

Automated Generation of Availability Models for SFCs: The case of Virtualized IP Multimedia Subsystem

Mario Di Mauro¹, Giovanni Galatro¹, Maurizio Longo¹, Arcangelo Palma¹, Fabio Postiglione¹, Marco Tambasco^{1,2}

¹*Department of Information and Electrical Engineering and Applied Mathematics (DIEM)*

University of Salerno (Italy)

{mdimauro,longo,fpostiglione}@unisa.it, {g.galatro1,a.palma27}@studenti.unisa.it

² *Co.Ri.Tel. - Research Consortium on Telecommunications*

Via Giovanni Paolo II, Fisciano (SA), Italy

marco.tambasco@coritel.it

Abstract—The reputation of network providers strongly depends on their ability to guarantee high performance levels of virtualized infrastructures, and to maintain strict Quality-of-Service (QoS) requirements, thus, the concept of “five nines” or high availability (HA) is critical. It means that, on average, the continuity of a provided service cannot be violated for more than about five minutes per year. In this direction, we propose a framework useful to design and model, from a HA perspective, the Service Function Chains (SFCs) whose chained software logic is embodied in many softwarized telco infrastructures (e.g. IP Multimedia Subsystem elected here as a representative use case). The proposed framework interacts with TimeNET tool, and offers interesting functionalities such as: *i*) generating stochastic models of SFCs based on the SRN (Stochastic Reward Nets) formalism; *ii*) deploying network scenarios via drag-and-drop operations for basic users, or modifying the underlying SRN models for advanced users; *iii*) setting a variety of parameters (mean-time-to-failure/repair, software/hardware specs, redundancy, etc.); *iv*) presenting availability results in tabular and/or graphical forms.

Index Terms—Availability model generation, Stochastic Reward Nets, Redundancy Optimization.

I. INTRODUCTION

Novel paradigms such as Network Function Virtualization (NFV) pose new challenges in terms of Service Level Agreements (SLAs) involving availability issues. As a matter of fact, the well assessed industry standard for carrier grade service availability [1]–[3] imposes the so-called “five nines” requirement, establishing that a telco system (even softwarized) could tolerate a maximum downtime of 5 minutes and 26 seconds per year. Among the impacted infrastructures, we have to certainly include the virtualized IP Multimedia Subsystem (vIMS) which plays a central role within the 5G domain for the convergence of telecommunication systems [4]. In practice, the vIMS can be considered a realization of a Service Function Chain (SFC), since it allows to build a specific service when the composing virtualized nodes (each one representing a

VNF) are traversed in a predetermined order [5]–[8]. From an availability viewpoint, one can interpret a vIMS as an ensemble of subsystems arranged in series/parallel configurations, where the series refers to the particular chained structure, and the parallel presumes the presence of redundant elements for high availability purposes. Our main idea is to reduce the effort needed to model an SFC by designing a dedicated model-driven framework. Such a framework is helpful to find out the optimal SFC (the optimal vIMS in our use case) which satisfies a desired availability target (e.g. “five nines”) with the minimum number of redundant resources. Precisely, the proposed tool allows to: *i*) handle an automated generation of stochastic models on the basis of some input parameters (Mean Time to Failure - MTTF, Mean Time to Repair - MTTR, etc.); *ii*) manage the transformation process in charge of translating high-level user specifications into templates feeding TimeNET, a powerful tool for Petri nets evaluation (freely downloadable from [9]); *iii*) compute the availability of the whole vIMS and of the component subsystems. The choice of stochastic models (in particular Stochastic Reward Nets - SRN) is in line with some recent and affine literature (see e.g. [10], [11]) and relies on the ability of counteract the state space explosion arising with classic Continuous-Time Markov Chain models when considering complex systems. Moreover, a Reliability Block Diagram (RBD) representation is exploited to describe the resulting series/parallel configurations.

The paper presents the following structure: Section II advances a list of related works where similar approaches have been followed; in Section III we propose a brief recap of IMS characteristics by focusing on a virtualized deployment (vIMS), and where a three-layered structure of each vIMS node is presented; Section IV is aimed at providing some details about the stochastic framework used for the availability modeling and evaluation, namely the SRN; Section V provides a description of the realized tool according a workflow perspective; finally, conclusions are advanced in Section VI.

II. RELATED WORK

The interest in affording availability issues related to modern network infrastructures is becoming more and more attractive in the field of service management [12]–[14]. Many attempts have been made in proposing tools, models, and techniques to cope with these intriguing aspects, as highlighted in the following bibliographic excursus. Authors in [15] advance a procedure in charge of automating the availability analysis of a media streaming application deployed on top of a middleware. The project relies on the availability management framework (AMF), and is more suitable for standard network architectures. By contrast, our tool has the flexibility of managing virtualized deployments which support the challenging design of nested structures.

Other works are instead focused on the proposal of domain-specific languages (DSL) aimed at supporting the workflow underlying the performability models generation. It is case of [16] and [17], where the automated models composition is based on the stochastic activity networks and stochastic Petri networks formalisms, respectively. In such cases, the attention is more paid to the language semantic rather than to implement to implement dedicated functionalities (e.g. the possibility of setting failure/repair parameters of a network system as in our case). Similarly, authors in [18] start from a high-level system description through UML language and then use SHARPE [19] for an availability evaluation, where, differently from our case, the interconnections have been neglected.

Yet based on the automated generation of availability models is the work in [20], where the introduced framework is specifically tailored on the wireless sensor networks (WSNs) and pertinent issues (e.g. routing protocols, schedule maintenance, battery exhaustion, etc.). Even in this case, the work does not address the problem of nested deployments (e.g. virtualized WSNs) which is crucial in our setting.

Generally speaking, our proposal espouses the availability model generation approach adopted in the aforementioned works, but, noteworthy differences arise. First, the advanced tool is specifically designed to cope with the availability of softwarized architectures exhibiting a chained structure (as almost always occurs in networked systems), and where each node is typically made of three nested layers: hardware, hypervisor, and software. Then, a high degree of customization allows to set/tune a variety of parameters ranging from adjusting MTTF/MTTR to defining different models of hardware, hypervisors, software each one with its own specs. Finally, a flexibility depending on the user expertise permits to compose evolved network scenarios through comfortable drag-and-drop operations (basic users), or to intervene on the underlying Stochastic Reward Networks to modify some structural properties of the model (advanced users). As far as we know, the proposed automated tool represents the first attempt in the field of availability management applied to softwarized infrastructures.

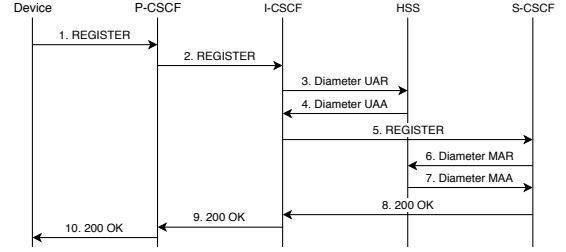


Fig. 1: Simplified representation of a Registration procedure within IMS domain.

III. AN EXEMPLARY SFC: THE VIRTUALIZED IP MULTIMEDIA SUBSYSTEM (VIMS)

IP Multimedia Subsystem was conceived as a framework in charge of providing access to multimedia IP-based services with guaranteed QoS. Such services (enriched communications, HD voice, etc.) are managed through SIP protocol, whose messages are exchanged among Call Session Control Function (CSCF) nodes: *Proxy* CSCF (P-CSCF), a SIP proxy allowing user devices to access the IMS domain; *Interrogating* CSCF (I-CSCF), a forwarder of SIP requests among nodes; *Serving* CSCF (S-CSCF), a node equipped with SIP routing functionalities. Finally, Home Subscriber Server (HSS) is an advanced database in charge of managing users' profiling and authorization services.

Usually, the signaling flow within IMS follows a predefined path. The Registration procedure shown in Figure 1 represents a classic example: the user device initiates the session by sending a REGISTER message to P-CSCF (1) aimed at getting the access to IMS domain; the REGISTER message is passed to I-CSCF (2) which obtains from the HSS the address of S-CSCF node in charge of managing the signaling flow. To perform such a procedure, two messages are involved: User Authentication Request (UAR) (3), and User Authentication Answer (UAA) (4). Once REGISTER message is transferred to the S-CSCF (5), a query containing information about user profile is passed to HSS through Message Authentication Request (MAR) (6) and Message Authentication Answer (MAA) (7). In case of a successful procedure ending, S-CSCF back-propagates a 200 OK message to user device (8), (9), (10).

It is easy to recognize that IMS nodes are typically traversed in a chained order, thus realizing an SFC-like arrangement. Moreover, each node is suitable to be virtualized, as often occurs within novel telco systems (see [21], [22]), resulting in a three-layered virtual node architecture composed by:

- Hardware (HW): represents the ensemble of physical parts (CPU, RAM, Hard Disk, etc.) that are often deployed in dedicated server farms;
- Hypervisor (HYP): the abstraction level in charge of optimizing the resources used by underlying software applications;
- Virtualized Network Function (VNF): represents the application which realizes a specific software functionality (P-CSCF, HSS, etc.);

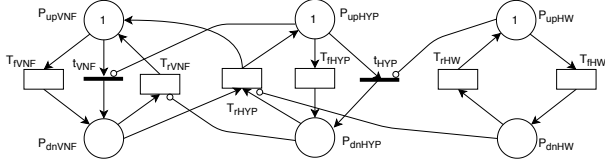


Fig. 2: SRN-based model representative of a generic vIMS node, according to the three-level structure.

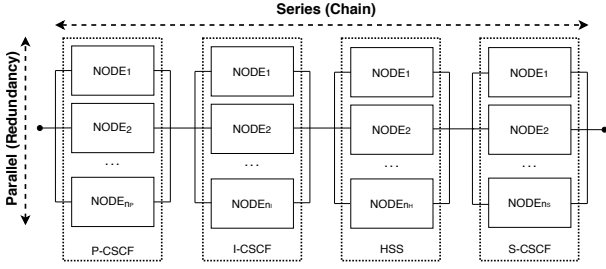


Fig. 3: RBD model for series/parallel representation of vIMS.

For the sake of brevity, in the rest of paper we shall indicate by vIMS the virtualized IMS infrastructure composed by nodes deployed according the described three-layered model.

IV. THE STOCHASTIC REWARD NET FORMALISM FOR vIMS AVAILABILITY EVALUATION

Before providing technical details about the realized framework, it is useful to anticipate some insights about the stochastic formalism adopted for the availability evaluation, namely the SRN [23]. Such a space state-based formalism is adopted to model, from a stochastic viewpoint, the failure/repair interactions occurring among single layers of a node. It admits a comfortable graphical representation with: *Places* (circles) which denote a condition (e.g. up/down) of a layer, and whose numbers inside are called *tokens* and are associated to a particular holding condition; *timed transitions* (unfilled and thick rectangles) which denote actions (e.g. a layer crashing) with exponentially distributed times (a common assumption in this field, see e.g. [24], [25]); *immediate transitions* (filled and thin rectangles) which designate quasi-instantaneous events. An SRN-based model of a generic vIMS node is depicted in Fig. 2, where each layer is supposed to be working since there is a token in each “up” place (P_{upVNF} , P_{upHYP} , and P_{upHW}). For instance, if the hypervisor fails, the transition T_{fHYP} is *fired* and token lying in P_{upHYP} will be transferred into P_{dnHYP} place. At the same time, t_{VNF} (connected with P_{upHYP} by a so-called *inhibitory arc*) is fired, thus provoking a VNF failure (token transferred from P_{upVNF} to P_{dnVNF}) as well, since it lies on the hypervisor layer. The whole SRN solution is equivalent to the evaluation of a *reward function*, a non-negative random process which can be mapped with the availability metric (see [23]). Precisely, given $Y(t)$ the reward function which amounts to 1 when the system is working at time t , and to 0 otherwise,

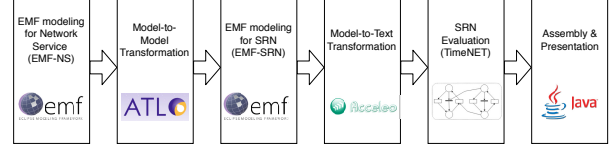


Fig. 4: Workflow of the realized tool.

it is possible to derive the instantaneous availability as

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

where: r_i (a.k.a. *reward rate*) is the value of $Y(t)$ across a certain distribution of tokens i (a.k.a. *marking*), $p_i(t)$ is the pertinent probability, and S is the set of markings. Since we are interested in the SRN steady-state evaluation, it can be obtained by (1) as follows:

$$A = \lim_{t \rightarrow +\infty} A(t) = \sum_{i \in S} r_i \cdot p_i, \quad (2)$$

where p_i represents the steady-state probability given by $p_i = \lim_{t \rightarrow +\infty} p_i(t)$. In summary, the SRN formalism is useful to model a single (three-layered) vIMS node in terms of failure and repair events. After this stage, an RBD representation is adopted to “compose” the whole SFC through series (chained structure) and parallel (for redundancy) nodes, as represented in Fig. 3. It is also useful to remark that, in our tool, SRN model of a vIMS node is transparent to a basic user, whereas it is accessible and adjustable to the advanced user, according to her/his specific needs.

V. SFC MODEL GENERATION

The proposed tool is realized from-the-scratch, and relies onto Eclipse Modeling Framework (EMF) [26] which provides runtime support to produce Java code starting from a meta model. Moreover, a specific plug-in for invoking TimeNET tool has been realized in order to evaluate SRN models. From a general perspective, the workflow representative of the realized tool is shown in Fig. 4, where the composing modules are described in the following subsections.

A. EMF-NS

This module allows to model, through EMF, the “skeleton” of an SFC composed of at least two VNFs connected in series (often known as Network Service - NS). By means of easy drag-and-drop operations, any user (basic or advanced) can setup the desired chain. At this stage, users can set system parameters such as: MTTF/MTTR, maximum number of replicas to use for redundancy, k -out-of- n values, and others.

B. Model-to-Model Transformation

In composing an NS, each single VNF needs to be mapped into a corresponding SRN. Such an operation is performed via Atlas Transformation Language (ATL), a model-to-model transformation language often adopted in the Model-Driven

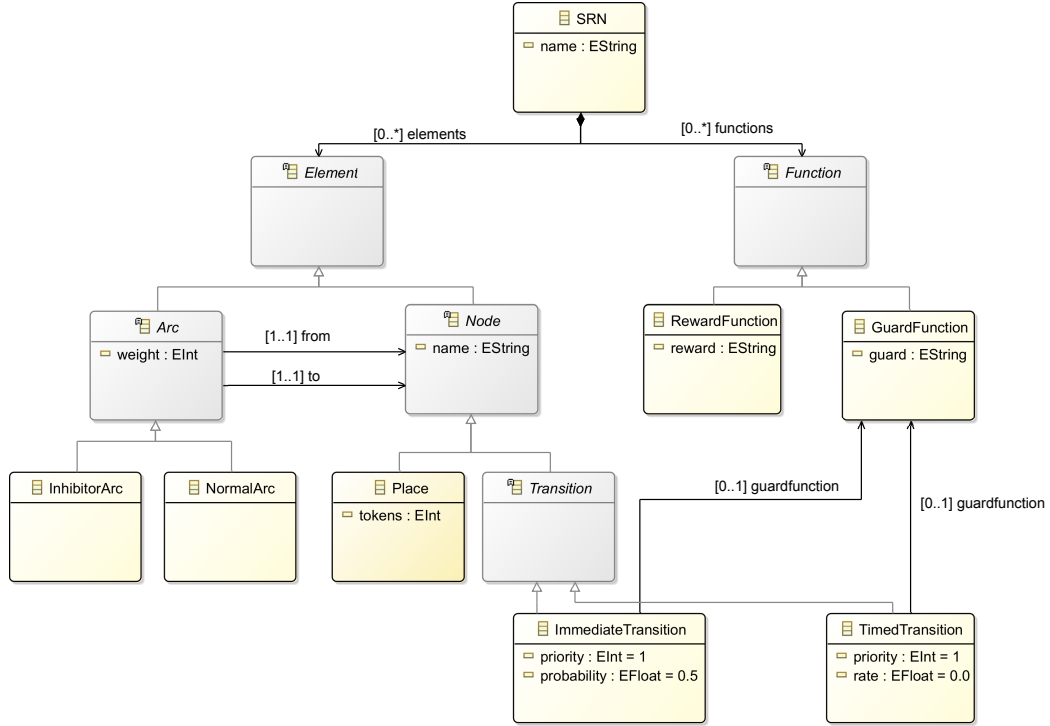


Fig. 5: UML class diagram (EMF-based) of the SRN depicted in Fig. 2.

Engineering (MDE), which allows to produce target models from a set of source models by means of specific semantic rules that we have designed and customized for our domain.

C. EMF-SRN

This module embeds a meta model representative of the SRN depicted in Fig. 2, and is based on the definition of classes and attributes according the standard description provided by the UML class diagram, as specified in Fig. 5. The root class is *SRN* (on top of Fig. 5) having an attribute *name* with type *EString*. Classes *Element* and *Function* are abstract (grey background) and can be specialized through *Arc/Node* and *RewardFunction/GuardFunction*, respectively. Classes such as *InhibitorArc* and *NormalArc* inherit functionalities from *Arc* class.

D. Model-to-Text Transformation

This module operates a transformation from a model to text by using *Acceleo* [28], a tool adhering to MOFM2T (Meta-Object Facility Model-to-Text) standard. In particular, the meta model specs are translated into textual code readable by *TimeNET*¹.

E. SRN Evaluation

TimeNET [29] is a tool adopted for the modeling and analysis of stochastic Petri nets, including SRNs. We have

¹This module is fully customizable if one is interested to interact with different availability evaluation tools.

designed a wrapper which allows to feed *TimeNET* with textual information embedding system parameters (MTTF/MTTR etc.) extracted during the EMF-SRN step. In practice, the invoked tool evaluates the steady-state availability associated to the SRN representative of a vIMS node by solving (2).

F. Assembly & Presentation

This module has a double function: *i*) it is in charge of evaluating the availability of the whole SFC starting from the steady state availability of a single node expressed by (2) (Assembly); *ii*) it proposes to the final user results either in tabular and in graphical forms with the support of Java-based utilities (Presentation). As regards the Assembly sub-module, a clarification is needed. Actually, *TimeNET* does not provide a simple way to evaluate the availability of a system made of components arranged in series (vIMS nodes) and in parallel (redundant replicas of vIMS nodes). Accordingly, we move such functionalities into the Assembly sub-module which, starting from the availability value associated to each SRN (namely, each vIMS node), builds an RBD series/parallel representation of the whole vIMS and evaluates the final availability. Remarkably, the number of resulting series/parallel configurations can be combinatorial since each vIMS node can have a different redundancy degree, and, then, a different number of redundant replicas. Finally, the overall availability of the vIMS is quickly obtained by multiplying single availabilities derived from (2). Starting from the obtained configurations, the Presentation sub-module is in charge of

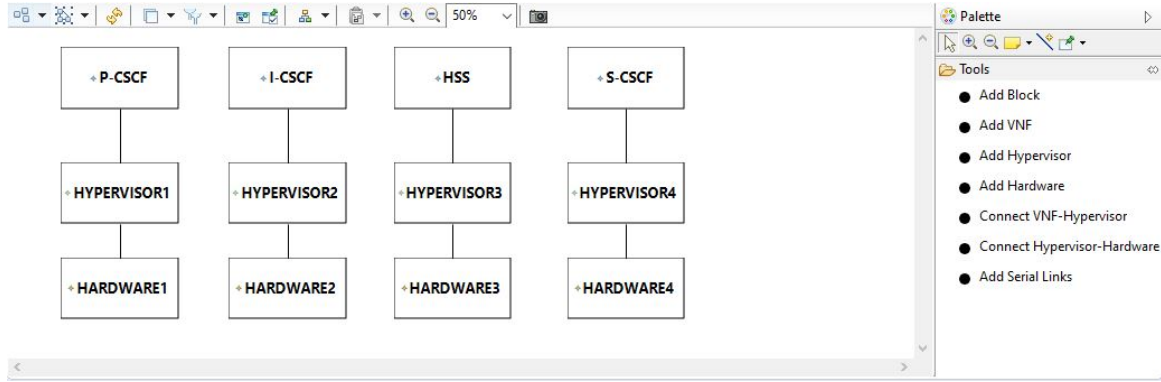


Fig. 6: Building an SFC (virtualized IMS in our example) via drag-and-drop operations.

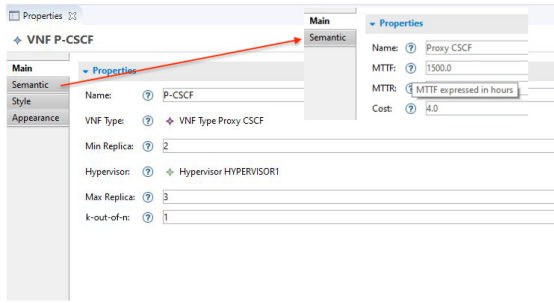


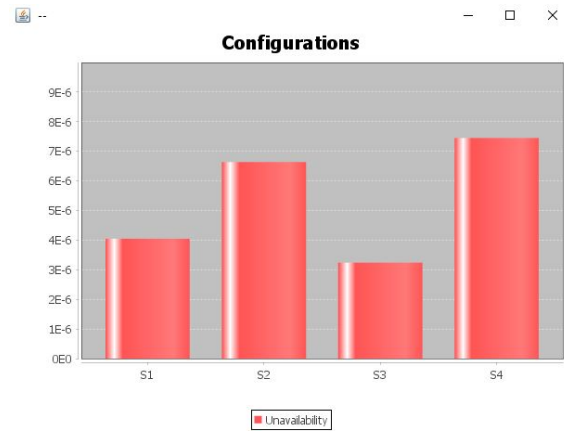
Fig. 7: Properties setting.

performing a user-guided exhaustive search to propose just those vIMS configurations whose availability value satisfies a certain requirement (e.g. “five nines”).

G. Graphical Appearance

The realized tool is equipped with a GUI for either basic or advanced users. Figure 6 shows the main screen where, through drag-and-drop operations, a user can build a three-layered node and connect via links the nodes to form an SFC. It is interesting to note that there is the possibility of customizing each layer by accessing to the property tab (see, for instance, Fig. 7). Here it is possible to set/modify some parameters of each layer. A generic user is authorized to access the following fields:

- *Name*: name of the VNF;
- *VNF Type*: the type of pre-charged VNFs (e.g. P-CSCF, HSS, etc.);
- *Min Replica*: the minimum number of node replicas to deploy. Deploying more replicas implies more redundancy (and, hence, availability) but higher costs;
- *Max Replica*: the maximum number of node replicas to deploy;
- *Hypervisor*: the Hypervisor type on which deploy the chosen VNF;
- *k-out-of-n*: this parameter is set when a node is required to work if and only if at least k of n replicas work.



(a) Graphical Output.

Configuration ID	Configuration Type	Availability
S1	[P-CSCF112, I-CSCF112, HSS124, S-CSCF213]	0.9999959523350834
S2	[P-CSCF112, I-CSCF113, HSS123, S-CSCF213]	0.9999933601841544
S3	[P-CSCF112, I-CSCF113, HSS124, S-CSCF213]	0.9999967604622317
S4	[P-CSCF112, I-CSCF112, HSS123, S-CSCF213]	0.9999925520597545

(b) Tabular Output.

Fig. 8: Outputs produced by the tool.

By contrast, Semantic Tab (see red arrow in Fig. 7) allows to access some layer parameters such as MTTF, MTTR, Layer Cost. From another panel (not shown here) it is possible to load templates of various layers. This is because a user can be interested in differentiating, for example, Hypervisor types (e.g. VMware, Virtualbox, etc.) on the basis of MTTF/MTTR and other specific parameters. Again, a dedicated panel (not shown here) reports the underlying SRN as presented in Fig. 2, so that an advanced user can modify it.

H. Outcomes

Let us now consider the results shown by the tool. Once a user has built the desired SFC (a vIMS in our example), the tool evaluates the availability of all configurations which, as remarked before, are combinatorial since each vIMS node

can have a different number of replicas lying in the interval $[Min\ Replica, Max\ Replica]$. Figure 8 shows the visual outputs produced by the tool where the user decided to visualize only a limited set of configuration satisfying the HA requirement. Precisely, Fig. 8(a) reports a bar graph of the steady-state Unavailability (1-Availability) of 4 exemplary configurations ($S_1, \dots, S_4$).

More details are shown in the corresponding tabular output (see Fig. 8(b)), where the exact value of availability is shown in the third column of the table, whereas the composition of configurations is reported in the second column (Configuration Type). For instance, the notation in the form HSS|2|3 indicates that virtualized HSS node has been considered with $k = 2$ and $n = 3$.

VI. CONCLUSIONS

In this work we introduced a tool useful to manage the availability of service chains, where the example of virtualized IP Multimedia Subsystem (vIMS) has been tackled. Its design adheres to the principles of Model-driven engineering (MDE) which focuses on the creation of conceptual models underlying the SFC application domain. Our framework can be used in the field of network management to design SFC structures compliant to the “five nines” availability requirements (max tolerated downtime per year of 5.26 seconds). The tool allows to generate probabilistic models of service chains by using the Stochastic Reward Net (SRN) formalism whose availability evaluation is performed by TimeNET. By means of a comfortable GUI allowing drag-and-drop operations it is possible to compose SFC-like scenarios (vIMS in our example) and to set a number of system specs/parameters (type of hypervisor, MTTF/MTTR, number of replicas etc.). An advanced user can also manipulate the underlying SRN (representative of a vIMS node) to modify its structural properties. We advise that the proposed tool is still under refinement since it requires some optimization to be released and then comfortably exploited by less experienced users.

REFERENCES

- [1] Implement a high availability architecture. Available online: https://www.ibm.com/garage/method/practices/manage/practice_high_availability/.
- [2] Delivering High Availability in Carrier Grade NFV Infrastructures. Available online: <https://www.vmware.com/files/pdf/techpaper/vmware-vcloud-nfv-high-availability.pdf>.
- [3] Cisco NFV infrastructure. Available online: <https://www.cisco.com/c/dam/en/us/solutions/collateral/service-provider/network-functions-virtualization-nfv-infrastructure/cisco-nfv-infrastructure-aag.pdf>.
- [4] M. Di Mauro, M. Longo, and F. Postiglione, “Availability Evaluation of Multi-tenant Service Function Chaining Infrastructures by Multidimensional Universal Generating Function,” *IEEE Trans. Serv. Comput.*, DOI: 10.1109/TSC.2018.2885748, 2018.
- [5] G. Davoli, W. Cerroni, C. Contoli, F. Foresta, F. Callegati, “Implementation of service function chaining control plane through OpenFlow,” in *IEEE Conference on Network Function Virtualization and Software Defined Networks*, pp. 1–4, 2017.
- [6] D. Borsatti, G. Davoli, W. Cerroni, C. Contoli, F. Callegati, “Performance of Service Function Chaining on the OpenStack Cloud Platform,” in *International Conference on Network and Service Management*, pp. 432–437, 2018.
- [7] A. M. Medhat, G. A. Carella, M. Pauls, T. Magedanz, “Orchestrating scalable service function chains in a NFV environment,” in *IEEE Conference on Network Softwarization (NetSoft)*, pp. 1–5, 2017.
- [8] M. Gharbaoui, C. Contoli, G. Davoli, G. Cuffaro, B. Martini, F. Paganelli, W. Cerroni, P. Cappanera, P. Castoldi, “Demonstration of Latency-Aware and Self-Adaptive Service Chaining in 5G/SDN/NFV infrastructures,” in *2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, pp. 1–2, 2018.
- [9] TimeNET project. Available online: <https://timenet.tu-ilmenau.de/>.
- [10] 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.
- [11] M. Di Mauro, G. Galatro, M. Longo, F. Postiglione, and M. Tambasco, “IP Multimedia Subsystem in a containerized environment: availability and sensitivity evaluation”, in *IEEE PVE-SDN19/Netsoft*, (Paris, France, Jun. 2019).
- [12] G. Su, and A. Iyengar, “A highly available transaction processing system with non-disruptive failure handling,” in *IEEE NOMS 2012*, pp. 409–416, 2012.
- [13] A. F. de la Cruz, J. P. Muñoz-Gea, P. Manzaneres-Lopez and J. Malgosa-Sanahuja, “Network failures support for traffic monitoring mechanisms in software-defined networks,” in *IEEE NOMS 2016*, pp. 691–694, 2016.
- [14] L. Zhang Y. Wang, X. Zhong, W. Li and S. Guo, “Resource-saving replication for controllers in multi controller SDN against network failures,” in *IEEE NOMS 2018*, pp. 1–7, 2018.
- [15] A. Kalso, M. Toeroe and F. Khendek, “Automating Service Availability Analysis: An Application to a Highly Available Media-Streaming Service,” in *IEEE Seventh International Conference on Software Security and Reliability Companion*, pp. 94–101, 2013.
- [16] L. Montecchi, P. Lollini and A. Bondavalli, “A Template-Based Methodology for the Specification and Automated Composition of Performance Models,” *IEEE Transactions on Reliability*, vol. 15, no. 4, pp. 1–17, 2019.
- [17] L. Montecchi, P. Lollini and A. Bondavalli, “A DSL-Supported Workflow for the Automated Assembly of Large Stochastic Models,” in *Tenth European Dependable Computing Conference*, pp. 82–93, 2014.
- [18] C. Leangsuksun, H. Song and L. Shen “Reliability Modeling Using UML,” in *Proceedings of the International Conference on Software Engineering Research and Practice*, pp. 259–262, 2003.
- [19] R. A. Sahner and K. S. Trivedi, “SHARPE: a modeler’s toolkit,” *Proceedings of IEEE International Computer Performance and Dependability Symposium*, pp. 58–59, 1996.
- [20] M. Cinque, D. Cotroneo and C. Di Martino, “Automated Generation of Performance and Dependability Models for the Assessment of Wireless Sensor Networks,” *IEEE Transactions on Computers*, vol. 161, no. 6, pp. 870–884, 2012.
- [21] J. Duan and C. Wu and F. Le and A. X. Liu and Y. Peng, “Dynamic Scaling of Virtualized, Distributed Service Chains: A Case Study of IMS,” *IEEE Journal on Selected Areas in Communications*, vol. 35, no. 11, pp. 2501–2511, 2017.
- [22] VoLTE Service based on vEPC and vIMS Architecture Available online: https://nfvwiki.etsi.org/index.php?title=VoLTE_Service_based_on_vEPC_and_vIMS_Architecture.
- [23] J.K. Muppala, G. Ciardo, and K.S. Trivedi, “Stochastic Reward Nets for Reliability Prediction,” in *Communications in Reliability, Maintainability and Serviceability*, pp. 9–20, 1994.
- [24] D. Bruneo, “A Stochastic Model to Investigate Data Center Performance and QoS in IaaS Cloud Computing Systems,” *IEEE Trans. Parallel Distrib. Syst.*, vol. 25, no. 3, pp. 560–569, 2014.
- [25] G. Nencioni, B. E. Helvik, P. E. Heegaard, “Including Failure Correlation in Availability Modeling of a Software-Defined Backbone Network,” *IEEE Trans. Netw. Serv. Man.*, vol. 14, no. 4, pp. 1032–1045, 2017.
- [26] Eclipse Modeling Framework (EMF). Available online: <https://www.eclipse.org/modeling/emf/>.
- [27] Atlas Transformation Language. Available online: <https://www.eclipse.org/atll/>.
- [28] Acceleo Project. Available online: <https://www.eclipse.org/acceleo/>.
- [29] R. German, C. Kelling, A. Zimmermann, and G. Hommel, “TimeNET: a toolkit for evaluating non-Markovian stochastic Petri nets,” *Performance Evaluation*, vol. 24, no. 1–2, pp. 69–87, 1995.