Availability Aware SFC Embedding in NFV: A Clustering Approach

Pegah Torkamandi
Department of Computer Engineering and Information Technology (CEIT)
Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran
pegah.torkamandi@aut.ac.ir

Siavash Khorsandi
Department of Computer Engineering and Information Technology (CEIT)
Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran
khorsandi@aut.ac.ir

Bahador Bakhshi
Department of Computer Engineering and Information Technology (CEIT)
Amirkabir University of Technology (Tehran Polytechnic)
Tehran, Iran
bbakhshi@aut.ac

***Abstract*— Network Function Virtualization (NFV) was recently proposed to improve the flexibility of network service provisioning and reduce the deployment costs by the decoupling the traditional network functions from the dedicated proprietary hardware equipment and turning them into the software-based virtualized entity, which is called a Virtual Network Function (VNF). Typically, the VNFs are concatenated together in a sequential order to form a Service Function Chains (SFCs) which provide specific services for users. Since failure of each VNF leads to breaking down the entire service chain, it is critical that service providers offer guaranteed and highly available SFCs against the failures. Previous works for simplicity assumed a completely available network infrastructure or just one failure in a same time; further, they increase availability of a SFC by adding dedicated backups which is costly. Unlike previous works, we investigate the problem of guarantying the required availability of SFCs against the multiple failures in a same time with minimum resource consumption. We proposed a new Availability-aware Clustered SFC Embedding (AC-SFC) algorithm which not only satisfies the availability requirement of SFC but also reduces the footprints of backup resource by using the Share Protection Cluster (SPC). Further, this problem is formulated as an Integer Linear Programming (ILP). We proposed a heuristic algorithm to overcome the ILP's complexity and new technique for calculating the availability of SPC. The numerical results show our heuristic algorithm works efficiency and reduces the network resource consumption in different network infrastructure topologies.**

Keywords: NFV, SFC, Availability, Network Virtualization, Resilience, Optimization

I. INTRODUCTION

The communication networks include some different parts such as the cloud service providers, enterprise networks, content delivery networks and mobile users that require high performance, availability, security, and so forth. In order to afford these problems, the modern networks deploy a wide range of the traditional network functions (middle boxes) to provide the advanced services such as the prevention systems, firewalls, gateways, and so on [1]. Each middle box is deployed in a specific hardware appliance to implement the specialized functions; it brings several draw backs, such as high CAPEX[1] and OPEX[2]costs, network function inflexibility, difficulty in scaling up and etc. As a result, these middle boxes whose failures have a strong impact on the network performance and availability, are costly [2]. Network Function Virtualization (NFV) is a new architectural paradigm to overcome these problems. NFV was proposed to not only reduce the costs but also improve the network management and flexibility for the network service provisioning. NFV uses the virtualization technology to decouple the traditional network functions from the dedicated proprietary hardware equipment and also turn the middle boxes into the software-based virtualized entities which referred as Virtual Network Function (VNF). VNFs will run on Virtual Machines (VMs) which are hosted by high volume servers, switches and storages. Moreover, the VNFs can share the same physical hardware to decrease the resource consumption.

Typically, incoming user traffic is often required to undergo a subset of Network Functions (NFs) in a specific order which is referred as Service Function Chain (SFC). Deploy of a given SFC in the NFV environment, is composed of two steps: i) VNF placement problem where the optimal number of required VNF instances and the Physical Machines (PMs) to host the VNFs are identified ii) Routing optimization problem to determine the optimal end-to-end paths over which traffic flows are routed to traverse the required set of placed VNFs [15].

While NFV leads to significant flexibility and control for network operators, it also brings more concerns for network resiliency, i.e., the ability of the network to provide and maintain an acceptable level of service in the face of failures and challenges to normal operation [3]. The server may fail due to the errors of the hardware (processor, memory, storage discs, etc.); further, there can be the software failures caused by the hypervisor or the VM instances.

In this work we concentrate on the servers' hardware failures. The main object of our researches is to provision the SFCs with the adequate availability guarantees in the NFV-enabled network infrastructure that if any problem occurs in the network infrastructure components, operation of the embedded SFCs is not affected. Moreover we proposed a new method and evaluate our method's results in different network infrastructure topologies.

The remaining of this paper is organized as follows. Section II reviews the related works and contributions. Section III includes the protection models and section IV describes the proposed scheme which includes four parts: system model,

[1] Capital EXpenditures
[2] Operational EXpenditures

978-1-7281-1508-5/19/$31.00 ©2019 IEEE

problem formulation, AC-SFC solution and heuristic algorithm. Section V contains the performance evaluation as well as a discussion of the results. Finally, in the concluding part, we wrap up this paper and discuss about future works.

II. Related works and Contributions

A. Related works

The references [5] and [6] assume a completely reliable/available NFV infrastructure (i.e., no network interruptions due to the hardware or software failures) which is not realistic. Moreover, their objective is to minimize the network resources consumption. Indeed, the authors of [7] and [2] consider the reliability/availability concerns in NFV and discuss the types of failures that may arise from both hardware (e.g., restart/shutdown of PMs) and software (e.g., misconfiguration of VMs) which lead to network service interruptions.

The references [8] to [10] augment the service chain requests with backup resources (e.g., one backup path or one dedicated backup resource) to resist against the failure in physical structure. Moreover, in works (e.g., [11]-[13]) ensure a desired level of end-to-end reliability. The main weakness in these suggested approaches are that they assume there is just single failure in physical infrastructure in a time, so the failure of several physical machines at the same time leads to breaking down entire the service chain.

On the other hand, the work [14] and [15] propose an incremental algorithm which add several dedicated backup resources to provision the service chains with considering the required reliability/availability. However this works waste many network resources due to the adding dedicated backup resources.

B. Contributions

- Different from the above-surveyed publications, we proposed a new method AC-SFC which clusters the SFC and uses the Shared Protection Cluster (SPC) to share the backup resource between the VNFs. So, our method not only guarantees the availability requirement against the multiple failures in the network infrastructure components but also reduces the footprint of backup.
- Moreover, we proposed a technique to compute the availability of a Shared Protection Cluster SFC (SPC-SFC) that it considers the number of the redundant and shared virtual nodes as the backup resources.
- Find a network infrastructure topology which decreases the resource consumptions.

III. Protection models and Illustrative example

A. Protection models

In this section, we first explained the two traditional protection models and then we proposed a new one called Availability-aware Clustered SFC (AC-SFC).

In order to calculate the availability of SFC in each schemes, we first need to calculate the availability of network's components. The availability of a component can be expressed using uptime followed by downtime, which can be characterized in terms of MTBBF and MTTR [4]. In our work, the availability of a component A being its long term probability of availability (1). Herein, only PM failures are considered and the failures of other network components (e.g., routers, switches, bridges, etc.) are neglected for simplicity. Per-PM failures are different and independent. Then, VNFs will inherit their respective host PMs' availability levels.

$$A = MTBF \times (MTBF + MTTR)^{-1} \quad (1)$$

1) No Protection SFC: The NP-SFC provisions the SFCs without any backup resource. Herein, the VNFs are connected in a serial manner in SFC. So, the SFC is available if all its VNFs are available at the same time (2).

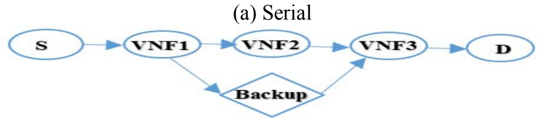
$$A_{SFC} = A_{VNF_1} \times A_{VNF_2} \times \dots \times A_{VNF_n} \quad (2)$$

2) Dedicated Protection SFC: The DP-SFC augments the SFC with the dedicated backup resources, that each backup resource of PM x is connected to the predecessor and successor of x to maintain the proper order in SFC, to improve the availability of the SFC. So, the availability of each VNF is that at least one instance of the VNF is available. Further, the availability of augmented SFC is that at least one instance k ($k \in F_j^T$ which is the set of VNF j instances) from each type j ($j \in F_i$ which is the set of VNFs in service i) of VNFs (3).

$$A_{SFC} = \prod\nolimits_{j\in F_i}\left[1 - \prod\nolimits_{k\in F_j^T}(1 - A_K)\right] \quad (3)$$

3) Shared Protection Clustered SFC: The aim of SPC-SFC is to reduce the footprint of backup resources through the using of more Shared Protection Cluster (SPC), which shares the backup resource between two VNFs along a SFC. we can share a backup resource between adjacent VNFs or non-adjacent VNFs. The share backup resource between adjacent VNFs will result in less resource consumption since the total number of links within the SFC is smaller. Further, in (Fig. 1) two chaining options are concerned; serial and parallel. Serial chaining will result in less resource consumption than parallel since the total number of links within the SFC is smaller.

(a) Serial



(b) Parallel

Fig. 1. Examples of backup resource connections

The SPC creates the dependency between VNFS's availability. So, in order to calculate the availability of SFC, which includes the SPCs. We cluster the SFC, which refer to the SPC-SFC, to three type clusters (Fig. 2):

a) No Protection Cluster: NPC includes one primary VNF. So, the availability of the NPC is equal to the VNF's availability.

b) Dedicated Protection Cluster: DPC includes one primary VNF and several dedicated backup. The availability of DPC is calculated based on (3).

c) Shared Protection Cluster: SPC includes two primary VNF1 and VNF2 and one shared backup resource SH-B. in order to calculate the availability of SPC (4), we defined its failure modes:

- A: simultaneous failure of VNF1 & SH-B
- B: simultaneous failure of VNF1 & VNF2
- C: simultaneous failure of VNF2 & SH-B
- D: simultaneous failure of VNF1 & SH-B & VNF2

$$A_{SPC} = 1 - P(AUBUCUD) \tag{4}$$

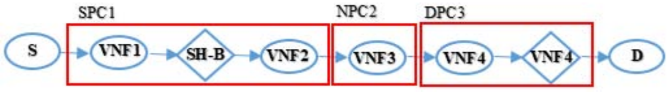


Fig. 2. Example of SPC-SFC

Whereas the SPC-SFC includes some independent clusters, so the availability of SPC-SFC is that all its clusters are available at the same time.

$$A_{SPC-SFC} = A_{C_1} \times A_{C_2} \times \ldots \times A_{C_n} \tag{5}$$

B. Illustrative Example

In order to achieve the availability above the 0.9 for the SFC two chaining options are concerned; Fig. 3 and Fig. 4. SPC-SFC (Fig. 3) will result in less network resource consumption than DP-SFC (Fig. 4) since the total number of backup resource is smaller due to the SPC. Moreover, their availabilities are so close to each other.

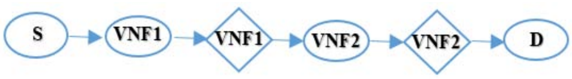


Fig. 3. Availability of DP-SFC = 0.98

Fig. 4. Availability of SPC-SFC = 0.97

IV. Proposed Scheme

In this section we described four parts: System model, AC-SFC Solution, ILP-based solution, Heuristic-based solution.

A. System model

1) Network Infrastructrue (NI): The NI consists two components: the PMs and the core transport network which includes routers links connecting the PM nodes. So, the links of the failed PM which pass through the router can be used. Moreover, the NI is modeled as a bidirectional graph G(N, L) where N and L represent the set of PMs and the set of physical links and the |N| and |L| denote the number of PMs and the number of links. The $B_m > 0$ represents the bandwidth capacity of a link $m \in l$. Each PM can host a set of VNFs. Each PM_k has a specific processing capacity $C_k > 0$ and is associated with a certain availability which is its long-term probability of availability.

2) Service Function Chain: Let S be the set of network services and F be the overall set of VNF types. The SFC includes a subset of VNFs F_i ($F_i \subseteq F$). Each VM executes only one VNF and allocates the capacity C_f ($f \in F$). the network service S_i ($i \in [1,|S|]_z$)[3] has a network bandwidth requireme $b_i > 0$ and an availability requirement $0 < \theta_{req}^{i} < 1$. Let Sr_i and D_i be the source and destination of the SFC. The f_{ij} ($j \in [1,|F_i|]_z$) is the j^{th} VNF which is traversed along the SFC S_i. Finally, each link m connects two VMs (x,y) which are hosted by (k1,k2) and the x = m.head and y = m.tail.

B. Problem Formulation

In this section, we illustrated our problem as an optimization model which was covered in [15]. Further, we add some new parameters and constraints.

1) Decision Variables:

$$y_{i,j}^{k} = \begin{cases} 1, & \text{if } f_{i.j} \text{ is hosted on } PM_k \\ 0, & \text{otherwise} \end{cases}$$

$$z_{i,j}^{m} = \begin{cases} a, & \text{if link } m \text{ is selected by } f_{i.j} \\ 0, & \text{otherwise} \end{cases}$$

$$w_{i,j}^{m} = \begin{cases} a, & \text{if and only if link } m \text{ connectes the } f_{i.j-1} \text{ and } f_{i.j} \\ 0, & \text{otherwise} \end{cases}$$

$$p_{i,j}^{k} = \begin{cases} 1, & \text{if fist instance of } f_{i.j} \text{ is hosted on } PM_k \\ 0, & \text{otherwise} \end{cases}$$

$$q_{i,j}^{m} = \begin{cases} a, & \text{if last instance of } f_{i.j} \text{ is hosted on } PM_k \\ 0, & \text{otherwise} \end{cases}$$

$(\forall i \in [1,|S|]_z,\ j \in [1,|F_i|]_z,\ m \in [1,|L|]_z,\ k \in [1,|N|]_z)$

2) Parameters:

$$sh_{i,j}^{k} = \begin{cases} 1, & \text{if } f_{i.j} \text{ shares the } PM_k \text{ with its adjacent VNF} \\ 0, & \text{otherwise} \end{cases}$$

$$ShF_{i,j} = \begin{cases} a, & \text{if } f_{i.j} \text{ shares backup with } VNF(a) \\ 0, & \text{if } f_{i.j} \text{ does not share backup} \end{cases}$$

$$ShC_{i,j} = \begin{cases} C_j, & \text{if } sh_{i,j}^{k} = 1 \text{ and } C_j \geq C_{ShF_{i,j}} \\ C_{ShF_{i,j}}, & \text{if } sh_{i,j}^{k} = 1 \text{ and } C_j < C_{ShF_{i,j}} \end{cases}$$

Define $x_{k,t_j} = 1$ if a function of type t_j is hosted on PM k. Otherwise $x_{k,t_j} = 0$. We define the VNF_Count_{ij} to be the number of VNF instances. 0 and $|F_i|$+1 indicate the source Sr_i and destination D_i of service i. Therefore, $p_{i,0}^{Sr_i} = 1$, $q_{i,0}^{Sr_i} = 1$ and $q_{i,|F_i|+1}^{Sr_i} = 1$.

3) Objective Function and Constraints:

$$\min\{\frac{1}{\sum_{m=1}^{|L|} B_m} \sum_{m=1}^{|L|} \sum_{i=1}^{|S|} \sum_{j=1}^{|F_i|} z_{ij}^{m} b_i\} \tag{6}$$

[3] |.| indicates the cardinality of a finite set and $[a,b]_z$ indicates the set of integers from a to b

Equation (6) is objective function which minimizes the bandwidth utilization.

$$\sum_{k=1}^{|N|} p_{i,j}^k = 1, \ \ (\forall i \in [1,|S|]_z, j \in [0,|F_i|+1]_z) \tag{7}$$

$$\sum_{k=1}^{|N|} q_{i,j}^k = 1, \ \ (\forall i \in [1,|S|]_z, j \in [0,|F_i|+1]_z) \tag{8}$$

$$p_{i,j}^k \le \sum_{m.tail=k} z_{i,j}^m \tag{9}$$
$$(\forall k \in [1,|N|]_z, i \in [1,|S|]_z, j \in [1,|F_i|+1]_z)$$

$$q_{i,j}^k \le y_{i,j}^k, \tag{10}$$
$$(\forall k \in [1,|N|]_z, i \in [1,|S|]_z, j \in [1,|F_i|]_z)$$

We defines some constraints for embedding the first and last instance of each VNF. Equations (7) and (8) guarantee that first or last instances are hosted only by one PM. Further, (9) ensures that if first instance of VNF embed on the PM_k so there must be an incoming link m toward the PM_k and (10) assume that if the PM_k is selected for embedding the last instance of VNF j so PM_k should hosting the VNF j.

$$\sum_{m.head=k} z_{i,j}^m - \sum_{m'.tail=k} z_{i,j}^{m'}\left(1-q_{i,j}^k\right) = q_{i,j-1}^k \tag{11}$$
$$(\forall k \in [1,|N|]_z, i \in [1,|S|]_z, j \in [1,|F_i|+1]_z)$$

$$\sum_{m.tail=k} z_{i,j}^m \le 1 \tag{12}$$
$$(\forall k \in [1,|N|]_z, i \in [1,|S|]_z, j \in [0,|F_i|+1]_z)$$

$$\sum_{m.head=k} z_{i,j}^m \le 1 \tag{13}$$
$$(\forall k \in [1,|N|]_z, i \in [1,|S|]_z, j \in [0,|F_i|+1]_z)$$

$$z_{i,j}^m z_{i,j}^{m'} = 0 \tag{14}$$
$$(\forall m, m' \in [1,|L|]_z, m.head = m'.tail$$
$$m.tail = m'.head, i \in [1,|S|]_z, j \in [0,|F_i|+1]_z)$$

$$z_{i,j}^m \ge zq_{i,j}^m \tag{15}$$

$$q_{i,j}^k \ge zq_{i,j}^m \tag{16}$$

$$zq_{i,j}^m \ge z_{i,j}^m + q_{i,j}^k - 1 \tag{17}$$

$$(\forall m \in [1,|L|]_z, k = m.tail, i \in [1,|S|]_z, \tag{18}$$
$$j \in [1,|F_i|+1]_z)$$

The VNF routing constraints can be formulated as the (11) to (18) which determine an optimal route between different instances of one VNF and flow conservation (11) and prevent of loop (14). The auxiliary binary variable $zq_{i,j}^m = z_{i,j}^m q_{i,j}^k$ is used for linearization of (11).

$$\left(1+sh_{i,j}^k\right)y_{i,j}^k \le x_{k,t_j} + sh_{i,j}^k x_{k,t_{ShF_{i,j}}}, \tag{19}$$
$$(\forall k \in [1,|N|]_z, i \in [1,|S|]_z \ j \in [1,|F_i|]_z)$$

$$\sum_{k=1}^{|N|} y_{i,j}^k == VNF_{Count_{ij}} \tag{20}$$
$$(\forall i \in [1,|S|]_z, j \in [1,|F_i|]_z)$$

$$y_{i,j}^k \le \sum_{m.head=k}^{|N|} z_{i,j}^m \tag{21}$$
$$(\forall k \in [1,|N|]_z, i \in [1,|S|]_z \ j \in [1,|F_i|+1]_z)$$

The availability constraints are formulated as the (19) to (21). The equation (19) insures that the PM_khosts the type of VNF j instance which is embedded on it. Moreover, if the VNF j instance is shared backup so the PM_k should host the VNF j and its adjacent VNF.

$$\sum_{i=1}^{|S|}\sum_{j=1}^{|F_i|}\left(1-sh_{i,j}^k\right)y_{i,j}^k C_{f_{ij}} + sh_{i,j}^k y_{i,j}^k ShC_{i,j} \le C_k \tag{22}$$

$$\sum_{i=1}^{|S|}\sum_{j=1}^{|F_i|} z_{i,j}^m b_i \le B_m, \ \ (\forall M \in [1,|L|]_z) \tag{23}$$

The equation (22) and (23) enforce the limit on the PM' processing resources and link bandwidth capacity. Further, (22) ensures that if PM_k is shared between VNF j and its adjacent VNF, so we reserve maximum capacity $ShC_{i,j}$ between two these VNFs.

$$w_{i,j}^m = q_{i,j-1}^{m.head} p_{i,j}^{m.tail} \tag{24}$$
$$(\forall m \in [1,|L|]_z, i \in [1,|S|]_z, j \in [1,|F_i|+1]_z)$$

$$w_{i,j}^m \le z_{i,j-1}^m \tag{25}$$
$$(\forall m \in [1,|L|]_z, i \in [1,|S|]_z, j \in [1,|F_i|+1]_z)$$

$$q_{i,j-1}^k p_{i,j}^k = 0 \tag{26}$$
$$(\forall k \in [1,|N|]_z, i \in [1,|S|]_z, j \in [1,|F_i|+1]_z)$$

The routing constraints between different VNFs of one network service are formulated as the (24) to (26). Further, the (24) and (26) are linearized like (11).

C. AC-SFC Solution

In this section the approach AC-SFC for embedding the SFC with guaranteeing the availability requirement is proposed. Moreover, the aim of the AC-SFC is to minimize the bandwidth and CPU[4] consumption.

Algorithm 1: Availability-aware Clustered SFC (AC-SFC)

```
Network Infrastructure (N) and Network Services (S) are given.
Create NPC clusters d_ic for each VNF in network services
VNF_COUNT_ij=1 (∀i ∈ [1,|S|]_z, ∀j ∈ [1,|F_i|]_z
flag = 1
While true do
    Embedding the SFCs on N
    for i = 1: |S| do
        if ∏_{c=1}^{|C|} A_ic < A_req then
            Select the cluster d_ic with the minimum availability;
            Improve Available Cluster (d_ic)
            flag = 0;
        end
    end
    if flag = 1 then
        break;
    end
    flag = 1;
end
```

The algorithm 1 first clusters all the input SFCs and creates the NPC for each primary VNF along the SFC. Then the clustered SFCs are embedded on the network infrastructure by using the optimal model in section B (Opt-AC-SFC) or heuristic algorithm in section D (H-AC-SFC). Next, the availability of each SFC are calculated based on (5). If the result meets the required availability then stop otherwise

[4] Central Processing Unite

improves the availability of network service through the improving the availability of a cluster, which has the minimum availability by the algorithm 2. This process is repeated as long as the availability of SFC meets the requested availability. Otherwise it stops.

In order to add more shared backup resource between adjacent. We consider function j+1 as an adjacent function for each odd function j ($j \in [1, |F_i|]_z$).

Algorithm 2: Improve Available Cluster (IAC)

Given cluster $d_{i,c}$ from Algorithm 1;
if $d_{i,c}\{f_{ij}\}$ is NPC and $j\%2 > 0$ *and* $(d_{i,y}\{f_{ij+1}\}$ is NPC or $|d_{i,y}| > 2$) **then**
  Convert $d_{i,c}$ to DPC $\{f_{i,j}, f'_{i,j}\}$; $VNF_{COUNT_{i,j}} = VNF_{COUNT_{i,j}} + 1$;
end
if $d_{i,c}\{f_{ij}\}$ is NPC and j%2 = 0 and ($d_{i,y}\{f_{ij-1}\}$ is NPC or $|d_{i,y}| > 2$) **then**
  Convert $d_{i,x}$ to DPC $\{f_{i,j}, f'_{i,j}\}$; $VNF_{COUNT_{i,j}} = VNF_{COUNT_{i,j}} + 1$;
end
if $d_{i,c}\{f_{ij}\}$ is NPC and j%2 > 0 and $d_{i,y}\{f_{ij+1}\}$ is DPC and $|d_{i,y}| = 2$ **then**
  Remove $d_{i,c}$;
  Convert $d_{i,y}$ to SPC $\{f_{i,j+1}, \mathrm{SH}(f'_{i,j+1}), f_{i,j}\}$;
  $sh^k_{i,j+1} = 1$; $ShF_{i,j+1} = j$; $ShC_{i,j+1} = \max\{C_j, C_{j+1}\}$;
end
if $d_{i,c}\{f_{ij}\}$ is NPC and j%2 = 0 and $d_{i,y}\{f_{ij-1}\}$ is DPC and $|d_{i,y}| = 2$ **then**
  Remove $d_{i,c}$;
  Convert $d_{i,y}$ to SHPC $\{f_{i,j-1}, \mathrm{SH}(f'_{i,j-1}), f_{i,j}\}$;
  $sh^k_{i,j-1} = 1$; $ShF_{i,j-1} = j$; $ShC_{i,j-1} = \max\{C_j, C_{j-1}\}$;
end
if $d_{i,c}\{f_{i,x}, \mathrm{SH}(f'_{i,x}), f_{i,y}\}$ is SHPC **then**
  Convert $d_{i,c}$ to two independent DPC $\{f_{i,x}, f'_{i,x}\}, \{f_{i,y}, f'_{i,y}\}$;
  $VNF_{Count_{i,y}} = VNF_{Count_{i,y}} + 1$; $sh^k_{i,x} = 0$;
end
if $d_{i,c}\{f_{ij}\}$ is DPC **then**
  Add one dedicated backup to cluster $d_{i,c}$;
  $VNF\_Count_{i,j} = VNF\_Count_{i,j} + 1$;
end

Algorithm 2 improves the availability of input cluster. So, it evaluates the type of cluster if the cluster is NPC and its adjacent cluster is DPC and has the backup for sharing then it merges two cluster to SPC. If the cluster is NPC and its adjacent cluster cannot share a backup resource or the cluster is DPC then it merges the cluster to DPC and add one dedicated backup. Finally, if the cluster is SPC, two clustering options are concerned; Fig. 5 and Fig. 6. The cluster in Fig. 5 includes the share backup and the dedicated backup. It includes six failure modes. So, like the (4) its availability is equal with 1-P(AUBUCUDUEUF). Further, the SPC in Fig. 6 converts into two DPCs. Whereas, the availability of two methods are so close together then we used the method in Fig. 6 due to that its availability is easier to calculate by (3) and (5).

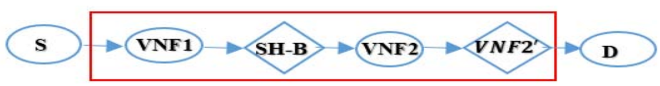


Fig. 5. Adds a backup resource to SPC and $A_c = 0.988$

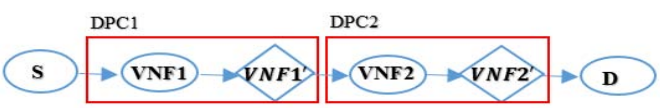


Fig. 6. Converts a SPC into two DPCs and $A_c = 0.980$

D. Heuristic Algorithm

In this section we introduce the heuristic solution (H-AC-SFC) which is an appropriate substitute for optimal model solution (Opt-AC-SFC) which is not responsible for embedding the large scale requested network services on the big scale network infrastructure.

Algorithm 3: Greedy Shortest Path Embedding-SFC (GSPE-SFC)

Network Infrastructure (N), Networked Service (S), $VNF\_COUNT_{i,j}$
for i=1 : |S| **do**
  src = Sr_i
  for j=1 : $|F_i|$ **do**
    for k=1 : $VNF\_COUNT_{i,j}$ **do**
      next = $f^k_{i,j}$;
      Dijkstra (src);
      Select a PM which has **<u>least bandwidth</u>** and **<u>max availability</u>** and enables to host next.
      Update N #decreases the BW and CPU consumption.
      src = next
      if j = $|F_i|$ and k = $VNF\_COUNT_{i,j}$ **then**
        Dijkstra(src , next);
        Update N
      end
    end
  end
end

We used the algorithm 3 in line 6 of algorithm 1 as our heuristic algorithm (H-AC-SFC). Algorithm 3 relies on the weighted-Dijkstra algorithm which starts from the source of each network then finds an appropriate PMs for hosting next VNF which has the enough computational resource and executes the VNF. This process is repeated until it reaches the network service's destination and all VNF instances have been assigned to PMs.

V. Performance Evaluation

In this section we evaluate the performance of the optimal model (Opt-AC-SFC) and the heuristic model (H-AC-SFC). All simulations are implemented and solved on a machine equipped with an Intel 2.3 GHz processor and 4 GB RAM and the Gurobi solver is used to solve the ILP model.

We evaluate the performance of the HAC-SFC in network infrastructure types: structured (lattice) and unstructured (NSFNET[5]) moreover, it is compared against the Greedy-NFV heuristic [15], which adds the dedicated backup resources to guarantee the availability or reliability of service networks. Bandwidth utilization, CPU resource utilization, Number of created VNF instances and Percentage of blocked network services are used as the evaluation metrics. The evaluation were repeated 10 times.

Three network infrastructures are considered; i) a small network composed of 8 PMs hosting 4 network services and ii) a medium network composed of 14 PMs hosting 10 to 20 network services and iii) a large network composed of 40 PMs hosting 20 to 50 network services. Each network service includes four VNFs (e.g., NS1: f_3→f_1→f_2→f_4). The

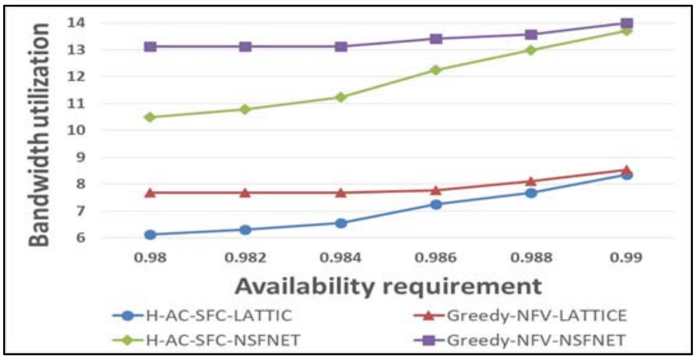
[5] National Science Foundation NETwork

CPU consumption for each VNF and the bandwidth requirements of the network services are randomly drawn from the range 2 to 4 (units). Each PM is assumed to be able host 1 to 3 VNFs. The availability of PMs are randomly generated between 0.92 and 0.96. The nodal computational capacities and the link bandwidths are randomly drawn from a range 20 to 30 units.

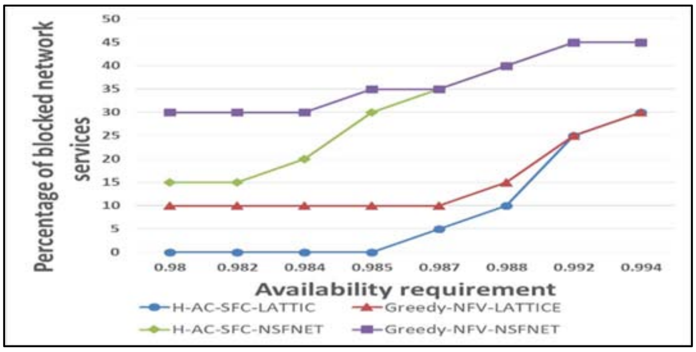
TABLE I. RESULTS FOR SMALL SCALE IN LATTICE NETWORK

Algorithm	Evaluation Factors			
	N of S	*Availability*	*BW utilization*	*N of VNF*
Opt-AC-SFC	4	0.99	14.52	47
H-AC-SFC	4	0.99	14.53	49

Table I compare the results of Opt-AC-SFC and H-AC-SFC with the availability requirement 0.99 in the small scale. As you see the result of our heuristic is so similar to the optimal model.



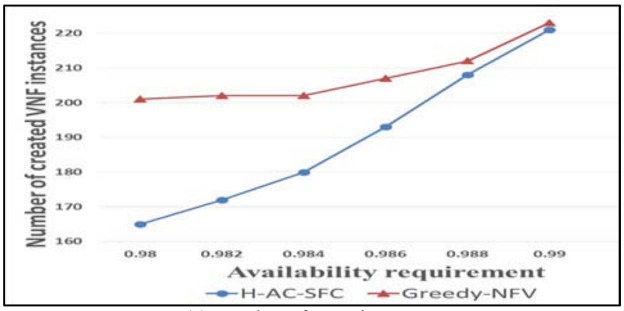
(a) Bandwidth utilization



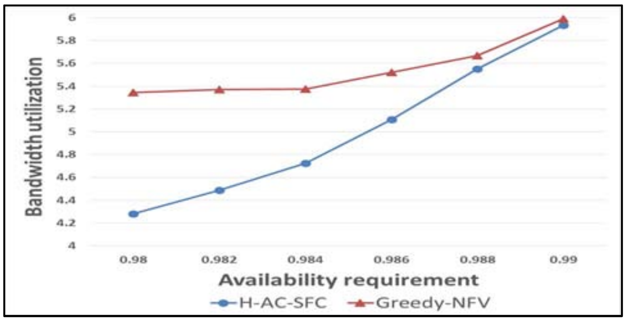
(b) Loss ratio

Fig. 7. Results for medium scale in different network topologies

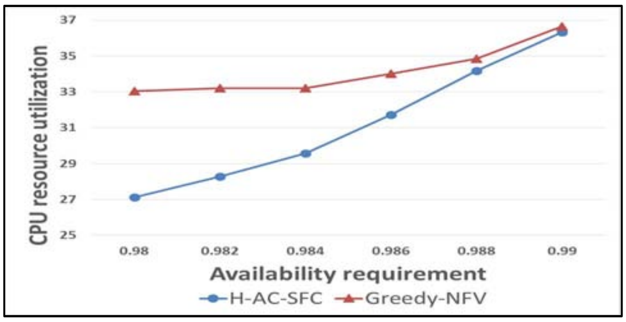
In the Fig. 7, we evaluate the results of H-AC-SFC and Opt-AC-SFC in the structured network topology (NSFNET) and unstructured network topology (lattice) in medium scale. As you see in the Fig. 7 the process of changing bandwidth utilization and loss ratio for different availability requirements in two topologies are similar and are explained in more detail in the next evaluation. Further, in the Fig. 7.a the bandwidth utilization for both of the heuristics in the lattice network is smaller than NSFNET network due to that the lattice network has high connectivity and degree of PMs which lead to embed the VNFs along the SFC close together and decrease the bandwidth consumption. Consequently; embedding the network services in the lattice network consumes less network resources and decrease the percentage of blocked network services (Fig. 7.b). In Fig. 7.b the number of input network services is considered 20. As a result, we receive better results in the lattice network topology.



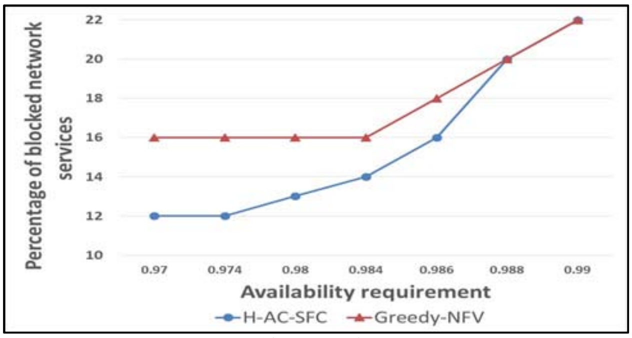
(a) Number of VNF instances



(b) Bandwidth utilization



(c) CPU utilization



(d) Loss ratio

Fig. 8. Results for big scale in lattice network topology

In the Fig. 8, we compare the results of our heuristic H-AC-SFC and Greedy-NFV in the big scale system model. As you see in Fig. 8 the process of changing number of VNF instances, bandwidth utilization, CPU resource utilization and loss ratio for different availability requirements include three types of changes:

A. Incremental changing

Whereas, a higher availability requirements can be satisfied by instantiating redundant backup resources so with increasing the availability requirements the number of created VNF instantiates will be increased (Fig. 8.a); moreover, these redundant backup resources need backup links to communicate with other VNFs along the service chain and CPU resource which lead to increase the bandwidth utilization (Fig. 8.b) and CPU utilization (Fig. 8.c). As a result; the network resource consumption will be increased in higher availability requirements which causes to have less available and free network resource for embedding network services and increase the percentage of blocked network services chart. We compare the results of loss ratio chart for 50 network services.

B. The results of H-AC-SFC is less than Greedy-NFV

Whereas, in our method we create the SPC to increase the availability of network service and share a backup resource between VNFs so it leads to decrease the number of VNFs instances (Fig. 8.a) and backup links; consequently, in our work, we have less bandwidth utilization (Fig. 8.b) and CPU utilization (Fig. 8.c) which causes to decrease network resource consumption and the loss ratio (Fig. 8.d).

C. Convergance

Whereas, in the high availability requirements, the number of VNF instances will increase further until that the most clusters along the network service are DPC and the number of SPC will decrease. As a result, in the high availability required, the difference between the results of the two heuristic will decreased.

Conclusion and Future work

In this paper we presented a novel Availability-aware Clustered SFC (AC-SFC) framework for embedding SFCs in the NFV-enabled network infrastructure which includes the physical machines with heterogeneous failure rate. AC-SFC uses the SPC to share backup resources between adjacent VNFs to decrease the footprint of backup resources, network resource consumption and the loss ratio. Further, to overcome the complexity of optimization model a greedy shortest-path-based heuristic (GSPE-SFC) is proposed. We simulate the H-AC-SFC method and compare with the Greedy-NFV heuristic in different network topologies and various factors to prove the efficiency of our suggested approach.

For future work, we aim to consider more constraints like end-to-end delay for mapping the service chain requests in NFV-enabled substrate network and extend our work with new methods for merging clusters to decrease more network resources.

References

[1] Sekar V, Ratnasamy S, Reiter MK, Egi N, Shi G. The middlebox manifesto: enabling innovation in middlebox deployment. InProceedings of the 10th ACM Workshop on Hot Topics in Networks ACM, p. 21, Nov 2011.

[2] Cotroneo D, De Simone L, Iannillo AK, Lanzaro A, Natella R, Fan J, Ping W. Network function virtualization: Challenges and directions for reliability assurance. InSoftware Reliability Engineering Workshops (ISSREW), 2014 IEEE International Symposium on, pp. 37-42, 2014.

[3] Han B, Gopalakrishnan V, Kathirvel G, Shaikh A. On the Resiliency of Virtual Network Functions. IEEE Communications Magazine., vol. 55, pp. 7, pp.152-7, 2017.

[4] Fan J, Guan C, Ren K, Qiao C. Guaranteeing Availability for Network Function Virtualization with Geographic Redundancy Deployment. 2015.

[5] Ahvar S, Phyu HP, Buddhacharya SM, Ahvar E, Crespi N, Glitho R. CCVP: Cost-efficient centrality-based VNF placement and chaining algorithm for network service provisioning. InNetwork Softwarization (NetSoft), 2017 IEEE Conference on, pp. 1-9. Jul. 2017.

[6] Mehraghdam S, Keller M, Karl H. Specifying and placing chains of virtual network functions. InCloud Networking (CloudNet), 2014 IEEE 3rd International Conference on , pp. 7-13, Oct. 2014.

[7] Govindan R, Minei I, Kallahalla M, Koley B, Vahdat A. Evolve or die: High-availability design principles drawn from googles network infrastructure. InProceedings of the 2016 ACM SIGCOMM Conference, pp. 58-72. Aug. 2016.

[8] Hmaity A, Savi M, Musumeci F, Tornatore M, Pattavina A. Virtual network function placement for resilient service chain provisioning. InResilient Networks Design and Modeling (RNDM), 2016 8th International Workshop on, pp. 245-252, Sep. 2016.

[9] Guo B, Qiao C, Wang J, Yu H, Zuo Y, Li J, Chen Z, He Y. Survivable virtual network design and embedding to survive a facility node failure. Journal of Lightwave Technology., vol. 32, no. 3, pp. 483-93, Feb. 2014.

[10] Zhang X, Phillips C, Chen X. An overlay mapping model for achieving enhanced QoS and resilience performance. InUltra Modern Telecommunications and Control Systems and Workshops (ICUMT), 2011 3rd International Congress on, pp. 1-7, Oct. 2011.

[11] Khan MM, Shahriar N, Ahmed R, Boutaba R. Simple: Survivability in multi-path link embedding. In2015 11th International Conference on Network and Service Management (CNSM), pp. 210-218, 2015.

[12] Kang J, Simeone O, Kang J. On the trade-off between computational load and reliability for network function virtualization. IEEE Communications Letters, vol. 21, no. 8, pp. 1767-70, 2017.

[13] Yang S, Wieder P, Yahyapour R. Reliable virtual machine placement in distributed clouds. InResilient Networks Design and Modeling (RNDM), 2016 8th International Workshop on, pp. 267-273, Sep. 2016.

[14] Qu Qu L, Assi C, Shaban K, Khabbaz M. A reliability-aware network service chain provisioning with delay guarantees in nfv-enabled enterprise datacenter networks. IEEE Transactions on Network and Service Management, vol. 14, no. 3, pp.554-68, 2017.

[15] Qu L, Assi C, Shaban K, Khabbaz M. Reliability-aware service provisioning in NFV-enabled enterprise datacenter networks. InNetwork and Service Management (CNSM), 2016 12th International Conference on 2016 Oct 31, pp. 153-159, Oct. 2016.