Modelling, Design and Optimization of Dependable Softwarized Networks for Industrial Applications

Petra Vizarreta
Chair of Communication Networks,
Technical University of Munich, Germany
petra.vizarreta@lkn.ei.tum.de

***Abstract*—The recent trend of Industry 4.0 promotes the concepts of "industrial internet and digital factory", which requires the transformation of local industrial networks, as well as wide area networks carrying industrial traffic. Softwarized network architectures, i.e., Software Defined Networking (SDN) and Network Function Virtualization (NFV), can aid this transition by providing a fine grained network traffic control and high degree of programmability, with open standards and protocol stack. While first studies and prototypes have demonstrated the feasibility of SDN/NFV based networks, a comprehensive study of dependability of softwarized networks is still missing in state of the art literature. Recent studies on reliability in softwarzed network have focused mainly on the network failures, while the reliability of software functions has been widely overlooked or oversimplified. Our research aims to close this gap, by providing the contribution in three following areas: *i*) Evaluation and forecasting of the reliability of control plane functions, *ii*) Characterization of failure dynamics between control and data plane, and *iii*) Design of robust wide area networks in the presence of software and network failures.**

I. Research overview

The recent trends of Industry 4.0, including Cyber Physical Systems (CPS) and Internet of Things (IoT), require high degree of automation of industrial systems, their tighter coupling and an efficient coordination. Industrial communication networks, which in the past have been developed as closed systems, rely on the proprietary protocol stacks, are not yet prepared for a seamless integration, due to the the lack of mechanisms for automated and secure exchange of information. Furthermore, the existing wide area networks have been tailored for the human-centric communication, and are not optimized for the M2M traffic requirements.

The recent concepts of network softwarization, i.e., Software Defined Networking (SDN) and Network Function Virtualization (NFV), enable a fine grained per-flow QoS control and high degree of programmability with open and extendible protocol stack. With SDN, the distributed control plane logic of forwarding devices, i.e., switches and routers, is moved to a software entity called SDN controller, effectively decoupling the control plane (e.g., path computation and traffic engineering) from data plane functions (i.e., switching). In NFV higher layer network functions, such as firewalls or intrusion detection systems, which are traditionally implemented in a specialized hardware, are replaced with modular software components deployed on commodity hardware. Such modular network functions can be further chained to provide fine grained traffic control, offering much greater flexibility. First field trials have shown the feasibility of the SDN/NFV based networks in operational industrial environment [1]. The contemporary performance evaluations typically focus on the scalability and response times, while the reliability, which is the key requirement for the wide spread adoption in industrial domains is overlooked or oversimplified. We aim to close this gap by providing answers to the following research questions:

- **RQ1**: Forecasting and evaluation of the reliability of control and management functions in softwarized networks
- **RQ2**: Characterization of failure dynamics between control and data plane in softwarized networks
- **RQ3**: Design of robust wide area networks in the presence of software and network failures

The rest of the paper is organized as following. Section II presents a brief summary of related work. In Section III we define the research problems, describe the proposed solution and present our contributions, open research questions and threats to validity, as well as the the status and planned timeline for the remainder of the PhD thesis. Section IV concludes the paper with summary of the presented ideas and expected impact for the software reliability community.

II. Related Works

In this section we present a brief overview of the relevant studies on dependability of softwarized networks.

Dependability of control plane elements: The existing studies in the state of the art literature [2]–[6] provided first insights on the control plane dependability. Majority of these studies model the controller reliability based on the similar systems, rather than empirical data from real life deployments or tesbed measurements, which led to some overly simplifying assumptions, such as the perfect synchronization between distributed controller replicas. Ros et al. [2] model the controller reliability as a static random Weibull variable, while the studies in [3], [4] treat software failures as a Poisson process. Impact of the controller workload on software failure rate was studied by Longo et al. [5], which is modelled with phase-type distribution. Sakic et al. [7] provide a more realistic model of RAFT consensus algorithms under different failure modes, complementing it with the measurements from an OpenDaylight testbed, showing that maintaining the consistency between distributed control plane elements is indeed a challenge. Another important deployment aspect that is not

addressed by the existing studies, is that the controller is needed only while serving a request, e.g., planning of the path for unknown packets. The arrival and serving rates of the particular service have a detrimental impact on the user perceived service availability [8]. Our goal is i) to provide models to evaluate and forecast the risk of the controller software outages and ii) a framework to characterize the impact of control plane outages on data plane services based on real controller platforms and realistic use case scenarios.

Design of reliable softwarized networks: Design and optimization problems in the context of softwarized networks combine several NP-hard problems, e.g., facility location problem, generalized assignment problem, knapsack, etc [9]. Different techniques for solving such combinatorial optimization problems have been proposed, such as Integer Linear Programming, greedy heuristic, meta-heuristic (e.g., Pareto Simulated Annealing), and game theory [10]–[12]. The main drawback of state of the art approaches to QoS-aware design of softwarized networks is that they either ignore completely or offer a very limited support for service availability differentiation. Our goal is to propose a comprehensive service availability model including software, hardware and network failures, as well as the adaptive service availability-aware network design and optimization strategies.

III. Research questions

A. Reliability of control plane elements (RQ1)

1) Problem definition: In softwarized networks the critical control plane tasks, e.g., network monitoring, route planning, installation of the flow rules, are outsourced to a software, i.e., SDN controller. The SDN controller assumes the role of network operating system, providing an integrated interface towards diverse set of networks and forwarding devices. The heterogeneity of the services and the speed of evolution of SDN controllers lead them be rather complex software components, which are inherently prone to bugs. The triggering of controller bugs can have catastrophic consequences for operational networks, as documented in post-mortem reports in SDN-based Google networks [13]. Despite the ubiquity of the software failure, the state-of-the art literature is still missing the models to accurately evaluate and estimate the risk of the software failures for operational SDN controllers.

2) Methodology and proposed solution: Software Reliability Growth Modelling (SRGM) is a statistical framework, used to model, evaluate and predict the reliability of the software components in the operational environment, based on the bug manifestation reports from the testing phase [14]. We leverage the fact that the majority of today's commercial SDN controllers is based on the open source platforms, ONOS and OpenDaylight, whose bug repositories are publicly available, and contain all the information required by SRGM. Once the best model to describe the stochastic behaviour of bug manifestation and bug removal processes is found and parametrized, it can be used to estimate how the software reliability metrics, such as the residual bug content, the expected failure intensity and conditional software reliability, change over time as the software matures.

3) Status and contributions: The results on the assessment of software maturity of SDN controllers with SRGM was published in an article [15], extending the results of the conference paper [16]. Our contribution presented in these studies can be summarized in the following:

- **Data collection and preprocessing**: We gather the empirical data from bug repositories of the two largest open source SDN platforms, and provide the high level statistics, e.g., fault densities, and distributions of the times to detect and resolve a bug.
- **Model selection (1)**: We find the best SRGM to describe the bug detection processes. We have found that all 3-parameter S-shaped models provide a good approximation of stochastic behaviour of bug detection.
- **Model selection (2)**: We also show that the bug resolution process for the first releases of ONOS controller can be described well with combination of the simple Non-Homogeneous Poisson Process (NHPP) models. More recent releases showed anomalies, e.g., sudden changes in debugging effort, that require more complex models, whose parameters are difficult to estimate in comparatively small data sets.
- **Evaluation of controller software reliability**: We used these stochastic models to estimate the optimal software release and software adoption time based on reliability and cost criteria, which is a well known application of SRGM framework. We show that at day of the official release of the recent ONOS controller, the residual bug content and failure intensity were relatively high, and that in order to achieve the reliability of 0.90 the software adoption time had to be postponed 4 months.
- **Early prediction of software reliability**: We propose the novel approach to improve the accuracy of early prediction of SRGM model parameters, based on the extrapolation of the behaviour of the consecutive controller releases.
- **Software maturity metrics**: We propose the software maturity metric, which is a scaled version of the expected software failure intensity. The proposed software maturity metrics can be used for the comparison of two alternative controller solutions, when the software reliability is the main criteria.

4) Open research questions: The parametric SRGM models have the problem of prediction accuracy in the early phase of the software life cycle. However, predicting the software reliability parameters at the end of the software lifecycle does not have much practical value. We foresee that the further research on improving the early predictability of the SRGM considering the non-parametric methods, e.g. artificial neural networks is an interesting research direction.

5) Threats to validity: The model parameters are derived from the empirical data found in the public issue trackers. However, the completeness and accuracy of such public repository is not guaranteed. The second limitation comes from the

inherent assumptions of SRGM, which assume independent times between the consecutive fault reports, and that every undetected fault contributes the same to the fault manifestation rate, in order to obtain analytically tractable results.

B. Failure dynamics between control and data plane (RQ2)

1) Problem definition: In softwarized networks, the simple redundancy of the control plane elements is not an efficient way to improve the reliability of the control plane, as it can only provide the environmental diversity counteracting some of the transient failures, while deterministic failures, such as error in the path computation module are shared between the replicas. Different failure modes of the SDN controllers occur different frequencies, have different recovery times, as well as different impact on the network applications and services provided and end users.

Our goal is to provide a comprehensive study of the failure dynamics between control and data plane in SDN, taking into account realistic interaction between controller and the services, as well as the more accurate model parameters based on the empirical data and testbed measurements.

2) Methodology and proposed solution: We have modelled the dynamics between the controller and network applications and services in the formalism of Stochastic Reward Nets (SRN). SRN are a stochastic extension of Petri Nets, and represent a powerful tool for dependability modelling [17]. In SRN modelling framework, the combination of markings in the places represents model states, the system state is changed upon the firing of the activities, and every state is associated with the corresponding reward. For simpler small size problems with Poisson and instantaneous transitions closed form solution can be derived. The large problem instances with general transition time distributions are solved using numerical approximations or discrete event simulation.

The parameters of the models will be derived from the empirical data, e.g., failure rates and impact on the controller downtimes, as well as the testbed measurements. The models of interaction between the controller, its environment and services will be derived from the real flow charts and sequence diagrams.

3) Status and contributions: The first concept on characterization of failure dynamics in SDN was presented in the conference paper [18]. The proposed model captured the effects of software reliability growth, software ageing, different software failure modes, as well as the external failures of the operating system (OS) and the computing hardware (HW), which have not been considered so far in the state-of-the art literature. We presented a case study to demonstrate the impact of different failure modes on the controller availability. Current contributions can be summarized in the following:

- We have shown that a single controller instance is not sufficient to achieve the availability of "5-nines". The analysis of the downtime distribution has shown that more than 80% of the failures have a downtime below 10 minutes, and the median is 3.6 minutes.
- The study has shown big differences in the frequency and the impact on downtime of the different failure modes considered in the model. We have observed that the software accounts for 84% of all the failures, but contribute to only one third of the controller's downtime.
- Sensitivity analysis has been performed to identify the most critical controller parameters w.r.t. its availability. We identified the hardware repair rate and the controller reload (the longest software recovery) to be the most important parameters, followed by the hardware failure and software ageing failure rates.
- Software ageing has been identified as one of the most important, and yet most uncertain factor in the controller's availability. We have studied how the relationship between the ageing rate and software ageing failures influences the impact factor of the ageing, and show in which operational range it has a non-negligible impact on software reliability.

Further extensions of this model are planned, to account for the imperfect synchronization process between controller replicas, and model the services with different arrival and serving rates. Once the models of the control plane and its services are finalized, we plan to perform a case study based the actual SDN controllers. The models will be used to compare different control plane designs, as well as to evaluate relevant dependability metrics, such as user perceived service availability and reliability.

4) Open research questions: The main challenge we faced has been the time consuming classification of the bug reports, which also requires profound expert knowledge to infer the problem nature, root cause or the resolution strategy based on the scarce free text descriptions. At the moment of writing, more than 7k bugs from development and deployment phase are recorded in the OpenDaylight bug repository. The recent study [19] has prosed an alternative approach for an automatic classification and the knowledge extraction from network incident reports based on Natural Language Processing (NLP).

5) Threats to validity: The threats to validity of the proposed approach are the reliability of estimated model parameters, selected level of model abstraction, as well as the representativeness of the use case scenarios.

C. Design of the robust wide area softwarized networks (RQ3)

1) Motivation and problem definition: The traffic between geo-distributed industrial sites typically operates over the commercial wide area networks, which provide the probabilistic guarantees on minimum guaranteed bit rate, maximum delay and connection availability, formalized in terms of Service Level Agreements (SLA). The existing work focused on bit rate and delay, while service availability in the context of wide area networks received little attention. The goal of this study is to devise adaptive service availability-aware resource management strategies for the fair coexistence of industrial and human-centric traffic.

2) Methodology and proposed solution: Our proposed solution consists of three steps. First we propose a comprehensive

SLA model for softwarized networks, including software, hardware and network failures. Then we propose a set of tunable heuristic algorithms that have a very good performance in a limited problem subspace. Finally, we design machine learning approach based on ensemble learning techniques for an automated configuration and selection of low level heuristics, to ensure the adaptive resource management.

3) Status and contributions: The first concept of QoS-driven network design reducing expenditures in softwarized wide area networks was presented in [20]. The service availability model accounting for the failures of network function software, supporting hardware, as well as the failures of underlying communication network. Two QoS-aware placement strategies are presented, an optimal solution based on the Integer Linear Programming (ILP) and an efficient heuristic to obtain near optimal solution for a particular case of German core network. The next study will focus on design and evaluation of machine learning based service composition in the context of softwarized networks.

D. Proposed timeline

The student is currently finishing the third year of her PhD studies. The study on reliability assessment of control plane elements (RQ1) was completed in March 2018 [15], [16]. The concept and models for characterization of failure dynamics between control and data plane (RQ2) have been presented in [18], and the final evaluation is expected by the end of 2018. The first results of strategies for network design and optimization of wide area softwarized networks (RQ3) have been published in [20]. The further development and extensions are planned for the first half of 2019. Submission of the dissertation and a defence is expected in the second half of 2019.

IV. Conclusions

Network softwarization is the necessary step in the evolution towards the next generation industrial networks, and dependability is the key feature for the industrial applications. Hence, it is of the utmost importance to develop the framework to accurately estimate the dependability of all of layers in softwarized networks. The main goal of this thesis is to advance the state of the art understanding of dependability of softwarized networks for industrial applications. However, *"network in a software"* is not just a novel use case for the software reliability community, but it also opens several new research questions: i) how to improve the early predictability of software reliability, ii) how to automate the classification and analyze the large corpus of software bugs with a network domain specific vocabulary and iii) how to model the interplay between the physical and software network components.

Acknowledgement

This work has been done at the Chair of Communication Networks, of Technical University of Munich, under the supervision of Carmen Mas Machuca and Wolfgang Kellerer.

References

[1] T. Mahmoodi, V. Kulkarni, W. Kellerer, P. Mangan, F. Zeiger, S. Spirou, I. Askoxylakis, X. Vilajosana, H. J. Einsiedler, and J. Quittek, "Virtuwind: virtual and programmable industrial network prototype deployed in operational wind park," *Transactions on Emerging Telecommunications Technologies*, vol. 27, no. 9, pp. 1281–1288, 2016.

[2] F. J. Ros and P. M. Ruiz, "Five nines of southbound reliability in software-defined networks," in *Proc. of the third workshop on Hot topics in software defined networking (HotNets)*. ACM, 2014, pp. 31–36.

[3] T. A. Nguyen, T. Eom, S. An, J. S. Park, J. B. Hong, and D. S. Kim, "Availability modeling and analysis for software defined networks," in *Proc.of 21st Pacific Rim International Symposium on Dependable Computing (PRDC)*. IEEE, 2015, pp. 159–168.

[4] G. Nencioni, B. E. Helvik, A. J. Gonzalez, P. E. Heegaard, and A. Kamisinski, "Availability modelling of software-defined backbone networks," in *Dependable Systems and Networks Workshop, 2016 46th Annual IEEE/IFIP International Conference on*. IEEE, 2016, pp. 105–112.

[5] F. Longo, S. Distefano, D. Bruneo, and M. Scarpa, "Dependability modeling of software defined networking," *Computer Networks*, vol. 83, pp. 280–296, 2015.

[6] A. J. Gonzalez, G. Nencioni, B. E. Helvik, and A. Kamisinski, "A fault-tolerant and consistent sdn controller," in *Proc. of Global Communications Conference (GLOBECOM)*. IEEE, 2016, pp. 1–6.

[7] E. Sakic and W. Kellerer, "Response time and availability study of raft consensus in distributed sdn control plane," *IEEE Transactions on Network and Service Management*, vol. 15, no. 1, pp. 304–318, 2018.

[8] D. Wang and K. S. Trivedi, "Modeling user-perceived service availability," in *Proc. of International Service Availability Symposium*. Springer, 2005, pp. 107–122.

[9] J. G. Herrera and J. F. Botero, "Resource allocation in nfv: A comprehensive survey," *IEEE Transactions on Network and Service Management*, vol. 13, no. 3, pp. 518–532, 2016.

[10] A. Hmaity, M. Savi, F. Musumeci, M. Tornatore, and A. Pattavina, "Protection strategies for virtual network functions placement and service chains provisioning," *Networks*, vol. 70, no. 4, pp. 373–387, 2017.

[11] S. Lange, A. Grigorjew, T. Zinner, P. Tran-Gia, and M. Jarschel, "A multi-objective heuristic for the optimization of virtual network function chain placement," in *Proc. of Teletraffic Congress (ITC 29)*. IEEE, 2017, pp. 152–160.

[12] L. Qu, M. Khabbaz, and C. Assi, "Reliability-aware service chaining in carrier-grade softwarized networks," *IEEE Journal on Selected Areas in Communications*, vol. 36, no. 3, pp. 558–573, 2018.

[13] R. Govindan, I. Minei, M. Kallahalla, B. Koley, and A. Vahdat, "Evolve or die: High-availability design principles drawn from Google's network infrastructure," in *Proc. of ACM SIGCOMM Conference*. ACM, 2016, pp. 58–72.

[14] M. R. Lyu *et al.*, *Handbook of software reliability engineering*. IEEE computer society press CA, 1996.

[15] P. Vizarreta, K. Trivedi, B. Helvik, P. Heegaard, A. Blenk, W. Kellerer, and C. M. Machuca, "Assessing the maturity of sdn controllers with software reliability growth models," *IEEE Transactions on Network and Service Management*, 2018.

[16] P. Vizarreta, K. Trivedi, B. Helvik, P. Heegaard, W. Kellerer, and C. Mas Machuca, "An empirical study of software reliability in SDN controllers," in *Proc. of 13th International Conference on Network and Service Management (CNSM)*. IEEE, 2017, pp. 1–9.

[17] K. S. Trivedi and A. Bobbio, *Reliability and Availability Engineering: Modeling, Analysis, and Applications*. Cambridge University Press, 2017.

[18] P. Vizarreta, P. Heegaard, B. Helvik, W. Kellerer, and C. Mas Machuca, "Characterization of failure dynamics in SDN controllers," in *Proc. of 9th International Workshop on Resilient Networks Design and Modeling (RNDM)*. IEEE, 2017, pp. 1–7.

[19] R. Potharaju, N. Jain, and C. Nita-Rotaru, "Juggling the jigsaw: Towards automated problem inference from network trouble tickets." in *Proc. of USENIX Symposium on Networked Systems Design and Implementation (NSDI)*, 2013, pp. 127–141.

[20] P. Vizarreta, M. Condoluci, C. M. Machuca, T. Mahmoodi, and W. Kellerer, "Qos-driven function placement reducing expenditures in nfv deployments," in *Proc. of International Conference on Communications (ICC)*. IEEE, 2017, pp. 1–7.