# Joint Diversity and Redundancy for Resilient Service Chain Provisioning

Abdelhamid Alleg, Toufik Ahmed, Mohamed Mosbah, and Raouf Boutaba, *Fellow, IEEE*

***Abstract*— Achieving network resiliency in terms of availability, reliability and fault tolerance is a central concern for network designers and operators to achieve business continuity and increase productivity. It is particularly challenging in increasingly virtualized network environments where network services are exposed to both hardware (e.g., bare-metal servers, switches, links, etc.) and software (VNF instances) failures. This increased risk of failures can severely deteriorate the quality of the deployed services and even lead to complete service outages. In this context, deploying services in operational networks often exacerbates the availability problem and requires considering availability of hardware and software components both individually and collectively. A key challenge in this perspective is the additional resources needed to achieve partial or full recovery after failures. In this paper, we propose a joint selective diversity and tailored redundancy mechanism to provision resilient services in an NFV framework. Diversity splits a single VNF into a pool of "N" active instances called replicas while redundancy provides "P" standby ready-to-use instances called backups. Based on an enhanced N+P model, we propose a placement solution of Service Function Chains (SFC) modeled as a Mixed Integer Linear Program (MILP). The proposed solution is designed to meet a target SFC availability level and, at the same time, to reduce the inherent cost due to diversity (overhead) and redundancy (backup resources). We evaluate the efficiency of the proposed solution through numerically and experimentally. Results demonstrate that our solution, not only, improves service resiliency by avoiding complete service outages but can also overcome network resource fragmentation.**

***Index Terms*— Network function virtualization, network resilience, redundancy, network reliability, placement and chaining, network diversity.**

## I. Introduction

RESILIENCY is considered as a fundamental design property of the Future Internet [1]; it defined as the ability of the network to provide and maintain an acceptable level of service in the face of failures and challenges to normal operation [2]. It is a Key Performance Indicator for Service Provider often quantified by different metrics such as reliability and availability. Both concepts may appear to be interchangeable, but they have different meanings. On one hand, reliability is generally expressed in terms of Mean Time Between Failures (MTBF) which estimates how long a system (e.g. a Service Chain, a VNF, physical node or link) performs its expected function properly before it fails. On the other hand, availability calculates the amount of time during which the system is in operational state, and it is influenced by both MTBF and Mean Time To Repair (MTTR). Therefore, two systems with the same MTBF and with distinct MTTRs will have different availabilities despite the fact that they have the same reliability.

Nowadays, networks are undergoing a major transformation through softwarization that will have a lasting impact on how network services will be designed, deployed and managed. Network Function Virtualization (NFV) heralds a new era for future networks where the flexibility provided by virtualized environments will replace the rigidity of traditional networks that are built from physical middleboxes. In traditional networks, ensuring resiliency to failures of a hardware middlebox is typically achieved by adding a set of secondary or redundant hardware middleboxes that remain in standby mode during failure-free operations resulting in a considerable cost. In an NFV environment, services rely on decoupled software (VNF instances deployed in VMs or Containers) and hardware (commodity servers) and are prone to the failure of both. In particular, software is notoriously buggy, hence VNFs are expected to fail more often compared to traditional hardware middleboxes typically subjected to rigorous testing and validation processes. Ensuring service resiliency is therefore more complex in NFV environments compared to traditional networks and require dedicated mechanisms to guarantee high availability while keeping costs (resource consumption, overhead, etc.) as low as possible.

Existing works on resilient NFV [3]–[5] focused on availability requirements and proposed models to select the best VNF placement to meet some availability requirements. However, resource optimization is not fully considered in current solutions especially when using redundancy techniques that are by nature resource-hungry. Most backup-based solutions can be expensive in terms of resource consumption since they rely on reserving resources that remain idle until a failure occurs. Such protection techniques tend to overprovision backup instances to cover improbable scenarios. For example, the possibility to lose all replicas of the same VNF at once causing a complete service blackout is very unlikely (especially when they are running on distinct servers) while the amount of redundant resources is calculated to cover this worst-case scenario.

In this paper, we propose a joint diversity and redundancy for resilient service chain provisioning. Toward this objective,

Manuscript received January 30, 2019; accepted January 28, 2020. Date of publication April 9, 2020; date of current version June 29, 2020. *Corresponding author: Abdelhamid Alleg.)*

Abdelhamid Alleg, Toufik Ahmed, and Mohamed Mosbah are with the LaBRI (UMR5800), CNRS, University of Bordeaux/Bordeaux INP, 33076 Bordeaux, France (e-mail: aalleg@labri.fr; tad@labri.fr; mosbah@labri.fr).

Raouf Boutaba is with the David R. Cheriton School of Computer Science, University of Waterloo, Waterloo, ON N2L 3G1, Canada (e-mail: rboutaba@uwaterloo.ca).

Color versions of one or more of the figures in this article are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSAC.2020.2986867

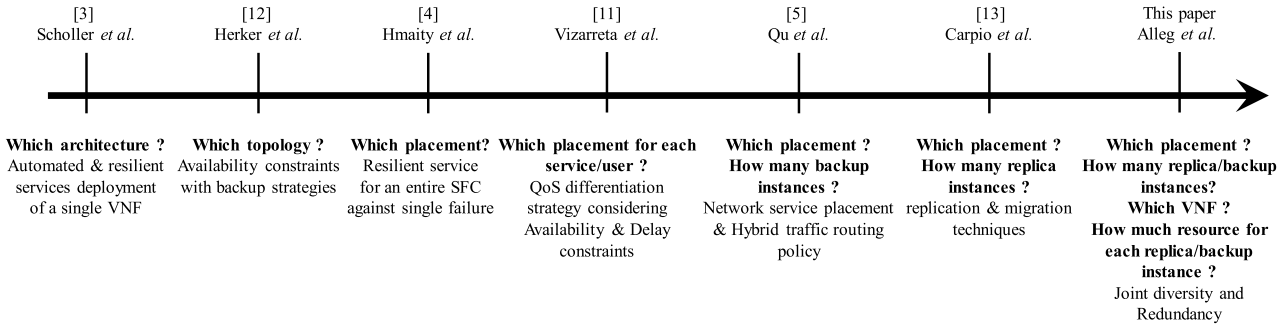


Fig. 1. Evolution of availability related work within NFV networks.

we devise an enhanced N+P model including a selective VNF diversity combined with tailored redundancy. The main idea underlying VNF diversity consists in replacing a single VNF instance by a pool of N thinner instances called replicas operating in parallel. These instances collectively and in a distributed manner perform the same network function as the original (i.e., diversified) VNF and collectively process at least the same amount of traffic as the original VNF.

Diversity is expected to avoid complete service outages. Specifically, when some replicas from the pool fail, the remaining replicas continue to provide a safe mode service, albeit at lower capacity. However, achieving diversity comes with additional overhead associated with the pool of replicas and at the same time requires a load balancing capability to automatically distribute the incoming traffic across multiple replicas. Redundancy is expected to improve reliability by provisioning backup instances that replace failed ones. Redundancy has an inherent cost in terms of the extra resources dedicated for the backup instances. Both diversity overhead and backup resources for redundancy must be minimized while meeting the target level of service availability.

To the best of our knowledge, this paper is first to adapt N+P model for handling a chain of virtualized network functions. The novelty of our approach also stems from our goal to ensure not only the availability of one VNF but the availability of the entire service chain. Finally, in our solution we don't just provide the N and P needed to reach a target availability level but we also determine the amount of resources allocated to each of the replica and backup instances in a way that minimizes the overall resource consumption. In other words, our solution avoids complete service outage (thanks to diversity) and allocates less resources to backup instances compared to a traditional N+P model. Specifically, the main contributions of this paper are as follows:

- We introduce a resiliency mechanism for NFV service chains based on a joint N+P diversity and redundancy model.
- We formulate and optimally solve the resilient Service Function Chain (SFC) placement problem using Mixed Linear Integer Programming (MILP).
- We ensure efficient resource utilization by means of a redundancy scheme tailored for specific VNFs.
- We evaluate the proposed model numerically and implement a proof of concept to demonstrate its feasibility.

The rest of this paper is organized as follows. Section II discusses related work. Section III introduces our joint diversity and redundancy model for resilient service function chain provisioning. In Section IV, we formulate and solve the resilient SFC placement problem. Section V numerically evaluates the performance of the proposed solution including comparison with the most relevant work in the literature. It also describes our proof of concept implementation of the joint diversity and redundancy mechanism for VNFs. Finally, conclusions and future work are presented in sectionVI.

## II. Related Work

Generally speaking, we can distinguish two main approaches to achieve resilience according to the type of network or more precisely to its softwarization level: an expensive but simple solution approach; and a complex but cost-effective solution approach. Indeed, in a traditional network exclusively composed of physical appliances, providers can simply deploy redundant hardware and extra capacity in order to handle failures. In this case, the ease of achieving resilience comes at a significant cost in terms of capital and operational expenditures. In turn, softwarized networks leveraging Network Function Virtualization (NFV) significantly reduce these costs but raise new challenges for achieving the same level of resiliency.

Several works (e.g., [6]–[9]) focused on the Virtualized Network Function (VNF) placement and chaining problem. They aim at finding the best locations for a set of VNFs that optimize different objectives such as minimizing the number of VNF instances, end-to-end delays or provisioning cost. However, this group of work assumed a completely reliable NFV infrastructure and did not consider service interruptions caused by hardware and/or software failures. Another group of work, depicted in Fig. 1 tackled directly the reliability problem within virtual environments and tried to ensure questions related to architecture, topology, placement (where to place, how many instances, etc.).

In this context, Cotroneo *et al.* [10] were among the first to discuss the reliability challenges of NFV infrastructures and investigated how risks caused by hardware or software failures can be assessed. Automated and resilient service deployment mechanism has been described by Scholler *et al.* [3]; specifically, an architecture for deploying complex multi-component services on a cloud infrastructure based on an information

model has been proposed. Their approach focused on the resilience of a single VNF. Hmaity *et al.* [4] proposed three protection schemes that take into consideration latency constraints for (1) end-to-end protection, (2) virtual-node protection and (3) virtual-link protection. The end-to-end protection strategy ensures resilience for the whole Service Function Chain (SFC) against single link or node failure but requires double the amount of resources compared to the unprotected scenario. Virtual-node protection is more efficient since it is designed as an off-site redundancy strategy in that all backup VNFs are placed in distinct locations from where their primary ones are hosted. However, the authors did not consider the heterogeneity of the VNFs in a chain (e.g., load balancers are more likely to fail than other types of VNFs) when deciding virtual node protection. Furthermore, the virtual-link protection may result in mapping both primary and backup virtual links on the same physical link which in turn results in weaker resiliency.

In the same vein, S. Herker *et.al* [11] developed a service chain embedding algorithm that considers service availability constraints and compared several backup strategies using different datacenter architectures. They provided insights on how to select the "best" data center topology that provides high availability. Reliance on VNF backups often results in longer routing paths and hence increased end-to-end delays. To overcome this problem, Vizarreta *et.al* [12] proposed two VNF placement strategies that minimize the service deployment cost for the operator without compromising the quality of service in terms of availability and end-to-end delays. However, the placement based on availability can result in resource wastage in that resources that do not satisfy the required availability threshold will never be used. This conflicts with the initial objective of cost effectiveness in terms of resource utilization. Similarly, Qu *et.al* [5] proposed a solution that provides an optimal network service placement using a hybrid traffic routing policy while balancing between bandwidth consumption and end-to-end delay performance.

Carpio and Jukan [13] presented a Linear Program (LP) model for active-active configuration based on both replication and migration techniques to improve service reliability. They also propose N-to-N configuration to speed-up recovery after server failures. Their solution is based on an implicit redundancy scheme that increases the resources allocated (called spare resources) to each replica to be able to process a part of the additional traffic that will be redirected to it, after a failure of another replica from the same group. The obtained results show that replication combined with migration can improve resource utilization without degrading reliability. Nevertheless, their recovery solution cannot support more than one replica failure per VNF (there are not enough spare resources to cover more than one outage). Moreover, the amount of spare resources depends on the number of replicas and increases dramatically when dealing with only two replicas similar to the solution proposed in [4].

In summary, previous work focused on flat protection strategies that are applied in a uniform manner regardless of the type or the importance of the VNF within a given SFC. The need for a selective protection strategy can be justified by the observation that the probability of a middlebox to fail depends on its type [14]. For example, [14] shows that load balancers exhibit high failure rates compared to other types of middleboxes. Also, the previous work systemically oversizing redundancy instances to cover worst case scenarios which are unlikely to happen in practice.

Motivated by open issues from related work, we propose, in this paper, a VNF placement and chaining based on joint diversity and redundancy approach that minimizes resource consumption while strengthening service resiliency. We believe that our proposed solution is a significant step further on N+P model in that it provides a selective diversity (N) by protecting the VNFs within a service function chain that are most prone to failure and a tailored redundancy by allocating the needed resources to backup instances covering most likely failure scenarios.

## III. VNF Diversity and Redundancy: N+P Model

In this section, we define the diversity and redundancy concepts that are used jointly to ensure better resilience. The term replica in our model refers to heterogeneous (or homogeneous) instances in terms of resource requirements that result from applying diversity to a VNF. Therefore, we avoid using the term duplicate because it indicates perfect similarity between instances i.e. that all instances have the same resource requirements which is not our cas.

### A. VNF Diversity (N)

A basic notion of diversity was introduced in [15] and considered as an essential means for providing resiliency. Software diversity was introduced as multi-version programing (N-version) to increase the reliability especially for critical embedded systems. Furthermore, the European Telecommunications Standards Institute (ETSI) [16] promotes designing and implementing diversity as a means for tolerating faults and discusses technical approaches aiming to automatically control software diversity. Our approach for VNF Diversity consists in replacing a single VNF by a set of thinner ***active*** *replicas* instances that can in a distributed manner perform the same Network Function (NF) and collectively process at least the same amount of traffic assigned originally to the VNF.

Our approach can be used to provision resilient service chains with a high availability and fault tolerance. However, applying diversity comes with a cost resulting from the necessity of integrating additional load balancer instances to distribute/redirect the incoming traffic across a pool of $N$ *replicas*. Also, if these instances collectively use resources as much as the initial VNF the principle of performance conservation would not be guaranteed, thus, an estimated overhead $H(v, N)$ is introduced to ensure that *N replicas* are as efficient as the targeted VNF (see equation (i)). The total amount of resources allocated to a pool of $N$ *replicas* including the load balancer instance $\Psi^{LB}$ and the overhead associated to diversity, is denoted $\pi(v)_{diversity}$ as presented in equation *(ii)* where $\Psi^{v}$ is the resource required by the tragated

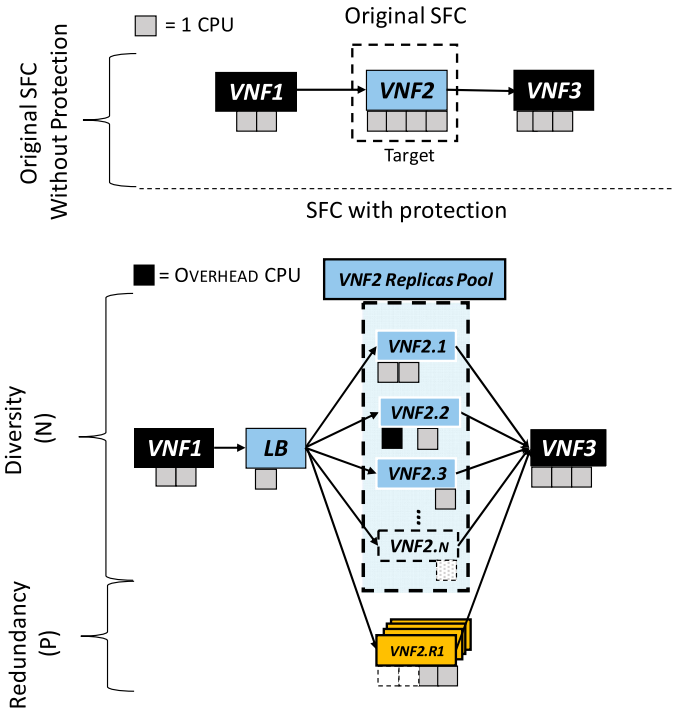


Fig. 2. Diversity and redundancy applied to a single VNF.

VNF $\boldsymbol{v}$.

$$H(v,N) = \Psi^v \cdot \frac{N-1}{2(N+1)} \qquad (i)$$

$$\pi(v)_{diversity} = \Psi^v + \Psi^{LB} + H(v,N) \qquad (ii)$$

In equation(i), we assume that the overhead equals to 0 when the number of replicas equals 1 (no diversity case) and tends to half amount of the original instance for high number of replicas. We introduced this overhead to consider a non-negligible disadvantage and to avoid a systematic raise of diversity level. We also assume that the resource required by each LB ($\Psi^{LB}$) remains fixed whatever the number of associated replicas. These equations remain an estimation and do not pretend quantifying precisely the diversity overhead.

Fig. 2 illustrates the idea of VNF Diversity. For simplicity, we show only CPU requirements, but resources may include processor, memory, storage, etc. In this example, an SFC composed of {VNF1, VNF2, VNF3} where diversity is applied on VNF2 which is replaced by a set of $N = 3$ *replicas* {VNF2.1, VNF2.2, VNF2.3}. However, the amount of requested resources (1, 2 or 3 CPU) of each *replica* is smaller than that of the original VNF (**4** CPU). The associated overhead (black squares) to the pool of 3 *replicas* is $H(VNF2,3) = \mathbf{4} \times \frac{2}{8} = 1\text{CPU}$. If we suppose that the LB instance needs 1 CPU, the total amount of resource allocated to the pool in this example is 6 CPU instead of the original **4** CPU.

Moreover, the proposed diversity is enhanced through adopting an offsite mapping strategy in order to guarantee a weak correlation between failures. This strategy consists of mapping *replicas* of the same pool into separated locations (different sites or different racks). The diversity as presented in this paper can handle both stateless and stateful VNFs. Though, without loss of generality, stateful feature is more complex to achieve since it requires using a shared volume to save pool states [17] and to maintain continuous synchronization. This volume needs to be duplicated to maintain *replicas* states. Such details related to network architectures are beyond the scope of our paper.

### *B. Redundancy Scheme (P)*

Redundancy consists of using additional instances that accomplish the same task knowing that a single instance is enough to fully execute this task. Therefore, integrating redundancy mechanism is *unavoidable* in NFV networks, in order to improve the existing availability level and to ensure service continuity despite failures. However, attention should be drawn to the fact that creating redundant VNFs will increase the length of the considered SFC and thus, consuming extra resources compared to none redundancy case.

Our proposed solution adopts a redundancy scheme that satisfies the requested availability while optimizing resource consumption. To this end, we propose a redundancy mechanism tailored for specific VNFs. For each target VNF, $P$ *backup* instances are provisioned in ***standby mode*** to utilize a customizable amount of resources. Thus, for a target VNF v that requires $\Psi^v$ resource units $(ru)$. e.g. $1ru = 1\text{CPU}$, the amount of resource $\pi$ assigned to the *backup* instances is expressed as:

$$\pi(v,\alpha) = \alpha \cdot \Psi^v \cdot P \qquad (iii)$$

where the redundancy parameter $\alpha \in [0,1]$ indicates the *fraction* of the resource allocated to each backup instance compared to the original VN. For exampl, in Fig. 2, when $P = \mathbf{2}$ and $\alpha = 0.5$, the VNF2 that requires 4 CPU will be covered by **2** *backup* instances. Each backup instance (VNF.R1 and VNF.R2) will get $0.5 \times 4\text{CPU} = 2\text{CPU}$. As a result, the total amount of resources assigned to redundancy is $\mathbf{2} \times 2\text{CPU} = 4\text{CPU}$.

Note that for a given number of backup instances $P$, the amount of resources allocated to redundancy depends on the redundancy parameter $\alpha$. When $\alpha = 1$ (full redundancy) denotes that each backup instance will use as much resource as the targeted VNF. One way to tailor redundancy can be done by assigning to each backup instance as much resource as the most demanding instance among a pool of $N$ *replicas*. Thus, for any targeted VNF that requires $\Psi^v$ resource units, the amount of resource $\pi_{Redundancy}$ assigned to the backup instances is expressed by:

$$\pi(v,\alpha)_{Redundancy} = Max\{\Psi_i^v\}_{i \in Pool(v)} \cdot P \qquad (iv)$$

The main difference between redundancy schemes lays in the amount of resources allocated to the redundant instances. It is clear that traditional redundancy ($\alpha = 1$) offers a systematic protection, since it is specially designed to cover *unlikely* multi-failure worst-case scenarios (e.g. VNF2.1, VNF 2.3 and VNF 2.3 fail simultaneously even they are mapped in different sites). Whereas, our proposed tailored redundancy provides a customized redundancy to the targeted VNF to cover the *most likely* failure scenarios, especially, knowing that diversity is designed to reduce multi-failure risk within a same pool due to offsite mapping strategy.

## IV. Problem Statement and Solution

In this section, we analyze closely the SFC availability when applying different configurations (series and parallel representations), then, we present the problem formulation of our joint diversity and redundancy for resilient service chain provisioning.

### A. Service Function Chain Availability

In NFV networks hardware and software are decoupled. However, in terms of resilience, we need to consider jointly both of them to form a whole system. Thus, the availability of each VNF running on a physical appliance must be expressed by both software and hardware availabilitie. The availability of a VNF instance $i$ hosted by the server $n$ is given by:

$$A_i = A_{vnf}^{software} \cdot A_n^{Hardware} \tag{1}$$

where $A_{vnf}^{software}$, the software availability of VNF, is related to software bugs, configuration errors, VNF complexity, etc. The $A_n^{hardware}$ is the hardware availability of the host $n$ that depends on its equipment quality among other factors. Furthermore, an SFC is considered as a complex system composed of reparable VNF where its availability is determined from the individual and collective availability of its components. Once an SFC is *deployed* the resultant availability will also depend on the availability of the hardware that host its software instances. In the following, we study SFC availability considering three possible way to apply diversity: *No Diversity, Full Diversity and Selective Diversity*.

In the first case, diversity is not applied on any VNF. Thus, the SFC is modeled as a series system ($\mathcal{S}$) since the failure of one VNF leads to the failure of the whole SFC regardless of whether it is linear or bifurcated. Indeed, a linear SFC is intuitively seen as series system while for a bifurcated SFC, the damage of a VNF in a branch will endanger the end-to-end service even if the remaining branches are undamaged. In case of full diversity, all VNFs in an SFC are concerned by diversity. Thus, each VNF is substituted by a pool of $N$ *replicas* VNFs which require individually less capacity than the original targeted VNF but behave collectively as the original one. For such case, the failure of one *replica* will not cause immediately in the failure of the entire SFC. Each targeted VNF, in this case, can be modeled as a parallel system ($\mathcal{P}$). In contrast to the aforementioned cases, applying selective diversity on an SFC means that some of its VNFs are targeted by diversity. Thus, such case can be modeled as a series-parallel system composed of a set of subsystems (parallel $\mathcal{P}$ and series $\mathcal{S}$) linked in series as depicted in Fig. 3.

Let consider $\mathbb{N}_P$ denotes the set of VNFs that are targeted by diversity where each VNF is modeled as a parallel subsystem $\mathcal{P}$. $\mathbb{N}_S$ is a set of VNFs which are not involved by diversity and is modeled a single series subsystem ($\mathcal{S}$ in Fig. 3). This latter is logically linked to one or many $\mathcal{P}$ subsystems that correspond to $\mathbb{N}_P$ elements. Consequently, the availability $A(SFC)$ in equation (2) of any SFC is written as the product of each VNFs availabilities $A_{vnf}$ composing this SFC. Also, each $A_{vnf}$ depends on whether diversity is applied or not (parallel or series system). The first term of equation (2)

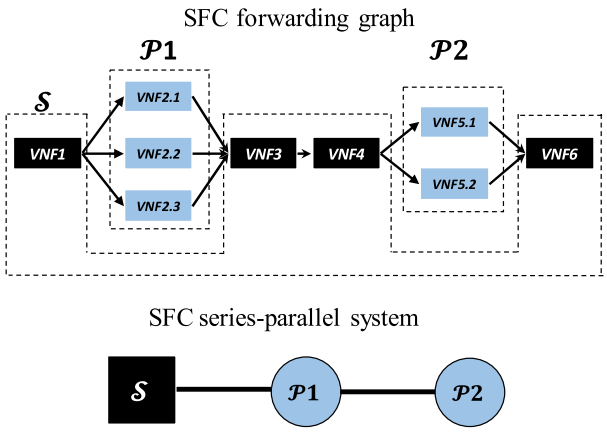


Fig. 3. Generic representation of SFC as a series-parallel system.

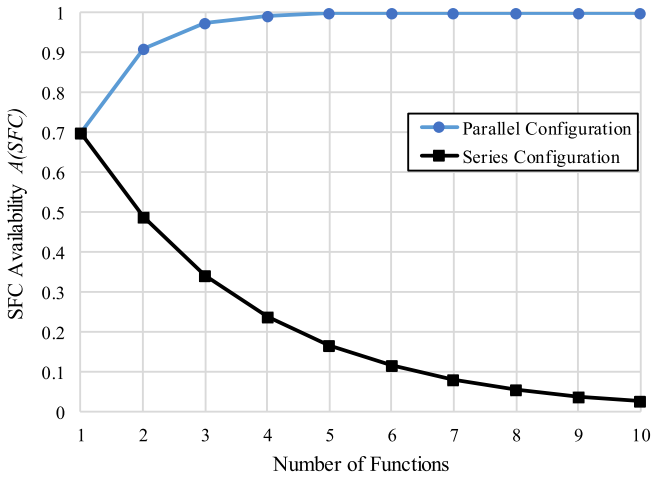


Fig. 4. Availability behavior for parallel and series configurations.

calculates the availability of VNFs that are not subject to diversity while the second term represents the availability of targeted VNFs.

$$A(SFC) = \prod_{vnf \in SFC} A_{vnf} = \prod_{i \in \mathbb{N}_S} A_i \cdot \prod_{P \in SFC} \left(1 - \prod_{j \in \mathbb{N}_P} (1 - A_j)\right) \tag{2}$$

This implies that the availability of two (or more) VNFs combined in series is always **lower** than their individual availabilities. Similarly, the availability of two (or more) VNFs combined in parallel is always **higher** than their individual availabilities. Fig. 4 shows, in one hand, that the availability of an SFC in series configuration (i.e. no diversity case) is decreasing when the number of its components increases and we notice that the resulting availability is lower than the initial individual availabilities ($A(SFC)$ tends to 0 when the number of components increase despite using components that have initially an availability $A_i = 0.7$). On the other hand, the mere fact of connecting these components (with same individual availability) in a parallel manner, significantly improves the

TABLE I
NFVI AND SFC NOTATION

| PAR. | DESCRIPTION |
|---|---|
| | **NFVI** |
| $\mathbb{G}_{\mathbf{P}}$ | NFVI graph |
| $\mathbb{N}_{\mathbf{P}}$ | Set of physical nodes in $\mathbb{G}_{\mathbf{P}}$ |
| $\mathbb{E}_{\mathbf{p}}$ | Set of physical links between physical nodes |
| $\Theta^{n}$ | Available resource at physical node $n \in \mathbb{N}_{\mathbf{P}}$ |
| $\delta^{(n,m)}$ | Available capacity of physical link $(n,m) \in \mathbb{E}_{\mathbf{p}}$ |
| | **SFC** |
| $\mathbb{G}_{\mathbf{V}}$ | SFCs graph |
| $\mathbb{N}_{\mathbf{V}}$ | Set of VNFs $v$ in $\mathbb{G}_{\mathbf{V}}$ |
| $\mathbb{E}_{\mathbf{V}}$ | Set of virtual links between VNFs in $\mathbb{G}_{\mathbf{V}}$ |
| $\mathbb{N}_{\mathbf{V}}^{\mathbf{s}}$ | Set of VNFs composing the SFC $s$ where $\mathbb{N}_{\mathbf{V}}^{s} \subseteq \mathbb{N}_{\mathbf{V}}$ |
| $\mathbb{E}_{\mathbf{V}}^{\mathbf{s}}$ | Set of Virtual links composing SFC $s$ where $\mathbb{E}_{\mathbf{V}}^{\mathbf{s}} \subseteq \mathbb{E}_{\mathbf{V}}$ |
| $\Psi^{v}$ | Required resources of VNF $v \in \mathbb{N}_{\mathbf{V}}^{s}$ |
| $\Omega^{(k,l)}$ | Required capacity of virtual link $(k,l)$ connecting VNFs $k$ and $l$ |
| | **DIVERSITY & REDUNDANCY** |
| $D_i^v$ | The $i^{th}$ *replica* instance of VNF $v \in \mathbb{N}_{\mathbf{V}}^{\mathbf{s}}$ |
| $R_i^v$ | The $i^{th}$ *backup* instance of VNF $v \in \mathbb{N}_{\mathbf{V}}^{\mathbf{s}}$ |
| $\Psi^{D_i^v}$ | Requested resources of the $i^{th}$ *replica* instance of VNF $v \in \mathbb{N}_{\mathbf{V}}^{\mathbf{s}}$ |
| $\Psi^{R_i^v}$ | Requested resources of the $i^{th}$ *backup* instance of VNF $v \in \mathbb{N}_{\mathbf{V}}^{\mathbf{s}}$ |
| $\Omega^{(D_i^k, D_j^l)}$ | Requested capacity of virtual link $(D_i^k, D_j^j)$ connecting the $i^{th}$ instance of VNF $k$ and the $j^{th}$ diversity instance of VNF $l$ |
| $\mathcal{E}_v$ | Indicates whether diversity is applied on VNF $v$. |
| $\Delta^{v}$ | Diversity level tolerated to the VNF $v \in \mathbb{N}_{\mathbf{V}}^{\mathbf{s}}$ |
| $M_n^{D_i^v}$ | Binary variable indicating if *replica* $D_i^v$ is mapped into $n$ |
| $M_n^{R_i^v}$ | Binary variable indicating if *backup* $R_i^v$ is mapped into $n$ |
| $M_n^{LB(v)}$ | Binary variable indicating if the LB associated to *replicas* pool of VNF $v$ is mapped into $n$ |
| $M_{(n,m)}^{(D_i^k, D_j^l)}$ | Binary variable indicating if the virtual link between replicas of VNFs $l$ and $k$ is mapped into physical link $(n,m)$ |

resultant availability of the whole SFC (full diversity case it tends to 1 using components with $A_i = 0.7$) especially when adding more instances.

### *B. Problem Formulation*

In this section, we propose a placement and chaining solution for resilient service chain provisioning that is built jointly on diversity and redundancy approach, modeled as a Mixed Integer Linear Program (MILP). The inputs of the MILP are the network characteristics (such as resource capacities and hardware availabilities) and SFC requirements (in terms of resources and software availabilities). The output represents the optimal solution for placement and chaining of VNFs that minimizes the resources consumption while meeting the availability requirement of each SFC.

We denote the set of SFC requests by $\Gamma$. Each SFC request $\mathbf{s} \in \Gamma$ is modeled as a subgraph $\mathbb{G}_{\mathbf{V}}^{\mathbf{s}}(\mathbb{N}_{\mathbf{V}}^{\mathbf{s}}, \mathbb{E}_{\mathbf{V}}^{\mathbf{s}})$ where $\mathbb{N}_{\mathbf{V}}^{\mathbf{s}}$ is a set of VNFs composing the $s$ and $\mathbb{E}_{\mathbf{V}}^{\mathbf{s}}$ is a set of directed edges called virtual links connecting these VNFs. In addition, each SFC $s$ defines a set of characterizing metrics that might be related to resource consumption, e.g. number of requested vCPUs, or network performance, e.g. latency or bitrate. Therefore, each VNF instance $v \in \mathbb{N}_{\mathbf{V}}^{\mathbf{s}}$ of SFC $s$, requires a predefined amount of resources (computing, memory, storage) denoted by $\Psi^{v}$. Similarly, each virtual link $(k,l) \in \mathbb{E}_{\mathbf{V}}^{\mathbf{s}}$ connecting two VNFs $k, l \in \mathbb{N}_{\mathbf{V}}^{\mathbf{s}}$ has a bandwidth requirement denoted $\Omega^{(k,l)}$.

In the context of diversity, a targeted VNF $v \in \mathbb{N}_{\mathbf{V}}^{\mathbf{s}}$ can be split into a set of *replicas* denoted by $D_i^v$. The number of these instances is defined according to diversity level denoted $\Delta^{v}$. Table I. summarizes the used notations. The optimization objective is formulated as follow:

$$Min\left(\sum_{i=1}^{N}\sum_{n\in\mathbb{N}_{\mathrm{P}}}\sum_{v\in\mathbb{N}_{\mathrm{V}}}\Psi^{D_i^v}\cdot M_n^{D_i^v}+\sum_{i=1}^{P}\sum_{n\in\mathbb{N}_{\mathrm{P}}}\sum_{v\in\mathbb{N}_{\mathrm{V}}}\Psi^{R_i^v}\cdot M_n^{R_i^v} + \sum_{n\in\mathbb{N}_{\mathrm{P}}}\sum_{v\in\mathbb{N}_{\mathrm{V}}}\Psi^{LB}\cdot M_n^{LB(v)}\right) \quad (3)$$

$M_n^{D_i^v}$ $(resp. M_n^{R_i^v})$ is a binary variable indicating whether *replica* instance $D_i^v$ (*resp. backup* instance $R_i^v$) is mapped into the physical node (PN) $n$. Also, $M_n^{LB(v)}$ indicates whether the load balancer instance (LB) associated to the pool of *replicas* of VNF $v$ is mapped into node $n \in \mathbb{E}_{\mathrm{p}}$. The optimization objective is subject to the following constraints:

$$\sum_{i=1}^{N}\sum_{v\in\mathbb{N}_{\mathrm{V}}}\left(\Psi^{D_i^v}\cdot M_n^{D_i^v}\right) + \sum_{i=1}^{P}\sum_{v\in\mathbb{N}_{\mathrm{V}}}\left(\Psi^{R_i^v}\cdot M_n^{R_i^v}\right) + \sum_{v\in\mathbb{N}_{\mathrm{V}}}\left(\Psi^{LB}\cdot M_n^{LB(v)}\right) \leq \theta^{n} \quad \forall n \in \mathbb{N}_{\mathrm{p}} \quad (4)$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{(k,l)\in\mathbb{E}_V}\left(\Omega^{(D_i^k,D_j^l)}\cdot M_{(n,m)}^{(D_i^k,D_j^l)}\right)\leq\delta^{(n,m)}$$
$$\forall(n,m)\in\mathbb{E}_{\mathrm{p}} \tag{5}$$

$$1\leq\sum_{i=1}^{N}\sum_{n\in\mathbb{N}_{\mathrm{p}}}M_n^{D_i^v}\leq\Delta^v\quad\forall v\in\mathbb{N}_{\mathrm{V}} \tag{6}$$

$$\sum_{i=1}^{N}\sum_{v\in\mathbb{N}_{\mathrm{V}}}\left(\Psi^{D_i^v}\cdot M_n^{D_i^v}\right)+\sum_{i=1}^{P}\sum_{v\in\mathbb{N}_{\mathrm{V}}}\left(\Psi^{R_i^v}\cdot M_n^{R_i^v}\right)\geq\Psi^v+H(v,N)$$
$$\forall v\in\mathbb{N}_{\mathrm{V}} \tag{7}$$

$$\sum_{i=1}^{N}\sum_{j=1}^{N}\sum_{(k,l)\in E_v}\left(\Omega^{(D_i^k,D_j^l)}\cdot M_{(n,m)}^{(D_i^k,D_j^l)}\right)\geq\Omega^{(k,l)}$$
$$\forall(k,l)\in\mathbb{E}_{\mathrm{V}} \tag{8}$$

$$\sum_{m\in\mathbb{N}_{\mathrm{p}}}M_{(n,m)}^{(D_i^k,D_j^l)}-\sum_{m\in\mathbb{N}_{\mathrm{p}}}M_{(m,n)}^{(D_i^k,D_j^l)}=M_n^{D_i^k}-M_n^{D_j^l}$$
$$\forall n\in\mathbb{N}_{\mathrm{p}},\ \forall\left(D_i^k,D_j^l\right)\in\mathbb{E}_{\mathrm{V}} \tag{9}$$

$$\sum_{i=1}^{N}M_n^{D_i^v}+\sum_{i=1}^{p}M_n^{R_i^v}\leq 1\quad\forall v\in\mathbb{N}_{\mathrm{V}}\ \forall n\in\mathbb{N}_{\mathrm{p}} \tag{10}$$

$$\sum_{n\in\mathbb{N}_{\mathrm{p}}}M_n^{D_i^k}\leq 1\quad\forall k\in\mathbb{N}_{\mathrm{V}} \tag{11}$$

$$\frac{\Psi^{D_j^l}}{\Psi^l}\geq\frac{\sum_{i=1}^{N}\sum_{k\in\mathbb{N}_{\mathrm{V}}}\Omega^{(D_i^k,D_j^l)}\cdot M_{(n,m)}^{(D_i^k,D_j^l)}}{\sum_{k\in\mathbb{N}_{\mathrm{V}}}\Omega^{(k,l)}}$$
$$\forall k,l\in\mathbb{N}_{\mathrm{V}}\ \forall n,m\in\mathbb{N}_{\mathrm{P}} \tag{12}$$

$$\Psi^{R_i^k}\geq\alpha\cdot\Psi^k\quad\forall k\in\mathbb{N}_{\mathrm{V}}\ \forall n\in\mathbb{N}_{\mathrm{p}} \tag{13}$$

$$\Omega^{\left(R_i^k,R_j^l\right)}\geq\beta\cdot\Omega^{(k,l)}\quad\forall R_i^kR_J^L\in\mathbb{N}_{\mathrm{V}}\ \forall n\in\mathbb{N}_{\mathrm{p}} \tag{14}$$

$$M_n^{D_i^k}\cdot(1-\varepsilon_k)\,\varepsilon_l=M_n^{LB(v)}\quad\forall n\in\mathbb{N}_{\mathrm{p}}\ \forall(k,l)\in\mathbb{E}_{\mathrm{V}} \tag{15}$$

$$\sum_{n\in\mathbb{N}_{\mathrm{p}}}A_v^{software}\cdot A_n^{Hardware}\cdot M_n^{D_i^v}\geq 1-\left(1-A^{Th}(s)\right)^{1/|\mathbb{N}_{\mathrm{V}}^{\mathrm{s}}|}$$
$$\forall s\in\Gamma\quad\forall v\in\mathbb{N}_{\mathbf{V}}^{\mathbf{s}} \tag{16}$$

Constraint (4) ensures that the resources allocated to VNF instances do not exceed the available resource $\theta^n$ of PN $n\in\mathbb{N}_{\mathrm{p}}$ while constraint (5) guarantees that the bandwidth required by the virtual links mapped into physical link $(n,m)\in\mathbb{E}_{\mathrm{P}}$ does not exceed its available capacity $\delta^{(n,m)}$. We ensure that each VNF $\boldsymbol{v}$ meets the specified diversity level by defining constraint (6) that allows at most $\Delta$ *replica* instances per VNF when applying diversity. Constraint (7) guarantees that the sum of resources allocated to *replica* instances $D_i^v$ of the same VNF $\boldsymbol{v}$ meets the expected performance by consuming enough resources *i.e.* the associated overhead $H(v,N)$ in addition to required resources of the targeted VNF $\boldsymbol{v}$. Similarly, Constraint (8) allows the conservation of the capacity initially required by a virtual link between two VNFs ($k$ and $l$) through all the virtual links connecting different *replica* instances ($D_i^k$ and $D_j^l$). Constraint (9) is the flow conservation constraint to enforce the condition that for each virtual link $(D_i^k,D_j^l)\in\mathbb{E}_{\mathrm{V}}$ there must exist a continuous path $(n,m)\in\mathbb{E}_{\mathrm{P}}$ allocated between the pair of PNs $n,m$, where *replica* instances $D_i^k,D_j^l$ have been mapped.

Constraint (10) states that *replicas* and *backup* instances of each VNF $\mathbf{v}$ must be mapped into distinct (separated) PNs. In other words, this constraint achieves offsite placement and prevents placing instances protecting the same targeted VNF on the same physical entity. Constraint (11) states that each *replica* $D_i^v$ must be mapped only once into the physical infrastructure. In other words, the whole amount of resources (Computer, Memory and Storage) allocated to $D_i^v$ must be provided by only one PN $n$. Constraints (12) ensures a fair load balancing between the amount of resource allocated to *replica* instances and the networking capacity allocated to the virtual links that connect them. Constraints (13) and (14) are used to determine respectively the amount of computing and networking resource allocated to *backup* instances. We notice that for $\alpha=1$ (resp. $\beta=1$) refers to the case of a full redundancy scheme. Also tailored redundancy can be achieved by varying $\alpha$ (resp. $\beta$) value between ]0, 1[ to customize the amount of resource allocated to the *backup* instances that cover the targeted VNF. Constraint (15) allows mapping LB instances to the appropriate node. These instances are used to distribute traffic across *replicas* pool. However, when diversity is not applied on a given VNF the LB instance is not needed.

The availability required by each SFC $\boldsymbol{s}$, $A^{Th}(s)$ is considered as a threshold parameter for our solution. However, using the equation (2) as a constraint in our model will lead to nonlinear model. For this reason, we choose to define an alternative linear constraint (16). This latter ensures that the resultant availability when mapping a *replica* instance $D_i^v$ in node $n$ depends on the availability threshold $A^{Th}(v)$ where $\boldsymbol{v}$ is the targeted instance that belongs to the SFC $\boldsymbol{s}$ and its requested availability is given by:

$$A^{Th}(v)=A^{Th}(s)^{1/|N_v^s|} \tag{17}$$

For example, if the requested availability of SFC $\boldsymbol{s}$ composed of 3 VNFs (in series) is $A^{th}(s)=0.900$ thus, the individual availabilities of its VNFs should be at least $A^{Th}(v)=0.9^{1/3}=0.965$.

The proposed joint diversity and redundancy solution is based on the MILP model to solve the problem of placement and chaining for resilient service provisioning. Depending on how the protection will be applied, we can derive different variants of the proposed solution. A part of the formulation in this section can be reduced to the well-studied NP hard Virtual Network Embedding (VNE) since VNE is used for placing the VNFs of a given SFC onto a given infrastructure, however our solution goes further and determines for each SFC the placement, the number and the size of its VNFs (including both replicas and backup instances) in order to achieve a resilient SFC deployment. Also, finding approximations to the VNE problem formulation is difficult and doing so for our more complicated problem formulation to achieving resilient SFC deployment is even more complicated."

## V. Numerical Evaluation and Proof of Concept Implementation

This section presents the system implementation and evaluation results of the proposed joint diversity and redundancy solution. The first subsection is dedicated to the performance evaluation. We implemented three versions of our solution:

- **ALLDIV**, applies diversity on every VNF of an SFC, without distinction and no matter their individual availabilities.
- **RNDIV**, applies diversity in a random manner to a set of VNFs in an SFC.
- **SELEDI** selects the most sensitive (prone to failure) VNFs based on their individual availabilities to be diversified.

Since our model uses a joint diversity and redundancy, thus, each VNF targeted by diversity is mechanically protected by backup instances. Also, all aforementioned variants use the same tailored redundancy scheme. These latter are compared with the most relevant works namely N-to-N model presented in [13]. Several criteria are used to evaluate the performance of all these variants: resource consumption ratio, the mean availability, the number of generated instances, service outage and rejected request ratio.

The second subsection discusses the online, offline and scalability issues on the proposed solution while the last subsection describes a proof of diversity and redundancy concept. Implementation details and experimental results are presented in order to discuss the feasibility and usefulness of such mechanism for a video streaming service deployed within NFV networks.

TABLE II
Simulation Parameters

| Parameters | Value range |
|---|---|
| **Number of VNF per service** | 3 |
| **Requested resource $\Psi^v$** | 5 % |
| **Available resources at PNs $\Theta^n$** | set at 100% |
| **Required capacity of virtual link $\Omega^{(k,l)}$** | 5 % |
| **Available capacity of PL $\delta^{(n,m)}$** | set at 100% |
| **VNF software availability** | random 0.9-0.999 |
| **PN availability** | random 0.9-0.999 |
| **Desired SFC availability threshold** | random 0.7-0.8 |

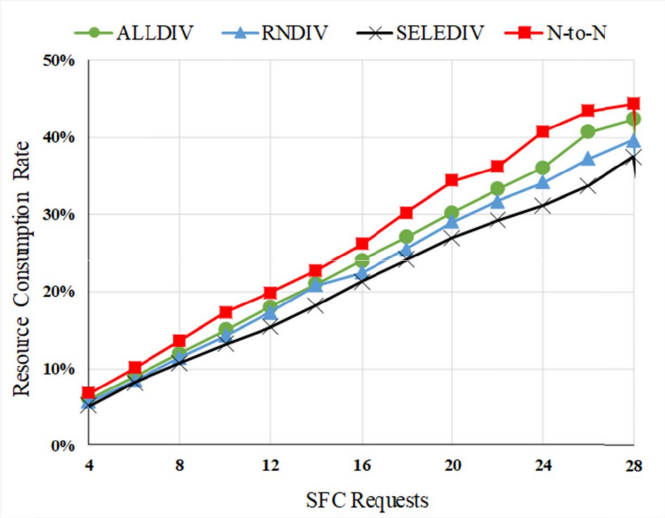


Fig. 5. Resource consumption rate when increasing the number of SFC requests.

### A. Numerical Evaluation

All our simulations are implemented using AIMMS [18] and Gurobi optimizer [19] is used to solve the MILP models on a machine equipped with an Intel 2.7 GHz processor and 16 GB RAM. We considered the NSFNET network as physical topology with 14 nodes and 22 bidirectional links. All physical nodes (PNs) can host VNFs and have the same capacity in terms of resource. Each PN has a random reliability between 0.9-0.999. All physical links have initially the same capacity. Additionally, we assume that each SFC consists of 3 VNFs connected by virtual links that require a portion of the bandwidth. We set availability requirement of every SFC randomly between 0.7-0.8 depending on the deployed service. It represents the availability level required by a given SFC as mentioned in its corresponding service-level agreement (SLA). During each simulation, the service requests are simultaneously deployed.All evaluations are repeated 10 times and the results show that the confidence intervals are negligible. Table II. summarizes simulation parameters.

*1) Resource Consumption:* Fig. 5 shows the resource consumption rate for different models in terms of deployed SFCs number. There is a clear trend of consuming more resources when dealing with an increasing number of SFCs regardless the used model. However, we can distinguish different consumption ratios between these models, SELEDIV exhibits the best performance compared to remain models which means that both diversity and redundancy are achieved using an optimal amount of resources. Unlike remain models, SELEDIV targets a limited set of VNFs (the most prone to failure) to apply protection which explains the gain on resource consumption.

Results show that RNDIV is more resource-efficient than ALLDIV and N-to-N since it targets less VNFs to protect. Also, ALLDIV consumes less resources compared to N-to-N since it tailors the redundancy instead of using implicit redundancy scheme. Particularly, as discussed before (in section II.B) N-to-N redundancy scheme can increase dramatically resource consumption when diversity equals to 2. When comparing SELEDIV and N-to-N, the obtained gain is up to 10% (for 24 SFC request) which is particularly significant when dealing with an increasing number of SFC requests. However, the resource consumption gain is not the only targeted goal in our proposed solution as we obtain performance enhancement in terms of availability and service outage.

Fig. 6 shows the mean resource consumption for all models in terms of diversity level. For the same set of 10 SFCs, the amount of consumed resources tends to slightly decrease when increasing the number of *replica* instances per VNF. ALLDIV have the worst performance compared to other models because it applies diversity and redundancy to all VNFs

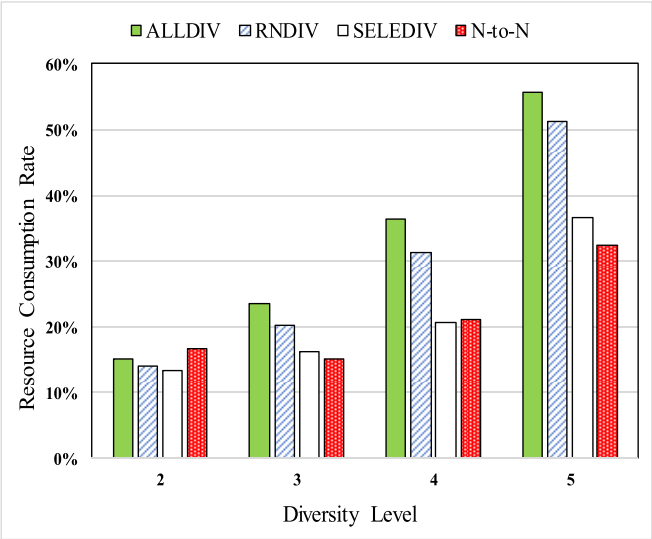


Fig. 6. Resource consumption rate when increasing the diversity level.

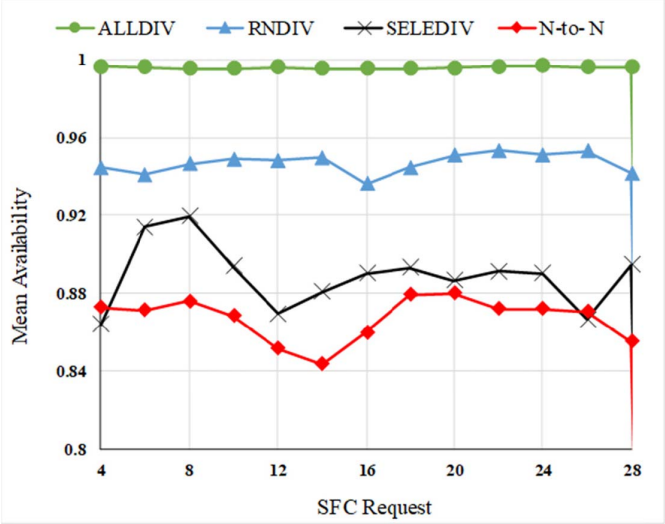


Fig. 7. Mean service availability when increasing the number of SFC requests.

which needs additional resources compared to the remain models. Both SELEDIV and N-to-N have *almost* similar behavior characterized by a low resource consumption since that both models protect less VNF compared to ALLDIV and RNDIV. The maximum gain obtained is up to 19% (ALLDIV vs. SELEDIV) while the minimum gain is almost close to zero for low diversity level.

Moreover, we note that diversity level directly impacts the amount of resource allocated to both backup instances (in the case of ALLDIV, RNDIV and SELEDIV) and spare resources for N-to-N case. Concretely, the allocated resource to the redundant instance is equal to maximum attributed resource to its *replicas* (see equation $(iv)$), while in N-to-N approach the spare resource (per *replica*) is equal to the number of *replicas* times the amount of resource allocated to one *replica,* divided by the number of remaining *replicas*.

Consequently, more diversity implies more resource consumption in terms of backup instance or spare resource. Though diversity is meant to improve the availability, it is important for service provider to seek for the accurate diversity level that allow to achieve a tradeoff between availability and resource consumption. In other words, even if under some conditions the resulting gain seems low (5%) it stills worthy to use our proposed solution as it allows reaching better availability level with less service outage and less resources consumption.

*2) Service Availability:* The availability is an important resilience indicator. Fig. 6 depicts a comparison between different models in terms of mean SFC availability when increasing the number of deployed services. We can notice that the mean availability provided by each model remains stable when increasing the number of SFCs. However, ALLDIV and RNDIV achieves better performance (nearly full availability $\approx 1$ in the case off ALLDIV) than SELEDIV and N-to-N. ALLDIV applies diversity on each VNF causing the rise of the entire SFC availability while RNDIV applies diversity randomly to SFCs. Both SELEDIV and N-to-Ntarget

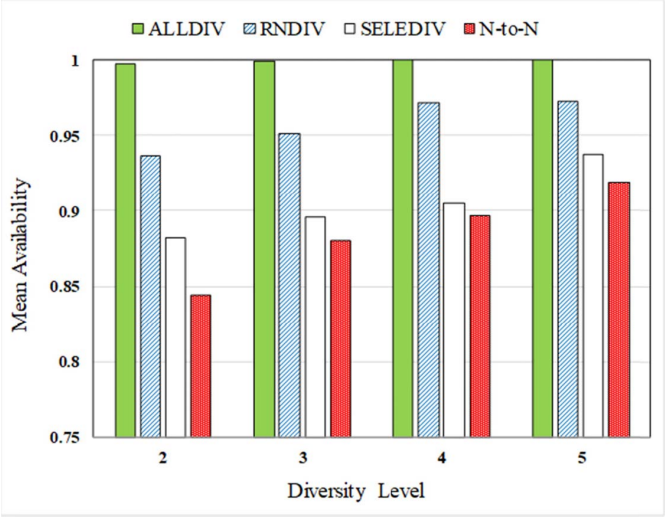


Fig. 8. Mean service availability when increasing the diversity level.

the most sensitive VNFs (low $A_i$) to enhance their individual availabilities to accurately reach the availability of their corresponding SFC. SELEDIV provides better availability since it uses backup instances which is not the case for N-to-N. Consequently, even if ALLDIV provides the best gain in terms of availability, we are more interested by accurately reaching the availability requested by each service than uselessly achieving a higher availability at the expense of available resources.

In Fig. 8, we plotted the mean SFC availability when applying different diversity level for a fixed set of SFC. In general, availability of a given VNF trends to rise when increasing the number of *replicas* (see Fig. 4). Therefore, in the case of ALLDIV, availability is high compared to the remaining models. This results from applying diversity indistinctively to all VNF without any selection. We notice also that RNDIV shows an irregular behavior when increasing diversity level since it randomly picks the VNFs which may initially have high or acceptable availabilities and ignores the

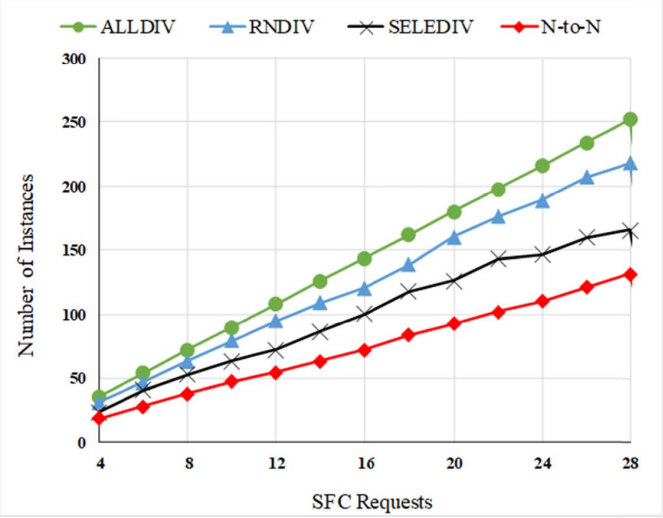


Fig. 9. Number of instances generated when increasing the number of SFC requests.

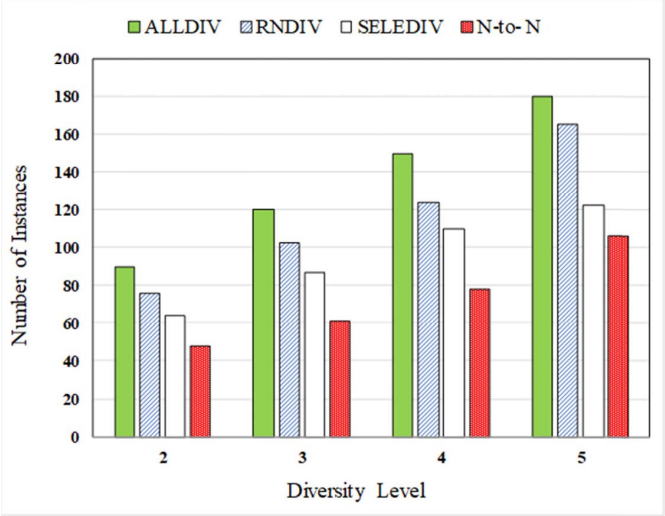


Fig. 10. Number of instances generated when increasing the diversity level.

vulnerable ones. The maximum gain in terms of availability was noted for low diversity level (diversity equal to 2).

However, we notice that systematically increasing diversity level may be useless since the required availability can be reached by applying low diversity level on specific VNFs, especially when we know from previous results that diversity level increases the resource consumption. Also, diversity level may impact the number of generated instances through the protection as presented in the following section

*3) Number of Instances:* This metric reflects the complexity of a model *i.e.* reaching the required resilience (meeting availability constraint) and impacts the placement time (more instances implies more time to place). Fig. 9 depicts the number of instances when increasing the number of deployed SFC. It is clear that the number of instances rises with the number of requests, although it is interesting to see the behavior of each model in response to such situation. As expected, ALLDIV generates a large number of instances compared to the other models that target a limited set of VNFs since ALLDIV applies all VNFs protection. The maximum gain in terms of generated instances is up to 36% when comparing SELEDIV with ALLDIV which is highly significant in this context.

Similarly, Fig. 10 shows the number of generated instances when increasing diversity level and thus placing more instances. This can lead to 1) unnecessarily increasing the execution time and 2) reducing feasible solutions space especially when applying offsite redundancy constraint.

*4) Service Outage Ratio:* We compared the service outage ratio for each model when facing different failure rates. A failure rate of 0.1 means that 10% of the most vulnerable pairs (*PN, VNF*) *resp. (PL,VL)* will fail. Thus, service outage ratio estimates the fraction of SFCs that were *almost* completely interrupted (for example loss of more than 70% of service capacity) due to a hardware/software failure. Fig. 11 shows that SELEDIV yields the lowest service outage ratio under different failure rates while the remain models seems more exposed to failure. Indeed, SELEDIV selects the most vulnerable SFCs and increases their respective availabilities using

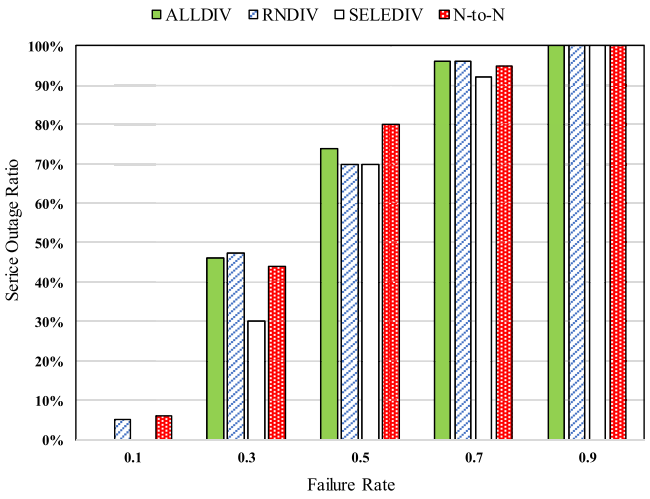


Fig. 11. Service outage ratio vs failure rate.

the proposed jointly diversity and redundancy mechanism to map them in PNs with a high availability (i.e. reduction of the risk of failure).

Protecting all SFCs in the case of ALLDIV implies using a large number of PNs (to meet offsite mapping constraint) including the ones with low availabilities (less reliable PNs). This will increase SFC service outage risk when increasing failure rate. Also, N-to-N does not use an explicit redundancy scheme (no backup instance just *replicas* with embedded spare resource) which makes it more prone to failure. Without diversity, the failure of a physical node that hosts a VNF of a given service will cause an immediate and a complete interruption of this service. However, with diversity case, the failure of a physical node will lead to a loss of a part of targeted VNF since the remaining *replicas* VNFs which are mapped elsewhere (due to offsite mapping strategy) guarantees the continuity of the service. Note that recovery mechanism may be more efficient when dealing with smaller instances (fast transfer, restoration and restoration of smaller contexts).

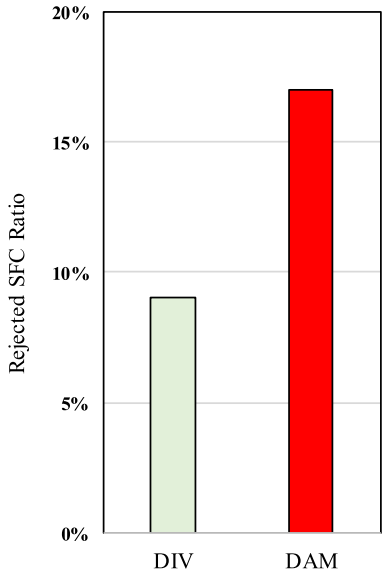


Fig. 12. SFC rejection rati.

Thus, the proposed joint VNF diversity and redundancy is designed to reduce the failure impact on service.

*5) Rejected Requests:* In order to observe diversity performance when facing resource fragmentation problem, we compare a pure diversity-based model (DIV) with a diversity agonistic model (DAM). Using the same set of 100 SFC, we measure the number of rejected requests by both models. A high rejection ratio indicates that these SFCs cannot fit in the underlaying available resource (even if globally there is enough resources) due to fragmentation problem that leads to an early exhaustion of network resources. As presented in Fig. 12, DIV rejects less requests than DAM. Indeed, DIV accepts 91% of the SFCs requests before starting rejection while DAM starts rejecting after reaching 83%. This difference in terms of acceptance rate is due to diversity that adapts VNFs requirements of each SFC to fit in the available fragmented resources. Whereas, DAM tries to map VNFs as they are, without any adaptation.

Indeed, some incoming SFC requests can be rejected despite the fact that there are enough available resources in the network to satisfy their requirements. In the example illustrated in Fig. 13, a network with a simple topology of 3 nodes (N1 …N3) in which we have already placed 2 SFC (SFC1 and SFC2) and an SFC3 is waiting to be placed. This SFC3 (*VNF7* → *VNF8* → *VNF9* with 8 CPUs are required) cannot be placed without diversity and will be rejected even if the network has enough resource to satisfy it (9 CPUs are available). The VNFs of SFC3, as they are shaped (sized), cannot fit into the available resources. This, deadlock situation, is what we call resource fragmentation problem. One way to address such resource fragmentation problem is to reshape the SFCs by replacing some VNFs (or all of them, depending on the situation) by a pool of thinner VNFs *replicas*. If we look closer, our proposed VNF diversity is naturally a proactive solution to resource fragmentation problem compared to some existing reactive solutions that use expensive migration mechanism.

Diversity takes advantage of the VNF *replicas* distribution in order to rescale the requirements of individual instances in terms of computing, memory and storage to suit with the actual network resources distribution. On this basis, instances requirements are adapted to the existing resources which leads to an efficient resource utilization.

### *B. Discussion: Online Placement and Scalability*

*1) Online SFC Placement:* In principle, the MILP can be executed in an online fashion processing each incoming SFC request when it arrives. However, this may not be practical considering its computational complexity. Our proposal is to execute our solution in a semi-online fashion by processing SFC requests in batches. The batches can be determined by selecting an appropriate time window for the execution of the solution. The time window can be fixed or variable depending on the number of SFC requests in the queue or scheduling constraints such as maximum waiting time as specified in the SLAs. We have not considered such factors as batch size or scheduling delay in our evaluation and proof of concept implementation since this requires a scheduling module (see Fig. 14) and a periodic updating of information about available resources. The scheduler handles the new SFC requests and those still in the waiting queue and selects among them the candidate SFCs that can be deployed on the infrastructure based on some objective function such as maximizing the number of accepted SFC requests, maximizing the revenue generated from the accepted SFCs, and/or minimizing the scheduling delay. These are indeed important considerations for the online enforcement and management of placement decisions and have a direct impact on its performance. In this paper, we have tried to focus on the resiliency aspect of SFC placement and assume we are given a batch of SFC requests as input and performed our evaluations accordingly. A complete system would require the implementation of a scheduler and the workflow as depicted in Fig. 14 is left for future work. In this workflow, the scheduler handles the new SFC requests and those still in the waiting queue and selects among them the candidate SFCs to be deployed on the infrastructure based on a predefined discipline (FIFO, minimum waiting time, etc.). Then, the MILP solver is executed to find a resilient placement of the selected SFCs on the underlying infrastructure. Once the placement and chaining are completed the available resources are updated before the placement of the subsequent batch of SFC requests.

*2) Solution Scalability:* It is worth noting that MILP-based solutions suffer from scalability issues in that they cannot handle large problem instances in a reasonable time. This is for instance the case for the general virtual network embedding problem where the size and number of virtual network requests as well as the size of the substrate (physical) network are very large. In the case of SFC placement however we assume that the size of the problem is manageable and hence warranting the computation of the optimal configuration. We believe that this assumption is reasonable for the following reasons. First. The number of SFC requests is limited compared to virtual network requests since an SFC is usually deployed as part of a network service processing aggregate

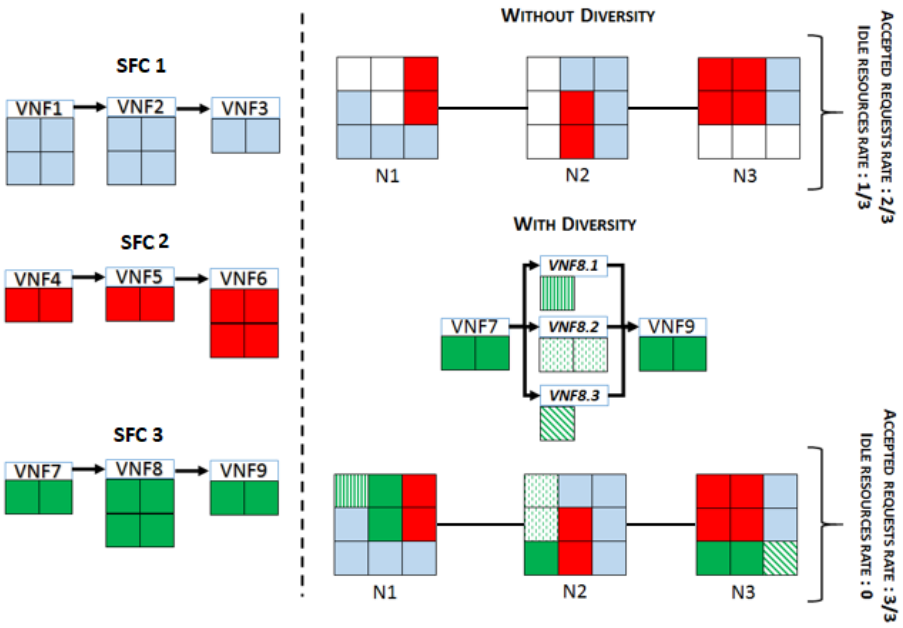


Fig. 13. Resource fragmentation problem.

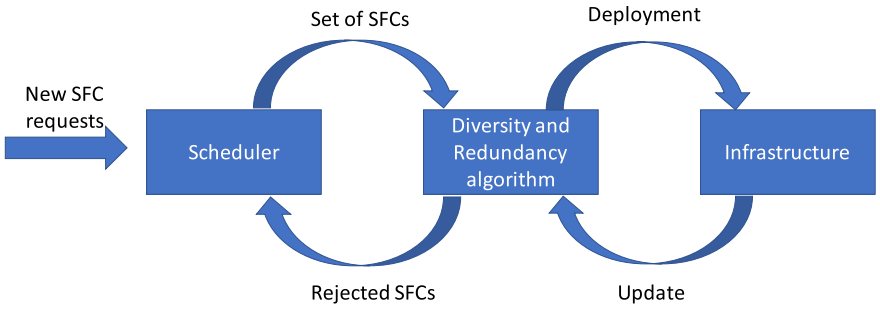


Fig. 14. Online SFC placement process.

traffic belonging to this service. Second, the size of SFCs is limited in the number of virtualized network functions deployed as part of the service chain. Indeed, the IETF Network and Service Chaining Working Group has several IETF drafts demonstrating SFC use-cases in operator networks [20], mobile networks [21], and data center networks [22]. The length of these SFCs as illustrated in these common use cases is between 3 and 7 virtualized network functions. Third, the size of the underlying infrastructure for the placement of the SFCs is also limited to the operator's edge and core clouds used for hosting the virtualized network functions of the chain. In contrast to virtual network embedding where many infrastructure nodes (i.e., all routers and switches) must be considered, the number of edge and core datacenters is significantly smaller. Finally, considering modern solvers and available computational capabilities these days (e.g., multicore machines with substantial RAM), it is possible to compute the optimal solution in a reasonable time without resorting to heuristic algorithms that are fast but achieve sub-optimal solutions or near optimal solutions in the best case.

### C. Proof of Concept Implementation

*1) Experiment Scenario:* The proposed joint diversity and redundancy of VNFs is implemented and tested in a virtual environment to realize a video streaming service. The test-bed topology as presented in Fig. 15 is based on the DigitalOcean Droplet [23] virtualization environment and the docker-machine tool [24] to ensure the deployment of the environment. Each node (droplet) is an Ubuntu 18.04 x64 server that hosts Docker engine to build, manage and containerize VNFs based on FFMPEG software to constitute

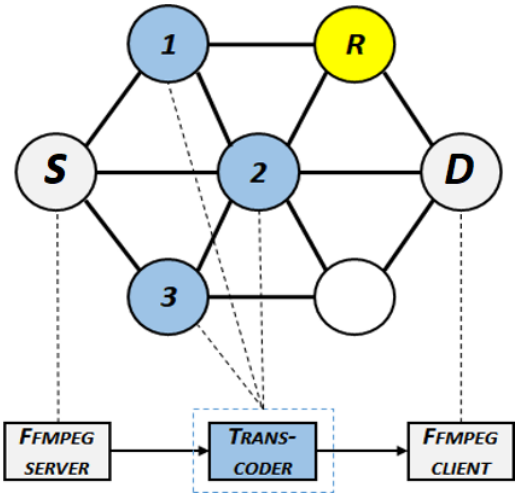


Fig. 15. VNF Diversity-Redundancy PoC topology.

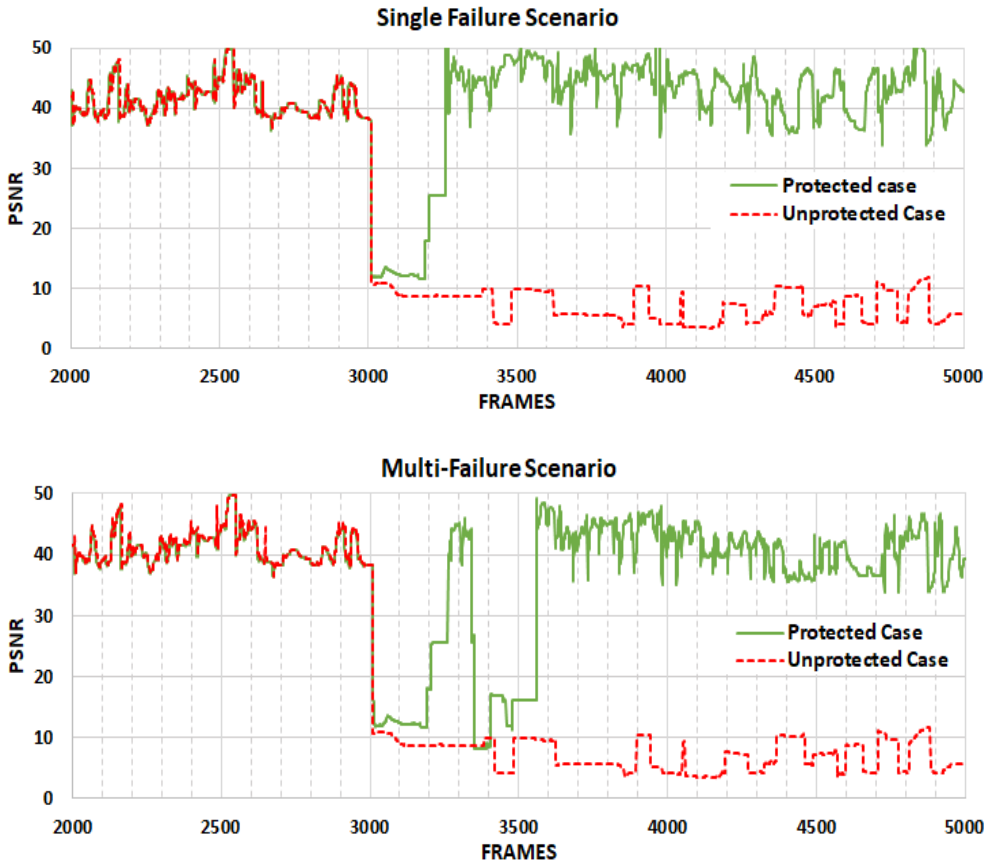


Fig. 16. PSNR Score Comparison for single and multi-failure scenario.

streaming function chain service. The Sever node (S) and Client node (D) host respectively *ffmpeg* server (with Load Balancer) and *ffmpeg* client container while core nodes host *ffmpeg* transcoder containers.

In this scenario, the video streaming service is presented as VNF forwarding graph with three VNFs based on FFMPEG software: Video server (4CPU), transcoder (8 CPU) and Video client (4CPU). On the server-side (*ffmpeg* server), video parameters are set to $480 \times 360$ and 24 fps. The transcoder uses H.264 codec with *ffmpeg* as VNF. After service deployment, these parameters are instantly captured and enforced to suit the client side (*ffmpeg* client). Each node is connected via virtual switches based on a set of *Openflow* rules that produce the desired network topology.

In order to evaluate QoS/QoE of the video streaming service, we use an objective quality assessment based on Peak Signal-to-Noise Ratio (PSNR) which compare the received video against the original transmitted one. PSNR is defined as the mean squared error between the original video frames at server side and the reconstructed ones at the client side.

Our proposed joint diversity and redundancy can improve the resiliency of this service. We use SELEDIV variant to select the *ffmpeg* transcoder as the diversity target. Consequently, this VNF will have three *replicas ffmpeg* transcoder for diversity (4CPU for *replica* 1, 2CPU *replica* 2 and 2CPU for *replica* 3) and one *replica* for redundancy (4 CPU for *replica* $R$) that are mapped to distinct nodes (blue ones for diversity instances and yellow node for redundancy) in order to ensure connection between *ffmpeg* server and client instances (grey nodes as shown in Fig. 15). The Server node (S) hosts also a weighted Round Robin Load Balancer where the number of connections with replicas over time is proportionate to their respective amount of allocated resources. For example, *replicas 1* can handle twice the traffic of *replicas 2* and *3*, and thus the load balancer should send two connections to *replica 1* for each connection sent to *replicas 2* and *3*. Since replicas are not uniform, weighted Round Robin LoadBalancer is suitable to implement our solution.

After deploying video streaming SFC, we simulated node failures and we performed traffic steering over the redundant *replicas*. We show hereafter how the joint diversity and redundancy can improve service resilience based on the PSNR score obtained from our measurement study.

*2) Results and Discussion:* Fig. 16 shows the PSNR score for protected and unprotected cases before, during and after failure in both single and multi-failure scenarios.

*a) Single failure scenario:* As presented in Fig. 16, before failure, i.e. between [0,3000] video frames, the PSNR scores in both protected and unprotected cases are almost similar around 43.3 dB on average. Though, once failure (node 2) occurs at nearly frame 3000, we notice divergent PSNR behaviors that reflect different video quality. In unprotected case, PSNR drops rapidly from 42 to 11 and stays under 10 until the end of measurement. This low PSNR score (average 6.56) recorded after failure means that the whole service is interrupted (black screen) because transcoding was ensured by a single instance running on node 2. In contrast, in the protected case, the PSNR drops to 12 before starting to recover until reaching its score

before failure *i.e.* 44 dB (from 12 to 25 than 44). The recovery of PSNR level, after failure, is due to *replicas* redundancy R that covered the lack of performance caused by the loss of *replicas* instance in node 2. Additionally, the PSNR gap between frame [3000,3260] indicates the time that elapsed between failure detection and traffic redirection towards the redundant *replicas* which were already instantiated and on a standby mode.

*b) Multi-failure scenario:* In this scenario, we deliberately cause two successive failures (node 2 then node 3) to observe the behavior of our solution when facing several failures. As presented in Fig. 16 from the beginning of experiment until the recovery from the first failure, PSNR scores are almost identical to the first scenario. However, after the second node failure (node 3), the PSNR drops to 9 before rising to 48 in the protected case. Regardless of the number of failures, unprotected case continues to provide a low PSNR score since its transcoder VNF is no longer operational. On the other hand, protected case recovers the initial video quality. Thus, even when two nodes are impacted, the service may be impaired but still working at acceptable level. The continuity is ensured by both the remaining *replica* (in node 1) that was not impacted by failures and then the redundant *replica* that has enough capacity to cover the lack of capacity (4 CPU).

Furthermore, it is quite unlikely that three or more independent failures occur simultaneously because of the offsite placement strategy (constraint 11) that guarantees mapping *replicas* of the same targeted into distinct physical nodes. Thus, in our experiment, even if the three instances (hosted by node 1, 2 and 3) fail, the backup instance R can ensure 50% of service (providing safe mode performance) which avoids a complete service outage.

## VI. Conclusion

This paper proposed a VNF placement solution that employs joint diversity and redundancy to achieve resilient service function chains. The originality of the proposed approach lies in its use of selective diversity to improve global service availability by targeting failure-prone VNFs and in its reliance on tailored redundancy for optimal resource utilization. Specifically, we developed several variants of our solution approach, namely ALLDIV, RNDIV and SELEDIV and evaluated them both numerically and experimentally. The analytical results demonstrate the efficiency of our solution in terms of resource consumption and service availability. The implemented prototype using a virtual environment for video streaming service demonstrates that the proposed solution can be used to avoid complete service outages. As part of our future work, we plan to devise online algorithms for adaptive placement that can identify over time the most sensitive VNFs and even predict possible failures using machine learning techniques.

## References


[1] J. P. G. Sterbenz, E. K. Çetinkaya, M. A. Hameed, A. Jabbar, S. Qian, and J. P. Rohrer, "Evaluation of network resilience, survivability, and disruption tolerance: Analysis, topology generation, simulation, and experimentation," *Telecommun. Syst.*, vol. 52, no. 2, pp. 705–736, Dec. 2011.

[2] J. P. G. Sterbenz *et al.*, "Resilience and survivability in communication networks: Strategies, principles, and survey of disciplines," *Comput. Netw.*, vol. 54, no. 8, pp. 1245–1265, Jun. 2010.

[3] M. Scholler, M. Stiemerling, A. Ripke, and R. Bless, "Resilient deployment of virtual network functions," in *Proc. 5th Int. Congr. Ultra Modern Telecommun. Control Syst. Workshops (ICUMT)*, Sep. 2013, pp. 208–214.

[4] 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.

[5] 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.

[6] R. Cohen, L. Lewin-Eytan, J. S. Naor, and D. Raz, "Near optimal placement of virtual network functions," in *Proc. IEEE Conf. Comput. Commun. (INFOCOM)*, Apr. 2015, pp. 1346–1354.

[7] 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.

[8] H. Moens and F. D. Turck, "VNF-P: A model for efficient placement of virtualized network functions," in *Proc. 10th Int. Conf. Netw. Service Manage. (CNSM) Workshop*, Nov. 2014, pp. 418–423.

[9] M. Xia, M. Shirazipour, Y. Zhang, H. Green, and A. Takacs, "Network function placement for NFV chaining in Packet/Optical datacenters," *J. Lightw. Technol.*, vol. 33, no. 8, pp. 1565–1570, Apr. 15, 2015.

[10] D. Cotroneo *et al.*, "Network function virtualization: Challenges and directions for reliability assurance," in *Proc. IEEE Int. Symp. Softw. Rel. Eng. Workshops*, Nov. 2014, pp. 37–42.

[11] S. Herker, X. An, W. Kiess, S. Beker, and A. Kirstaedter, "Data-center architecture impacts on virtualized network functions service chain embedding with high availability requirements," in *Proc. IEEE Globecom Workshops (GC Wkshps)*, Dec. 2015, pp. 1–7.

[12] P. Vizarreta, M. Condoluci, C. M. Machuca, T. Mahmoodi, and W. Kellerer, "QoS-driven function placement reducing expenditures in NFV deployments," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2017, pp. 1–7.

[13] F. Carpio and A. Jukan, "Improving reliability of service function chains with combined VNF migrations and replications," Nov. 2017, *arXiv:1711.08965*. [Online]. Available: https://arxiv.org/abs/1711.08965

[14] P. Gill, N. Jain, N. Nagappan, P. Gill, N. Jain, and N. Nagappan, *Understanding network failures in data centers*, vol. 41, no. 4. New York, NY, USA: ACM, 2011, p. 350.

[15] Y. Deswarte, K. Kanoun, and J.-C. Laprie, "Diversity against accidental and deliberate faults," in *Proc. Comput. Secur., Dependability, Assurance, From Needs Solutions*, Jul. 1998, pp. 171–181.

[16] ETSI Industry Specification Group (ISG) NFV, "Network functions virtualisation (NFV); reliability; report on models and features for end-to-end reliability," NFV, Tech. Rep. ETSI GS NFVREL 1, 2016.

[17] N. Gray, C. Lorenz, A. Mussig, S. Gebert, T. Zinner, and P. Tran-Gia, "A priori state synchronization for fast failover of stateful firewall VNFs," in *Proc. Int. Conf. Networked Syst. (NetSys)*, Mar. 2017, pp. 1–6.

[18] J. Bisschop, *AIMMS Optimization Modeling*. Morrisville, NC, USA: Lulu.com, 2006.

[19] Optimization-Gurobi. (2012). *Gurobi Optimizer Reference Manual*. [Online]. Available: http://www.gurobi.com/

[20] O. Huang *et al.* (Sep. 2014). *Service Function Chaining (SFC) General Use Cases*. [Online]. Available: https://tools.ietf.org/html/draft-liu-sfc-use-cases-08

[21] W. Haefner, J. Napper, M. Stiemerling, D. R. Lopez, and J. Uttaro, "Service function chaining use cases in mobile networks," *Internet-Draft*, Apr. 2016. [Online]. Available: https://tools.ietf.org/html/draftietf-sfc-use-case-mobility-06

[22] S. Kumar, M. Tufail, S. Majee, C. Captari, and S. Homma, "Service function chaining use cases in data centers," IETF, Fremont, CA, USA, 2015.

[23] *Docs Home?: DigitalOcean Product Documentation*. Accessed: Dec. 17, 2018. [Online]. Available: https://www.digitalocean.com/docs/

[24] *Docker Machine | Docker Documentation*. Accessed: Dec. 17, 2018. [Online]. Available: https://docs.docker.com/machine/

**Abdelhamid Alleg** received the degree in computer science engineering from the Polytechnic School, Bordj el Bahri, Algeria, in 2009, and the Ph.D. degree from Bordeaux University, France, in 2019.

He performed his research activities at the LaBRI Lab-UMR 5800, CNRs, University of Bordeaux, France. His topics of interest are in the field of multisensory data processing, networking and service deployment, and linear programming applied to resource management in virtualized networks and functions.

**Toufik Ahmed** is currently a Full Professor with ENSEIRB-MATMECA School of Engineers, Institut Polytechnique de Bordeaux (Bordeaux INP). He performs his research activities at the LaBRI Lab-UMR 5800, CNRs, University Bordeaux, Bordeaux INP. His main research activities investigate network function virtualization, software-defined network, and end-to-end Quality of Service (QoS) management and provisioning for wired and wireless next generation networks. He has worked on a number of national and international projects, and he is serving as a TPC member for international conferences and journals.

**Mohamed Mosbah** received the Ph.D. degree from the University of Bordeaux in 1993. He is currently a Full Professor of computer science with the Polytechnic Institute of Bordeaux, France. He carries his research at the LaBRI, a research laboratory in computer science common with the University of Bordeaux and CNRS, where he is the Deputy Director. His research interests include distributed algorithms and systems, formal models, security, intelligent transportations systems, and ad hoc and sensor networks. He participated in several national and European research projects, including collaborations with industry. He wrote more than 80 research articles published in international journals and conference proceedings, and he is involved in various technical program committees and organizations of many international conferences.

**Raouf Boutaba** (Fellow, IEEE) received the M.Sc. and Ph.D. degrees in computer science from Sorbonne University, Paris, in 1990 and 1994, respectively. He is currently a University Chair Professor with the David R. Cheriton School of Computer Science, University of Waterloo, Canada, and holds an INRIA International Chair in Nancy, France. His research interests are in the areas of network management and resource virtualization. He is a fellow of the Engineering Institute of Canada, the Canadian Academy of Engineering, and the Royal Society of Canada.