

Availability Evaluation of Multi-tenant Service Function Chaining Infrastructures by Multidimensional Universal Generating Function

Mario Di Mauro, *Member, IEEE*, Maurizio Longo, *Member, IEEE*, Fabio Postiglione

Abstract—The Network Function Virtualization (NFV) paradigm has been devised as an enabler of next generation network infrastructures by speeding up the provisioning and the composition of novel network services. The latter are implemented via a chain of virtualized network functions, a process known as Service Function Chaining. In this paper, we evaluate the availability of multi-tenant SFC infrastructures, where every network function is modeled as a multi-state system and is shared among different and independent tenants. To this aim, we propose a Universal Generating Function (UGF) approach, suitably extended to handle performance vectors, that we call Multidimensional UGF. This novel methodology is validated in a realistic multi-tenant telecommunication network scenario, where the service chain is composed by the network elements of an IP Multimedia Subsystem implemented via NFV. A steady-state availability evaluation of such an exemplary system is presented and a redundancy optimization problem is solved, so providing the SFC infrastructure which minimizes deployment cost while respecting a given availability requirement.

Index Terms—Service Function Chaining, Network Function Virtualization, Availability Analysis, Universal Generating Function, Redundancy Optimization, Multi-State Systems.



1 INTRODUCTION

IN the era of fifth generation (5G) telecommunication systems, the design, management and deployment of complex architectures have dramatically boosted, due to the increasing demand of network resources by more and more connected devices as smartphones, laptops, tablets, sensor networks and other kinds of smart objects. Facing these issues, several telecom operators have established an industry specifications group providing guidelines for Network Functions Virtualization (NFV) [1]. NFV is the network concept aiming to virtualize the whole class of network node functions (routers, firewalls, load balancers and others) into building blocks that may be interconnected to create communication services. The resulting architecture includes a set of Virtualized Network Functions (VNFs), conveniently arranged to create innovative network services or to define new *as-a-service* models such as VNFaaS [2], often in conjunction with Software Defined Networking (SDN) [3] aimed at controlling the composition logic [4] and governing some security mechanisms [5]. New services can be designed by means of the so-called Service Function Chaining (SFC) process, which consists in selecting specific VNFs to be connected and traversed in a predefined order [6], [7].

In this paper, we present an availability analysis of an SFC infrastructure guaranteeing the so-called “five nines” availability requirement (no more than 5 minutes and 26 seconds system downtime per year) as invoked in typical Service Level Agreements (SLAs). In particular, we focus on

a multi-tenant SFC architecture, where several operators (*aka* tenants) share the existing VNFs to provide specific services. Such VNFs are prone to failure (and consequent repair) activities, that could alter in many ways the overall SFC functioning. For instance, a VNF under repair and, hence, temporarily out-of-service, might have a huge impact on one tenant but minimal influence on another one.

Our analysis exploits the remarkable properties of the Universal Generating Function (UGF), a formalism originally introduced in [8]. In particular, we propose an extended version that we call Multidimensional UGF (MUGF), useful to handle multidimensional performance figures.

The paper is organized as follows. In Section 2 we advance a general perspective of the considered problem. Section 3 presents a review of the most significant related work. Some details about the SFC paradigm, along with a brief description of NFV architecture, are given in Section 4. In Section 5 we outline a multi-state performance model of an SFC architecture, which accounts for failures and repair actions. In Section 6 we study the steady-state availability of the system by introducing the novel MUGF approach. In Section 7 we develop the solution of a redundancy optimization problem for an exemplary SFC infrastructure representing a multi-tenant 5G telecommunication system. Finally, Section 8 draws the main conclusions and indicates further research prospects.

2 MOTIVATIONS AND PROBLEM STATEMENT

Network operators are moving towards NFV-based infrastructures to efficiently reduce deployment efforts and to expedite the provisioning of new services. Perfectly inserted in an NFV ecosystem, SFC refers to a technique for selecting

• M. Di Mauro, M. Longo, F. Postiglione are with the Department of Information and Electrical Engineering and Applied Mathematics (DIEM), University of Salerno, 84084, Fisciano, Italy.
E-mail: {mdimauro,longo,fpostiglione}@unisa.it

“network elements” to be traversed in a predefined order to provide a specific service. Consequently, IP packets of a data flow are processed in a sequential manner by a series of network service functions (e.g., network address translator, load balancer, firewall, deep packet inspector) that are implemented as VNFs.

An exemplary use case is offered in the present work, where the SFC is composed by virtualized nodes of an IP Multimedia Subsystem (IMS), a key infrastructure deployed in the core networks of next generation telecommunication systems. In this regard, we want to highlight that, being mainly focused on an availability problem, we consider, for the sake of simplicity, a high-level perspective of the IMS service chain, as often contemplated in the technical literature on SFC infrastructures (e.g., [9], [10]).

One of the most valuable advantages provided by SFCs in terms of cost reduction concerns the possibility to share network functionalities among different operators. Such is the case of a *multi-tenant* SFC infrastructure, where VNF resources are (not necessarily equally) allocated for each operator, and hence posing a resource sharing problem [11]. This problem also arises in technological scenarios, as described in guidelines proposed by NEC Corporation [12], where Long Term Evolution (LTE) telecommunication nodes are shared (once virtualized) among different mobile providers.

In real operations, VNFs are affected by hardware and software faults that reduce resources allocated to one or more tenants, and, typically, some repair actions are executed. Consequently, performance levels exhibited by a multi-tenant SFC vary along time and can differ from one operator to another.

For each single tenant, an SFC is considered available when it guarantees a given performance level to that tenant, a condition depending on usable resources. In the presence of faults, it is crucial to conceive some redundancy methods, that, obviously, entail a trade-off between costs and availability targets.

The availability analysis of a system characterized by different performance levels can benefit of a Multi-State System (MSS) representation [13]. In particular, the availability of MSS complex systems, composed in turn by different MSS subsystems, can be faced by the UGF approach. Such an approach allows to characterize the overall system performance distribution, and thus its availability, by composing the performance distributions of its subsystems via some appropriate operators, in a computationally efficient way [14]. However, it can only be used to assess one-dimensional performance characteristics. On the other hand, in a multi-tenant SFC scenario, the performance levels of the operators vary from one operator to the other, thus, multivariate performance characteristics must be considered to evaluate the entire system availability.

In the present work, we address all of the aforementioned issues by offering three original contributions: *i*) we model a multi-tenant SFC infrastructure as an MSS by conveniently combining some composition operators; *ii*) we propose an extended version of the UGF technique, referred to as Multidimensional UGF (MUGF), dealing with performance vectors, and then applicable to complex scenarios such as those represented by multi-tenant network

architectures; *iii*) we perform an availability analysis and solve a redundancy optimization problem in a realistic scenario of a virtualized 5G telecommunication infrastructure, as a profitable example of a multi-tenant SFC. Finally, a sensitivity analysis is carried out to assess the robustness of the considered system with respect to variation of critical parameters value.

3 RELATED WORK

In the last years, the scientific community has devoted an increasing interest to the issue of availability assessment of novel cloud-oriented architectures [15]. In this section, without pretence to be exhaustive, we present a review of recent papers that have addressed problems affine to ours. In many cases, the availability problems in cloud infrastructures are solved by proposing algorithms based on optimal allocations of virtual backup resources in order to prevent possible faults of main elements, but without considering (or partially considering) a failure/repair model. The authors in [16], for example, analyze the availability problem (with regards to the minimum number of off-site backup VNFs to be deployed) of an SFC whose model includes only failures actions but not repair activities. Similarly, in [17] the problem of distributing VNF replicas between the primary and backup paths to maximize the SFC’s availability has been tackled through a heuristic algorithm where a failure/repair model is not addressed. The work in [18], although not considering the chaining structure of VNFs, examines the problem of providing service availability with bandwidth guarantees through the deployment of redundant virtual machines (VMs) in a multi-tenant environment. The focus there is on the design of a protection plan where each backup VM should protect one or more primary VMs in case of failure, and no VM failure/repair model is included, being the design of the optimal backup infrastructure out of the scope.

Even when a failure/repair model is present, the availability evaluation typically encompasses non-MSS models as in [19], where an availability analysis of container-based architectures is carried on by applying some non-state-space and state-space models. Similarly, the authors of [20] perform an availability analysis of cloud infrastructures by exploiting the Stochastic Petri Net framework aimed at finding optimal redundancy, considering non-MSS models of failure/repair activities.

Approaches exploiting the UGF function (to handle multi-state system representations) are adopted in [21] and [22], where Markov and semi-Markov models are used, respectively, to characterize a single tenant IMS architecture. Another UGF-based method is employed to model physical and virtual machine failures in [23], where a single tenant cloud-based environment is implicitly assumed.

Consequently, trying to fill in the gap in existing literature, in this paper we describe the novel MUGF method (previously just sketched in [24]) in order to enclose in an unified framework the ability of handling complex systems modeled by MSSs (as VNFs in a multi-tenant environment), and the capacity to evaluate, through series/parallel MUGF operators, the availability of a *chained* system (such as an SFC infrastructure).

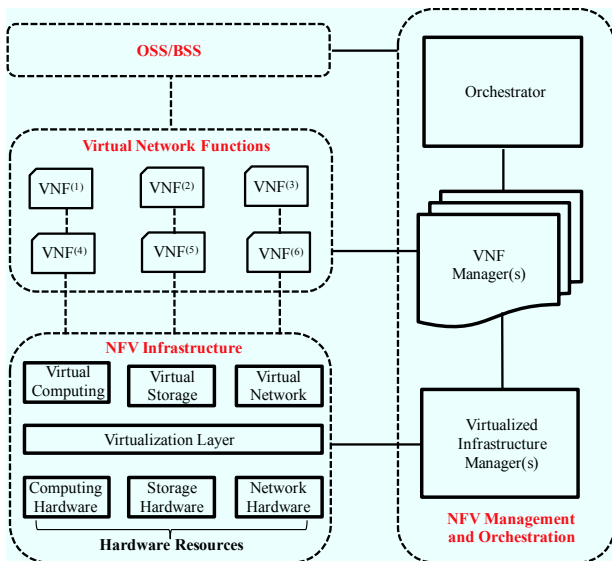


Fig. 1. The reference framework of the NFV architecture.

4 SERVICE FUNCTION CHAINING IN AN NFV ENVIRONMENT

In this Section we introduce the NFV architecture and detail a multi-tenant SFC infrastructure in line with the current standardization process.

4.1 The NFV architecture

NFV solutions offer such benefits as: *i) cost saving*, resulting from the use of generic and cheaper hardware platforms rather than dedicated and costly ones; *ii) scalability*, meant as the possibility of increasing or reducing the utilization of network equipments; *iii) flexibility*, achieved by faster deployment procedures of new services; *iv) security*, enforced by separation and isolation of network functions. In the NFV architectural framework the functional blocks are assorted in three domains, as per Fig. 1:

- *Virtual Network Functions*, containing all the VNFs, namely the virtualized instances of network functions (routers, firewalls, load balancers, etc.), replacing traditional appliances;
- *NFV Infrastructure*, containing all the hardware and software components useful to build VNFs, possibly distributed across several locations;
- *NFV management and orchestration*, containing the modules in charge of managing the VNFs functions, e.g. allocation of computing resources, storage and network connectivity to VNFs, root cause analysis, collection of fault information.

Such domains are supervised by the Operating and Business Support Systems (OSS/BSS) in order to guarantee network performances, customer support and charging/billing operations.

Being interested in a multi-tenant SFC model, we specifically focus on VNFs for multi-operator scenarios and on the NFV infrastructure domain, according to the framework in Fig. 1, where OSS/BSS and NFV Management and Orchestration domains are not considered for our purposes.

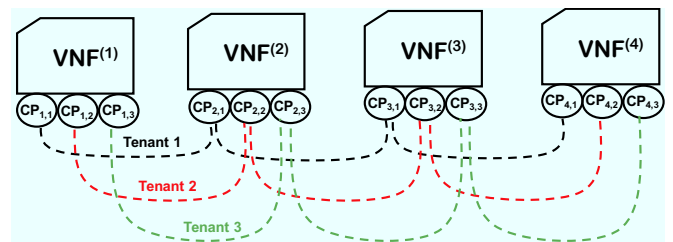


Fig. 2. A multi-tenant SFC architecture with 4 VNFs shared among 3 tenants, each one accessing dedicated VNF instances via Connection Points (CPs). $CP_{m,i}$ refers to the connection point of $VNF^{(m)}$ traversed by tenant i , $i = 1, \dots, K$.

4.2 A generic multi-tenant SFC model

Some providers are experimenting the deployment of multi-tenant SFCs where virtual resources are shared among different service operators. Three remarkable examples are reported below.

A commercial solution of a service chain designed for the LTE mobile telecommunication world has been presented in [12], where information flows of different telecom operators traverse common virtualized nodes. Some specifications about the virtualization (i.e. the deployment as VNFs chain) of main LTE nodes are also provided, such as Mobile Management Entity (vMME), Serving and Packet Data Network Gateway (vS/P-GW), Home Subscriber Server (vHSS), Policy Control and Charging Rules Function (vPCRF).

An exemplary implementation of a multi-tenant telecommunication system is offered by the Gateway Core Network (GWCN) [25], [26] for mobile networks, wherein more than one tenant share a consistent part of the underlying network infrastructure, so that, a tradeoff between security needs and cost savings arises. Obviously, the GWCN design has to be carefully planned by guaranteeing a satisfactory degree of isolation between tenants for security reasons (indeed, GWCN is often implemented among operators having strong commercial agreements).

Another example (inspiring the experiment in Section 7) is given by the novel 5G telecommunication network scenario in [27], proposing a virtualized IP Multimedia Subsystem architecture composed by virtualized Call Session Control Function (CSCF) nodes shared among different service providers.

Stemming from the analysis of the above examples, it is possible to derive a useful abstraction of a multi-tenant SFC infrastructure, that can be modeled as a chain of VNFs accessed by different operators through dedicated interfaces called Connection Points (CPs).

One such multi-tenant SFC system is shown in Fig. 2 where four VNFs are shared among three tenants. Each tenant i ($i = 1, \dots, K$) has its own dedicated access to the $VNF^{(m)}$ via connection point $CP_{m,i}$.

It is worth noting that, in this example, all tenants exploit the *same* SFC (the 4 VNFs traversed in the same order), but, in principle, they could also share a subset of VNFs or simply traverse them in a diverse order, resulting in *different* SFCs. In the latter case (not considered in this work), a classifier dispatching different flows to corresponding SFCs might be needed, and it should be added at the beginning of the SFC model.

5 AVAILABILITY ANALYSIS OF A SERVICE CHAIN BASED ON AN NFV INFRASTRUCTURE

We recall that an MSS is characterized by a finite number of states representing as many performance levels. For instance, a binary system is the simplest MSS with only two different states: perfect working and total failure.

An SFC can be regarded as an MSS, where the component VNFs are MSS subsystems as well. Firstly, we propose a performance model for a single VNF and then we provide an approach to evaluate the availability of a multi-tenant SFC system.

5.1 A VNF multi-state performance model

In Fig. 3, we sketch a multi-state performance model of a single VNF of a multi-tenant SFC system serving K different operators. We consider a typical performance metric for telecommunication service operators, namely the number of service requests that the VNF segment devoted to a single tenant is able to manage, typically referred to as *servicing capacity*. The proposed approach, however, can be easily extended to other performance metrics of interest.

From an availability point of view, we propose to model a single VNF as composed by the following three layers (see Fig. 3):

- a *service software layer*, representing some (identical) software instances that implement the VNF serving logic and work in parallel. Each instance is modeled as a component with two states: “active” (i.e. perfect functioning) and “failed” (i.e. complete failure). An instance has a servicing capacity equal to γ when active, and to 0 if failed. Tenant i is supposed to manage n_i software instances and to balance the load among them, where n_i is selected according to some performance and availability requirements ($i = 1, \dots, K$). For example, if a single instance in a virtualized LTE signaling node can manage up to 1000 requests, a tenant with a performance requirement of 5800 requests needs 6 software instances at least;
- a *virtualization layer*, also known as *hypervisor*, representing an element able to manage communications between hardware resources and software modules for each tenant i , and accessed via the corresponding CP. It is modeled as a two-state model: “active” or “failed”;
- a *hardware layer*, embodying hardware resources in the NFV Infrastructure domain (see Fig. 1). Similar to the virtualization layer, a two-state model is assumed.

Furthermore, we suppose that: *i*) software, virtualization layer and hardware failures are independent Homogeneous Poisson Processes (HPPs) with rates λ_{s_i} (for tenant $i = 1, \dots, K$), λ_v and λ_h , respectively; *ii*) software, virtualization layer and hardware repair times are independent and exponential random variables with rates μ_{s_i} ($i = 1, \dots, K$), μ_v and μ_h , respectively.

By arranging in each state all the numbers of active software instances managed by the tenants, the resulting

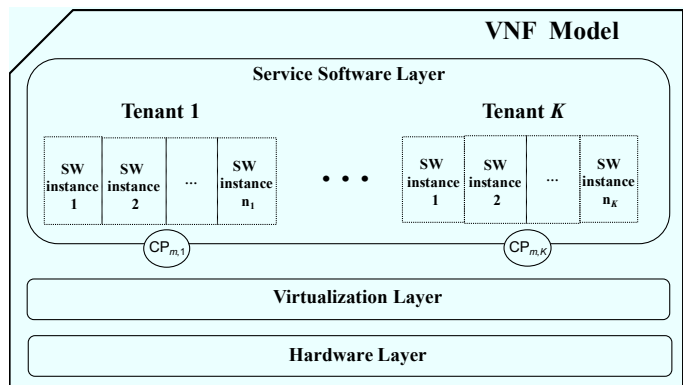


Fig. 3. Schematic representation of VNF hosting K tenants. The service software layer is represented by a set of tenants, each one managing some instances. The virtualization layer represents the interface with underlying physical resources (hardware layer).

VNF multi-state model is the homogeneous Continuous-Time Markov Chain (CTMC) depicted in Fig. 4, where:

- A generic state j is described by the K -dimensional vector $\alpha_j = (\alpha_{1,j}, \dots, \alpha_{K,j})$, where $\alpha_{i,j} \in 0, 1, \dots, n_i$ is the number of active software instances managed by tenant i in that state; the initial state vector (n_1, \dots, n_K) refers to a fully working system where the maximum number of active software instances for each tenant is available. For example, the vector $(n_1, \dots, n_i - 1, \dots, n_K)$ refers to a state where a single software instance devoted to tenant i is down.
- The K -dimensional state vector $(0, 0, \dots, 0)$ refers to a state where no software instance is active, whether the hypervisor is working or not.
- The state *Virtualization Layer Failure* (VLF) refers to the hypervisor failure condition causing the crash of all states except for the Hardware Layer Failure (defined below), as indicated by the dashed arrows with λ_v rate in Fig. 4. From VLF state, a single transition to the initial state is assumed, because a repairing action of the hypervisor is typically concluded by a complete element restoration. Such an operation usually requires a different activity (with mean duration $1/\mu_v$).
- The state *Hardware Layer Failure* (HLF) pertains to the hardware failure condition compelling the collapse of hypervisor and all running instances, simultaneously. The corresponding performance vector is $(0, 0, \dots, 0)$, as well. Also the HLF state can be reached by any other state, as pointed by the dashed arrows with λ_h rate in Fig. 4. As in the previous case, from HLF state, a single transition to the initial state is supposed, because a repairing action of hardware layer is presumed to be concluded by a complete VNF rehabilitation. Such an operation requires a technical activity with mean duration $1/\mu_h$.

It is useful to clarify that, according to the model in Fig. 3, the failure of the hypervisor or the hardware has effects only on the instances running on the considered physical host, without compromising instances of other VNFs running on different physical devices, possibly located in different geographical regions.

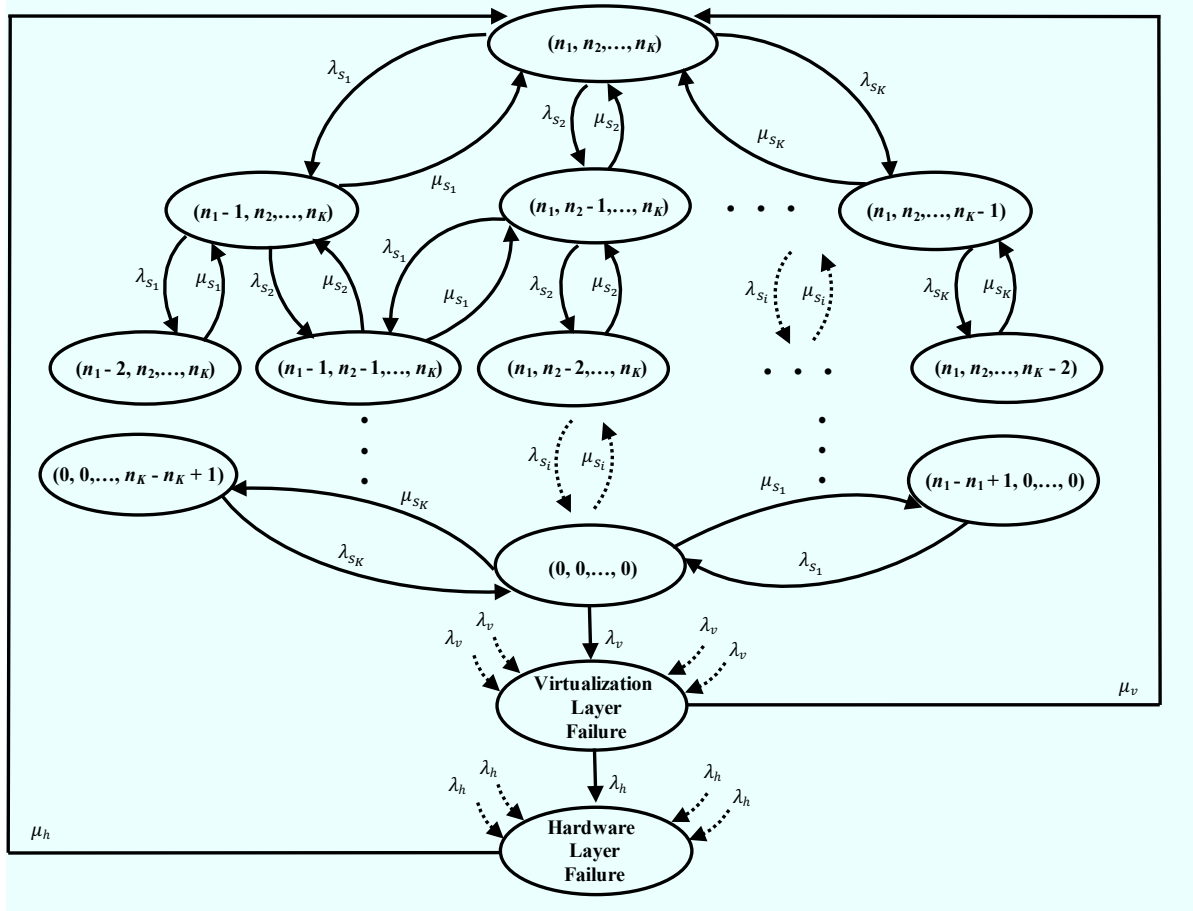


Fig. 4. Multi-state model of a generic VNF. Each state contains the number of active software instances for all K tenants.

Each state of the CTMC corresponds to a K -dimensional *performance vector* containing the total VNF serving capacity to all tenants. Thus, the *performance level* $g_{i,j}$ offered by the VNF for tenant i in state j can be defined as

$$g_{i,j} = \gamma \cdot \alpha_{i,j}. \quad (1)$$

The set containing all possible performance levels of a single VNF is

$$\mathcal{G} = \{\mathbf{g}_1, \mathbf{g}_2, \dots, \mathbf{g}_N\}, \quad (2)$$

where $\mathbf{g}_j = (g_{1,j}, \dots, g_{K,j}) = \gamma \cdot \alpha_j$ is the performance vector in state j , and N is the number of states given by

$$N = \prod_{i=1}^K (n_i + 1) + 2. \quad (3)$$

Therefore, the VNF performance level at time $t \geq 0$ is modeled by the vector stochastic process $\mathbf{G}(t) = (G_1(t), \dots, G_K(t)) \in \mathcal{G}$ with (state) probability vector $\mathbf{p}(t) = (p_1(t), \dots, p_N(t))$ at time t , where $p_j(t) = \Pr\{\mathbf{G}(t) = \mathbf{g}_j\}$, for $j = 1, \dots, N$. Given an initial probability vector a time $t = 0$, $\mathbf{p}(t)$ is derived at $t \geq 0$ by solving the system of differential equations [28]

$$\frac{d\mathbf{p}(t)}{dt} = \mathbf{p}(t)\mathbf{Q}$$

together with the normalization condition $\sum_{j=1}^N p_j(t) = 1$, where \mathbf{Q} is the infinitesimal generator matrix [29] of the CTMC shown in Fig. 4. Being the performance model $\mathbf{G}(t)$

an ergodic CTMC, a unique steady-state probability distribution $\mathbf{p} = (p_1, \dots, p_N)$ is the solution of $\mathbf{p}\mathbf{Q} = \mathbf{0}$, where

$$p_j = \lim_{t \rightarrow \infty} \Pr\{\mathbf{G}(t) = \mathbf{g}_j\}, \quad j = 1, \dots, N. \quad (4)$$

Thus, the discrete random vector $\mathbf{G} = (G_1, \dots, G_K)$, representing the asymptotic behavior of $\mathbf{G}(t)$ as $t \rightarrow \infty$, has values in the set (2) with probabilities (4). The collection of pairs $\{p_j, \mathbf{g}_j\}$, $j = 1, \dots, N$, completely determines the steady-state performance behavior (in terms of serving capacity) of a VNF.

5.2 Availability of the SFC

An availability model for an SFC is built by considering that: *i*) the SFC works when all VNFs are operative (see Fig. 2); thus, a *series* connection among them is in force; *ii*) in order to cope with faults and to guarantee high availability requirements, some redundancy must be introduced. We adopt *parallel* redundancy for each VNF, where some load balancing solutions are also assumed to exploit simultaneously all the parallel VNFs. The final series/parallel availability model for an SFC is shown in Fig. 5, where $\text{VNF}^{(m,l)}$ is the parallel node l of VNF m composing the SFC system, and the *subsystem* m is the parallel redundant configuration of VNF m . A multi-tenant SFC system is supposed to be *available* when each tenant provides to its customers a required performance level (also referred to as *demand*): this calls for the introduction of a K -dimensional demand vector

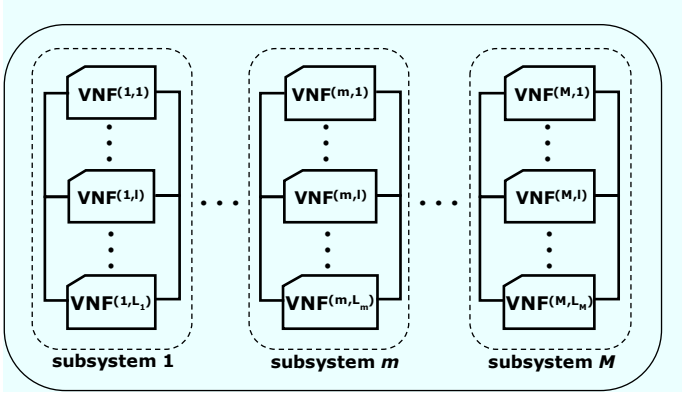


Fig. 5. A multi-tenant SFC infrastructure where parallel configuration for each VNF element is considered. $\text{VNF}^{(m,l)}$ refers to parallel node l ($l = 1, \dots, L_m$) of subsystem m ($m = 1, \dots, M$).

$\mathbf{W}(t) = (W_1(t), \dots, W_K(t))$. To cope with a given demand in the presence of random failures via parallel redundancy, the whole SFC can be conveniently modeled as a series-parallel system with *flow dispersion* [13], indicating that any of the parallel elements is able to handle service requests. Therefore, the performance levels of a subsystem composed by parallel VNFs is given by the sum of the performance vectors provided by a single VNF. Finally, the series connection among redundant functionalities imposes that the SFC system performance be limited by the subsystem with the lowest performance level for each tenant. Focusing on tenant i , the performance level at time t of the series-parallel SFC system is:

$$G_i^S(t) = \min_{m \in \{1, \dots, M\}} \sum_{l=1}^{L_m} G_i^{(m,l)}(t), \quad (5)$$

where $G_i^{(m,l)}(t)$ represents the performance level of tenant i exhibited by parallel element l of subsystem m at time t , and L_m is the number of parallel nodes of subsystem m .

Let $\mathbf{G}^S(t) = (G_1^S(t), \dots, G_K^S(t))$ be the SFC system performance process, namely the vector stochastic process collecting all tenants performance (5), $\forall i = 1, \dots, K$. Accordingly, $\mathbf{G}^S(t)$ can be represented for $t \rightarrow \infty$ by a (discrete) random vector $\mathbf{G}^S = (G_1^S, \dots, G_K^S)$ with a number of (vector) states equal to

$$J = \prod_{m=1}^M J^{(m)}, \quad (6)$$

where $J^{(m)} = \prod_{l=1}^{L_m} N^{(m,l)}$ is the number of states of the subsystem m , and $N^{(m,l)}$ is given by (3) for each parallel element l of m , where $n_i^{(m,l)}$ software instances are considered for tenant i .

Let $\mathbf{g}_j^S = (g_{1,j}^S, \dots, g_{K,j}^S)$ be the performance level vector j of the SFC, and $p_j^S = \lim_{t \rightarrow \infty} \Pr\{\mathbf{G}^S(t) = \mathbf{g}_j^S\}$ its limiting probability as $t \rightarrow \infty$. The collection of pairs

$$\{p_j^S, \mathbf{g}_j^S\}, j = 1, \dots, J \quad (7)$$

completely determines the steady-state performance behavior of the whole SFC system. It is worth noting that a direct solution of the CTMC representing the whole MSS of the SFC system is unfeasible, due to the huge state space cardinality J in (6). However, hierarchical approaches, like that

proposed in Section 6, can help to reduce the required computational burden in finding the steady-state performance distribution of the overall SFC and, then, its steady-state availability. This latter can be determined by introducing the *instantaneous availability* [13] $A^S[t, \mathbf{W}(t)]$, namely the probability that the system at $t > 0$ is in one of the *acceptable states*, defined as the states where performance of tenant i is not less than demand $W_i(t)$ for each $i = 1, \dots, K$, viz.

$$A^S[t, \mathbf{W}(t)] = \Pr\{G_i(t) - W_i(t) \geq 0, \forall i = 1, \dots, K\}. \quad (8)$$

As t grows, the system initial state has a vanishing influence on its availability. Therefore, given a constant demand level $\mathbf{W}(t) = \mathbf{w} = (w_1, \dots, w_K)$, the *steady-state availability* $A^S(\mathbf{w}) = \lim_{t \rightarrow \infty} A^S(t, \mathbf{w})$ can be computed by means of the pairs (7) as

$$A^S(\mathbf{w}) = \sum_{j=1}^J p_j^S \cdot \mathbf{1}(g_{i,j}^S \geq w_i, \forall i = 1, \dots, K), \quad (9)$$

where $\mathbf{1}(\mathcal{C}) = 1$ if condition \mathcal{C} holds true and 0 otherwise.

6 AVAILABILITY EVALUATION OF A MULTI-TENANT SFC SYSTEM

A convenient procedure to compute the steady-state distribution (hence the availability or other dependability metrics of complex MSS systems) is based on the UGF method. It is a hierarchical approach that avoids handling the huge overall state-space model of a complex system (which likely turns out insoluble in most practical configurations) by combining the distributions of its components through some suitable operators amenable to numerical implementations with affordable complexity [13]. In case of series-parallel systems, series and parallel operators are needed. For the sake of clarity, we start by presenting the UGF of an SFC system used by a single tenant ($K = 1$), where scalar performance levels are involved. Then, we generalize the approach to $K > 1$ by introducing the Multidimensional UGF.

6.1 UGF for single-tenant SFC

The UGF of the (steady-state) performance distribution $G^{(m,l)}$ of parallel node l of subsystem m is the polynomial-shape function (also referred to as *u-function*) defined as:

$$u^{(m,l)}(z) = \sum_{j=1}^{N^{(m,l)}} p_j^{(m,l)} z^{g_j^{(m,l)}}, \quad (10)$$

where $g_j^{(m,l)}$ is the performance level in state j , and $p_j^{(m,l)}$ is the corresponding steady-state probability.

In order to characterize the *u-function* of a system described by a series/parallel availability model, two operators can be adopted: an operator π to build the *u-function* of subsystems connected in parallel, and an operator σ to calculate the *u-function* of elements interconnected in series. Recall that, under the assumption of adopting load balancing techniques for parallel nodes [22], a subsystem constituted by multi-state components with flow dispersion is obtained. According to [21], the *u-function* of subsystem

m with L_m parallel nodes is expressed by the π operator that represents the product of the UGFs for each node, namely

$$\begin{aligned} u^{(m)}(z) &= \pi \left(u^{(m,1)}(z), \dots, u^{(m,L_m)}(z) \right) \\ &= \prod_{l=1}^{L_m} \left(\sum_{j_l=1}^{N^{(m,l)}} p_{j_l}^{(m,l)} z^{g_{j_l}^{(m,l)}} \right) \\ &= \sum_{j_1=1}^{N^{(m,1)}} \dots \sum_{j_{L_m}=1}^{N^{(m,L_m)}} \left[\left(\prod_{l=1}^{L_m} p_{j_l}^{(m,l)} \right) z^{\sum_{l=1}^{L_m} g_{j_l}^{(m,l)}} \right] \\ &= \sum_{j=1}^{J^{(m)}} p_j^{(m)} z^{g_j^{(m)}}. \end{aligned} \quad (11)$$

On the other hand, the UGF of a series configuration is defined (for a couple of subsystems) as:

$$\sigma \left(u^{(1)}(z), u^{(2)}(z) \right) = \sum_{j=1}^{J^{(1)}} \sum_{h=1}^{J^{(2)}} p_j^{(1)} p_h^{(2)} z^{\min\{g_j^{(1)}, g_h^{(2)}\}}, \quad (12)$$

where the minimum arises since the element exhibiting the lowest performance level represents the bottleneck in a series-connected system. By applying (12) recursively, the UGF of M subsystems composing the SFC system in Fig. 5 is easily derived as:

$$u^S(z) = \sigma \left(u^{(1)}(z), u^{(2)}(z), \dots, u^{(M)}(z) \right), \quad (13)$$

whose UGFs are provided by (11), that can be recast as

$$u^S(z) = \sum_{j=1}^J p_j^S z^{g_j^S}, \quad (14)$$

where J is given by (6).

Through (14), the performance levels g_j^S and the steady-state probabilities p_j^S are simply read out as the exponents of z and the respective coefficients. The steady-state availability $A^S(w)$ is then obtained by (9), particularized with $K = 1$ and scalar demand w .

6.2 The Multidimensional UGF

As said before, the availability evaluation of a multi-tenant SFC signaling system involves vector random processes describing performance of all tenants using the system. In order to preserve the benefits of the UGF approach, we propose the *Multidimensional* UGF (MUGF) that accounts for the case $K > 1$, useful to handle performance vectors such as $\mathbf{G}^{(m,l)}$ and \mathbf{G} . Indeed, the MUGF approach allows to efficiently combine the collection of pairs $\left\{ p_j^{(m,l)}, \mathbf{g}_j^{(m,l)} \right\} \forall m, l$, arising from the model in Section 5.1, toward assessing the steady-state performance and availability of the whole SFC. The main concept underlying our proposal is a "dimension-wise" extension of the π and σ operators.

Accordingly, in keeping with the definition of multivariate probability generating function, we define the MUGF $u(\mathbf{z})$ of the K -dimensional random vector \mathbf{G} , with values in

the set $\{\mathbf{g}_1, \dots, \mathbf{g}_J\}$ and probabilities in the set $\{p_1, \dots, p_J\}$, as

$$u(\mathbf{z}) = \sum_{j=1}^J p_j \prod_{i=1}^K z_i^{g_i^{j,j}}, \quad (15)$$

where $\mathbf{g}_j = (g_{1,j}, \dots, g_{K,j})$ and $\mathbf{z} = (z_1, \dots, z_K)$.

As a result, the MUGF $u^{(m)}(\mathbf{z})$ of a subsystem m , composed by L_m parallel VNFs with flow dispersion ¹, amounts to the following extension of (11):

$$\begin{aligned} u^{(m)}(\mathbf{z}) &= \pi \left(u^{(m,1)}(\mathbf{z}), \dots, u^{(m,L_m)}(\mathbf{z}) \right) \\ &= \prod_{l=1}^{L_m} \left[\sum_{j_l=1}^{N^{(m,l)}} \left(p_{j_l}^{(m,l)} \prod_{i=1}^K z_i^{g_{i,j_l}^{(m,l)}} \right) \right] \\ &= \sum_{j_1=1}^{N^{(m,1)}} \dots \sum_{j_{L_m}=1}^{N^{(m,L_m)}} \left(\prod_{l=1}^{L_m} p_{j_l}^{(m,l)} \right) \prod_{i=1}^K z_i^{\sum_{l=1}^{L_m} g_{i,j_l}^{(m,l)}} \\ &= \sum_{j=1}^{J^{(m)}} p_j^{(m)} \prod_{i=1}^K z_i^{g_i^{(m)}}, \end{aligned} \quad (16)$$

where $u^{(m,1)}(\mathbf{z}), \dots, u^{(m,L_m)}(\mathbf{z})$ represent the MUGFs of nodes composing subsystem m with $J^{(m)}$ different states, characterized by performance levels vectors $\mathbf{g}_j^{(m)} = (g_{1,j}^{(m)}, \dots, g_{K,j}^{(m)})$ and occurrence probability $p_j^{(m)}$.

Once drawn $u^{(m)}(\mathbf{z})$ by (16), the MUGF $u^S(\mathbf{z})$ of a multi-tenant SFC system is computed by the following series operator

$$u^S(\mathbf{z}) = \sigma \left(u^{(1)}(\mathbf{z}), u^{(2)}(\mathbf{z}), \dots, u^{(M)}(\mathbf{z}) \right), \quad (17)$$

which can be elaborated by applying recursively the K -dimensional version of the binary operator (12), viz.:

$$\begin{aligned} \sigma \left(u^{(1)}(\mathbf{z}), u^{(2)}(\mathbf{z}) \right) &= \sum_{j=1}^{J^{(1)}} \sum_{h=1}^{J^{(2)}} p_j^{(1)} p_h^{(2)} \prod_{i=1}^K z_i^{\min\{g_{i,j}^{(1)}, g_{i,h}^{(2)}\}}, \end{aligned} \quad (18)$$

where $J^{(1)}$ ($J^{(2)}$) is the number of performance vectors \mathbf{g}_j (\mathbf{g}_h), whose probability is $p_j^{(1)}$ ($p_h^{(2)}$), of subsystem m_1 (m_2).

Finally, $u^S(\mathbf{z})$ can be recast, like (14), as

$$u^S(\mathbf{z}) = \sum_{j=1}^J p_j^S \prod_{i=1}^K z_i^{g_i^{j,j}}, \quad (19)$$

where J is given by (6). Therefore, $u^S(\mathbf{z})$ is a polynomial-shape function in K indeterminates z_1, \dots, z_K , where each term provides the performance vector $\mathbf{g}_j^S = (g_{1,j}^S, \dots, g_{K,j}^S)$ (exponents of z_i), and its steady-state probability p_j^S (corresponding coefficient). It is worth noting that the sum in (19) collects all the terms with the same exponents $\mathbf{g}_{i,j}^S$ (by summing the corresponding probabilities), thus the resulting number of the effective performance levels can be much less than J .

1. A "parallel" version of the MUGF operator has also been proposed for the case of parallel systems without flow dispersion [24].

The steady-state availability $A^S(\mathbf{w})$ of the multi-tenant SFC system is finally provided by (9).

Once granted an expression for $A^S(\mathbf{w})$, it becomes possible [30] to address some redundancy optimization problem by exhaustive search or other known techniques, such as genetic algorithms [13], [31].

Letting $C^{(m,l)}$ be the cost of parallel node l in subsystem m , the overall cost of the multi-tenant SFC configuration $\mathbf{l} = (L_1, \dots, L_M)$ is

$$C^S(\mathbf{l}) = \sum_{m=1}^M \sum_{l=1}^{L_m} C^{(m,l)}. \quad (20)$$

A problem of interest when designing a multi-tenant SFC service is to devise the configuration \mathbf{l}^* that minimizes the total cost of deployment while satisfying a certain steady-state availability requirement A_0 . Given the set $\mathcal{L}^S = \{\mathbf{l} : A^S(\mathbf{w}, \mathbf{l}) \geq A_0\}$ of the possible configurations satisfying the steady-state availability condition, the formal solution of the problem amounts to:

$$\mathbf{l}^* = \arg \min_{\mathbf{l} \in \mathcal{L}^S} C^S(\mathbf{l}). \quad (21)$$

7 A NUMERICAL EXAMPLE

In this section, we provide an example of availability analysis focusing on a cutting-edge deployment of a virtualized SFC infrastructure: the IP Multimedia Subsystem [27], [32]. IMS has been elected by the telecommunication industry as the enabling technology of 5G networks, providing a huge variety of IP-based services ranging from real-time multimedia (i.e. phone calls) to web messaging. The IMS signaling network functionalities are called Call Session Control Functions (CSCFs), and are distributed among three servers that communicate mainly by exchanging Session Initiation Protocol (SIP) messages: the *Proxy* CSCF (PCSCF), acting as an interface between a user and the IMS network; the *Serving* CSCF (SCSCF), performing some core functions such as session and routing control or user registration management; the *Interrogating* CSCF (ICSCF), forwarding SIP requests or responses to the appropriate SCSCF. Another key element is the *Home Subscriber Server* (HSS), an advanced database containing users' profiles that can be queried by means of Diameter, a specific protocol.

By exploiting the SFC paradigm, IMS can be deployed as a service chain named virtualized IMS (vIMS).

In this example, we focus on the call set-up procedure between two mobile phones. The considered scenario is represented in Fig. 6(a), where the signaling flow, originated by a calling User Equipment (UE₁), traverses the IMS servers in an ordered way to reach a called UE (UE₂). In particular, the SIP request of UE₁ is forwarded by the server PCSCF (the first contact point of IMS network) towards SCSCF₁. In order to reach UE₂, SCSCF₁ forwards such a request to server I that, after querying HSS, can identify server SCSCF₂ in charge of managing the network area where UE₂ is located. Upon completion of the call set-up procedure, UE₁ and UE₂ can establish a multimedia session, e.g. an audio/video call.

Figure 6(b) shows the same system in a multi-tenant SFC environment, where a chain of parallel vIMS elements (all

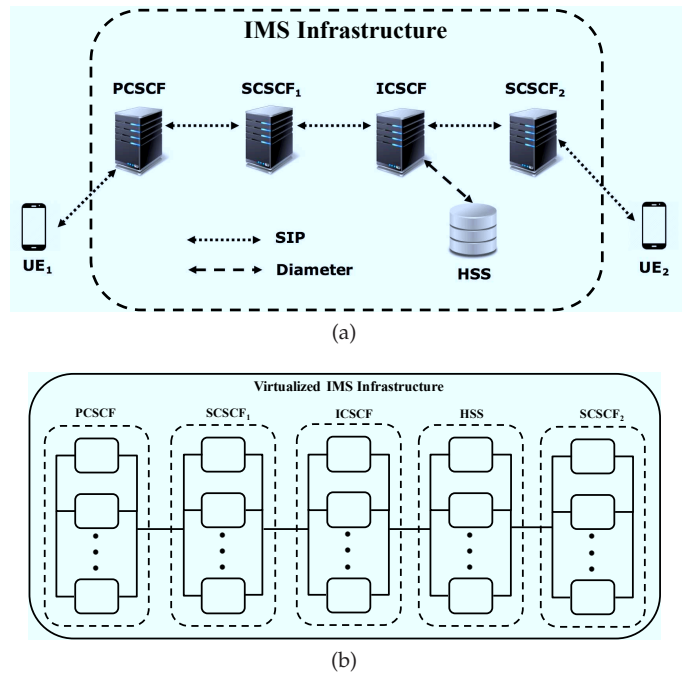


Fig. 6. (a) The IMS signaling network for call set-up establishment between two UEs. (b) A multi-tenant vIMS representation, where parallel redundancy for each IMS server is considered.

shared among the various operators) has been introduced in order to achieve the required redundancy for high availability of the call set-up service.

To exemplify the MUGF approach introduced in Section 6, we compute the steady-state availability of a minimal deployment cost of the vIMS system, by solving the redundancy optimization problem (21) with a given (steady-state) availability requirement.

We assume for simplicity that the nodes composing the vIMS series-parallel system have one and the same performance model like that proposed in Section 5.1, where relevant parameters are: number of states $N^{(m,l)} = N$; failure and repair rates for the hardware layer $\lambda_h^{(m,l)} = \lambda_h$ and $\mu_h^{(m,l)} = \mu_h$, respectively; failure and repair rates for the virtualization layer $\lambda_v^{(m,l)} = \lambda_v$ and $\mu_v^{(m,l)} = \mu_v$, respectively; failure and repair rates for the software instances $\lambda_{s_i}^{(m,l)} = \lambda_{s_i}$, $\mu_{s_i}^{(m,l)} = \mu_{s_i}$, respectively; number of the software instances implementing the service logic for a given tenant i $n_i^{(m,l)} = n_i$, $\forall i = 1, \dots, K$. We assume also that all nodes have one and the same cost, that we suppose, with no lack of generality, equal to 1, i.e. $C^{(m,l)} = 1$, $\forall l \in \{1, \dots, L_m\}$, with m indexing the set $\{\text{PCSCF}, \text{SCSCF}_1, \text{ICSCF}, \text{HSS}, \text{SCSCF}_2\}$. However, these assumptions can be easily generalized to more realistic scenarios, with different performance characteristics (due to services offered by IMS servers) and costs imposed by different manufacturers.

In our exemplary scenario, $K = 2$ telecom operators (tenants) are using the vIMS system and offer different service levels to their customers. Precisely, we assume a (constant) demand vector $\mathbf{w} = (w_1, w_2) = (15000, 25000)$ sessions. Given the serving capacity $\gamma = 10000$ sessions, we choose, somewhat arbitrarily, $n_1 = \lceil w_1/\gamma \rceil = 2$ and $n_2 =$

$\lceil w_2/\gamma \rceil = 3$, as these are the smallest integers allowing to satisfy the demand without redundancy and without faults. The resulting MSS performance model of a vIMS node (thus representing a VNF) is a special case of the CTMC depicted in Fig. 4, with $N = (n_1 + 1)(n_2 + 1) + 2 = 14$ different states (after applying (3)) described by a 2-dimensional performance vector containing the number of the active software instances for each tenant. Such an exemplary MSS is depicted in Fig. 7, where, in accordance with the model presented in Fig. 4, a fully working system is embodied in the state (2, 3), whereas, a failed system is simultaneously accounted by states (0, 0), VLF and HLF. Failure and repair rates of software instances are assumed the same for both tenants and in line with telecommunication experts' hints and with pertinent literature [33]: $\lambda_{s_1} = \lambda_{s_2} = 1.587 \times 10^{-6} \text{ s}^{-1}$ (corresponding to 1 fault every 175 hours) and $\mu_{s_1} = \mu_{s_2} = 5.556 \times 10^{-4} \text{ s}^{-1}$ (corresponding to a mean repair time of 30 minutes). Failure and repair rates of the virtualization part are $\lambda_v = 1.047 \times 10^{-7} \text{ s}^{-1}$ (corresponding to 1 fault every 2654 hours) and $\mu_v = 1.667 \times 10^{-4} \text{ s}^{-1}$ (corresponding to a mean repair time of 100 minutes), respectively. Finally, failure and repair rates of the hardware layer are $\lambda_h = 4.630 \times 10^{-9} \text{ s}^{-1}$ (corresponding to 1 fault every 60000 hours) and $\mu_h = 3.472 \times 10^{-5} \text{ s}^{-1}$ (corresponding to a mean repair time of 8 hours), respectively. By solving this CTMC with $N = 14$ states for $t \rightarrow \infty$ as described in Sect. 5.1, the steady-state performance distribution (in terms of the number of call set-up sessions) for a single virtualized node is computed, and is given by the collection of pairs $\{p_j, g_j\}$, where p_j is the probability (4) correlated to the performance level g_j in the set (2) and where \mathbf{Q} is given by (22), in which, for brevity, $\sum \mu_s$ denotes $\sum_{i=1}^2 \mu_{s_i}$ and the like. Table 1 summarizes the obtained performance levels and pertinent state probabilities. It is useful to note that $p_1 + p_2 + p_3$ refers to (0, 0) performance vector since HLF, VLF and the state accounting for all failed software instances admit the same performance vector (namely, a completely failed system).

The corresponding MUGF $u(\mathbf{z}) = u(z_1, z_2)$ is reported in (23).

In order to meet the "five nines" availability requirement for the vIMS, we set $A_0 = 1 - 10^{-5}$ and solve numerically the optimization problem (21) by an exhaustive search approach, having imposed a uniform limitation of 4 redundant nodes for each server. A routine (available upon request for non commercial use), written in Mathematica[®], implements the proposed MUGF approach: it evaluates the steady-state availability of the vIMS system (9) in every redundancy configuration by computing the MUGF (19) via parallel and series operators (16) and (18), respectively. Then, the numeric algorithm finds the minimal cost configuration(s) \mathbf{l}^* . In this use case, 5 (almost) equivalent solutions were found: they correspond to \mathbf{l}^* consisting of the combinations of 4 (arbitrarily chosen) subsystems with 3 redundant nodes, and the remaining subsystem with 2 redundant nodes (we briefly indicate such a configuration by $\mathbf{l}^* = (2, 3, 3, 3, 3)$).

The MUGF $u^I(\mathbf{z})$ of \mathbf{l}^* is reported in (24), where the probabilities and the performance levels of the acceptable states (those with performance levels greater than 15000 and 25000 for the first and the second tenant, respectively) are underlined and highlighted (in green in the electronic version). The resulting steady-state availability, in terms of the

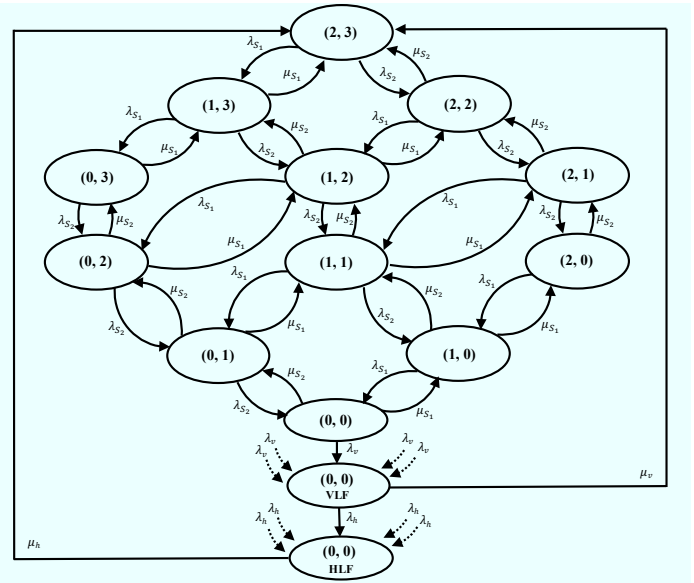


Fig. 7. A multi-state model of a virtualized IMS node (VNF) with 2 tenants; the tenant 1 is supposed to manage 2 software instances and the tenant 2 is supposed to manage 3 software instances.

TABLE 1
Steady-state probabilities and performance levels in terms of the number of call set-up sessions of a virtualized node.

State probabilities	Probability values	Performance levels
$p_1 + p_2 + p_3$	7.608×10^{-4}	(0, 0)
p_4	6.617×10^{-11}	(0, 10000)
p_5	2.316×10^{-8}	(0, 20000)
p_6	8.107×10^{-6}	(0, 30000)
p_7	6.617×10^{-11}	(10000, 0)
p_8	2.316×10^{-8}	(10000, 10000)
p_9	8.108×10^{-6}	(10000, 20000)
p_{10}	2.838×10^{-3}	(10000, 30000)
p_{11}	2.316×10^{-8}	(20000, 0)
p_{12}	8.107×10^{-6}	(20000, 10000)
p_{13}	2.838×10^{-3}	(20000, 20000)
p_{14}	0.9935	(20000, 30000)

number of call set-up sessions managed by the vIMS system, is computed as the sum of the probabilities of the acceptable states and is equal to $A^I(\mathbf{w}, \mathbf{l}^*) = 0.999993413$ while the cost is $C^I(\mathbf{l}^*) = 14$. An availability analysis performed by considering some alternative redundant configurations \mathbf{l} of vIMS servers offers the results shown in Fig. 8, where, for a more comfortable visualization, the unavailability $1 - A^I(\mathbf{w}, \mathbf{l})$ is reported. The horizontal dashed line represents the target steady-state unavailability $1 - A_0 = 10^{-5}$. It is readily seen that configurations $\mathbf{l}_1 = (2, 2, 2, 2, 2)$ (with $C^I(\mathbf{l}_1) = 10$), $\mathbf{l}_2 = (2, 2, 2, 3, 3)$ (with $C^I(\mathbf{l}_2) = 12$) and $\mathbf{l}_3 = (2, 2, 3, 3, 3)$ (with $C^I(\mathbf{l}_3) = 13$) do not meet the availability requirement. Remarkably, adding just 1 extra redundant component, the configuration $\mathbf{l}_4 = (3, 3, 3, 3, 3)$

$$\mathbf{Q} = \begin{pmatrix}
-\mu_h & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_h \\
\lambda_h - \lambda_h - \mu_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \mu_v \\
\lambda_h & \lambda_v & -\lambda_h - \lambda_v - \sum \mu_s & \mu_{s1} & \mu_{s2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_h & \lambda_v & \lambda_{s1} & -\lambda_v - \lambda_h - \lambda_{s1} - \sum \mu_s & 0 & \mu_{s1} & \mu_{s2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_h & \lambda_v & \lambda_{s2} & 0 & -\lambda_h - \lambda_v - \lambda_{s2} - \sum \mu_s & 0 & \mu_{s1} & \mu_{s2} & 0 & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_h & \lambda_v & 0 & \lambda_{s1} & 0 & -\lambda_h - \lambda_v - \lambda_{s1} - \mu_{s2} & 0 & 0 & \mu_{s2} & 0 & 0 & 0 & 0 & 0 & 0 \\
\lambda_h & \lambda_v & 0 & \lambda_{s2} & \lambda_{s1} & 0 & -\lambda_h - \lambda_v - \sum \lambda_s - \sum \mu_s & 0 & \mu_{s1} & \mu_{s2} & 0 & 0 & 0 & 0 & 0 \\
\lambda_h & \lambda_v & 0 & 0 & \lambda_{s2} & 0 & 0 & -\lambda_h - \lambda_v - \lambda_{s2} - \sum \mu_s & 0 & \mu_{s1} & \mu_{s2} & 0 & 0 & 0 & 0 \\
\lambda_h & \lambda_v & 0 & 0 & 0 & \lambda_{s2} & \lambda_{s1} & 0 & -\lambda_h - \lambda_v - \sum \lambda_s - \mu_{s2} & 0 & 0 & \mu_{s2} & 0 & 0 & 0 \\
\lambda_h & \lambda_v & 0 & 0 & 0 & 0 & \lambda_{s2} & \lambda_{s1} & 0 & -\lambda_h - \lambda_v - \sum \lambda_s - \sum \mu_s & 0 & \mu_{s1} & \mu_{s2} & 0 & 0 \\
\lambda_h & \lambda_v & 0 & 0 & 0 & 0 & 0 & \lambda_{s2} & 0 & 0 & -\lambda_h - \lambda_v - \lambda_{s2} - \mu_{s1} & 0 & \mu_{s1} & 0 & 0 \\
\lambda_h & \lambda_v & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{s2} & \lambda_{s1} & 0 & -\lambda_h - \lambda_v - \sum \lambda_s - \mu_{s2} & 0 & \mu_{s2} & 0 \\
\lambda_h & \lambda_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{s2} & \lambda_{s1} & 0 & -\lambda_h - \lambda_v - \sum \lambda_s - \mu_{s1} & \mu_{s1} & 0 \\
\lambda_h & \lambda_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{s2} & \lambda_{s1} & 0 & -\lambda_h - \lambda_v - \sum \lambda_s & \mu_{s1} \\
\lambda_h & \lambda_v & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & 0 & \lambda_{s2} & \lambda_{s1} & -\lambda_h - \lambda_v - \sum \lambda_s & 0
\end{pmatrix} \quad (22)$$

$$\begin{aligned}
u(\mathbf{z}) = & 7.608 \times 10^{-4} + 6.617 \times 10^{-11} z_1^{10000} + 2.316 \times 10^{-8} z_1^{20000} + 6.617 \times 10^{-11} z_2^{10000} \\
& + 2.316 \times 10^{-8} z_1^{10000} z_2^{10000} + 8.107 \times 10^{-6} z_1^{20000} z_2^{10000} + 2.316 \times 10^{-8} z_2^{20000} \\
& + 8.108 \times 10^{-6} z_1^{10000} z_2^{20000} + 2.838 \times 10^{-3} z_1^{20000} z_2^{20000} + 8.107 \times 10^{-6} z_2^{30000} \\
& + 2.838 \times 10^{-3} z_1^{10000} z_2^{30000} + 0.9935 z_1^{20000} z_2^{30000}
\end{aligned} \quad (23)$$

$$\begin{aligned}
w^I(\mathbf{z}) = & 5.806 \times 10^{-7} + 1.011 \times 10^{-13} z_1^{10000} + 3.540 \times 10^{-11} z_1^{20000} + 3.149 \times 10^{-18} z_1^{30000} \\
& + 5.412 \times 10^{-16} z_1^{40000} + 1.011 \times 10^{-13} z_2^{10000} + 3.540 \times 10^{-11} z_1^{10000} z_2^{10000} \\
& + 1.239 \times 10^{-8} z_1^{20000} z_2^{10000} + 2.204 \times 10^{-15} z_1^{30000} z_2^{10000} + 3.789 \times 10^{-13} z_1^{40000} z_2^{10000} \\
& + 3.540 \times 10^{-11} z_2^{20000} + 1.239 \times 10^{-8} z_1^{10000} z_2^{20000} + 4.338 \times 10^{-6} z_1^{20000} z_2^{20000} \\
& + 1.157 \times 10^{-12} z_1^{30000} z_2^{20000} + 1.990 \times 10^{-10} z_1^{40000} z_2^{20000} + 1.239 \times 10^{-8} z_2^{30000} \\
& + 4.338 \times 10^{-6} z_1^{10000} z_2^{30000} + 1.519 \times 10^{-3} z_1^{20000} z_2^{30000} + 5.403 \times 10^{-10} z_1^{30000} z_2^{30000} \\
& + 9.287 \times 10^{-8} z_1^{40000} z_2^{30000} + 1.654 \times 10^{-15} z_2^{40000} + 1.158 \times 10^{-12} z_1^{10000} z_2^{40000} \\
& + 6.079 \times 10^{-10} z_1^{20000} z_2^{40000} + 1.418 \times 10^{-7} z_1^{30000} z_2^{40000} + 2.438 \times 10^{-5} z_1^{40000} z_2^{40000} \\
& + 3.858 \times 10^{-13} z_2^{50000} + 2.701 \times 10^{-10} z_1^{10000} z_2^{50000} + 1.418 \times 10^{-7} z_1^{20000} z_2^{50000} \\
& + 3.310 \times 10^{-5} z_1^{30000} z_2^{50000} + 5.690 \times 10^{-3} z_1^{40000} z_2^{50000} + 6.632 \times 10^{-11} z_2^{60000} \\
& + 4.643 \times 10^{-8} z_1^{10000} z_2^{60000} + 2.438 \times 10^{-5} z_1^{20000} z_2^{60000} + 5.690 \times 10^{-3} z_1^{30000} z_2^{60000} \\
& + 0.9870 z_1^{40000} z_2^{60000}
\end{aligned} \quad (24)$$

(with $C^I(\mathbf{l}_4) = 15$) achieves the much larger availability value of 0.9999999646. Thus, although exceeding the specifications, configuration \mathbf{l}_4 might be more appealing to the network designer. We remark that the MUGF approach circumvents the computational burden inherent in the baseline CTMC approach. Indeed, considering our example with $N^{(m,l)} = N = 14$, the state space of the CTMC is worth $J = 14^{\sum_{m=1}^5 L_m} = 14^n$, being n the number of nodes in the system. Thus, the optimization problem (21) amounts to the solution of 4^5 linear systems of J equations, with J ranging between $14^5 \cong 5.4 \cdot 10^5$ and $14^{20} \cong 8.4 \cdot 10^{22}$; in particular, for the optimal configuration $\mathbf{l}^* = (2, 3, 3, 3, 3)$, $J = 14^{(\sum_{m=1}^4 3)(\sum_{m=1}^1 2)} = 14^{14} \cong 1.1 \cdot 10^{16}$. Needless to say, the baseline approach is unfeasible with these numbers. On the other hand, the MUGF technique takes advantage of a

hierarchical decomposition of the problem as follows: 1) the steady-state distribution of each component VNF, namely a CTMC with $N = 14$ states, is computed by solving a system of N equations; 2) the computed distributions are combined via the series/parallel operators, which involve the standard algebraic manipulations in (16), (17) and (18). Coming to our example, step 1) of MUGF approach amounts to the solution of a system of $N = 14$ equations that requires about 10 ms for each VNF on a notebook based on an Intel Core i74960 HQ CPU@2.60GHz; whereas step 2) requires, for each configuration, an average time of 73 ms on the same platform, and hence about 75 s to complete the exhaustive search over 4^5 redundancy configurations. After having determined the solution \mathbf{l}^* , we have performed a sensitivity analysis aimed at evaluating the robustness of vIMS system

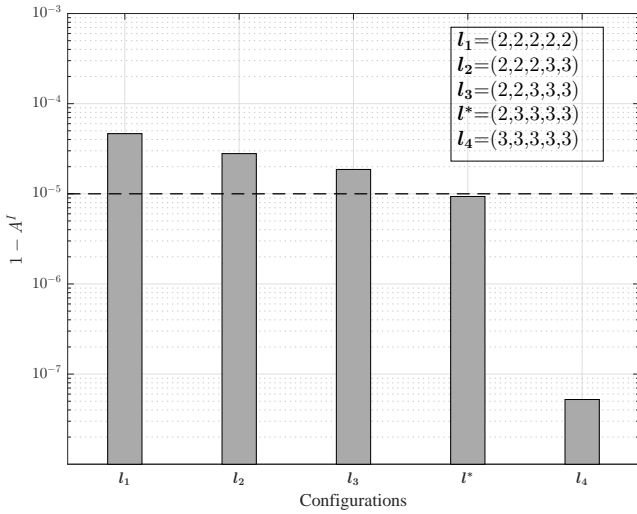


Fig. 8. Unavailability $1 - A^I$ of the virtualized IMS infrastructure with five exemplary redundant configurations l . The horizontal dashed line is the required steady-state unavailability $1 - A^I = 10^{-5}$. The minimal cost configuration guaranteeing the “five nines” condition is l^* .

with respect to deviation of some critical parameters from their nominal values. In particular, the panel of Figs. 9 shows the influence of failure rates $\lambda_s, \lambda_v, \lambda_h$, and repair rates μ_s, μ_v, μ_h (all expressed in terms of their reciprocals) on the overall system availability. In each figure, a circle (in red in the electronic version) points out the nominal value of the parameter under analysis, whereas the horizontal dashed line represents the “five nines” limit value.

Figures 9(a), 9(c), 9(e) highlight that l^* configuration still meets the “five nines” availability requirement for slightly higher fault rates up to (approximately) 1 fault every 167 hours, 1 fault every 2527 hours and 1 fault every 44440 hours, for software instances, virtualization layer and hardware layer, respectively. Similarly, Figs. 9(b), 9(d), 9(f) show that the nominal values $1/\mu_s, 1/\mu_v, 1/\mu_h$ can be relaxed up to (approximately) a mean repair time of 31.6 minutes, 105 minutes and 10.7 hours, for software instances, virtualization layer and hardware layer, respectively, still satisfying the “five nines” condition.

Finally, we have analyzed the effects of variations of w around its nominal value (15000, 25000), by determining the respective optimal configurations l^* and the pertinent availability values. Table 2 reports the results corresponding to increments or decrements to the initial value in blocks of 5000 for both tenants. As expected, the more demanding w , the greater the number of redundant elements and SFC cost are, and vice versa.

8 CONCLUSIONS

Today, the service composition is becoming a common practice for network and telecommunication operators desiring to boost the provisioning of novel services. In this spirit, the Service Function Chaining (SFC), supported by virtualization concepts introduced by Network Function Virtualization (NFV), proposes an infrastructure built on virtualized network functions (or VNFs) to be traversed in an ordered way aimed at providing specific services. In

TABLE 2
Effects of variation of w .

w	l^*	$A^I(w, l^*)$
(15000, 25000)	(2, 3, 3, 3, 3)	0.999990659
(20000, 20000)	(2, 2, 3, 3, 3)	0.999990022
(20000, 30000)	(2, 3, 3, 3, 3)	0.999990659
(10000, 30000)	(2, 2, 3, 3, 3)	0.999990114
(10000, 20000)	(2, 2, 2, 2, 2)	0.999996982

many applications, the VNFs belonging to the service chain are intended to host many instances of different operators (or tenants), resulting in a multi-tenant environment.

In this paper we have afforded an availability analysis of a multi-tenant SFC infrastructure by offering a three-fold contribution. First, we have modeled a multi-tenant SFC infrastructure as a multi-state system by conveniently combining series and parallel operators. Then, we have proposed an extended version of the Universal Generating Function (UGF) technique, referred to as Multidimensional UGF (MUGF), useful to cope with performance vectors applicable to complex multi-tenant network scenarios. Finally, we have performed an availability analysis in a realistic scenario of a virtualized IP Multimedia Subsystem (vIMS), a state-of-the-art deployment of a multi-tenant SFC infrastructure. The vIMS steady-state availability has been computed by choosing the number of call set-up requests (handled by the system for each operator with different performance requirements) as performance vector. Accordingly, given a service demand vector and a high system availability target, a parallel redundancy optimization problem has also been solved by a computationally efficient routine implementing MUGF approach, and the network configurations minimizing the cost expressed in terms of number of deployed nodes have been identified. Future work will be devoted to include in the model more sophisticated dependencies among nodes behavior, typically present in real world scenarios.

REFERENCES

- [1] European Telecommunications Standards Institute, “Network Functions Virtualisation: An Introduction, Benefits, Enablers, Challenges and Call for Action,” Oct, 2012 [Online]. Available: https://portal.etsi.org/nfv/nfv_white_paper.pdf, accessed: 2018-10-01.
- [2] D. Cotroneo, R. Natella, and S. Rosiello, “NFV-throttle: An overload control framework for network function virtualization,” *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 4, pp. 949–963, 2017.
- [3] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner, “OpenFlow: Enabling innovation in campus networks,” *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, 2008.
- [4] J. Matias, J. Garay, N. Toledo, J. Unzilla, and E. Jacob, “Toward an SDN-enabled NFV architecture,” *IEEE Commun. Mag.*, vol. 53, no. 4, pp. 187–193, 2015.
- [5] S. Taha Ali, V. Sivaraman, A. Radford, and S. Jha, “A survey of securing networks using Software Defined Networking,” *IEEE Trans. Rel.*, vol. 64, no. 3, pp. 1086–1097, 2015.
- [6] Internet Engineering Task Force (IETF), “Service Function Chaining Architecture,” Oct, 2015 [Online]. Available: <https://tools.ietf.org/html/rfc7665>, accessed: 2018-10-01.

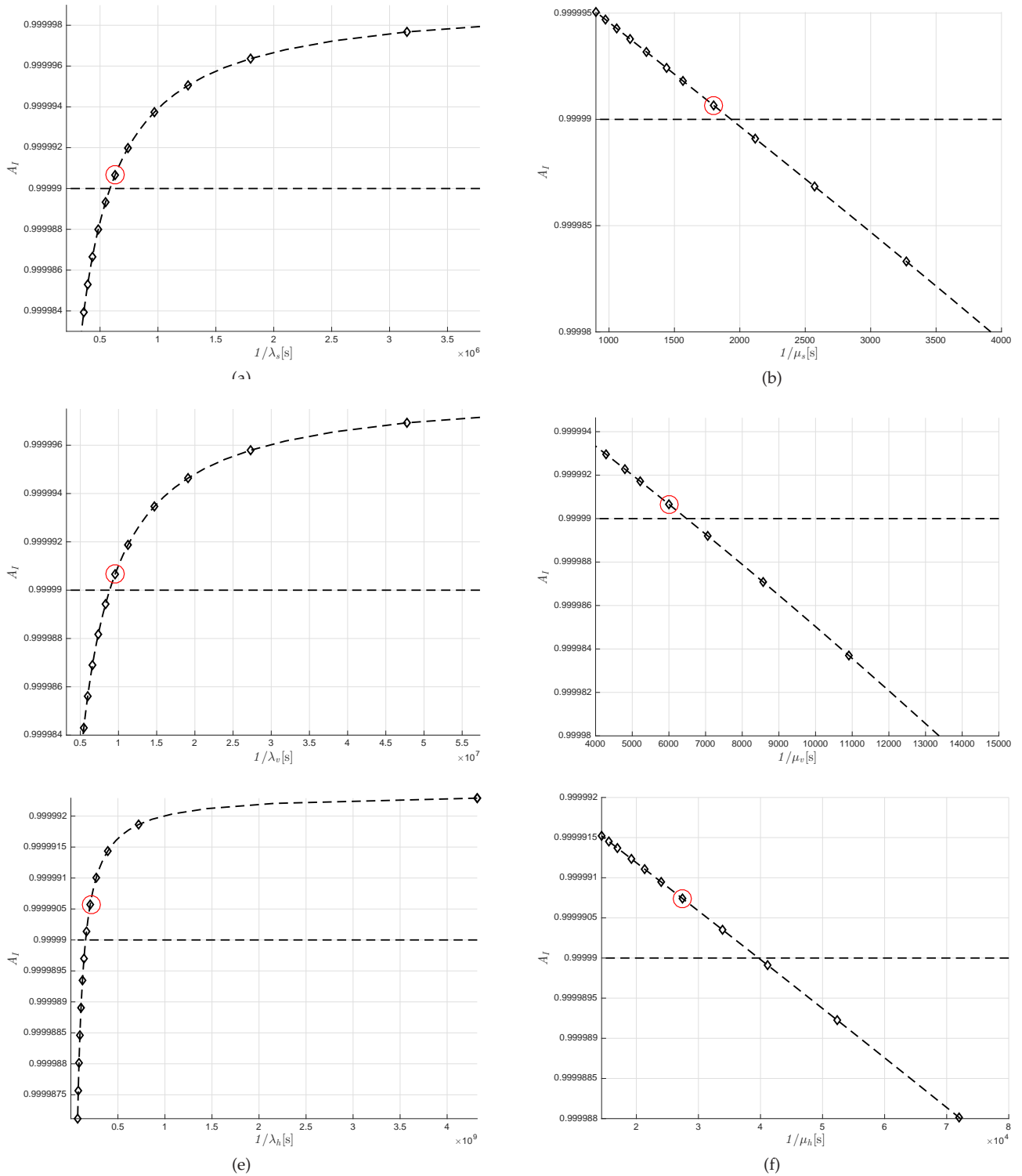


Fig. 9. Influence on the overall vIMS infrastructure of: software instances failure rate (a), software instances repair rate (b), virtualization layer failure rate (c), virtualization layer repair rate (d), hardware layer failure rate (e), hardware layer repair rate (f). Nominal values are circled in red.

- [7] Intel, "Evaluating Dynamic Service Function Chaining for the Gi-LAN," 2016 [Online]. Available: www.intel.com/content/dam/www/public/us/en/documents/white-papers/evaluating-dynamic-service-function-chaining-for-the-gilan-paper.pdf, accessed: 2018-10-01.
- [8] I. A. Ushakov, "A universal generating function," *Sov. J. Comput. Syst. Sci.*, vol. 24, no. 5, pp. 37–49, 1986.
- [9] A.S. Sendi, Y. Jarraya, M. Pourzandi, and M. Cheriet, "Efficient Provisioning of Security Service Function Chaining Using Network Security Defense Patterns," *IEEE Trans. Serv. Comput.*, vol. PP, no. 99, pp. 1–1, 2016.
- [10] J. Sun, G. Zhu, G. Sun, D. Liao, Y. Li, A.K. Sangaiah, M. Ramachandran, and V. Chang, "A Reliability-Aware Approach for Resource Efficient Virtual Network Function Deployment," *IEEE Access*, vol. 6, pp. 18238–18250, 2018.
- [11] S. Mehraghdam, M. Keller, and H. Karl, "Specifying and placing chains of virtual network functions," in *Proc. IEEE 3rd International Conference on Cloud Networking (CloudNet)*, pp. 7–13, 2014.
- [12] NEC Corporation, "NEC Virtualized Evolved Packet Core - vEPC," 2014 [Online]. Available: <https://networkbuilders.intel>.

- com/docs/vEPC_white_paper_w.cover_final.pdf, accessed: 2018-10-01.
- [13] G. Levitin and A. Lisnianski, *Multi-state system reliability: assessment, optimization and applications*. Singapore: World Scientific, 2003.
- [14] G. Levitin, *The Universal Generating Function in Reliability Analysis and Optimization (Springer Series in Reliability Engineering)*. Secaucus, NJ, USA: Springer-Verlag New York, Inc., 2005.
- [15] R. Ghosh, F. Longo, F. Frattini, S. Russo, and K. S. Trivedi, "Scalable analytics for iaas cloud availability," *IEEE Trans. Cloud Comput.*, vol. 2, no. 1, pp. 57–70, 2014.
- [16] J. Fan, C. Guan, Y. Zhao, and C. Qiao, "Availability-aware mapping of service function chains," in *IEEE INFOCOM 2017 - IEEE Conference on Computer Communications*, pp. 1–9, 2017.
- [17] J. Kong, I. Kim, X. Wang, Q. Zhang, H. C. Cankaya, W. Xie, T. Ikeuchi, and J. P. Jue, "Guaranteed-availability network function virtualization with network protection and VNF replication," in *GLOBECOM 2017 - 2017 IEEE Global Communications Conference*, pp. 1–6, 2017.
- [18] H. A. Alameddine, S. Ayoubi, and C. Assi, "An efficient survivable design with bandwidth guarantees for multi-tenant cloud networks," *IEEE Trans. Netw. Service Manag.*, vol. 14, no. 2, pp. 357–372, 2017.
- [19] S. Sebastio, R. Ghosh, and T. Mukherjee, "An availability analysis approach for deployment configurations of containers," *IEEE Trans. Serv. Comput.*, vol. PP, no. 99, pp. 1–1, 2018.
- [20] E. Sousa, F. Lins, E. Tavares, P. Cunha, and P. Maciel, "A modeling approach for cloud infrastructure planning considering dependability and cost requirements," *IEEE Trans. Syst., Man, Cybern., Syst.*, vol. 45, no. 4, pp. 549–558, 2015.
- [21] M. Guida, M. Longo, and F. Postiglione, "Performance evaluation of IMS-based core networks in presence of failures," in *Proc. IEEE GLOBECOM 2010*, (Miami, USA, Dec. 6-10 2010), pp. 1–5, 2010.
- [22] M. Guida, M. Longo, F. Postiglione, K. S. Trivedi, and X. Yin, "Semi-Markov models for performance evaluation of failure-prone IP multimedia subsystem core networks," *Proc. Inst. Mech. Eng. O J. Risk Reliab.*, vol. 3, pp. 290–301, 2013.
- [23] P. Sun, D. Wu, X. Qiu, L. Luo, and H. Li, "Performance analysis of cloud service considering reliability," in *2016 IEEE International Conference on Software Quality, Reliability and Security Companion (QRS-C)*, pp. 339–343, 2016.
- [24] M. Di Mauro, M. Longo, and F. Postiglione, "Performability evaluation of Software Defined Networking infrastructures," in *Proc. 11th Int. Conference on Performance Evaluation Methodologies and Tools, Valuetools'16*, (Taormina, Italy, October 25-28 2016), pp. 88–95, ACM, 2016.
- [25] European Telecommunications Standards Institute, "Universal Mobile Telecommunications System (UMTS); LTE; Network sharing; Architecture and functional description," Jan, 2013 [Online]. Available: https://www.etsi.org/deliver/etsi_ts/123200_123299/123251/11.04.00_60/ts_123251v110400p.pdf, accessed: 2018-10-01.
- [26] NEC White Paper, "RAN Sharing: NEC's approach towards Active Radio Access Network Sharing," 2013 [Online]. Available: http://www.ranworldevent.com/_media/PDFs/necs_approach_towards_active_ran_sharing.pdf, accessed: 2018-10-01.
- [27] Ericsson Review, "Virtualizing network services - the telecom cloud," 2014 [Online]. Available: <https://www.ericsson.com/assets/local/publications/ericsson-technology-review/docs/2014/er-telecom-cloud.pdf>, accessed: 2018-10-01.
- [28] K. S. Trivedi, *Probability and Statistics with Reliability, Queuing and Computer Science Applications*. New York: John Wiley & Sons, Inc., 2 ed., 2002.
- [29] G. Rubino and B. Sericola, "Sojourn times in finite markov processes," *Journal of Applied Probability*, vol. 26, no. 4, pp. 744–756, 1989.
- [30] I. A. Ushakov, "Optimal standby problems and a universal generating function," *Sov. J. Comput. Syst. Sci.*, vol. 25, no. 4, pp. 79–82, 1987.
- [31] G. Levitin, A. Lisnianski, H. Ben-Haim, and D. Elmakis, "Redundancy optimization for series-parallel multi-state systems," *IEEE Trans. Rel.*, vol. 47, no. 2, pp. 165–172, 1998.
- [32] J. Garcia-Reinoso, I. Vidal, P. Bellavista, I. Soto, and P. A. Aranda Gutierrez, "Transparent reallocation of control functions in IMS deployments," *IEEE Commun. Mag.*, vol. 54, no. 1, pp. 106–113, 2016.
- [33] R. d. S. Matos, P. R. M. Maciel, F. Machida, D. S. Kim, and K. S. Trivedi, "Sensitivity analysis of server virtualized system availability," *IEEE Trans. Rel.*, vol. 61, no. 4, pp. 994–1006, 2012.



Mario Di Mauro received the Laurea degree in electronic engineering from the University of Salerno (Italy) in 2005, the M.S. degree in networking from the University of L'Aquila (Italy) jointly with the Telecom Italia Centre in 2006, and the Ph.D. degree in information engineering in 2018 from University of Salerno. He was a Research Engineer with CoRiTel (Research Consortium on Telecommunications, led by Ericsson Italy) and then a Research Fellow with University of Salerno. He has authored several scientific papers, and holds a patent on a telecommunication aid for impaired people. His main fields of interest include: network availability, network security, data analysis for telecommunication infrastructures.



Maurizio Longo received the MSEE degree from Stanford University, CA, USA, in 1977, and the Laurea degree in electronic engineering from the University of Napoli (Italy), in 1972. He is currently with the University of Salerno (Italy) as Full Professor of Telecommunications and the Director of the CoRiTel (Research Consortium on Telecommunications) Lab., having also served as the Department Dean and the Chairman of the Graduate School of Information Engineering. He held academic positions also with the University Federico II (Napoli), the Parthenope University (Napoli), the University of Lecce and the Aeronautical Academy. In 1986 - 1987 and 1990, he was on leave with Stanford University, as a Foromez Fellow and as a NATO-CNR Senior Fellow. He has authored over 180 papers in international journals and conference proceedings, mainly in the fields of telecommunication networks and statistical signal processing.



Fabio Postiglione is currently an Assistant Professor of Applied Statistics with the Dept. of Information and Electrical Engineering and Applied Mathematics (DIEM) at University of Salerno (Italy). He received his Laurea degree (summa cum laude) in Electrical Engineering and his Ph.D. degree in Information Engineering from University of Salerno in 1999 and 2005, respectively. His main research interests include degradation analysis, lifetime estimation, reliability and availability evaluation of complex systems (telecommunication networks, fuel cells), Bayesian statistics and data analysis. He is/was involved in several EU-funded FP7/H2020 research projects on degradation analysis, lifetime estimation and diagnosis of fuel cells. He is member of the LIGO-VIRGO Collaboration devoted to gravitational waves observation on topics related to data analysis. He has authored over 100 papers, mainly published in international journals.