Service Function Chaining deployed in an NFV environment: an availability modeling

M. Di Mauro, M. Longo, F. Postiglione
Dept. of Information Engineering, Electrical Engineering
and Applied Mathematics (DIEM)
University of Salerno, Italy
{mdimauro,fpostiglione,longo}@unisa.it

G. Carullo, M. Tambasco
Research Consortium on Telecommunications
(CoRiTeL)
University of Salerno, Italy
{giuliana.carullo,marco.tambasco}@coritel.it

***Abstract*—Nowadays, network and telecommunication operators require flexible and dynamic models to deploy new services in a fast, reliable and cost saving way. The Service Function Chaining (SFC) design is particularly suited to meet such needs, especially in conjunction with the Network Function Virtualization (NFV) paradigm that adds a noteworthy elasticity during the SFC deployment phase. Accordingly, SFC is realized by means of a composition of Virtualized Network Functions (VNFs) aimed at providing some specific services. We consider, from an availability point of view, an SFC-based architecture with an aim to find out the optimal configuration guaranteeing the so-called "five nines" availability requirement, as demanded in the telecommunication systems. The availability analysis is carried out by exploiting a hierarchical model where a Reliability Block Diagram describes high level dependencies in the SFC implementation, while Stochastic Reward Nets are adopted to model the probabilistic behavior of single blocks. In particular, the SFC availability has been evaluated by performing a steady-state analysis, while a sensitivity analysis of some critical parameters allowed us to analyze in depth the whole system robustness.**

***Index Terms*—Network Function Virtualization, Service Function Chaining, Stochastic Reward Nets, Availability analysis.**

I. Introduction

The design of 5G network architectures, functions and protocols, is currently driven by high availability and reliability requirements expected to support mobile networks, applications and devices for the next generation services. The main challenge involving the network and telecommunication operators is to fulfill said requirements while mitigating the related costs. In this scenario, the Software Defined Networking (SDN) and Network Function Virtualization (NFV) paradigms appear the most suitable approaches aimed at exploiting the benefits drawing from the 5G adoption. SDN relies on the separation between network control plane and data forwarding plane by exploiting the OpenFlow protocol [1]. On the other hand, NFV has been conceived to fully exploit the advantages of virtualization mechanisms applied to the networking world by decoupling the network functionalities from underlying hardware, according to a Cloud Computing model. A valuable example of interworking between these paradigms is the Service Function Chain (SFC), namely an ordered list of network elements (e.g. routers, firewalls, etc.) linked together to provide a specific macro service. In this scenario, the network appliances can evolve towards Virtualized Network Functions (VNFs) by forming a so-called Forwarding Graph (VNF-FG) where an SDN controller can be used to dynamically address the VNF-FG itself on the basis of the inputs specified by the Orchestrator in the NFV architecture. Let us stress that, as the fault of a single piece of the chain could jeopardize the overall SFC functionality, the availability, i.e., the probability of a system being available when called upon for use, becomes a key issue for an accurate design and deployment. The main contribution of this paper is the availability evaluation of an SFC implementing a network service, modeled as a forwarding graph where two exemplary working conditions are considered. This kind of analysis has the goal of determining the optimal SFC configuration guaranteeing the so-called "five nines" availability requirement (no more than 5 minutes and 26 seconds downtime per year) as invoked in typical service level agreements of network operators. The availability analysis is carried out by exploiting a hierarchical model relying on two different formalisms: *(i)* the Reliability Block Diagrams (RBDs), a combinatorial model used to represent the high level setting of the SFC and *(ii)* the Stochastic Reward Networks (SRNs), a state-space model accounting for the probabilistic behavior of the underlying structure of single blocks. The rest of the paper is organized as follows. In Section II, some related works are presented, while in Section III more details about the reference architectural scenario are provided. Sections IV describes an availability model of the SFC scenario along with the adopted formalism. Section V reports the experimental results in a realistic scenario along with a sensitivity analysis of the most critical parameters. Section VI ends this paper by providing conclusions and considerations about future work.

II. RELATED WORK

The scientific community is devoting a noteworthy interest in reliability and availability issues related to the novel network and telecommunication infrastructures realized according to the SDN and NFV paradigms, intended as two sides of the same coin [2]. As a matter of fact, some guidelines about the application of reliability models to the new generation networks have been disclosed by European Telecommunications Standard Institute (ETSI) [3]. In this section, a non exhaustive *excursus* about the more relevant literature concerning the cited aspects is offered. An analytic framework for reliability evaluation of NFV deployment is proposed in [4] where the authors describe four algorithms to solve the Minimum Total Failure Removal (MTFR) problem. Being the virtualization concept a milestone for the 5G infrastructures, a pioneering work about the availability issues involving virtualized systems appears in [5], where a sensitivity analysis approach (based on the Markov reward models) to find the parameters deserving more attention for improving the availability of the system has been performed. A modeling strategy based on a hierarchical approach for cloud infrastructure planning appears in [6]; such a strategy allows to select the best cloud infrastructure according the dependability and cost requirements. In [7], the authors propose a reliability analysis of the SDN controller hosting a certain number of virtual telecom operator instances, by solving a redundancy optimization problem by the Universal Generating Function (UGF), a technique allowing to find a complex multi-state system performance distribution by using efficient algebraic procedures to combine performance distributions of systems. Moreover, the availability of the core element of the NFV architecture (implemented through the OpenStack platform), namely the Virtualized Infrastructure Manager (VIM), has been assessed in [8] where a SRN model for the VIM element has been proposed. Some backup strategies for VNF service chains along with algorithms for their resilient embedding in a data-center environment have been proposed in [9]. Again, high availability and service reliability issues are faced in [10] where a framework to guarantee service resilience along with efficient restoration procedures in carrier cloud environments has been introduced. In line with these recent trends, the principal contribution of the present paper is to characterize an NFV-based Service Function Chaining structure by mainly leveraging SRN, a state-space model adopted to perform SFC availability analysis. Due to the intrinsic system complexity, some crucial issues arise in operating with classical Markov chains, as the state space rapidly reaches intractable sizes. On the other hand, SRN provides a formalism able to reduce the model size, by capturing the dynamical behavior of the system.

III. The Service Function Chaining model

According to a general definition [11], an SFC is an ordered set of functions responsible for specific treatments applied to packet flows. A service function can operate at various layers of the TCP/IP protocol stack and, being a logical component, can be implemented by a virtual element or can be embedded in a physical network equipment. In the NFV context, the SFC can be easily interpreted as a VNF Forwarding Graph (VNF-FG), a set of VNFs conveniently traversed in order to offer certain services in a virtualized environment. In such a scenario, the SFC can be exploited in conjunction with some key elements of the network infrastructure [3] as *i)* the SDN controller useful to create and manage different paths traversing the SFC when some network criteria are matched and, *ii)* the Virtualized Infrastructure Manager (VIM) in charge of managing and deploying the virtual resources needed for a correct functioning of the VNFs composing the SFC. Such a flexibility allows network operators to provide, in a fast and flexible way, multiple solutions properly tailored to specific requirements [12]. Figure 1 depicts a common scenario with N interconnected VNFs where, for the sake of simplicity, the functionalities are grouped in two domains: the *Service Function Domain* and the *Control Domain*. The former comprises the set of VNFs that provide the specific network service. The latter refers to a part of management infrastructure in charge of controlling and monitoring the work of VNFs. In our work we focus on the Service Function Domain.

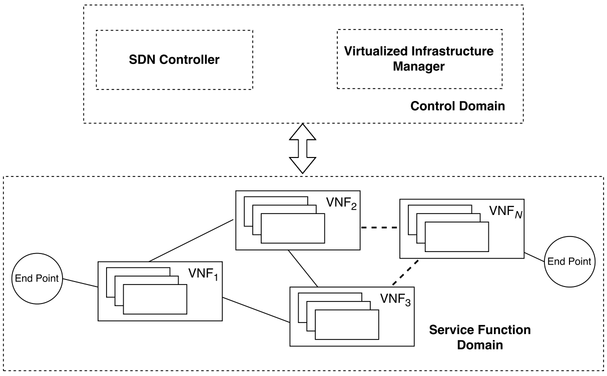

Fig. 1. Service Function Chain implementation in an NFV environment.

IV. SFC Availability Model

In this section, we provide an availability model of the Service Function Chain, remarking again that the modeling of Control Domain is out of our scope. For the sake of clarity, we first give an introductory explanation about the exploited methodologies and then we analyze the designed model. The adopted hierarchical approach relies on a two-level representation that mixes combinatorial models, such as RBDs [13], and state-space models as the SRNs [14]. The RBD model represents the system through a set of blocks and is able to catch working or failure conditions in terms of structural composition of the VNFs in the SFC, as exemplified in Fig. 2. Such a top-level representation admits a concise description of the system by referring to a series configuration of blocks, so that the SFC can be considered as fully working when every component block is working. The low-level representation relies on SRN as detailed in the following subsection.

A. Stochastic Reward Nets formalism

The SRN model stems from Markov Reward Model (MRM), which in turn extends the classical Continuous-Time Markov Chain (CTMC) by adding a reward rate to each state. A problem with MRMs and, in general, with state-space based models, is the huge growth of said space in real situations. SRNs achieve a more compact description by identifying regularities, i.e. repetitive structures in the underlying system, thus allowing an automated generation of the underlying MRMs [15]. An SRN describes a complex systems as a bipartite directed graph in which a *place* (represented

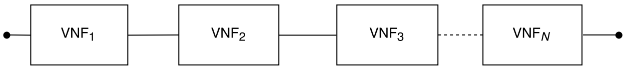

Fig. 2. RBD representation of a Service Function Chain.

by a circle) specifies a condition (e.g., the system is *up* or *down*), a *transition* (represented by a rectangle) denotes an action (e.g., the system crashes) and *arcs* are directed edges connecting places and transitions. The term stochastic alludes to the probabilistic delays introduced by transitions, since transition times are assumed as exponentially distributed. A *token* (represented as a number or depicted as a small circle in a place) indicates an holding condition and it is moved from a place to another when a transition is *fired*. In essence, the SRN model captures the dynamics of the system by evaluating the distribution of tokens as time elapses. Such a distribution is referred to as *marking* and indicates the possible assignment of tokens to all places within the underlying Petri Net.

The *reward function* $X(t)$ is a non-negative random process representing system conditions, whose value varies according the desired measure (dependability, performance, availability and so forth [16]). In case of availability assessment, $X(t)$ is typically defined as: $X(t) = 1$ if the system is working (up condition) at t, $X(t) = 0$ otherwise (down condition). Consequently, the instantaneous availability $A(t)$ can be evaluated as the expected reward function at time t, viz.

$$A(t) = Pr\{X(t) = 1\} = E(X(t)) = \sum_{i \in S} r_i p_i(t), \quad (1)$$

where S indicates the state space (in other words the set of markings in the SRN) that can be split in a subset of up states S_u (with reward rate $r_i = 1$) and a subset of down states S_d (with reward rate $r_i = 0$), while $p_i(t)$ represents the probability of the system being in state i.

The SRN of a single VNF is shown in Fig. 3 while the resulting SFC will be obtained at RBD level.

We propose to model a VNF as a three-layer structure with: *i)* an hardware layer (henceforth HW); *ii)* a software layer (henceforth SW) representing the appliance running on top of the VNF; *iii)* a Virtual Machine Monitor (henceforth VMM) or *hypervisor*, in charge of managing the virtual resources. Besides, a provisioning mechanism is assumed to manage hardware, software and virtual resources on an "as needed" basis by adding or removing VNFs.

In Fig. 3 we distinguish the following elements:

- *Places* (circles): P_{up} takes into account the working condition where hardware, software and virtual resources are fully working, and where the n inside represents the number of *tokens*, namely the number of initial working VNFs replicas. In this setting, the total number of working replicas accounts for two contributions: L replicas supposed to share a time-varying load, and M (extra) replicas added to match the high-availability requirement that is time-varying itself; thus we let $n = L + M$. When used, the notation $\#P_k$ indicates the number of tokens in the generic place P_k. Place P_L models the condition of having L replicas, where the token L inside indicates the initial number of needed L replicas. The number of tokens in P_p, instead, models the number of VNF replicas during the provisioning state. Places P_{fvmm} and P_{fsw} model a VMM or a software fault condition respectively, requiring a restoring procedure (e.g., a reboot); P_{fvmm1} and P_{fsw1} model two *vanishing conditions* indicating that a token gets immediately transferred to other places; place P_{fsw2} models a tough software fault condition where the intervention of a repairman is required aimed at returning to the working condition P_{up}; P_m models a *migration* condition indicating that critical faults (ascribed to HW or VMM) imply a migration, viz., the resource moving to another location without any state loss.
- *Timed Transitions* (unfilled rectangles): are indicated by T_x (x is the event associated to the timed transition); a timed transition is characterized by an exponentially distributed time with a "firing rate" as parameter. If the firing rate depends on the number of tokens in the starting place, a *Place Dependent Transition* (PDT) is needed, that is denoted by a symbol $\#$ appearing close to it; in our model all the PDTs depend on the starting place.
- *Immediate Transitions* (thin and filled rectangles): are indicated by t_y (y is the event associated to the immediate transition), and characterized by a zero transition time ("firing time"), and take into account instantaneous actions.

B. SRN evolutionary model

We now outline the evolution of the SRN of a VNF model under the assumption that the probability of failure during repair time is negligible. The provisioning and the deprovisioning phases have been characterized in terms of Scale-Out (S-O) and Scale-In (S-I) operations, respectively.

When an S-O operation is needed (i.e., adding replicas), the transition t_{so} is fired and a token enters place P_p. The *inhibitory arc* from P_p to t_{so} prevents multiple provisioning at the same time.

In P_p the replica is requested but not working yet until the transition T_p is fired and the token enters P_{up} place. It is worth noting that the provisioning time governing T_p, depends on the intermediate operations performed by the NFV orchestrator to make the replica be fully available, such as selecting virtual images or allocating and deploying resources.

When an S-I procedure is needed (i.e., removing replicas) the transition t_{si} is fired. The *inhibitory arc* from P_p to t_{si} prevents S-I operations during the provisioning phase. The guard functions g_1 and g_2 have been introduced with the following purpose: g_1 prevents S-I operations when the total number of needed replicas in the system, say N_{tot}, undergoes the number of tokens $L+M$, while g_2 prevents S-O operations when N_{tot} overcomes the same value, where N_{tot} = $\#P_p$ + $\#P_{up}$ + $\#P_{fvmm}$ + $\#P_{fsw}$ + $\#P_{fsw2}$ + $\#P_m$.

In the boxed submodel, we account for the process of adding or removing replicas L on behalf of T_{add} or T_{rem} transitions,

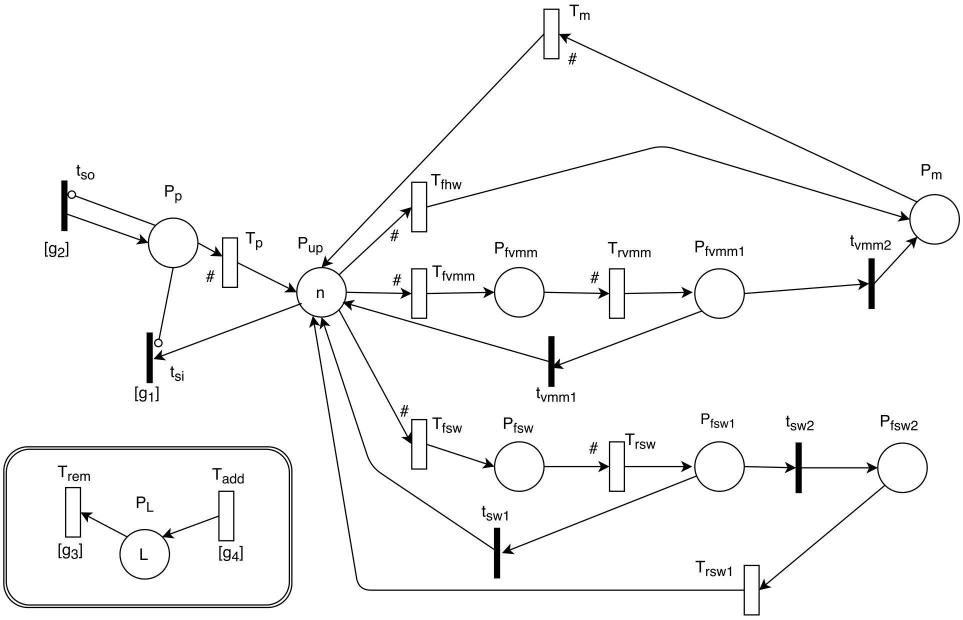

Fig. 3. SRN model of a single Virtualized Network Function.

respectively. Guard function g_3 prevents that the number of tokens in P_L overcomes the maximum number of replicas (say L_{max}) whereas guard function g_4 prevents that the number of tokens in P_L undergoes the minimum number of replicas (say L_{min}) needed for the system to work. All the guard functions are summarized in Table I

Consider an initial fully working system with n tokens in P_{up}: in case of an HW failure, the transition T_{fhw} is fired and a token is removed from the input place P_{up} and deposited in the output place P_m, where a migration governed by transition T_m is needed to return in P_{up}. If the VMM part fails, T_{fvmm} is fired and the token passes from P_{up} to P_{fvmm}; in order to recover the fault, the transition T_{rvmm} is fired and P_{fvmm1} is reached. In such a *vanishing* condition, the SRN does not spend any time but two alternatives are possible: *i)* the VMM soft restoring procedure succeeds with probability c_{vmm} (with c_{vmm} coverage factor for VMM) and the token comes back to P_{up} once t_{vmm1} is fired; *ii)* the procedure is unsuccessful with probability $(1 - c_{vmm})$ and the token is moved to P_m place once t_{vmm2} is fired.

In case of a SW fault, T_{fsw} is fired and the token is transferred from P_{up} to P_{fsw}; similar to the previous case, a soft restoring procedure starts and place P_{fsw1} is reached as T_{rsw} is fired. With probability c_{sw} the procedure is successful, and P_{up} is reached once transition t_{sw1} is fired; on the contrary, P_{fsw2} is reached with probability $(1 - c_{sw})$ once t_{sw2} is fired. In this case a summoned repairman is supposed to intervene in reason of specificity of software fault and, after repair, the token comes back to P_{up} once T_{rsw1} is fired. Recalling that a *marking* identifies the distribution of tokens in the various places of the SRN model, let r_i be the reward rate assigned to marking i and $p_i(t)$ the probability of SRN being in marking i at time t. Then, since the markings are mutually exclusive, the instantaneous availability $A(t)$ can be computed similar to (1) as the expected reward function at time t, viz.

$$A(t) = \sum_{i \in I} r_i p_i(t), \tag{2}$$

where I identifies the set of *tangible markings* (markings where no immediate transitions are enabled). The reward rate r_i associated with the tangible marking i is given by

$$r_i = \begin{cases} 1 & \text{if } (\#P_{up} \geq \#P_L) \\ 0 & \text{otherwise,} \end{cases}$$

specifying that the system is in a working state when the total number of working replicas (or tokens in P_{up}) is not less than the number of L replicas (or tokens in P_L). Accordingly, the steady-state availability for a single VNF can be obtained from (2) for long runs as $t \to \infty$ and can be expressed as:

$$A_{VNF} = \lim_{t \to +\infty} A(t) = \sum_{i \in I} r_i p_i, \tag{3}$$

where p_i is the steady-state probability given by $p_i = \lim_{t \to +\infty} p_i(t)$. In our representation, the SFC system is composed by N VNFs, each one described by the same SRN model, being connected in series according to the RBD representation in Fig. 2. Thus, the overall system availability can be expressed as

$$A_{SFC} = \prod_{j=1}^{N} A_{VNF}(j), \tag{4}$$

where A_{VNF} is derived by (3).

TABLE I
GUARD FUNCTIONS DEFINED ON THE SRN MODEL

Guard Function	Value
g_1	1 if $N_{tot} > L + M$, 0 otherwise
g_2	1 if $N_{tot} < L + M$, 0 otherwise
g_3	1 if $L < L_{max}$, 0 otherwise
g_4	1 if $L > L_{min}$, 0 otherwise

V. A NUMERICAL EXPERIMENT

In this section, the methods above discussed are applied toward obtaining numerical results in a realistic scenario by using SHARPE [17], a toolkit providing a powerful language and solution models for the analysis of reliability of complex systems. In Table II, numerical values derived from both technical literature (e.g. [5]) and telecommunication expertise are reported. The goal of the present analysis is to characterize the system in terms of number of *extra* replicas M satisfying the "five nines" availability requirement under two different operating conditions: c_1 and c_2, introduced to take into account different load conditions, for instance night and day. In condition c_1, the load is supposed to be shared among two replicas ($L_{c1} = 2$) whereas in condition c_2 the load is supposed to be shared among three replicas ($L_{c2} = 3$). In other words a *scaling procedure* is assumed, as typically adopted in the cloud environments, where the number of replicas varies accordingly to the load conditions, resulting in a time-varying need to meet the availability requirement. We want to remark that the number of conditions and respective VNF replicas considered in our setting are merely illustrative with no lack of generality. Let M_{c1} and M_{c2} be the number of extra replicas per VNF in c_1 and c_2 condition, respectively.

We distinguish three configurations (A, B, and C) representative of three possible deployments. Configuration A is characterized by 1 extra replica in c_1 ($M_{c1} = 1$) and 1 extra replica in c_2 ($M_{c1} = 2$). In configuration B we suppose 1 extra replica in c_1 and 2 extra replicas in c_2. Finally, the configuration C is distinguished by 2 extra replicas both for c_1 and for c_2 conditions. Figure 4 reports the main results for $N = 4$ VNFs where, for visualization comfort, we show the steady-state unavailability of the Service Function Chain structure ($1 - A_{SFC}$), by considering the three configurations. In configuration A, the system reaches an unacceptable availability outcome quantified as $A_{SFC} = 0.99998249$. Configuration B, instead, is characterized by a value of $A_{SFC} = 0.99999098$ that fulfills the demanded requirement. Finally, configuration C offers a slightly greater availability, with a value $A_{SFC} = 0.99999102$, but it is paid in the coin of deploying one more extra replica in configuration c_1. Therefore, the availability analysis in long runs reveals that the optimal configuration guaranteeing the "five nines" requirement with the minimal number of deployed extra replicas M, is the configuration B. The availability is also evaluated for $N = 3$ and $N = 5$.

The results are then summarized in Table III, where the first column indicates the number of considered VNFs in the Network Service, the second column specifies the configurations in terms of L and M replicas, and the third column shows the corresponding system availability. Table III reveals that, in case of $N = 3$, configuration B is the optimal one (similar considerations for $N = 4$ case hold). In the scenario with $N = 5$, none of the considered configurations satisfies the desired requirement, and additional replicas are required.

TABLE II
INPUT PARAMETERS FOR THE SRN REPRESENTING THE VNF

Parameter	Description	Value
$1/\lambda_{hw}$	mean time for hardware failure	60000 hours
$1/\lambda_{sw}$	mean time for software failure	3000 hours
$1/\lambda_{vmm}$	mean time for hypervisor failure	5000 hours
$1/\mu_{sw}$	mean time for fast software repair	7 minutes
$1/\mu_{sw1}$	mean time for tough software repair	2 hours
$1/\mu_{vmm}$	mean time for hypervisor repair	10 minutes
$1/\alpha_p$	mean time for provisioning	20 minutes
$1/\alpha_{mig}$	mean time for migration	20 minutes
$1/\alpha_s$	mean time for scaling (SI/SO) procedures	12 hours
c_{sw}	coverage factor for software repair	0.98
c_{vmm}	coverage factor for hypervisor repair	0.99

TABLE III
SUMMARY OF AVAILABILITY RESULTS FOR A NETWORK SERVICE WITH $N = 3, 4, 5$ DEPLOYED VNFS, AND VARIOUS (L, M) REPLICAS

VNFs (N)	Shared load replicas (L) and Extra replicas (M)	A_{SFC}
$N = 3$	$L_{c1} = 2,\ L_{c2} = 3,\ M_{c1} = 1,\ M_{c2} = 1\ \ (A)$	0.99998687
$N = 3$	$L_{c1} = 2,\ L_{c2} = 3,\ M_{c1} = 1,\ M_{c2} = 2\ \ (B)$	0.99999323
$N = 3$	$L_{c1} = 2,\ L_{c2} = 3,\ M_{c1} = 2,\ M_{c2} = 2\ \ (C)$	0.99999327
$N = 4$	$L_{c1} = 2,\ L_{c2} = 3,\ M_{c1} = 1,\ M_{c2} = 1\ \ (A)$	0.99998249
$N = 4$	$L_{c1} = 2,\ L_{c2} = 3,\ M_{c1} = 1,\ M_{c2} = 2\ \ (B)$	0.99999098
$N = 4$	$L_{c1} = 2,\ L_{c2} = 3,\ M_{c1} = 2,\ M_{c2} = 2\ \ (C)$	0.99999102
$N = 5$	$L_{c1} = 2,\ L_{c2} = 3,\ M_{c1} = 1,\ M_{c2} = 1\ \ (A)$	0.99997811
$N = 5$	$L_{c1} = 2,\ L_{c2} = 3,\ M_{c1} = 1,\ M_{c2} = 2\ \ (B)$	0.99998872
$N = 5$	$L_{c1} = 2,\ L_{c2} = 3,\ M_{c1} = 2,\ M_{c2} = 2\ \ (C)$	0.99998878

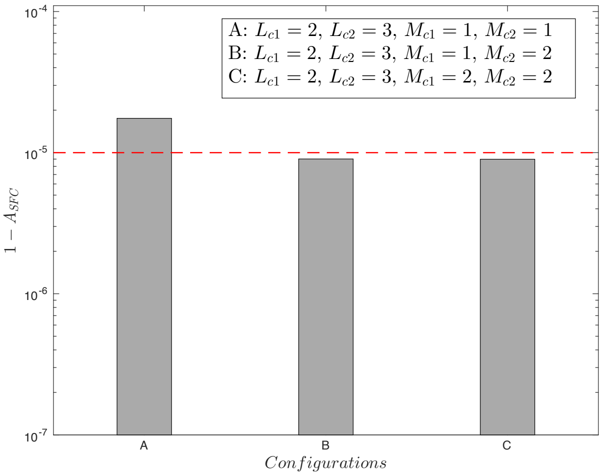

Fig. 4. Unavailability $1 - A_{SFC}$ of the system (with $N = 4$ VNFs) for three possible configurations (A, B and C). The red horizontal dashed line represents the "five nines" requirement ($1 - A_{SFC} = 10^{-5}$). Configuration B is the optimal one, by guaranteeing the "five nines" requirement.

Besides, a sensitivity analysis concerning the robustness of the system with respect to deviations of some parameters from their nominal values has been assessed. Our analysis is aimed

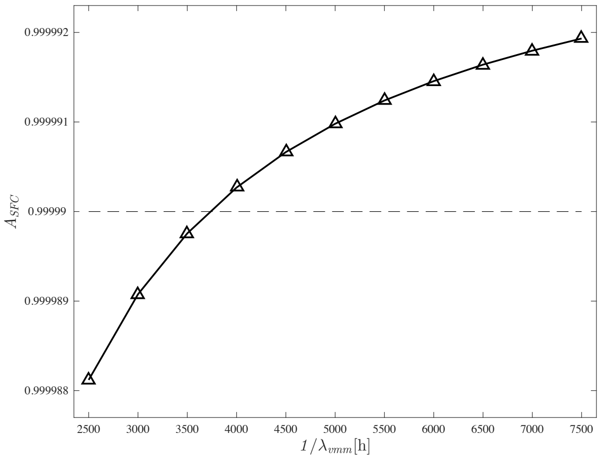

Fig. 5. Influence of the hypervisor failure rate on the overall SFC. The horizontal dashed line indicates the “five nines” availability requirement.

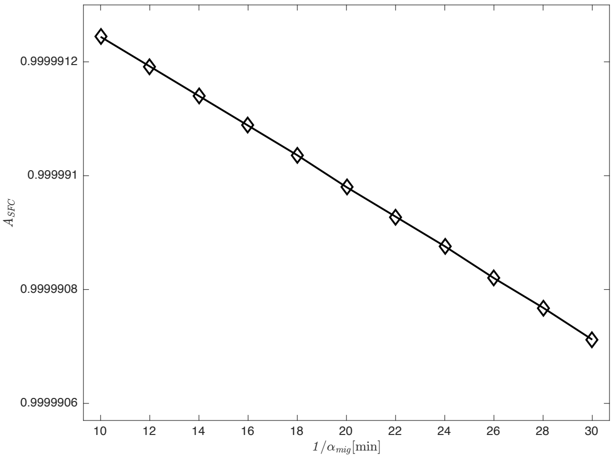

Fig. 6. Influence of the migration rate on the overall SFC.

at evaluating how the system availability, in case of configuration B, is affected by varying two exemplary parameters: *i)* λ_{vmm}, that is the hypervisor failure rate modeled by the transition T_{fvmm}, and *ii)* α_{mig}, the rate of transition T_m that governs the migration process activated when a hardware or hypervisor fault occur. Figure 5 shows that the working hypothesis of 5000 hours for the hypervisor mean time of failure can be reduced to 3700 hours with no side effects on the desired condition. Figure 6, instead, shows how the migration rate influences the steady-state availability. In this case, the nominal value of 20 min can be relaxed beyond the value of 30 min without side effects on high availability requirement.

VI. Conclusions

Recently, we face a growing demand by telecommunication and network operators for methods and models that allow speedy and inexpensive deployment of innovative services. In this context, Service Function Chaining (SFC) is able to satisfy such requirements by benefiting from Network Function Virtualization (NFV), the cutting-edge network paradigm aiming to speed up the deployment phase of network infrastructures. In this paper we characterize an SFC implementation based on NFV, by exploiting a two-level hierarchical model that combines Reliability Block Diagrams and Stochastic Reward Nets approaches. Such a modeling approach allows to compute the steady-state availability of the system to find out the optimal SFC configuration that guarantees the “five nines” availability requirement of different exemplary working conditions. Besides, the system robustness with regard to variations of two critical parameters has been evaluated. Future work will be devoted to the analysis of other failure mechanisms, such as the concurrent failures of VNFs replicas.

References

[1] N. McKeown, T. Anderson, H. Balakrishnan, G. Parulkar, L. Peterson, J. Rexford, S. Shenker, and J. Turner. Openflow: Enabling innovation in campus networks. *CCR SIGCOMM*, 38(2):69–74, 2008.

[2] S.T. Ali, V. Sivaraman, A. Radford, and S. Jha. A survey of securing networks using Software Defined Networking. *IEEE Transactions on Reliability*, 64(3):1086–1097, 2015.

[3] (ETSI). Network Functions Virtualisation (NFV) reliability; report on models and features for end-to-end reliability. Technical report, 2016.

[4] J. Liu, Z. Jiang, N. Kato, O. Akashi, and A. Takahara. Reliability evaluation for NFV deployment of future mobile broadband networks. *IEEE Wireless Communications*, 23(3):90–96, 2016.

[5] R. De S. Matos, P. Maciel, F. Machida, K. Dong Seong, and K. Trivedi. Sensitivity analysis of server virtualized system availability. *IEEE Transactions on Reliability*, 61(4):994–1006, 2012.

[6] E. Sousa, F. Lins, E. Tavares, P. Cunha, and P. Maciel. A modeling approach for Cloud infrastructure planning considering dependability and cost requirements. *IEEE Transactions on Systems, Man, and Cybernetics*, 45(4):549–558, 2015.

[7] M. Di Mauro, F. Postiglione, and M. Longo. Reliability analysis of the controller architecture in Software Defined Networks. In L. Podofillini, B. Sudret, E. Stojadinovic, B. Zio, and W. Kröger, editors, *Safety and Reliability of Complex Engineered Systems*, pages 1503–1510. Taylor & Francis Group, 2015.

[8] M. Di Mauro, F. Postiglione, M. Longo, R. Restaino, and M. Tambasco. Availability evaluation of the virtualized infrastructure manager in Network Function Virtualization environments. In L. Walls, M. Revie, and T. Bedford, editors, *Risk, Reliability and Safety: Innovating Theory and Practice*, pages 2591–2596. Taylor & Francis Group, 2017.

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

[10] T. Taleb, A. Ksentini, and B. Sericola. On service resilience in cloud-native 5g mobile systems. *IEEE Journal on Selected Areas in Communications*, 34(3):483–496, 2016.

[11] Service Function Chaining (SFC) Architecture. https://tools.ietf.org/html/rfc7665/.

[12] NGMN Alliance. 5G white paper. Technical report, 2015.

[13] W. Kuo and Z. Ming. *Optimal Reliability Modeling: Principles and Applications*. Wiley, 2002.

[14] J. K. Muppala, G. Ciardo, and K.S. Trivedi. Stochastic Reward Nets for reliability prediction. In *Communications in Reliability, Maintainability and Serviceability*, pages 9–20, 1994.

[15] G. Bolch, S. Greiner, H. De Meer, and K.S. Trivedi. *Queueing Networks and Markov Chains: Modeling and Performance Evaluation with Computer Science Applications*. Wiley-Interscience, New York, NY, USA, 1998.

[16] J.K. Muppala, M. Malhotra, and K.S. Trivedi. *Markov Dependability Models of Complex Systems: Analysis Techniques*. Springer Berlin Heidelberg, 1996.

[17] Robin A. Sahner and K.S. Trivedi. Reliability modeling using SHARPE. *IEEE Transactions on Reliability*, 36(2):186–193, 1987.