Abstract

Telecom operators worldwide are witnessing squeezed profit margins mainly due to hyper-competition. Hence, new business models/strategies are needed to help operators reduce Operational and Capital Expenditures. In this context, the Network Function Virtualization (NFV) paradigm, which consists of running Virtual Instances of Network Functions (NFs) in Commercial-Off-The-Shelf (COTS) hardware, represents a solid alternative. Virtual Network Functions (VNFs) are then concatenated together in a sequential order to form Service Chains (SCs) that provide specific Internet services. In this paper we study different approaches to provision SCs with resiliency against single-link and single-node failures. We propose three Integer Linear Programming (ILP) models to jointly solve the problem of VNF placement and traffic routing, while guaranteeing resiliency against single-link and/or single-node failures. Specifically, we focus on the trade-off between the conflicting objectives of meeting SCs latency requirements and consolidating as much as possible VNFs in NFV-capable nodes. We show that providing resiliency against both single-link and single-node failures comes at twice the amount of resources in terms of NFV-capable nodes, and that for latency-critical services providing resiliency against single-node failures comes at the same cost with respect to resiliency against single-link and single-node failures. Finally, we detract important insights about the deployment of bandwidth intensive SCs.

1 Introduction

Network operators rely on hardware appliances to provide Internet service. Each Internet service is usually provided thanks to the adoption of a purpose-built hardware that implements specific network functions (i.e., Firewalls, Nat, IDPS, etc.) within the network [1].

From the cost point of view, telecom operators are witnessing a decrease of the revenue-per-bit, which is envisioned to be even lower to the cost-per-bit, due to the competition from Over The Top (OTTs). The applications introduced by OTTs (i.e., VoIP) leave the Internet Service Provider (ISP) responsible only for transporting the information, hence contributes heavily in their revenue decrease. Network Functions Virtualization (NFV) is a new architectural paradigm that was proposed to improve the flexibility of network service provisioning and reduce the time to market of new services [2]. NFV can revolutionize how network operators design their infrastructure by leveraging virtualization to separate software instances from hardware appliances, and decoupling functionalities from locations for faster service provisioning. NFV supports the instantiation of Virtual Network Function (VNFs) through software virtualization techniques and runs them on Commercial-Off-The-Shelf (COTS) hardware. Hence, the virtualization of network functions opens the way to the provisioning of new services without the installation of new equipment. It is clear that NFV brings a whole new dimension to the landscape of telecommunication industry market due to the possibility of reducing capital investments, energy consumption by consolidating network functions, and by introducing tailored services based on customers needs. Moreover, NFV simplifies service deployment by exploiting the concept of *service chaining* [3]: a Service Chain (SC) is a sequential concatenation of VNFs and/or hardware appliances to provide a specific Internet service (e.g., VoIP, Web Service, etc.) to the users. Deploying NFV solutions in operational networks require solving multiple issues related to performance, availability, security and survivability. One important key design in an NFV framework is the ability of the NFV-Management and Orchestration (NFV-MANO) component (see Fig. 3.1) to ensure service continuity. Such objective translates into many requirements that the Virtual Network Function Infrastructure (NFVI) must satisfy, among which resiliency and geo-redundancy requirements. Hence, the deployment of SCs must meet a given resiliency level and aim to consolidate as much as possible the VNFs within

NFV-nodes (i.e., With *NFV-nodes*, we refer to those nodes in the physical network than can be used to instantiate VNFs), as an indiscriminate distribution of VNFs instances would lead to an increase of the costs. In this paper we address the issue of resiliency against single-link/node failures. We tackle this problem with the aim to investigate the trade-off between latency requirements and the amount of resources required in terms of *NFV-nodes*. To this objective, we propose three different Integer Linear Programming (ILP) models to jointly solve the VNF placement and routing problems with service chaining while guaranteeing resiliency against single-link failures, single-node failure and single-node/link failures.

The main contributions of this work are the following:

- We show the amount of resources needed, for each resilient design, and compare them with the unprotected scenario, for SCs with different latency and bandwidth requirements.
- We observe that traffic processing by VNFs causes a variation of data-rate, represented by compression factors, and include this aspect into the optimization framework.
- We investigate the trade-off between node consolidation and the average hop count, for different resilient design scenarios.
- We solve the ILP models considering the conflicting objective of VNFs consolidation within *NFV-nodes* and load balancing and derive important insights for the deployment of different SCs.

Numerical results indicate that, generally providing resiliency against single-node/link failures comes at the same cost as the resiliency against single-node failures. Moreover, for latency stringent SCs, we find that in order to provide resiliency against single-link failures the operator must place backup VNFs in physical disjoint locations. In addition, for SCs with loose latency requirement, we observe that a trade-off between the average length of primary and backup path and exists. Finally, we analyze the effect of bandwidth requirement of two SCs with the same latency requirements and find that balancing the load on physical link is beneficial for small values of node capacity, expressed in terms of CPU cores it is equipped with.

The rest of this paper is organized as follows. Section 2 discusses the NFV and the service-chaining concept and overviews related works. Section 3 discusses general requirements for resiliency and failures models in NFV, as per standards guidelines. In Section 4 we present the network model used, while in Section 5 we present the resilient design scenarios and discuss their failure prevention potential. In Section 6 the resilient SCs provisioning problem is formally stated and the ILP models proposed to solve it are shown. In Section 7 we present the case-studies and show the obtained numerical results. Finally, conclusions and future works are discussed in Section 8.

2
Related work

NFV is still a concept under standardization. Currently, a number of standardization activities in the NFV area are carried by ETSI and IETF [4] [5]. In the following we present a collection of works related to the Service Chaining problem, with a specific emphasis on NFV reliability.

2.1
The Service Chaining problem

The problem of embedding SCs into a physical infrastructure consists in solving the placement of VNFs and traffic routing problems. It can be considered as an extended version of two NP-hard problems: Virtual Network Embedding (VNE) [6],[7] and Location-Routing Problem (LRP) [8]. The similarity with VNE resides in the fact that SCs can be considered as *virtual networks* characterized by a chain topology where VNFs represent virtual nodes, chained together through virtual links that must be mapped to a physical path.The similarity with LRP consists in jointly considering the problem of finding the optimal placement of VNFs, among a set of potential locations, along with the routing between VNFs. The LRP combines this two planning tasks and solves them with the objective to reduce costs of nodes, edges or paths. Regarding the differences, the Service chaining problem require that the routing of traffic between the VNFs occurs according to a specific ordered sequence. Moreover, the sharing of VNFs between multiple SCs increase the number of combinatorial possibilities for the embedding of the SCs.

Several works dealing with the VNF placement and routing problems appeared in literature. Ref. [9] formalizes the VNF and SC concepts and develops an ILP model for the optimal placement of VNF and SCs. In [10] An extended version of the model considers that the upscaling of an existing VNF introduces additional cost, whereas hosting multiple VNFs within the same physical nodes introduces context switching costs. Our model leverages and extends both the above mentioned works by including resiliency aspects. In [11] an online algorithm that considers jointly the Virtual Machine (VM) placement and routing is proposed. Finally, authors in [12] formulate and ILP and a greedy heuristic for the VNF placement and routing

problem, including traffic compression/decompression constraints, and adopting two different forwarding latency regimes. The obtained results draw interesting considerations on NFV deployment strategies. However, this work assumed a completely reliable NFV infrastructure, which is not realistic. Authors in [13] focus on the deployment of VNFs in a hybrid environment where some NFs are virtualized and others use specialized hardware appliances. Finally, authors in [14] propose an ILP and a game theory model to capture the competition on physical resources between network function instance allocation and routing. However, they do not consider any resiliency aspects.

2.2 Reliable NFV deployment

Authors in [15] descrive some NFV-related reliability issues and discusses the types of failures that may arise from both hardware (i.e., shutdown of physical machine, hardware issues) and software (i.e., cyber attacks, bugs, etc.). A more detailed discussion on reliability challenges in NFV network scenarios can be found in [5]. Network reliability in NFV-enabled networks is a new problem whose resolution has not yet attained maturity even though few preliminary works have already appeared. Ref. [16] addresses the problem of Joint Topology Design and Mapping (JTDM) in a Telco Cloud (TC) environment. The authors propose an efficient heuristic algorithm that leverages the feedback obtained from mapping the critical sub-topologies of an SFC, to better coordinate and jointly optimize the VNF combination and SFC mapping. They extend such algorithm with dedicated and shared protection scheme and compare the results with a baseline scenario (i.e., unprotected). However, they do not consider latency requirements on the SFC and the processing delay introduced from the sharing of VNFs. Ref. [17] presents a framework for reliability evaluation of NFV deployment, and three heuristic algorithms to identify the minimum number of physical and logical nodes which removal lead to the failure of an NFV deployment. Ref. [18] proposes software Defined Networking (SDN) and NFV benchmarking test metrics for performance and reliability from the perspective of the operators. Ref. [19] presents ILP and heuristic solutions that exploits multiple backup nodes for the purpose of provisioning each of the supported network services with reliability guarantees. However, in this work authors focus only on the failures that might happen within the hardware hosting the VNF and, unlike in our work, discard the possibility of link failures and the failure of other network elements within the nodes. Moreover they assume that the backup VNF are placed in different physical machines than those hosting the primary VNF, but in the same physical location, whereas our models consider also scenarios with disjoint physical locations between primary and backup VNFs. Ref. [20] presents a VM placement method to achieve redundancy against host-server failures with a minimum set of servers. The idea is to minimize the resources while ensuring a certain protection level. With respect to our work no consideration is made on the resource sharing and the performance requirements of the VNFs that run on the VMs. Moreover, the authors focus only on failures that

occur within the physical nodes, while we include also failures of physical links. Finally, Ref. [21] proposes a model to describe the components of services along with a management system to deploy such information model, with the objective to provide an automated and resilient deployment. Apart from the differences in the general approach, authors in [21] focus on resiliency of a single VNF, whereas we consider the resiliency of the whole SC.

3
NFV architecture and resiliency guidelines

In the following we introduce the architecture used in this study and we highlight the role of some of its primary components. Successively, we discuss few of the relevant resiliency guidelines and illustrate the possible VNF failure models, as per [5].

3.1
NFVI

The NFV architecture, shown in Fig. 3.1, is a combination of both hardware and software resources which make up the environment in which VNFs are deployed. The physical resources include COTS hardware, on top of which virtual resources are abstracted. The abstraction is achieved through the virtualization layer (based on hypervisor) which decouples the virtual resources from the underlying physical resources. NFV-MANO provides the necessary functionalities to provision VNFs, and all the related operations such as configuration, orchestration, and life-cycle management, etc. Moreover, MANO plays an important role in achieving a resilient NFV deployment. In the following we discuss standard guidelines for a resilient deployment of VNFs and show the different failure models that arise, with the virtualization of NFs.

3.2
General requirements for resiliency

Different Internet services have different requirements in terms of service continuity and maximum tolerated latency. For instance, in case of a Web-Service, outages lasting seconds are tolerable and the user typically initiates retries, whereas in the telecom domain (i.e., phone calls) outages must last less than a certain expected level (i.e., few milliseconds). In the NFV framework, not every network function has the same requirements for resiliency. Consequently the virtualization of NFs needs to fulfill multiple design criteria, such as service continuity, automated recovery from failures, prevent single point of failures in the NFV infrastructure as well in the under-

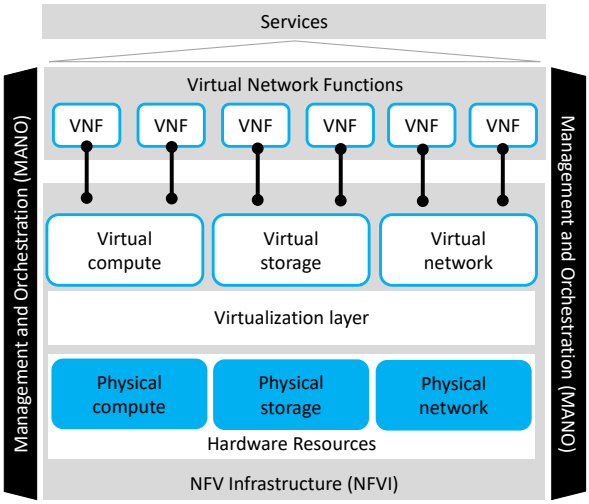


Figure 3.1 Simple illustration of NFV architecture

lying infrastructure. Here below we present some important resiliency requirements, according to ETSI guidelines [5]:

- The virtualized Network Function (VNF) needs to ensure the availability of its part of the end-to-end service, just as in the case of a non-virtualized NF.
- The whole NFV framework must not contain single point of failure with the potential to endanger service continuity. Thus, mechanisms to recreate any failed component to its state prior to failure, and to support recovery from total component failure must exist.
- The NFVI shall provide the necessary functionality to enable high availability at the VNF level, such as failure notification and remediation.

Besides the relative availability of a service, the impact of failures is also an important aspect for network providers in terms of service continuity. To limit the potential of failure impacts, the VNF limitations in terms of number of parallel users allowed, parallel transactions to be handled, etc. must be accurately defined. Our models include follows such guidelines and analyzes their impact on different network parameters.

3.3 VNF failure modes

Depending on the type of VNF deployment, the impact of failure will vary, hence the survivability method differs. In Fig. 3.2 we show the non-virtualized deployment of NF (option 1). The straightforward approach to virtualize such environment

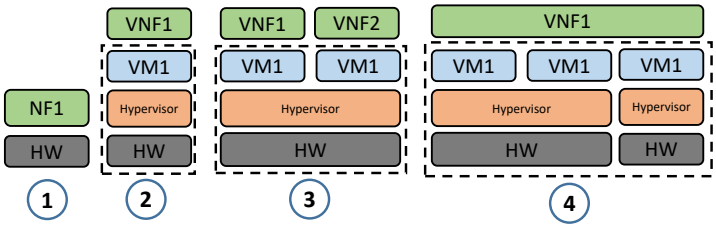


Figure 3.2 Deployment options of VNFs along with classical approach

is to take the network function software and run into a VM image and execute it on virtual resources provided by the hypervisor (option 2). This scenario add new failure mode to the existing ones since the failure on the supervisor do not exist in the *"box-model"*. In addition, to achieve high hardware utilization, the physical resources are sliced into multiple virtual machines. Hence, different VNFs can be hosted by the same hardware. This design might cause performance degradation if the VNF1 and VNF2 resources isolation does not work. Finally, a VNF that is composed by multiple VNFs can be hosted on different VMs running on the same physical hardware, or on different physical hardware. Again, new failures modes are introduced due to possible simultaneous failures of multiple VNF components caused by the failure of the underlying hardware or connectivity failures between VNF components (i.e., physical node/link failures). In this work we assume that the VNFs are running on physical machines according to option 2 or option 3. Hence, the failure of the physical nodes (i.e., hardware or hypervisor failures), would cause the failure of all the VMs running in that specific node. We also assume that the VNFI components do not constitute a single point of failure. In sec. 5 we discuss the possible redundancy modes to protect state-full and state-less VNFs and illustrate the possible protection designs for each category of VNFs.

4
Service chains and network model

4.1
Network model

We model the physical network as a directed graph composed of a set of physical nodes (which can host VNFs or only act as forwarding nodes) and a set of physical links representing the set of fiber links. Each physical link is associated with a bandwidth capacity. The physical nodes equipped with COTS hardware are referred to as *NFV-nodes* and can have different amount of processing capacity in terms of number of CPU cores that they are equipped with.

4.2
Service chains model

Service chains are composed by sequential concatenation of multiple VNFs. To deploy a SC, an operator needs to find the right placement of VNFs into the *NFV-nodes* in the physical network and chain them through a physical path. Different SCs can share multiple VNFs and different VNFs can be placed into the same physical *NFV node*. As shown in Fig. 4.1, two SCs composed of different VNFs have both as start point the physical node v_1 and as end point the physical node v_6. In addition, VNF1 is shared among the two SCs and mapped to physical node v_2 which shall be equipped with enough processing capacity to host such VNF. Finally, we assume that each VNF is assigned one CPU core into a VM.

4.3
VNF model

Generally, a VNF is an abstracted object that performs operations on input traffic. Each VNF has a processing capability which corresponds to the number of CPU Cores that are assigned to the VM that host that VNF. Moreover, we assume that each service corresponds to one SC modeled through a simple line graph composed

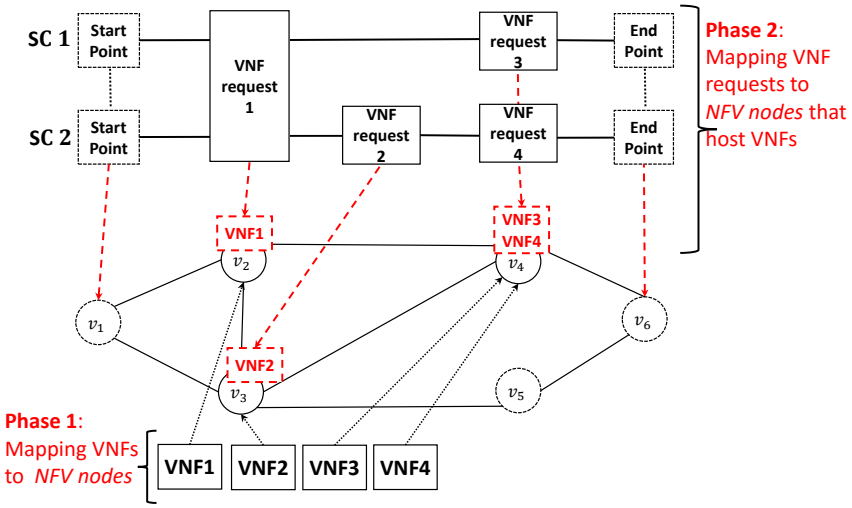


Figure 4.1 Two service chains, each having different VNFs, embedded in the physical network.

by a pair of start/end-points, a set of virtual nodes representing the VNFs and a set of virtual links chaining consecutive VNFs requests within the SC[1)]. In order to simplify the modeling, the concept of requests are decoupled from the VNFs that compose the Service chains. In other words, as shown in Fig. 4.1 (phase 1 and 2), a SC is considered as a chain of VNF requests. In order to deploy SCs in the network, VNF instances are mapped to *NFV-nodes* (phase 1) and successively, VNF requests are mapped to those *NFV-nodes* that hosts the requested VNFs (phase 2). The same apply for the mapping of end-points, which we assume have fixed location,known a priori, and that they cannot host VNFs. Furthermore, we assume the each SC serves aggregated traffic of a set of users requesting a specific service from a specific physical location.

1) We use the term *virtual node* to indicate the start/end point and the VNFs composing the SC and refer to to the segment used to chain two consecutive VNFs within the same SC as *virtual link*.

5
Resilient design protection schemes

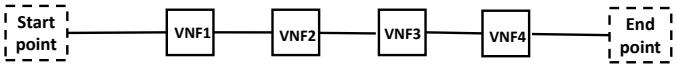
In this section we discuss the possible redundancy strategies for a resilient SC provisioning against single-node, single-link and single node/link failures.

5.1
On-Site Redundancy

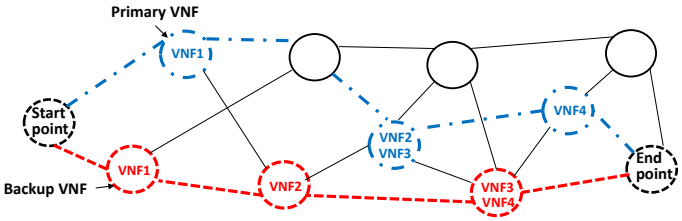
Critical VNFs supporting critical services and customers require fast switchover to backup VNFs in order to ensure availability. In order to ensure latency expectation, backup VNFs need to be instantiated on-site (i.e., Centralized Redundancy). Critical VNFs may necessitate a 1+1 level of redundancy while less critical function can tolerate a 1:1 redundancy. The main benefits from a centralized redundancy is to reduce switchover time, which allow to speed up the recovery process, and reduce the amount of VNF internal state information that need to be transfered from primary to backup VNFs. Note that this approach does not provide resiliency against node failures, since primary and backup VNFs share the same physical location.

5.2
Off-Site Redundancy

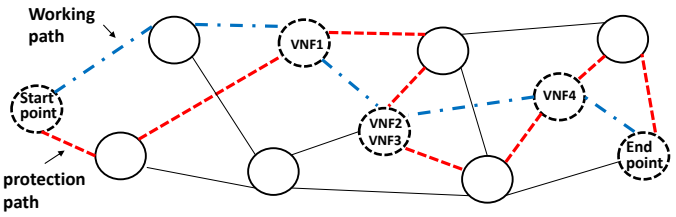
An off-site redundancy architecture involves having redundant VNFs placed in (hot or cold) standby mode in selected remote locations or NFVI nodes in the network operator's serving region. The intent is to instantiate them when there are failed VNFs in many NFVI-Points-of-Presence (NFVI-PoP). Moreover, this approach can guarantee resiliency against link and node failures since backup VNFs do not share the same physical locations as primary VNFs. Hence, based on the service criticalness and the resiliency guarantees targeted the operator can choose between an on-site or an off-site redundancy approach [5].



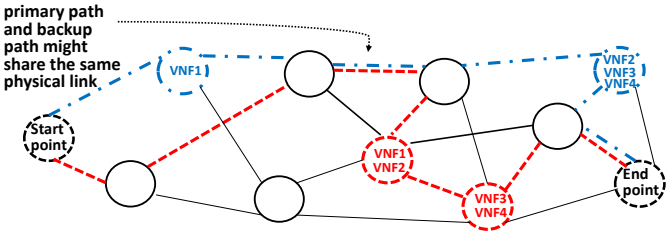
(a) Service chain to be embedded



(b) End-to-end protection



(c) Virtual-link protection



(d) Virtual-node protection

Figure 5.1 Proposed protection schemes.

In this work we propose three resiliency protection schemes. The first consists of an end-to-end protection of the entire SC. The idea behind such design is to have a SC that is resilient against single-link and single-node failures. To achieve such goal a primary SC is embedded in the physical network to support the related service in normal conditions and it is protected through a backup SC which has its VNFs embedded in different physical locations. The physical paths used to chain primary and backup VNFs must be node disjoint. Fig. 5.1(b) shows an example of such protection scheme, where a the SC illustrated in Fig. 5.1(a), composed fo four VNFs, is to embedded into the physical network. This protection scheme can be considered as an *Off-site redundancy* strategy since all backup VNFs are instantiated in different locations from where the primary ones are hosted. In this case, both redundancy strategies 1+1 and 1:1 are possible, depending on the service latency requirement and operators' design objective in terms of resource utilization. Note that both primary and backup physical paths resulting from the embedding must meet the latency requirement of the service. We refer to this protection strategy as *End-to-end protection (E2E-P)*.

The second protection scheme can be considered as an *redundancy* protection scheme, with the objective to protect the virtual links used to concatenate the VNFs of a certain SC. Hence, providing resiliency against physical link failures.

Each virtual link of the SCs is embedded through two physical paths, one primary path and one backup path, which must not share any physical link, while different primary/backup virtual links of the same SC can share common physical links. An example of such scenario is shown in Fig. 5.1(c). We refer to this protection scheme as *Virtual-link protection (Vl-P)*.

Finally, the last protection scheme provides resiliency against single-node failure. Each VNF composing the SC is instantiated in two disjoint physical locations, whereas the physical paths used to concatenate the primary and backup VNFs might share physical links. This protection scheme suits operators' need when failures occur in nodes with higher probability with respect to links. An example of this scenario is shown in Fig. 5.1(d). We refer to this scenario as *Virtual-node protection (Vn-P)*.

6
problem statement

In the following, we formally state the problem of resilient SCs provisioning and show the ILP models used to design each protection design scenario

6.1
Modeling the physical topology

We model the physical network as a directed graph $G = (V, E)$ where V represents the set of physical nodes $v \in V$, which can host VNFs or act as forwarding nodes, while E represents the set of physical links $(v, v') \in E$ which model high-capacity fiber links. Each physical link is associated with a latency contribution due to signal transmission and propagation, denoted with $\lambda(v, v')$ and a bandwidth capacity $\beta(v, v')$. The physical nodes equipped with COTS hardware are referred as *NFV-nodes* and can have different amount of processing capacity in terms of number of Virtual machine that they can host. Finally, we consider a processing-related latency $\omega(v) : v \in V$, introduced by *NFV-nodes*. This latency contribution is proportional to the number of SCs sharing the same VNF, hence, if a VNF is shared among a high number of SCs, the context switching latency would impact more the total latency.

6.2
VNF and service chains Modeling

Generally, a VNF is an abstracted object that performs operations on input traffic. Each VNF $f \in F$ has a processing capability which corresponds to the number of CPU Cores that are assigned to the VM that host the VNF f. We assume that a VNF shared among different SCs must run on a VM with enough capacity in terms of CPUs.

Moreover, we assume that each service corresponds to one SC modeled through a simple line graph $S^c = (E^c \cup U^c G^c)$ where E^c is the set of end-points of the SC, U^c is the set of VNF requests u, while G^c is the set of virtual links (u, u') chaining requests u and $u' \in U^c$. In order to simplify the modeling the concept of

requests are decoupled from the actual network functions that compose the Service chains. In other words VNFs are mapped to requests through a mapping parameter γ_u^c that specify the network function $f \in F$ requested by request $u \in U^c$, while requests are mapped to physical nodes through a decision variable. The same apply for the mapping of end-points, which we assume are fixed location and known a priori. Furthermore, we assume the each SC serve a set of users requesting a specific service from a specific physical location, and that each virtual link composing the SC is characterized by a bandwidth requirement $\gamma(u, u') : u, u' \in U^c, c \in C$. In addition, each SC is associated with a maximum tolerated latency, referred to as $\phi(c) : c \in C$.

Table 6.1 Parameters description for the ILP model

Parameter	Domain	Description
η_u^c	$c \in C, u \in U^c$	Physical start/end point where u is mapped for SC c
γ_u^c	$c \in C$ $u \in G^c$	Network function requests u for SC c, $\gamma_u^c \in F$
$\beta_{v,v'}$	$(v, v') \in E$	Bandwidth capacity of physical link (v, v')
$\lambda_{v,v'}$	$(v, v') \in E$	Latency of physical link (v, v')
$\omega_v \in E$	$v \in V$	contest switching latency of node v.
$\tau_u^c \in F$	$c \in C, u \in U^c$	VNF f requested by request u in the SC c
ϕ^c	$c \in C$	Maximum tolerated latency for SC c
$N_{req}(f)$	$f \in F$	Maximum number of requests of different SCs that VNF f can handle
$N_{VM}(v)$	$v \in V$	Maximum number of virtual machines that node v can host
M		Big-M parameter

6.3 ILP models

We now formulate the ILP models for resilient placement of VNFs. In Tab. 6.1 and Tab. 6.2 we summarize the parameters and the variables used. Given a physical topology, a set of SCs to be deployed in the network, we want to find the optimal placement of VNFs such that:

- The number of *VNF nodes* is minimized.
- Latency requirements of SCs are met.
- Resiliency is achieved according to the goals of the above mentioned scenarios (see Fig. 5.1 of section 5).

Table 6.2 Variables description for the ILP models

Variable	Domain	Description
$m^c_{u,v} \in \{0,1\}$	$c \in C, u \in U^c v \in V$	Binary variable equal to 1 iff the primary VNF request u of SC c is mapped to physical node v
$n^c_{u,v} \in \{0,1\}$	$c \in C, u \in U^c v \in V$	Binary variable equal to 1 iff the backup VNF request u of SC c is mapped to physical node v
$x^c_{v,v',x,y,u,u'} \in \{0,1\}$	$c \in C, (v,v') \in E, x \in V, y \in V, (u,u') \in G^c$	Binary variable equal to 1 iff the physical link (v,v') belongs to the path between x and y where primary VNFs requests u and u' for SC c are mapped, otherwise 0
$y^c_{v,v',x,y,u,u'} \in \{0,1\}$	$c \in C, (v,v') \in E, x \in V, y \in V, (u,u') \in G^c$	Binary variable equal to 1 iff the physical link (v,v') belongs to the path between x and y where backup VNFs requests u and u' for SC c are mapped, otherwise 0
$i_{f,v} \in \{0,1\}$	$f \in F$,$v \in V$	Binary variable equal to 1 iff VNF f is hosted by physical node v otherwise 0
$a_v \in \{0,1\}$	$v \in V$	Binary variable equal to 1 iff node v hosts at least one VNF.

6.3.1 Objective function

$$Minimize \sum_{v \in V} a_v \tag{6.1}$$

We consider three types of constraints to solve this problems, namely: Placement constraint, routing constraints and performance constraints. Due to space limitation we show only the constraints for the E2E-P protection scenario and give a brief description of what differs in the other two scenarios, Vl-P and Vn-P.

6.3.2 Placement constraints

Constraints (6.2a) and (6.2b) force each primary/backup VNF to be mapped to one single node. Equations 6.2c) and (6.2d state that a corresponding VNF f is mapped to physical node v only if there is a primary/backup VNF request. Constraint (6.2e) enforces that primary and backup VNF request u cannot be mapped to the same node (node disjointness).

$$\textstyle\sum_{v \in V} m^c_{u,v} = 1 \qquad \forall c \in C, u \in U^c \tag{6.2a}$$

$$\textstyle\sum_{v \in V} n^c_{u,v} = 1 \qquad \forall c \in C, u \in U^c \tag{6.2b}$$

$$i_{f,v} \leq \sum_{u \in U^c : \gamma^c_u = f} m^c_{u,v} + n^c_{u,v} \qquad \forall f \in F, v \in V \tag{6.2c}$$

$$\sum_{u \in U^c : \gamma^c_u = f} m^c_{u,v} + n^c_{u,v} \leq M \cdot i_{f,v} \qquad \forall f \in F, v \in V \tag{6.2d}$$

$$m^c_{u,v} + n^c_{u,v} \leq 1 \qquad \forall u \in U^c, c \in C, v \in V : v \neq \eta^c_u \tag{6.2e}$$

6.3.3 Routing constraints

Constraints (6.3a) [(6.3b)] ensure that a physical link (v, v') can belong to a path between two nodes x and y for a virtual link (u, u') of the SC c only if two consecutive primary [backup] VNF requests u and u' are mapped to these nodes, respectively. Note that equations (6.3a)-(6.4d) contain products of binary variables that we linearize in order to solve the ILP models.

$$w^c_{v,v',x,y,u,u'} \leq m^c_{u,x} \cdot m^c_{u',y} \qquad \forall c \in C, (v, v') \in E, x, y \in V, (u, u') \in G^c \tag{6.3a}$$

$$p^c_{v,v',x,y,u,u'} \leq n^c_{u,x} \cdot n^c_{u',y} \qquad \forall c \in C, (v, v') \in E, x, y \in V, (u, u') \in G^c \tag{6.3b}$$

$$\sum_{(x,v)\in E:x,y\in V} w^c_{x,v,x,y,u,u'} \cdot m^c_{u,x} \cdot m^c_{u',y} = 1 \quad \forall c \in C, (u,u') \in G^c \tag{6.4a}$$

$$\sum_{(v,y)\in E:x,y\in V} w^c_{v,y,x,y,u,u'} \cdot m^c_{u,x} \cdot m^c_{u',y} = 1 \quad \forall c \in C, (u,u') \in G^c \tag{6.4b}$$

$$\sum_{(x,v)\in E:x,y\in V} p^c_{x,v,x,y,u,u'} \cdot n^c_{u,x} \cdot n^c_{u',y} = 1 \quad \forall c \in C, (u,u') \in G^c \tag{6.4c}$$

$$\sum_{(v,y)\in E:x,y\in V} p^c_{v,y,x,y,u,u'} \cdot n^c_{u,x} \cdot n^c_{u',y} = 1 \quad \forall c \in C, (u,u') \in G^c \tag{6.4d}$$

During the mapping of primary/backup VNF requests on a physical path between x and y incoming links for the node x are not considered, constraint (6.5a), and no outgoing link for node y is considered (constraint (6.5b)

$$\sum_{(v,x)\in E:v\in V} w^c_{v,x,x,y,u,u'} = \sum_{(v,x)\in E:v\in V} p^c_{v,x,x,y,u,u'} = 0 \quad \forall c \in C, x \in V, y \in V : x \neq y, (u,u') \in G^c \tag{6.5a}$$

$$\sum_{(y,v)\in E:v\in V} w^c_{y,v,x,y,u,u'} = \sum_{(y,v)\in E:v\in V} p^c_{y,v,x,y,u,u'} = 0 \quad \forall c \in C, x \in V, y \in V : x \neq y, (u,u') \in G^c \tag{6.5b}$$

$$\sum_{(v,w)\in E:v\in V} w^c_{v,w,x,y,u,u'} = \sum_{(w,v')\in E:v\in V} w^c_{w,v',x,y,u,u'} \quad \forall c \in C, w \in V, x, y \in V : x \neq w, y \neq w, (u,u') \in G^c \tag{6.5c}$$

$$\sum_{(v,w)\in E:v\in V} p^c_{v,w,x,y,u,u'} = \sum_{(w,v')\in E:v\in V} p^c_{w,v',x,y,u,u'} \quad \forall c \in C, w \in V, x, y \in V : x \neq w, y \neq w, (u,u') \in G^c \tag{6.5d}$$

$$\sum_{(v,w)\in E:v\in V} w^c_{v,w,x,y,u,u'} \leq 1 \quad \forall c \in C, w \in V, x, y \in V : x \neq w, y \neq w, (u,u') \in G^c \tag{6.5e}$$

$$\sum_{(v,w)\in E:v\in V} p^c_{v,w,x,y,u,u'} \leq 1 \quad \forall c \in C, w \in V, x, y \in V : x \neq w, y \neq w, (u,u') \in G^c \tag{6.5f}$$

$$\sum_{(u,u')\in G^c} w^c_{v,v',x,y,u,u'} + p^c_{v,v',x,y,u,u'} \leq 1 \quad \forall c \in C, x, y, v, v' \in V : (v,v') \wedge (v',v) \in E \tag{6.5g}$$

Constraints (6.5c)-(6.5f) are transit constraints for primary/backup VNF requests. In particular, constraints (6.5c) and (6.5d) ensure that for any intermediate node ω within the physical path between x and y, if one of the incoming links belong to the primary/backup physical path, then also one of its outgoing links belong to the

physical path. While constraints (6.5e) [(6.5f)] avoid the use of multiple incoming [outgoing] links of the intermediate node. Finally, constraint (6.5g) ensures that a physical link (v, v') is whether part of the primary physical path or in the backup physical path used for the embedding of all VNF request of SC c.

Constraints (6.6a)-(6.6b) select the active *NFV-nodes*. A node is considered active if it hosts at least one single VNF. Constraint (6.6c) ensures that link capacity is not exceeded, whereas constraints (6.6d) and (6.6e) compute the context switching latency contribution σ_w^c and σ_p^c for primary and backup embedding of SC c, respectively. The maximum latency of primary/backup embedding of SC c are constrained in (6.6f)-(6.6g). Finally, the maximum number of CPU cores that *NFV-node* v can host is bounded by (6.6h), and the number of parallel requests that a given VNF can serve is constrained in (6.6i).

6.3.4 Latency and capacity constraints

$$\sum_{f \in F} i_{f,v} \leq M.a_v \qquad \forall v \in V \tag{6.6a}$$

$$a_v \leq \sum_{f \in F} i_{f,v} \qquad \forall v \in V \tag{6.6b}$$

$$\sum_{\substack{c \in C \\ (u,u') \in G^c \\ x,v \in V}} (w^c_{v,v',x,y,u,u'} + p^c_{v,v',x,y,u,u'}) \cdot \beta_{u,u'} \leq C_{v,v'} \qquad \forall (v, v') \in E \tag{6.6c}$$

$$\sigma_w^c = \sum_{v \in V, u \in U^c} m^c_{u,v} \cdot \omega_v \qquad \forall c \in C \tag{6.6d}$$

$$\sigma_p^c = \sum_{v \in V, u \in U^c} n^c_{u,v} \cdot \omega_v \qquad \forall c \in C \tag{6.6e}$$

$$\sum_{\substack{x,v \in V \\ (u,u') \in G^c \\ (v,v') \in E}} (w^c_{v,v',x,y,u,u'} \cdot \lambda_{v,v'}) + \sigma_w^c \leq \phi_c \qquad \forall c \in C \tag{6.6f}$$

$$\sum_{\substack{x,v \in V \\ (u,u') \in G^c \\ (v,v') \in E}} (p^c_{v,v',x,y,u,u'} \cdot \lambda_{v,v'}) + \sigma_p^c \leq \phi_c \qquad \forall c \in C \tag{6.6g}$$

$$\sum_{f \in F} i_{f,v} \leq N_{VM}(v) \qquad \forall v \in V \tag{6.6h}$$

$$\sum_{\substack{c \in C \\ u \in U^c : \gamma_u^c = f}} m^c_{u,v} + n^c_{u,v} \leq N_{req}(f) \qquad \forall v \in V, f \in F \tag{6.6i}$$

6.4 Additional modeling constraints

In the following we illustrate the constraints used to model the *Vl-P* and *Vn-P*.

Virtual-link Protection

With respect to the *E2E-P*, the *Vl-P* scenario ensures that the primary and backup physical path used to map a certain virtual link of a SC do not share any physical link and avoid closed loops. This is ensured using the constraints in eq (6.5g) and eq (6.7a)-(6.7b). See Tab. 6.3.

$$w^c_{v,v',x,y,u,u'} + w^c_{v',v,x,y,u,u'} \leq 1 \quad \forall c \in C(u,u') \in G^c x, y \in V : x \neq y, (v,v') \in E \tag{6.7a}$$

$$p^c_{v,v',x,y,u,u'} + p^c_{v',v,x,y,u,u'} \leq 1 \quad \forall c \in C(u,u') \in G^c x, y \in V : x \neq y, (v,v') \in E \tag{6.7b}$$

$$\sum_{\substack{x,y \\ (u,u')}} \sum_{(v,v')} (w^c_{v,v',x,y,u,u'} + p^c_{v,v',x,y,u,u'}) \cdot \lambda_{v,v'} + \sigma^c_p \leq \phi_c \quad \forall c \in C \tag{6.7c}$$

Regarding the placement of primary/backup VNFs, since they share the same physical location, in order to reduce the problem complexity, we use only one placement variable ($m^c_{u,v}$) to indicate the placement of both primary and backup VNFs. However, we assume that each of these VNFs is placed within a different physical machine. Regarding the physical paths, the latency constraint should be met independently from source to destination. An illustrative example is provided in Fig. 6.1. Consider the embedding of the SCs, shown in 6.1(a). We assume that the embedding process resulted in the *VNF1* placed in one node and *VNF2* and *VNF3* consolidated in the second node. According to the *Vl-P* every pair of nodes from start to end points are connected using a pair of disjoint paths (i.e., red and blue paths). The embedding of the virtual links can result in one single physical link carrying the primary and back up embedding of different virtual links. Hence, different physical paths can be used to transport the traffic from the start-point to the end-point. In Fig. **??** (case 1) the failure of a physical link causes the failure of the primary virtual link between the start-point and VNF1. The backup path (dashed lines) must meet the latency requirement. Similarly, in Fig. 6.1(a) and Fig. 6.1(b) we assume that the failure of one physical link causes the failure of the backup path of the first virtual link and the primary path of the second virtual link. In this case, two possible end-to-end paths are possible (dashed lines in case 2 and case 3) and both of theses options must satisfy latency requirement. Eq (6.7c) ensures that the latency requirements is met in all three cases. Please note that the paths between the starting point and a *VNF* node or between two consecutive *VNF* nodes are multi-hop paths. Intermediate nodes were omitted in the figure for the sake of simplicity.

Virtual-node Protection

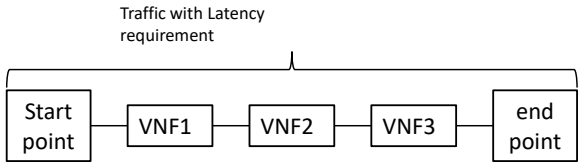
For the Vn-P scenario, only the node disjointness constraint apply and no disjointness constraints between primary/backup physical paths are needed since they can share physical links. In addition, eq (6.2c), eq (6.2d) and eq (6.6c) are substituted by the following constraints:

$$i_{f,v} \leq \sum_{u \in U^c : \gamma_u^c = f} m_{u,v}^c \qquad \forall f \in F, v \in V \tag{6.8a}$$

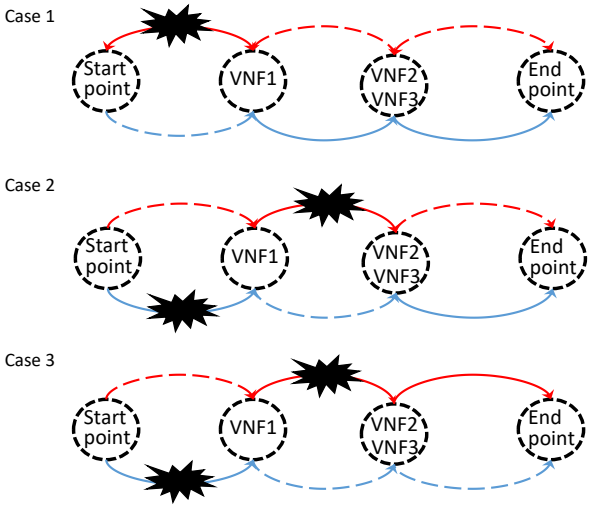
$$\sum_{u \in U^c : \gamma_u^c = f} m_{u,v}^c \leq M \cdot i_{f,v} \qquad \forall f \in F, v \in V \tag{6.8b}$$

$$\sum_{\substack{c \in C \\ (u,u') \in G^c \\ x,v \in V}} w_{v,v',x,y,u,u'}^c \cdot \beta_{u,u'} \leq C_{v,v'} \forall (v,v') \in E \tag{6.8c}$$

Please refer to Tab. 6.3 for a detailed description of the constraints used in each design scenario.



(a) Service chain with latency requirment



(b) Possible backup paths used in case of the failure of a physical link

Figure 6.1 Possible backup paths in the *VI-P* design scenario

Table 6.3 ILP formulations of the proposed protection scenarios

	Unprotected	*End-to-end protection*	*Vl-protection*	*Vn-protection*
Objective function	Minimize $\sum_{v \in V} a_v$			
Constraints	(6.2a) (6.3a) (6.4b) (6.4c) (6.5a)-(6.5c) (6.5e) (6.6a) (6.6b) (6.6f) (6.8a)-(6.8c)	(6.2a)-(6.2e) (6.3a)-(6.3b) (6.4a)-(6.4d) (6.5a)-(6.5g) (6.6a)-(6.6i)	(6.2a) (6.3a)-(6.3b) (6.4a)-(6.4d) (6.5a)-(6.5g) (6.6f)-(6.6i) (6.7a)-(6.7c)	(6.2a)-(6.2e) (6.3a)-(6.3b) (6.4a)-(6.4d) (6.5a)-(6.5f) (6.6a)-(6.6i)

6.5 Problem complexity

In this section we compute the total number of variables and constraints of the each design scenario. The number of variables of the *E2E* and the *Vn-P* is the same, differs slightly in case of the *Vl-P* and *unprotected* as no backup of nodes or nodes/links is required. The number of constraints is slightly different in each scenario. However, this difference does not affect the overall complexity, which is the same for all designs, and is in the order of $O(|G^c| \cdot |E'| \cdot |C| \cdot |V|^2)$.[1)] Below, we compute the number of variables and constraints, for the proposed design scenarios, based on the values of the unprotected scenario

$$N_{vars\_Unpro} = |V| \cdot (|C|\dot{|}U^c| + |C||E'||V||G^c| + |F| + 1) \tag{6.9a}$$

$$N_{const\_Unpro} = |C| \cdot (|U^c| + |E'||V|^2|G^c| + 2|G^c| + 3 \cdot |V|^2|G^c| + 1) + 2|V| \cdot (2|F| + 1) + |E'| \tag{6.9b}$$

$$N_{vars\_E2E} = N_{vars\_Unpro} + \alpha_{e2e} \tag{6.9c}$$

$$N_{const\_E2E} = N_{const\_Unpro} + \beta_{e2e} \tag{6.9d}$$

$$N_{vars\_vl-P} = N_{vars\_Unpro} + \alpha_{vl-p} \tag{6.9e}$$

$$N_{const\_vl-P} = N_{const\_Unpro} + \beta_{vl-p} \tag{6.9f}$$

$$N_{vars\_vn-P} = N_{vars\_Unpro} + \alpha_{vn-p} \tag{6.9g}$$

$$N_{const\_vn-P} = N_{const\_Unpro} + \beta_{vn-p} \tag{6.9h}$$

1) Please note that, for resolution purposes and in order to allow multiple VNFs of the same SCs to be hosted in the same *NFV* node, we assumed that each of these nodes has a self-loop link with infinite bandwidth. such self-loops links were included in the complexity computation by considering that $E' = E \cup N$, as the number of self-loop links is equal to the number of physical nodes.

where:

$$\alpha_{e2e} = |V| \cdot (|C|\dot{|}U^c| + |C||E'||V||G^c|) \tag{6.10a}$$

$$\beta_{e2e} = |C| \cdot (|U^c| + |E'||V|^2|G^c| + 2|G^c| + 3) \tag{6.10b}$$
$$+ |V| \cdot (|U^c + |V|^2|G|^2 + |V||E'| + |F| + 1)$$

$$\alpha_{vl-p} = \alpha_{e2e} - |C||E'||V|^2|G^c| \tag{6.10c}$$

$$\beta_{vl-p} = \beta_{e2e} + (|V|^2 \cdot (|C||G^c| + 1)) \tag{6.10d}$$

$$\alpha_{vn-p} = \alpha_{e2e} \tag{6.10e}$$

$$\beta_{vn-p} = \beta_{e2e} + 2|V|^2|G|^2 \tag{6.10f}$$

7
Case study and results

In this section we present and discuss the results of the ILP models shown in section 6. To solve the ILP problems we used CPLEX 12.6.1.0 installed on hardware platform equipped with 8×2 GHz processor and 8 GB of RAM. In order to evaluate the impact of latency requirements on the protection scenarios we investigated the embedding of four types of services chains, with different processing requirements and latency constraints, namely: *Web Service* (WS), *Video-Streaming* (VS), *VoIP* and *Online-Gaming* (OG). The maximum end-to-end tolerated latency for these services has been set to 500 ms for Web-service, 100 ms for both Video Streaming and VoIP and 60 ms for Online-Gaming on the line of [10]. Tab. 7.1 shows the VNFs composing the SCs, their bandwidth requirements and maximum allowed latency.

Table 7.1 Performance Requirements for the Service Chains

Service Chain	Chained VNFs	β	ϕ_c
Web-Service	NAT-FW-TM-WOC-IDPS	100 kbit/s	500 ms
Video Streamnig	NAT-FW-TM-VOC-IDPS	4Mbit/s	100ms
VoIP	NAT-FW-TM-FW-NAT	64kbit/s	100ms
Online-Gaming	NAT-FW-VOC-WOC-IDPS	50 kbit/s	60 ms

NAT: *Network Address Translator*, FW: *Firewall*, TM:*Traffic Monitor*, WOC: *WAN Optimization Controller*, IDPS: *Intrusion Detection Prevention System*, VOC: *Video Optimization Controller*

We consider heterogeneous and homogeneous traffic scenarios. In the heterogeneous scenario, 5 different SCs requests randomly selected from the SCs in Tab. 7.1, are considered. The type of SCs in this case is randomly selected at each ILP run. In the homogeneous scenario, 5 SCs requests of the same type are considered. The start/end-points, for both traffic scenarios, are randomly selected for each SC requests, at each ILP run. Moreover, we assume that all physical nodes can act as *NFV-nodes* and that the start/end points of SCs requests cannot host VNFs.

As for the physical topology, we considered the NSFNET network with 14 nodes

and 22 bidirectional links. Each *NFV node* is assumed to have the same capacity in terms of CPU cores. We set the context switching delay to 4 ms per VNF [10][22] [1)] and set the link capacity to be equal to 2.5 Gbps. Since the bandwidth requirements of the SCs are lower than the available bandwidth on the links, to constrain the bandwidth resources we assume that each SC aggregates the traffic of 2000 users. We also set the maximum number of parallel requests that a VNF can serve equal to 1, and assume that the bandwidth requirement of virtual links chaining VNFs varies according to a compression factor [2)] that ranges between 0.5 and 1 [12]. Such value is randomly selected at each ILP run. The results, shown in Fig. 7.1, were obtained averaging the results of 10 instances, solved within 5% of the optimal solution, for each value of *NFV-node*'s capacity, each protection scenario and considering different start/end points pairs, at each ILP run.

Figures from Fig. 7.1(a) to Fig. 7.1(e) show the average number of active *NFV-nodes* needed to support of the proposed protection scenarios for different values of node capacity (number of CPU cores it can host), for the Web-service, VoIP, Video Streaming, Online-Gaming and heterogeneous traffic scenarios, respectively. In the following we analyze the effect of latency and node capacity for the different traffic scenarios.

Impact of latency

Fig. 7.1(a) presents the number of active nodes for the less stringent SC in terms of latency (WS). We observe that all protection scenarios are possible and that the Vl-P scenario activates the same amount of Unprotected Scenario. We note that a SC with low requirements on latency can be protected against single-link failures (Vl-P) with no additional *NFV-nodes* with respect to the *Unprotected* case (baseline). On the other hand, providing protection against both single-link and single failure (E2E-P) requires the activation of around twice the amount of *NFV-nodes* when node capacity is greater than 6 CPU cores per *NFV-node*. Finally, increasing the capacity by a factor of five reduces the number of active *NFV-nodes* by 33% in case of off-site redundancy protection (E2E-P, Vn-P) and 80% in case of on-site redundancy protection (Vl-P). We also observe that the amount of resources required to supply end-to-end protection (E2E-P) is almost the same with respect to protection against single-node failures (Vn-P), independently from capacity values, meaning that in case the operator choses to place backup VNFs off-site, the protection against both link and node failures comes at the same cost, in terms of *NFV-nodes*, with respect to protection against node failures.

Fig. 7.1(b) and Fig. 7.1(c) show the results obtained by solving the VNF placement of VoIP and Video Streaming (VS) SCs, which have an average latency requirement. For both SCs, we observe that all scenarios are possible except for the Vl-P scenario,

1) Note that the provisioning of SCs introduces other latency contributes due to the upscaling of the capacity of VNFs and hypervisor processing of the VNF requests [10].

2) We used random compression factors given that no reference is available for such values.

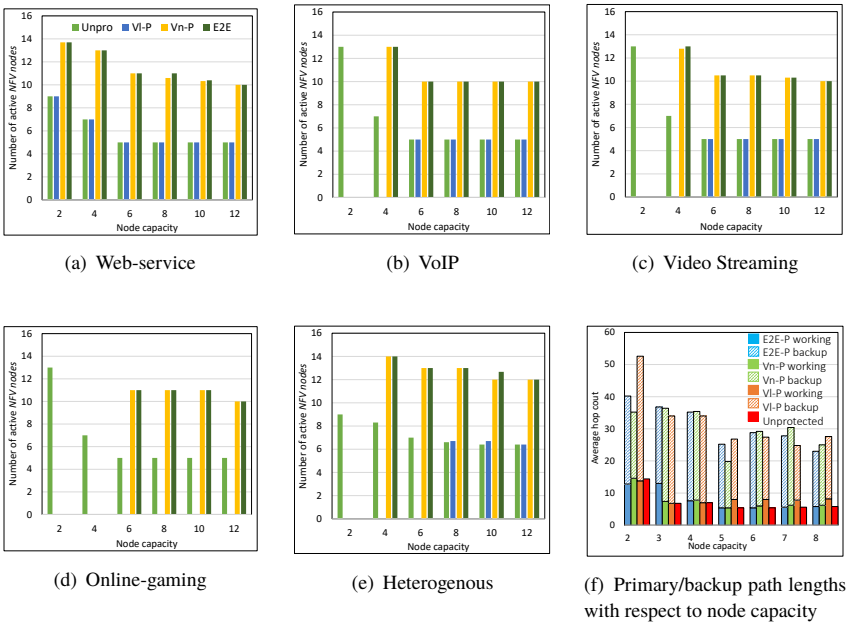
which leads to infeasible solution for node capacity values less than 6 CPU cores per *NFV-node*. This is mainly due to the fact that Vl-P has a stringent link disjointness constraint that in case of VNFs distribution among high number of nodes to increase the latency of physical paths needed to chain the VNFs and consequently violate the latency constraint.

Fig. 7.1(d) shows the results obtained when the SCs with the most stringent latency requirement (OG) are embedded into the network. We observe that, for node capacity values greater than 6 CPU cores, all scenarios are possible except for the Vl-P scenario which is infeasible independently from node capacity. For capacity values less than 6 CPU cores, we observe that all protection strategies lead to an infeasible solution. This means that, for latency stringent SCs, to provide protection against node/link failures, each *NFV-node* must be equipped with at least a minimum number of CPU cores. Moreover, for E2E-P and Vn-P, doubling the capacity lead to a tiny decrease of active *NFV-nodes* (around 10%), mainly due to the fact that increasing consolidation causes the context switching latency to increase and leads to violation of the latency constraint. Finally, since the only feasible protection sceanrios are E2E-P and Vn-P, the operator is constrained to place backup VNFs “Off-site" to provide resiliency against only single-link failures, when only latency critical SCs are deployed (in our case OG).

Finally, for the heterogeneous traffic scenario, shown in Fig. 7.1(e), all protection scenarios are possible starting from 8 CPU cores per *NFV-node* and lead to infeasible solution at 2 CPU cores per *NFV-node*. With respect to the OG case, we observe that when SCs with different requirements are deployed, protection against single-link failures on-site can be provided starting from 8 CPU cores per *NFV-node*. In general we observe that the heterogeneous traffic scenario requires more resources with respect to the homogeneous scenarios, but meets latency requirements while allowing a better VNF consolidation. This means that deploying SCs with different latency requirements and can guarantee resiliency with a small number of CPU cores per *NFV-node*, and consequently less failure impact within *NFV-nodes*.

Effect of node capacity

In terms of capacity, we observe that for WS (Fig. 7.1(a)) and heterogeneous deployment of SCs (Fig. 7.1(e)) increasing node capacity, the off-site redundancy protection strategies decrease the number of active *NFV-nodes* from 69% up to 96% for WS and from 52% up to 120% for heterogeneous deployment, with respect to the unprotected scenario. Wherease, for the on-site redundancy protection scenario (Vl-P), we observe that increasing the capacity more than 6 and 8 CPU cores, for WS and heterogeneous traffic scenarios, respectively, does not bring any benefit in terms of consolidation. This is mainly due to the fact that consolidation of VNFs is limited by the context switching latency. The same claim is valid for the VoIP and VS SCs (Fig. 7.1(b), 7.1(c)), where we observe that increasing node capacity more than 6 CPU cores per node does not effect the amount of active *NFV-nodes*, where it leads to a feasible solution. Finally, comparing the outcome obtained for



(a) Web-service (b) VoIP (c) Video Streaming

(d) Online-gaming (e) Heterogenous (f) Primary/backup path lengths with respect to node capacity

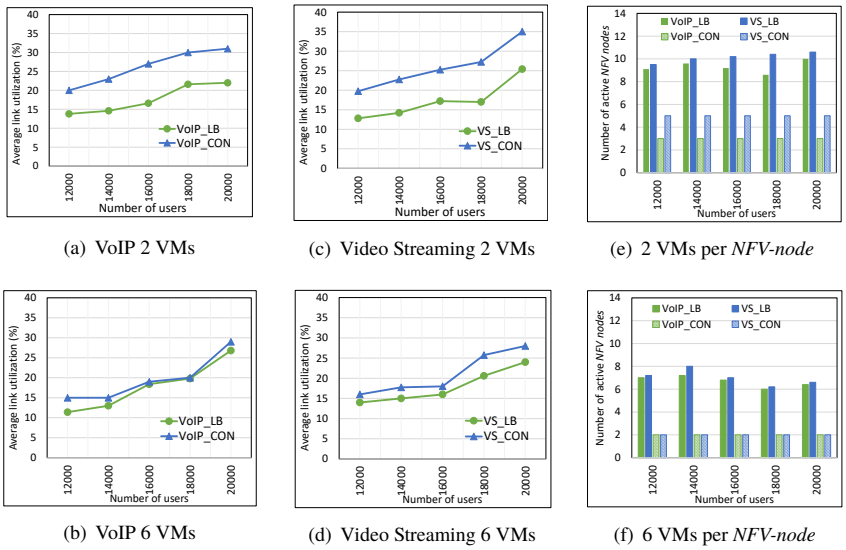
Figure 7.1 Comparison of the proposed protection scenarios for different latency requirements

VoIP and VS SCs, shown in Fig. 7.1(b) and Fig. 7.1(c) we observe that the impact of the different bandwidth requirements of both SCs is slightly noticeable (around 2%). Later in this section we relax the constraint on the number of parallel requests that a VNF instance can serve and evaluate the impat of bandwidth requirement on both SCs, under different values of node capacity and different optimization targets. In general, for both traffic scenarios, we observe that VNF consolidation is limited by latency, as consolidating more VNFs into less nodes would increase the impact of context switching latency.

Impact of node capacity on the average hop count

We analyzed the impact of node capacity on the average length of primary/backup physical paths of all proposed protection strategies. In Fig.7.1(f) we show the primary/backup paths lengths when 2 Web Service (WS) SCs are deployed. These results were obtained by averaging the paths lengths of 5 start/end point pairs randomly selected and tested for all protection scenarios. We observe that at the increasing of node capacity the length of the primary path does not change significantly, for all protection strategies. For backup paths, we observe that increasing node capacity

does not mean reducing backup paths lengths. This is shown by the fact that allowing more than 5 CPU cores per *NFV-node* does not reduce the average backup path length, meaning that a trade-off between consolidation of VNFs and the average path length exist.



(a) VoIP 2 VMs (c) Video Streaming 2 VMs (e) 2 VMs per *NFV-node*

(b) VoIP 6 VMs (d) Video Streaming 6 VMs (f) 6 VMs per *NFV-node*

Figure 7.2 result comparison of two objective functions when deploying 2 SCs with same latency requirement and different bandwidth requirements

Impact of different optimization targets

We consider the VNF placement of two different SCs, with the same latency requirements but different bandwidth requirements (VoIP and Video Streaming), when the optimization target is to consolidate VNFs (CON) and when optimizing with the objective of balancing the load on physical links (LB). We run the ILP model for E2E-P for different number of users and different values of node capacity. The objective here is to analyze the effect of increasing number of users (we consider that the 2 SCs aggregate the traffic all the users at each ILP run) on the average link occupation and on the number or active *NFV-nodes* when different objective functions are targeted. In addition, for this set of experiments, we assume that the data-rate between different VNFs of the same SC is fixed, and relax the constraint in eq (6.6i), which limits the maximum number of parallel requests that a VNF instance can serve.

To solve the ILP model with load balancing objective, we use the same formulation in 6 for the E2E-P, define an integer variable $\mu \in [0, 1]$ to account for the maximum load of any edge and substitute the objective function (eq 6.1) and the link capacity

constraint (eq 6.6c) with the objective function and link capacity constraints in eq 7.1a and eq 7.1b, respectively:

$$Minimize \quad \mu \tag{7.1a}$$

$$\sum_{\substack{c\in C \\ (u,u')\in G^c \\ x,v\in V}} (w^c_{v,v',x,y,u,u'} + p^c_{v,v',x,y,u,u'}) \cdot \beta_{u,u'} \leq C_{v,v'} \cdot \mu \quad \forall (v,v') \in E \tag{7.1b}$$

In Fig. 7.2(a) and Fig. 7.2(c) we show the average link occupation for the VoIP and VS SCs and compare the average number of active *NFV-nodes* for both optimization targets in Fig. 7.2(e), when each *NFV-node* is equipped with 2 CPU cores. In case of VoIP, LB shows a decrease in average link utilization from 28% up to 38% at the expense of triplicating the number of *NFV-nodes* with respect to results obtained in CON. Whereas, in case of VS, LB decreases the load on physical link from 27% up to 37% while doubling the number *NFV-nodes* with respect to CON. We also observe that the number of *NFV-nodes* activated for both VoIP and VS under CON does not change, independently from the number of users. Generally, we observe that the VoIP activates 60% less *NFV-nodes* than VS. Instead when targeting LB, VoIP activates up to 10% *NFV-nodes*, which is due to the different bandwidth requirements of VoIP and VS.

In the second set of experiments we increase the node capacity by factor of three and show the average link occupation for VoIP and VS in Fig. 7.2(b), and 7.2(d), respectively. We observe that when deploying VoIP, both objective functions lead to the same average link utilization, at high number of users, as shown in Fig. 7.2(f). Whereas, in case of VS, the gain obtained from LB ranges between 12% and 20%, as shown in Fig. 7.2(b). In terms of *NFV-nodes*, the increase of node capacity translates into better consolidation. Both SCs, reduce the amount of active *NFV-nodes* under CON by a more than 3 times and activate the minimum amount of *NFV-nodes* needed to support E2E-P. This is mainly due to the fact that whenever a VNF instance is activated is can be used by all SCs requesting it. This causes the paths used to concatenate the VNFs to be longer, which explains also the reason why both SCs achieve low gain from LB when node capacity is high.

We generally, observe that LB is beneficial for low values of node capacity, while CON brings more benefit when *NFV-nodes* are equipped with a higher number of CPU cores.

8 Conclusion

In this work we proposed three different protection strategies to provide resilient SCs deployment against single-node, single-link, single-node/link failures. We reported the formulation for all the design scenarios, solved the ILP models considering a small number of SCs with different latency requirements, and found that a trade-off between node capacity and latency of the deployed SCs. Moreover, We analyzed the effect of *NFV-nodes* capacity on the average primary/backup paths lengths. Finally, we solved one of the proposed ILP models considering two different SCs with equal latency constraints and different bandwidth requirements, under two conflicting objectives to analyze the effects of bandwidth requirements on the distribution of VNFs. In our small-scale scenario, we found that:

- In order to provide resiliency to SCs against single-link and single-node failures twice the number of *NFV-nodes* are needed with respect to the unprotected scenarios and the case where only single-link failures are targeted.
- Increasing node capacity does not cause the reduction of the average path lengths.
- Bandwidth intensive SCs benefit more from consolidation when the node capacity is high, while load balancing is beneficial at small values of node capacity.

Future steps of this work aim at solving the problem of SCs provisioning under dynamic conditions, while targeting the optimization of different cost functions. We also aim at extending the proposed models with a shared protection scheme.

Acknowledgment

This article is based upon work from COST Action CA15127 ("Resilient communication services protecting end-user applications from disaster-based failures - RECODIS") supported by COST (European Cooperation in Science and Technology).

References

1 Sherry, J., Hasan, S., Scott, C., Krishnamurthy, A., Ratnasamy, S., and Sekar, V. (2012) Making middleboxes someone else's problem: network processing as a cloud service. *ACM SIGCOMM Computer Communication Review*, **42** (4), 13–24.

2 Mijumbi, R., Serrat, J., Gorricho, J.L., Bouten, N., Turck, F.D., and Boutaba, R. (2016) Network function virtualization: State-of-the-art and research challenges. *IEEE Communications Surveys Tutorials*, **18** (1), 236–262, doi:10.1109/COMST.2015.2477041.

3 Halpern, J. and Pignataro, C. (2015) Service Function Chaining (SFC) Architecture, *Tech. Rep.*.

4 Network Functions Virtualisation (2014) Draft ETSI GS NFV-SEC 001 v0.2.1 (2014-06).

5 ETSI (2015) GS NFV-REL 001 v1. 1.1: Network functions virtualisation(nfv); resiliency requirements, *Tech. Rep.*, ETSI industry Specfication Group (ISG) Network Functions Virtualisation (NFV).

6 Fischer, A., Botero, J.F., Till Beck, M., De Meer, H., and Hesselbach, X. (2013) Virtual Network Embedding (VNE): A survey. *Communications Surveys & Tutorials, IEEE*, **15** (4), 1888–1906.

7 Rahman, M.R. and Boutaba, R. (2013) SVNE: Survivable virtual network embedding algorithms for network virtualization. *IEEE Transactions on Network and Service Management*, **10** (2), 105–118, doi:10.1109/TNSM.2013.013013.110202.

8 Prodhon, C. and Prins, C. (2014) A survey of recent research on location-routing problems. *European Journal of Operational Research*, **238** (1), 1–17.

9 Mehraghdam, S., Keller, M., and Karl, H. (2014) Specifying and placing chains of virtual network functions, in *IEEE 3rd International Conference on Cloud Networking (CloudNet)*, IEEE, pp. 7–13.

10 Savi, M., Tornatore, M., and Verticale, G. (2015) Impact of processing costs on service chain placement in network functions virtualization, in *IEEE Conference on Network Function Virtualization and Software Defined Network (NFV-SDN)*, pp. 191–197, doi:10.1109/NFV-SDN.2015.7387426.

11 Jiang, J.W., Lan, T., Ha, S., Chen, M., and Chiang, M. (2012) Joint vm placement and routing for data center traffic engineering, in *IEEE Conference on Information Communication (INFOCOM)*, IEEE, pp. 2876–2880.

12 Addis, B., Belabed, D., Bouet, M., and Secci, S. (2015) Virtual network functions placement and routing optimization, in *2015 IEEE 4th International Conference on Cloud Networking (CloudNet)*, pp. 171–177, doi:10.1109/CloudNet.2015.7335301.

13 Moens, H. and De Turck, F. (2014) VNF-P: A model for efficient placement of virtualized network functions, in *10th International Conference on Network and Service Management (CNSM)*, IEEE, pp. 418–423.

14 Lin, T., Zhou, Z., Tornatore, M., and Mukherjee, B. (2016) Demand-aware network function placement. *Journal of Lightwave Technology*, **34** (11), 2590–2600,

doi:10.1109/JLT.2016.2535401.

15 Cotroneo, D., Simone, L.D., Iannillo, A.K., Lanzaro, A., Natella, R., Fan, J., and Ping, W. (2014) Network function virtualization: Challenges and directions for reliability assurance, in *2014 IEEE International Symposium on Software Reliability Engineering Workshops*, pp. 37–42, doi:10.1109/ISSREW.2014.48.

16 Ye, Z., Cao, X., Wang, J., Yu, H., and Qiao, C. (2016) Joint topology design and mapping of service function chains for efficient, scalable, and reliable network functions virtualization. *IEEE Network*, **30** (3), 81–87, doi:10.1109/MNET.2016.7474348.

17 Liu, J., Jiang, Z., Kato, N., Akashi, O., and Takahara, A. (2016) Reliability evaluation for nfv deployment of future mobile broadband networks. *IEEE Wireless Communications*, **23** (3), 90–96, doi:10.1109/MWC.2016.7498079.

18 Kim, T., Koo, T., and Paik, E. (2015) Sdn and nfv benchmarking for performance and reliability, in *2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, pp. 600–603, doi:10.1109/APNOMS.2015.7275403.

19 Qu, L., Assi, C., Shaban, K., and Khabbaz, M. (2016) Reliability-aware service provisioning in nfv-enabled enterprise datacenter networks, in *2016 12th International Conference on Network and Service Management (CNSM)*, pp. 153–159, doi:10.1109/CNSM.2016.7818411.

20 Machida, F., Kawato, M., and Maeno, Y. (2010) Redundant virtual machine placement for fault-tolerant consolidated server clusters, in *IEEE Network Operations and Management Symposium (NOMS)*, pp. 32–39, doi:10.1109/NOMS.2010.5488431.

21 Scholler, M., Stiemerling, M., Ripke, A., and Bless, R. (2013) Resilient deployment of virtual network functions, in *the 5th International Congress on Ultra Modern Telecommunications and Control Systems and Workshops (ICUMT)*, IEEE, pp. 208–214.

22 Li, C., Ding, C., and Shen, K. (2007) Quantifying the cost of context switch, in *Proceedings of the workshop on Experimental computer science (WECS)*, ACM, p. 2.