference between the robust and non-robust solutions. Key to note is that the efficiency in the access (without robustness) is low – of the order of 40% at low loads. The efficiency of a robust solution at low-loads is similar. However, the efficiency in the access rises significantly when there is no uncertainty at high-loads. This is key to determining CO sizes. When robustness is added, the efficiency rises gradually, implying that the access is unable to support uncertainty in traffic without support from the core.

Fig. 5 presents a plot of additional (excess) network cost required for high-availability as a function of load. In the figure, we compare the optimization model to Algorithm 1 pertaining to this cost increase. It is observed that cost lowers with better server availability. However, beyond availability of 0.99, there is no further improvement. On average, the heuristic algorithm is 13% worse (more expensive) than the optimization formulation.

In Fig. 6, the maximum load that can be supported by the system for different server availability values is presented. As the server availability increases, there is a spurt in the load that can be supported for non-robust conditions (i.e., no uncertainty). This is because the number of parallel connections required to maintain high-availability drastically drops. Subsequently, as the gap between server and telecom availability reduces, the load supported matures to a steady-state value. The key takeaway is that by simply adjusting the architecture to design for availability, it is possible to create a high-availability framework from moderately reliable servers. In this regard, the over-provisioning of bandwidth is

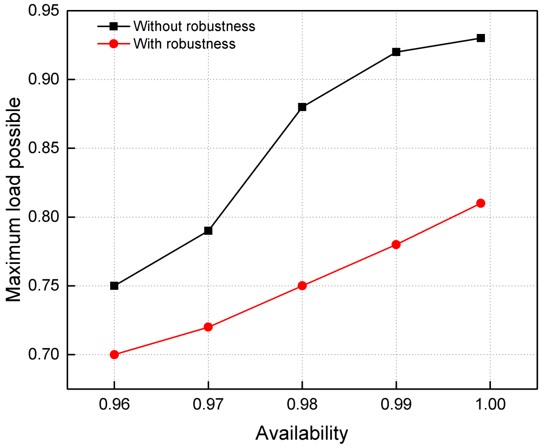


FIGURE 6. **Impact of server's availability on the maximum network load supported.**

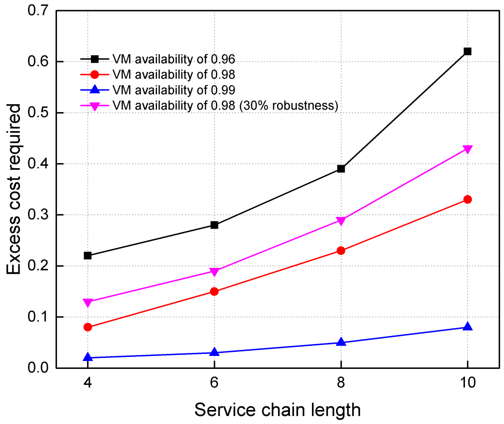


FIGURE 7. **Impact of service function chain length on cost.**

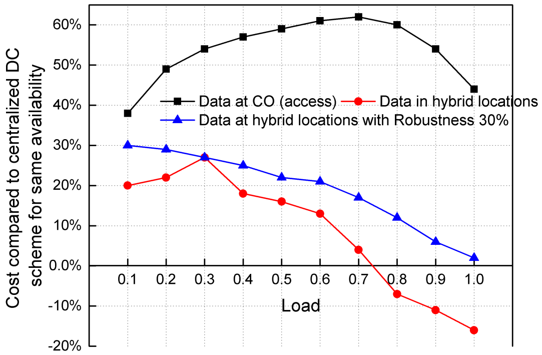


FIGURE 8. **Cost comparison of schemes with centralized DC scheme.**

key. We achieved the result with 12 free ports for uplink in the ToR switches. However, if we keep more than 12 ports for uplink at the TORs, the success does not rise in a linear fashion, implying higher cost. The lower curve in Fig. 6 shows the impact of robustness on maximum load supported as a function of server availability. We observe that unlike the spurt in supported load with the increased availability of servers, there is only a linear gradual increase in supported load when we deploy the robust formulation. To support the uncertainty, there is an overall reduction of about 11% load that can be supported.

Shown in Fig. 7 is a variation of the plot in Fig. 5, with consideration of service-chain length. It is interesting to observe that as long as there is no requirement of geo-redundancy, and only a need that the availability is above a threshold, the size of the chain has minimal impact on the network design. In this figure, we consider excess server costs with given different VM availability measures (0.96, 0.98 and 0.99). The non-linearity increases with lower availability value. This means that the number of excess VMs required to circumvent lower availability is more than when the VM availability is high. One would expect a linear increase, but due to the atomicity of the load (fractional VM utilization) and uncertainty in traffic, the increase turns out to be non-linear. We also considered the impact of uncertainty through robustness constraints shown in eqs. (24-32). As can be seen, there is a sizable impact of uncertainty. Even for high-availability servers (0.98 availability), to support the robustness of 25% peak-to-average, we require 36% more servers and hence a major cost increase. This result shows that there is no quick way to plan a network with uncertainty. To make matters worse there is a non-linear increase of servers with SFC length, which makes the design complicated.

In Fig. 8, the results evaluate where to place a service-chain in terms of cost-incurred as a function of network load. Here, we consider the base case of keeping all the VMs in the centralized DCs and compare two options: keeping the data at all the COs and a hybrid strategy in which the optimization technique places the primary VM at the CO and the backup at the central DC. When we keep the VMs at the COs, there is an initial increase in cost as there is a linear addition of servers with increasing load. However, the server utilization at COs is not linear with load – there is a catch-up process between adding servers and filling these up with VMs. The catch-up process stabilizes at medium load (0.4-0.7) (slope of around 0.2). Subsequently, the cost reduces as the installed capacity of the servers is now put to good use. This is a sweet spot for large networks. In the hybrid case, the cost initially increases compared to a centralized solution. This is because COs need to be populated with servers and VMs. Subsequently, the cost decreases because centralized entities are used instead of entities at the edge. At a load of 0.7, the hybrid cost is lower than that of the centralized DC – this is because the utilization is better in the edge with a high load. The cost then stabilizes at full capacity to a value of about 20% less than the centralized option. With uncertainty, however, the cost of the hybrid option tapers to the same as the centralized option. Here, we assume 30% of the connections can be at peak value and the peak-to-average granularity is 25%. With uncertainty, the cost of the hybrid scheme is initially 20% less than the scheme with data at the COs, and thereafter tapers to the same cost as the full centralized scheme. Hence we conclude that it makes more sense to keep data in the core of the network

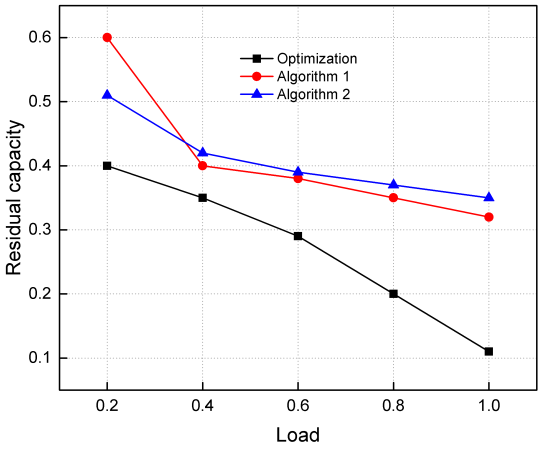


FIGURE 9. Average residual server capacity with load.

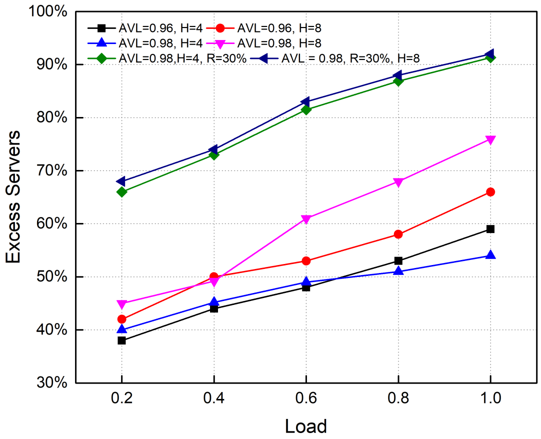


FIGURE 10. Excess cost considering robustness.

than at the access, especially if the data is seen to be dynamic and requests are uncertain and at low/medium loads.

Fig. 9 shows the comparison of the first two algorithms with the optimization formulation regarding the residual server usage of the schemes. For this graph, we first ran a traffic generator for 2000-time-epochs, defining an epoch where at least 2.5% of the traffic connections change in magnitude along with source-destination pair. From this, we calculate worst-case traffic on each source-destination pair and average these out as a function of time. We use this average value to compute the traffic matrix T. This is the input to the optimization model. The two algorithms do not make any time-epoch based assumptions. The fact is that we have cut a corner by assuming that one or multiple VMs in combination perfectly fit a server. This is to preserve the sanctity of Algorithm 1.

In Fig. 10, we plot the excess number of servers (as a percentage) for obtaining 99.9% overall SFC availability for different sized SFCs as a result of the optimization model along with robustness. There are 6 curves in the figure. Naturally, the best performance is for the case where server availability is the highest (0.98), and SFC size is lowest (4-VNFs), which may be collocated on the same server or across adjacent servers. The curve for slightly lower availability (0.96) and a similar number of VNFs (H = 4) per SFC is also almost similar with a 6% divergence at high-loads. The next two curves are for a higher number of VNFs (H = 8), and server availability of 0.98 and 0.96 respectively. We see that the excess server plots are initially grouped together, but then there is a fast divergence between the lower and higher availability server plots. The average difference between the lower and higher (0.96 and 0.98) availability servers in terms of excess servers required is about 12% at high loads. The final two curves are when we incorporate a 30% uncertainty in traffic with a peak-to-average value of 25%. On average, 67% excess servers are required for meeting the uncertainty in traffic. For this curve, we assume the SFC size of 4 and 8 and availability of 0.98. The key takeaway from this plot is that it is the SFC size which is more prevalent than availability, especially when the availability numbers are fairly high (>0.96). This plot is instructive towards understanding how to design the network when there are disparate sized SFCs.

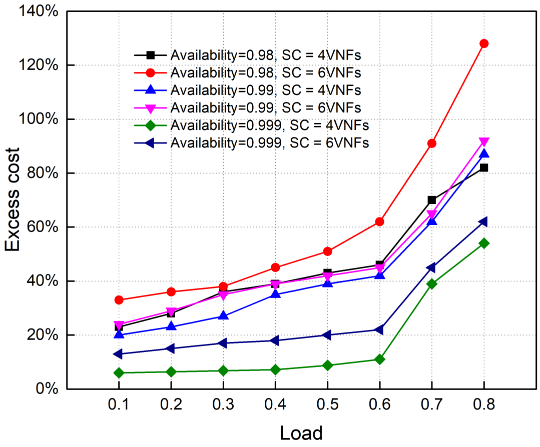


FIGURE 11. Excess cost considering availability of the complete SFC.

The results in Fig. 11 show the excess cost for achieving complete SFC-level availability. These results are directly due to eqs. (13)-(18) that focus on achieving availability for the entire SFC. To obtain this result, we ran the optimization program 100 times at each load value. VNFs are assumed to have average availability from 0.98 to 0.999 as shown in the figure. A SFC can contain 4 or 6 VNFs, whereby the target is to achieve a combined SFC availability of 0.999. The cost of VNFs is assumed to be one-fourth of the corresponding PNF cost [10]. Shown in the figure is a plot of the excess cost when we consider entire SFC availability in the design. Naturally, the low availability of the VNFs results in higher costs to maintain the SFC availability. There is a significant cost difference between VNFs of 0.99 and 0.999 availability value, while comparatively there is not much difference between VNFs of 0.99 and 0.98 availability. Another important observation is that the cost increases exponentially with increasing load. This is because of the number of times the VNFs have to be instantiated across the network, to keep the total SFC

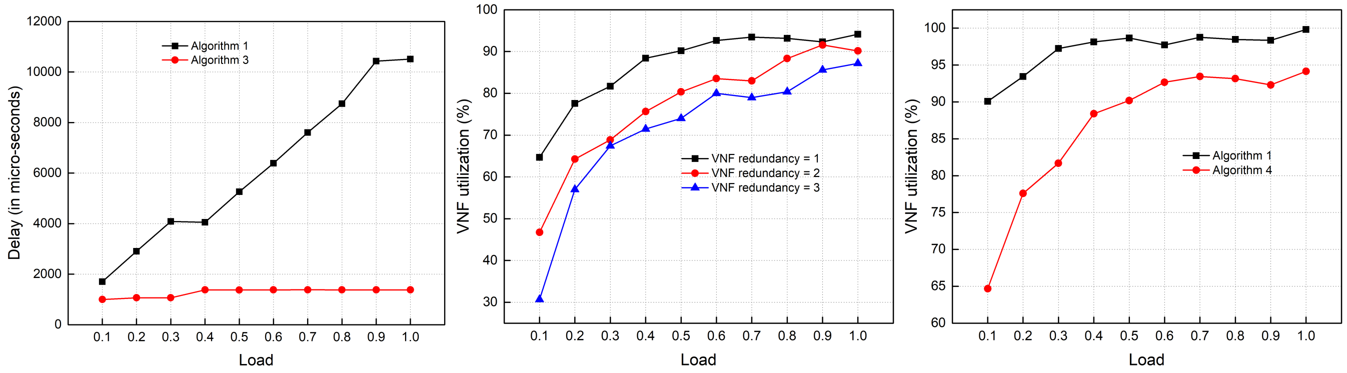


FIGURE 12. a) SFC delay with load (left), b) VNF utilization achieved by Algorithm 4 with different VNF redundancy values (middle), c) Comparison of Algorithms 1 and 4 based on VNF utilization (right).

traversing delay below 10ms (our simulation upper bound). A design takeaway is that it is much more advantageous from a cost perspective to reduce the gap between target SFC availability and VNF availability irrespective of the number of VNFs in the SFC. In fact, this finding holds even for large SFCs of 8 and 10 VNFs. For extremely high loads, the excess cost for achieving target SFC availability is almost 3× of VNF cost. Though PNF is 4× the VNF cost, the perceived cost post incorporating availability is only 1× more.

To show the effectiveness of Algorithm 3 for latency optimization, we compared Algorithm 3 with Algorithm 1 in terms of end-to-end SFC delay in Fig.12a. Here, we have only considered SFC delay within a DC. Switches in the DC are assumed to have latency in the range of 10-200 microseconds loosely proportional to the load. Since Algorithm 3 attempts to place VNFs of the SFCs in close proximity of each other, it outperforms Algorithm 1.

In Fig. 12b, we show the results obtained from Algorithm 4, which is designed for provisioning dynamic traffic requests in the NFV infrastructure. The algorithm attempts to optimize the VNF utilization by assigning an incoming request to one of already instantiated VNFs (having enough capacity to accommodate the new request). In order to analyze the effect of redundancy of VNF instances, we experimented with three different cases: 1) without using any redundancy of VNFs; 2) with two VNF replications; 3) with three VNFs replications to circumvent low-availability. The results show that at lower loads, VNF utilization is low as assigned requests do not fill the instantiated VNFs. However, VNF utilization increases with an increase in the load and stabilizes at a load of around 80%. It is pertinent to note that with the higher values of VNF redundancy, VNF utilization is lower as each request has to be replicated in multiple disjoint VNF instances that lead to lower utilization of corresponding VNFs. This result implies that providing high-availability VNFs will result in high-cost for service providers.

Algorithm 4 is an extension of Algorithm 1 as both attempt to maximize VNF/server utilization. Hence, we compared their VNF utilization capabilities in Fig. 12c. Algorithm 1 is designed for a static case, where traffic is known ahead in time, while Algorithm 4 deals with dynamic traffic requests. As a result, Algorithm 1 outperforms Algorithm 4 at each load-point. It is pertinent to note that at lower loads, the difference between both of these algorithms is high because of the dynamic nature of traffic requests. At low loads, there would not be enough requests to fill all the instantiated VNFs in Algorithm 4 as generated traffic requests randomly choose SFCs. Algorithm 1 assigns traffic requests by creating groups that will suffice to fill a VNF. With the increase in load, Algorithm 4 catches up with Algorithm 1, as Algorithm 4 attempts to place new requests on any existing partially utilized VNF instance leading to high VNF utilization. Despite this catching-up, there is a 6% difference at high-loads between Algorithm 1 and Algorithm 4.

VI. RELATED WORK

The Network Function Virtualization (NFV) paradigm enables flexibility, programmability and implementation of traditional network functions, in the form of Virtual Network Functions (VNFs) [32], [43]. Today, cloud service providers use Virtual Machines (VMs) for the instantiation of VNFs in data-centers. To instantiate multiple VNFs in a typical scenario of Service Function Chains (SFCs), many important objectives need to be met simultaneously.

VNF placement and service-chaining is a well-researched area. Authors in [2], [33], [35], [48] proposed solutions for VNF placement and service-chaining by formulating the problem as a constrained optimization model. In [5] is an optimization model for constrained shortest path routing for VNF forwarding graphs as well as a neighborhood search scheme to meet SLOs. These approaches assume a static set of service requests as an input to the model. Authors in [11], [25], [34] proposed approaches for VNF placements in a dynamic environment. In [6], [44] are VNF placement approaches based on predicting future resource requirements. In [7] is a VNF placement scheme that considers user's mobility and dynamic demands as well as end-to-end latency. Authors in [54] proposed an elastic network service chain, which utilizes a fine-grained hybrid scaling method to achieve both NFV efficiency and scalability. In [41] is a genetic algorithm for resource allocation in NFV. The algorithm handles initial placement as well as scaling of VNFs for additional resources. In [26] is a fully collaborative hybrid resource allocation algorithm. The resources are allocated to edge servers and cloud environments. Edge servers can be part of mDCs while cloud environments can be viewed as bDC.

In [14] is an approach that considers dynamic traffic due to VNF migration. The proposed ILP attempts to minimize the aggregated cost of optical circuit reconfiguration, cloud and bandwidth resources. Similar to our work, they have also considered optical interconnections between NFV DCs. In [13] is a strategy for long-term VM allocation to handle peak traffic. The proposed ILP model attempts to minimize the cost of bandwidth, cloud and VNF deployment. However, our work is fundamentally different from [13], [14] as we aim to design a highly-available NFV-enabled service provider network.

The approaches mentioned above do not take into consideration the availability requirements of the service. The service availability is a factor of utmost importance since the NFV premise consists of different hardware, software and network components that are prone to failures [22]. We have considered a real service provider network and included all relevant components while calculating SFC availability.

Researchers have proposed different approaches to provide high-availability cloud DC services. In [47] is an approach to select the best availability zones for a request that satisfies QoS requirements. In [53] is a formulation that uses mean-time-to-failure (MTTF) and mean-time-to-recover (MTTR) metrics to identify the availability of VMs. The proposed approach ensures that at least one set of VMs is available during Shared-Risk Node Group (SRNG) failures. Our work goes beyond this minimum requirement towards achieving availability at a value that is pre-decided (say five 9s).

ETSI presented guidelines for the NFV reliability design in [16]. The document showcases various methods and techniques for designing NFV services with high-availability. Further, the document emphasizes the role of redundancy in designing a highly-available NFV infrastructure. In [9] is a use-case of NFV for resiliency. The work in [12] considers different VNF placement strategies for SFC availability in the NFV data-center using redundancy. The focus of [12] is on creating analytical models for reliability scenarios. In [40] authors proposed an optimization model for VNF placement and traffic routing with an objective of maximizing the SFC reliability. They have also proposed a greedy shortest path based heuristic approach for VNF placement considering the availability of an SFC. In [17] is a heuristic that assigns backups greedily to VNFs within an SFC. However, these works are limited to a data-center and do not consider other elements of a provider network and the effect of factors such as geo-redundancy. More importantly, server license cost and latency also need to be considered, which is one of the premises of our heuristic work.

In [31] authors presented online algorithms for availability-aware VNF placements. The scope of their work is rather limited to the mobile-edge cloud (MEC). In [46] is a two-phase model that captures structural dependencies and failure dynamics of an NFV-enabled network. The authors have considered the availability of different IT and network equipment to realize high-available NFV infrastructure. They have evaluated service unavailability as a function of failure rates of disparate constituents of an NFV infrastructure. In contrast, we aim to design a service provider network that provides high-available services. In [8] is a VNF placement approach considering high-availability. Authors presented an ILP and a resource-efficient VNF placement for the high-availability deployment of a cluster over geo-distributed clouds. The work in [8] does not considers a provider network setting and other SLOs such as delay.

RABA [55] is an approach for shared and dedicated protection of VNFs that attempts to minimize resource consumption while ensuring high-availability. Similarly, our optimization model also considers maximizing service provider profit by minimizing additional backups while ensuring high-available services. In [50] is an approach that considers path and VNF protection to minimize resource usage. In [1] is a VNF placement approach that considers availability as one of the constraints. The proposed ILP model aims to maximize MTTF of VNFs and physical machines, part of a network service. The primary difference between our scheme and these approaches is that we consider designing a service provider network while providing SLOs such as availability and delay.

In [52] is a delay-sensitive and availability-aware VNF scheduling algorithm. The proposed method is an integer non-linear formulation. In contrast, due to the uniqueness of our optimization formulation, we have modeled both availability and delay in a linear program. Additionally, we have also considered the intricacies of a service provider network. The work in [51] is comparable to our work and discusses a trade-off between deployment cost and availability, presenting a genetic algorithm apart from a linear programming model with availability constraints. However, the primary difference is that our optimization model designs a cost-effective network considering the availability of SFCs.

In [15], the authors proposed two heuristics for highly-available SFC design, whereby the main idea is to increase SFC availability. The issue with the proposed approach is that while locally optimizing resources, there is a possibility of getting a globally cost-inefficient solution. Authors in [4] proposed an optimization model for guaranteed availability of services by replicating VNFs. The model assigns the service requests to master and slave VNFs. Master VNFs are the active VNF instances, whereas slave VNFs will be active once the master VNF fails.

In [23] is a MILP approach as well as an algorithm that considers delay and locality constraints for VNF placement within a DC. The objective of their approach is to minimize the communication delay between VNF instances to enhance end-to-end QoS of the SFC. They have not considered providing high-availability SFCs in

TABLE 4. Comparison between our proposal and relevant related works.

Work	Traffic uncertainty considered?	Dynamic traffic scenario considered?	Traffic request's delay considered?	Service provider network considered?	Availability of traffic requests considered?	Multi-layer solution?	Series-parallel combinations of VMs for availability calculation?
Fan et al. [17]	No	No	Yes	No	Yes	No	No
RABA [55]	No	Yes	No	No	Yes	No	No
Yang et al. [52]	No	No	Yes	No	Yes	No	No
Qu et al. [40]	No	Yes	Yes	No	Yes	No	No
Eramo et al. [14]	No	Yes	No	No	No	No	No
Yala et al. [51]	No	No	No	No	Yes	No	No
Our Proposal	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**	**Yes**

their optimization model. Authors in [30] presented an availability-aware VNF deployment model in DCs. For improving resource utilization, redundant VNFs are shared and multi-tenancy is used. In [30], authors formulated an ILP with an availability constraint. However, they have mentioned that because of non-linearity introduced by the availability constraint, the model cannot be solved. Hence they have proposed a separate heuristic – joint deployment and backup scheme (JDBS). In [49] is a VNF placement scheme with a consideration of availability in DCs. They have proposed an affinity-based placement algorithm to reduce load on physical links. Similar to [30], [49] also formulated the availability-aware VNF placement problem; however, authors do not solve the problem because of induced non-linearity in the availability constraint. We have used a smart data structure, the one-hot vector, that does not induce any non-linearity, while keeping the problem convex.

Authors in [22] explained the challenges for resilient VNF infrastructure models and showed that VNFs are not carrier-class. In [24] is proposed VNF placement strategies that take resiliency into account. The work is limited to the resiliency of an SFC under different failure scenarios such as DC node and link failure. In contrast, our work is availability centric and not only considers node and link failures but also takes into account the VM and server failures. It is important to note that VM and server failures are more prominent in an IT cloud. In [45] are fault prevention and failure recovery schemes for service chains. In [45] is achieved an acceptable level of network path survivability and a fair allocation of resource between different demands in the event of faults or failures. In [36] is an approach to detect failures by monitoring physical-layer statistics in NFV environments. In [28] is an approach using a self-adaptive paradigm (Brownout) aware load-balancing for correlated and cascading failures in a cloud.

REINFORCE [29] is a framework for providing VNF failure detection and failover mechanisms. Upon VNF failures, REINFORCE attempts to minimize state transfers to the standby VNFs. REINFORCE is complementary to our approach. It can be integrated with our approach for enabling state transfers to redundant VNFs. In [38] is an SFC failover mechanism that uses OpenFlow group tables to restore service in less than 50ms. In [19] is an approach for providing high-availability SFCs using rejuvenation and live migration. The proposed approach follows a preventive maintenance scheme while considering MTTF and MTTR of different cloud components.

Our optimization model aims to maximize the service provider's profit. VNF licensing and pricing will play an important role in inculcating NFV. In [37] is a dynamic pricing algorithm to increase the revenue for infrastructure providers that gives insights into VNF pricing. We have presented a shorter version of this work, without robust optimization and no heuristics in [42]. In Table 4, we compare our proposal and the main related works.

VII. CONCLUSION

In this paper, we designed a service provider network that is spread across the access, metro and core networks using disparate network components and low-availability VNFs. We formulated a constrained optimization model whose objective is revenue maximization subject to availability measures that OTT services demand. The constraints of robustness that facilitate traffic churn were provided. A realistic regional/metro service provider was modeled. Three algorithms were proposed for optimizing various parameters of interest to a service provider while considering high-availability service requests. These algorithms attempt to maximize server utilization and minimize VNF licensing cost and delay. To handle dynamic traffic, a fourth algorithm was also proposed. Simulation results show comparative data for efficiency, latency and server utilization and validate our optimization model as well as algorithms. As future work, we want to consider that the bandwidth between VNFs for the same service function chain is not assumed to be constant. This makes the system design significantly complicated on account of meta-data that can now be tagged to flows between VNFs in a service function chain. We also want to consider various queuing models for calculating latency.

REFERENCES

[1] Y. Alahmad and A. Agarwal, ''VNF placement strategy for availability and reliability of network services in NFV,'' in *Proc. 6th Int. Conf. Softw. Defined Syst. (SDS)*, Jun. 2019, pp. 284–289.

[2] F. Bari, S. R. Chowdhury, R. Ahmed, R. Boutaba, and O. C. M. B. Duarte, ''Orchestrating virtualized network functions,'' *IEEE Trans. Netw. Service Manage.*, vol. 13, no. 4, pp. 725–739, Dec. 2016.

[3] A. Ben-Tal, L. E. Ghaoui, and A. Nemirovski, *Robust Optimization*, vol. 28. Princeton, NJ, USA: Princeton Univ. Press, 2009.

[4] M. Casazza, P. Fouilhoux, M. Bouet, and S. Secci, ''Securing virtual network function placement with high availability guarantees,'' in *Proc. IFIP Netw. Conf. (IFIP Netw.) Workshops*, Jun. 2017, pp. 1–9.

[5] D. Chemodanov, F. Esposito, P. Calyam, and A. Sukhov, ''A constrained shortest path scheme for virtual network service management,'' *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 1, pp. 127–142, Mar. 2019.

[6] X. Cheng, Y. Wu, G. Min, and A. Y. Zomaya, ''Network function virtualization in dynamic networks: A stochastic perspective,'' *IEEE J. Sel. Areas Commun.*, vol. 36, no. 10, pp. 2218–2232, Oct. 2018.

[7] R. Cziva, C. Anagnostopoulos, and D. P. Pezaros, ''Dynamic, latency-optimal vNF placement at the network edge,'' in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2018, pp. 693–701.

[8] T.-X. Do and Y. Kim, ''Topology-aware resource-efficient placement for high availability clusters over geo-distributed cloud infrastructure,'' *IEEE Access*, vol. 7, pp. 107234–107246, 2019.

[9] S. Dobson, D. Hutchison, A. Mauthe, A. Schaeffer-Filho, P. Smith, and J. P. G. Sterbenz, ''Self-organization and resilience for networked systems: Design principles and open research issues,'' *Proc. IEEE*, vol. 107, no. 4, pp. 819–834, Apr. 2019.

[10] J. Donovan and K. Prabhu, *Building the Network of the Future: Getting Smarter, Faster, and More Flexible with a Software Centric Approach*. Boca Raton, FL, USA: CRC Press, 2017.

[11] S. D'Oro, L. Galluccio, S. Palazzo, and G. Schembra, ''A game theoretic approach for distributed resource allocation and orchestration of softwarized networks,'' *IEEE J. Sel. Areas Commun.*, vol. 35, no. 3, pp. 721–735, Mar. 2017.

[12] A. Engelmann and A. Jukan, ''A reliability study of parallelized VNF chaining,'' in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2018, pp. 1–6.

[13] V. Eramo and F. G. Lavacca, ''Optimizing the cloud resources, bandwidth and deployment costs in multi-providers network function virtualization environment,'' *IEEE Access*, vol. 7, pp. 46898–46916, 2019.

[14] V. Eramo and F. G. Lavacca, ''Proposal and investigation of a reconfiguration cost aware policy for resource allocation in multi-provider NFV infrastructures interconnected by elastic optical networks,'' *J. Lightw. Technol.*, vol. 37, no. 16, pp. 4098–4114, Aug. 15, 2019.

[15] J. Kong, I. Kim, X. Wang, Q. Zhang, W. Xie, H. C. Cankaya, N. Wang, T. Ikeuchi, and J. P. Jue, ''Guaranteed-availability network function virtualization in inter-datacenter networks,'' in *Proc. Opt. Fiber Commun. Conf.*, 2018, pp. 1–3.

[16] ETSI. *Network Functions Virtualisation (NFV); Reliability; Report on Models and Features for End-to-End Reliability*. Accessed: Jun. 22, 2018. [Online]. Available: http://www.etsi.org/deliver/etsi_gs/NFV-REL/001_099/003/01.01.01_60/gs_nfv-rel003v010101p.pdf

[17] J. Fan, C. Guan, Y. Zhao, and C. Qiao, ''Availability-aware mapping of service function chains,'' in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Atlanta, GA, USA, May 2017, pp. 1–9.

[18] R. Govindan, I. Minei, M. Kallahalla, B. Koley, and A. Vahdat, ''Evolve or die: High-availability design principles drawn from Googles network infrastructure,'' in *Proc. Conf. ACM SIGCOMM Conf. (SIGCOMM)*, 2016, pp. 58–72.

[19] E. Guedes and P. Maciel, ''Stochastic model for availability analysis of service function chains using rejuvenation and live migration,'' in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops (ISSREW)*, Oct. 2019, pp. 211–217.

[20] A. Gumaste and S. Akhtar, ''Evolution of packet-optical integration in backbone and metropolitan high-speed networks: A standards perspective,'' *IEEE Commun. Mag.*, vol. 51, no. 11, pp. 105–111, Nov. 2013.

[21] A. Gumaste, T. Das, S. Sharma, and A. Kushwaha, ''How much NFV should a service provider adopt?'' *J. Lightw. Technol.*, vol. 35, no. 13, pp. 2598–2611, Jul. 1, 2017.

[22] B. Han, V. Gopalakrishnan, G. Kathirvel, and A. Shaikh, ''On the resiliency of virtual network functions,'' *IEEE Commun. Mag.*, vol. 55, no. 7, pp. 152–157, Jul. 2017.

[23] H. Hawilo, M. Jammal, and A. Shami, ''Network function virtualization-aware orchestrator for service function chaining placement in the cloud,'' *IEEE J. Sel. Areas Commun.*, vol. 37, no. 3, pp. 643–655, Mar. 2019.

[24] A. Hmaity, M. Savi, F. Musumeci, M. Tornatore, and A. Pattavina, ''Virtual network function placement for resilient service chain provisioning,'' in *Proc. 8th Int. Workshop Resilient Netw. Design Modeling (RNDM)*, Sep. 2016, pp. 245–252.

[25] P. Hong, K. Xue, and D. Li, ''Resource aware routing for service function chains in SDN and NFV-enabled network,'' *IEEE Trans. Services Comput.*, early access, Jun. 22, 2018, doi: 10.1109/TSC.2018.2849712.

[26] H. Huang, ''Collaborative resource allocation over a hybrid cloud center and edge server network,'' *J. Comput. Math.*, vol. 35, no. 4, pp. 423–438, Jun. 2017.

[27] A. S. Jacobs, R. L. do Santos, M. F. Franco, E. J. Scheid, R. J. Pfitscher, and L. Z. Granville, ''AMNESiA: Affinity measurement platform for NFV-enabled networks,'' in *Proc. IFIP/IEEE Symp. Integr. Netw. Service Manage. (IM)*, May 2017, pp. 899–900.

[28] C. Klein, A. V. Papadopoulos, M. Dellkrantz, J. Dürango, M. Maggio, K.-E. Årzén, F. Hernández-Rodriguez, and E. Elmroth, ''Improving cloud service resilience using brownout-aware load-balancing,'' in *Proc. IEEE 33rd Int. Symp. Reliable Distrib. Syst.*, Oct. 2014, pp. 31–40.

[29] S. G. Kulkarni, G. Liu, K. K. Ramakrishnan, M. Arumaithurai, T. Wood, and X. Fu, ''REINFORCE: Achieving efficient failure resiliency for network function virtualization-based services,'' *IEEE/ACM Trans. Netw.*, vol. 28, no. 2, pp. 695–708, Apr. 2020.

[30] D. Li, P. Hong, K. Xue, and J. Pei, ''Availability aware VNF deployment in datacenter through shared redundancy and multi-tenancy,'' *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 4, pp. 1651–1664, Dec. 2019.

[31] J. Li, W. Liang, M. Huang, and X. Jia, ''Reliability-aware network service provisioning in mobile edge-cloud networks,'' *IEEE Trans. Parallel Distrib. Syst.*, vol. 31, no. 7, pp. 1545–1558, Jul. 2020.

[32] Y. Li and M. Chen, ''Software-defined network function virtualization: A survey,'' *IEEE Access*, vol. 3, pp. 2542–2553, 2015.

[33] M. C. Luizelli, L. R. Bays, L. S. Buriol, M. P. Barcellos, and L. P. Gaspary, ''Piecing together the NFV provisioning puzzle: Efficient placement and chaining of virtual network functions,'' in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 98–106.

[34] T. Lukovszki, M. Rost, and S. Schmid, ''Approximate and incremental network function placement,'' *J. Parallel Distrib. Comput.*, vol. 120, pp. 159–169, Oct. 2018.

[35] S. Mehraghdam, M. Keller, and H. Karl, ''Specifying and placing chains of virtual network functions,'' in *Proc. IEEE 3rd Int. Conf. Cloud Netw. (CloudNet)*, Oct. 2014, pp. 7–13.

[36] M. Miyazawa, M. Hayashi, and R. Stadler, ''VNMF: Distributed fault detection using clustering approach for network function virtualization,'' in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2015, pp. 640–645.

[37] B. Naudts, M. Flores, R. Mijumbi, S. Verbrugge, J. Serrat, and D. Colle, ''A dynamic pricing algorithm for a network of virtual resources,'' *Int. J. Netw. Manage.*, vol. 27, no. 2, p. e1960, Mar. 2017.

[38] H. B. Nguyen, N.-T. Dinh, J. Oh, and Y. Kim, ''An openflow-based scheme for service Chaining's high availability in cloud network,'' in *Proc. Int. Conf. Inf. Commun. Technol. Converg. (ICTC)*, Oct. 2019, pp. 805–807.

[39] L. Peterson, A. Al-Shabibi, T. Anshutz, S. Baker, A. Bavier, S. Das, J. Hart, G. Palukar, and W. Snow, ''Central office re-architected as a data center,'' *IEEE Commun. Mag.*, vol. 54, no. 10, pp. 96–101, Oct. 2016.

[40] L. Qu, C. Assi, K. Shaban, and M. J. Khabbaz, ''A reliability-aware network service chain provisioning with delay guarantees in NFV-enabled enterprise datacenter networks,'' *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 3, pp. 554–568, Sep. 2017.

[41] W. Rankothge, F. Le, A. Russo, and J. Lobo, ''Optimizing resource allocation for virtualized network functions in a cloud center using genetic algorithms,'' *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 2, pp. 343–356, Jun. 2017.

[42] S. Sharma, A. Kushwaha, A. Somani, and A. Gumaste, ''Designing highly-available service provider networks with NFV components,'' in *Proc. 28th Int. Conf. Comput. Commun. Netw. (ICCCN)*, Jul. 2019, pp. 1–9.

[43] K. Shiomoto, ''Research challenges for network function virtualization—Re-architecting middlebox for high performance and efficient, elastic and resilient platform to create new services,'' *IEICE Trans. Commun.*, vol. E101.B, no. 1, pp. 96–122, 2018.

[44] Q. Sun, P. Lu, W. Lu, and Z. Zhu, "Forecast-assisted NFV service chain deployment based on affiliation-aware vNF placement," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Washington, DC, USA, Dec. 2016, pp. 1–6.

[45] M. M. Tajiki, M. Shojafar, B. Akbari, S. Salsano, M. Conti, and M. Singhal, "Joint failure recovery, fault prevention, and energy-efficient resource management for real-time SFC in fog-supported SDN," *Comput. Netw.*, vol. 162, Oct. 2019, Art. no. 106850.

[46] B. Tola, G. Nencioni, and B. E. Helvik, "Network-aware availability modeling of an end-to-end NFV-enabled service," *IEEE Trans. Netw. Service Manage.*, vol. 16, no. 4, pp. 1389–1403, Dec. 2019.

[47] M. Unuvar, Y. Doganata, M. Steinder, A. Tantawi, and S. Tosi, "A predictive method for identifying optimum cloud availability zones," in *Proc. IEEE 7th Int. Conf. Cloud Comput.*, Jun. 2014, pp. 72–79.

[48] A. Varasteh, M. De Andrade, C. M. Machuca, L. Wosinska, and W. Kellerer, "Power-aware virtual network function placement and routing using an abstraction technique," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2018, pp. 1–7.

[49] M. Wang, B. Cheng, and J. Chen, "Joint availability guarantee and resource optimization of virtual network function placement in data center networks," *IEEE Trans. Netw. Service Manage.*, vol. 17, no. 2, pp. 821–834, Jun. 2020.

[50] M. Wang, B. Cheng, S. Zhao, B. Li, W. Feng, and J. Chen, "Availability-aware service chain composition and mapping in NFV-enabled networks," in *Proc. IEEE Int. Conf. Web Services (ICWS)*, Jul. 2019, pp. 107–115.

[51] L. Yala, P. A. Frangoudis, G. Lucarelli, and A. Ksentini, "Cost and availability aware resource allocation and virtual function placement for CDNaaS provision," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 4, pp. 1334–1348, Dec. 2018.

[52] S. Yang, F. Li, R. Yahyapour, and X. Fu, "Delay-sensitive and availability-aware virtual network function scheduling for NFV," *IEEE Trans. Services Comput.*, early access, Jul. 9, 2019, doi: 10.1109/TSC.2019.2927339.

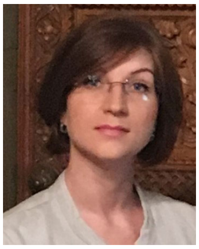
[53] S. Yang, P. Wieder, R. Yahyapour, S. Trajanovski, and X. Fu, "Reliable virtual machine placement and routing in clouds," *IEEE Trans. Parallel Distrib. Syst.*, vol. 28, no. 10, pp. 2965–2978, Oct. 2017.

[54] H. Yu, J. Yang, C. Fung, R. Boutaba, and Y. Zhuang, "ENSC: Multi-resource hybrid scaling for elastic network service chain in clouds," in *Proc. IEEE 24th Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Dec. 2018, pp. 34–41.

[55] J. Zhang, Z. Wang, C. Peng, L. Zhang, T. Huang, and Y. Liu, "RABA: Resource-aware backup allocation for a chain of virtual network functions," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2019, pp. 1918–1926.

[56] Y. Zhang, Y. Li, K. Xu, D. Wang, M. Li, X. Cao, and Q. Liang, "A communication-aware container re-distribution approach for high performance VNFs," in *Proc. IEEE 37th Int. Conf. Distrib. Comput. Syst. (ICDCS)*, Jun. 2017, pp. 1555–1564.

SIDHARTH SHARMA received the M.Tech. degree in computer science and engineering from the National Institute of Technology, Rourkela, India, in 2015. He is currently pursuing the Ph.D. degree with the Department of Computer Science and Engineering, IIT Bombay, Mumbai, India. His research interests include software-defined networks and network function virtualization.



ANNA ENGELMANN (Graduate Student Member, IEEE) received the M.Sc. degree in computer and communications systems engineering from the Technische Universität Braunschweig, Germany, in 2014, where she is currently pursuing the Dr.Ing. degree. Her current research interests include theory and performance of communication networks, protocols, routing, cryptography and coding, and reliability evaluation of network systems and services.

ADMELA JUKAN received the Dr.tech. degree *(summa cum laude)* in electrical and computer engineering from the Technische Universität Wien, the M.Sc. degree in information technologies from the Politecnico di Milano, Italy, and the Dipl.Ing. degree from the Fakultet Elektrotehnike i Racunarstva (FER), Zagreb, Croatia.

She was a Research Faculty Member with the Institut National de la Recherche Scientifique (INRS) and Georgia Tech (GaTech). She was a Visiting Scientist with Bell Labs, Holmdel, NJ, USA, from 1999 to 2000, and the University of Illinois at Urbana–Champaing (UIUC), in 2004. In 2015, she was invited for sabbatical at MIT. From 2002 to 2004, she was the Program Director of computer and networks system research at the National Science Foundation (NSF), Arlington, VA, USA. She is currently a Chair Professor of communication networks with the Technische Universität Carolo-Wilhelmina zu Braunschweig, Braunschweig, Germany.

Dr. Jukan was a recipient of an Award of Excellence for the BMBF/CELTIC project "100Gb Ethernet" and the IBM Innovation Award, in 2009. She has chaired and co-chaired several international conferences, including the IEEE/ACM IWqoS, IEEE ANTS, IFIP ONDM, IEEE ICC, and IEEE GLOBECOM. She was an elected Chair of the IEEE Optical Network Technical Committee (ONTC), from 2014 to 2015. She serves as a Senior Editor for the IEEE JOURNAL OF SELECTED AREAS IN COMMUNICATIONS. She is a Co-Editor-in-Chief of the Elsevier Journal on *Optical Switching and Networking* (OSN). She is the Dean of Studies for a joint degree program between computer science and engineering (IST). She was an elected Distinguished Lecturer of the IEEE Communications Society, from 2015 to 2017.

ASHWIN GUMASTE (Senior Member, IEEE) received the Ph.D. degree, in 2003.

He was the Institute Chair Associate Professor, from 2012 to 2015, and the JR Isaac Chair Assistant Professor, from 2008 to 2011, at IITB. He has held positions with Fujitsu Laboratories, USA, from 2001 to 2005. He has also worked with Cisco Systems, from 2000 to 2001. He was a consultant to Nokia Siemens Networks, where he focused on the emerging NGPON2. He has also held positions with MIT, from 2008 to 2009, the Lawrence Berkeley National Labs, in 2015, and Iowa State University, in 2009. He has been an Institute Chair Professor with the Department of Computer Science and Engineering, IIT Bombay, Mumbai, since 2005. His work on light trails has been widely referred, deployed, and recognized by both industry and academia. His recent work on omnipresent Ethernet has been adopted by tier-1 service providers and also resulted in the largest ever acquisition between any IIT and the industry. This has led to a family of transport products under the premise of Carrier Ethernet Switch Routers. His team has built a terabit capable SDN white box for the Ministry of Defense. He has 25 granted U.S. patents and has published about 175 papers in referred conferences and journals. He has authored three books in broadband networks. For research and development contributions, he received the S. S. Bhatnagar Award, in 2018; the DST Swaranajayanti Fellowship, in 2013; the Department of Atomic Energy's-SRC Outstanding Research Investigator Award, in 2010; the Department of Space's Vikram Sarabhai Research Award, in 2012; the IBM Faculty Award, in 2012, the National Academy of Sciences in India, the NASI-Reliance Industries Platinum Jubilee Award, in 2016; the Indian National Academy of Engineering's (INAE) Young Engineer Award, in 2010; and the INAE Fellow, in 2018.