

A Study on the Dependability of Software Defined Networks

Jianqing Wu, Yan Huang, Jiaming Kong, Qin Tang, Xin Huang

Xi'an Jiaotong-Liverpool University, Suzhou 215123, P.R. China

{Jian.Wu14, Yan.Huang12, Jiaming.Kong12, Qin.Tang13}@student.xjtlu.edu.cn,
Xin.Huang@xjtlu.edu.cn

Keywords: Software defined networking, model checking

Abstract. In this paper, we demonstrate how probabilistic model checking can be applied to a study of dependability analysis for a Software-Defined Network with the PRISM tool for probabilistic model checking. Based on probabilistic model checking, the Software-Defined Network is modelled using large and complex Markov chains. In order to improve the reliability of the system, we propose a multi-controller architecture. The results of the designed system model are verified and visualized using PRISM.

Introduction

Software-Defined Networking (SDN) is a new norm for networks, which allows the network administrators to gain unprecedented programmability, automation, and network control. It decouples the network control so that the system becomes more suitable for dynamic computing, as well as the storage needs for the data [1]. With the wide spread of SDN, its management and control can be quite complex and affect its reliability.

We design a formal model of SDN. In addition, probabilistic model checking technique is used to validate the reliability of the SDN. Probabilistic model checking is able to model and analyze complex stochastic systems [2]. It calculates the rate of occurrence of certain events by setting up the corresponding properties so as to validate the reliability of the system with numerical computation [3].

This paper is organized as follows. Section II explains probabilistic model checking techniques. The SDN formal model is given in Section III. Section IV provides the model checking results. Finally, the conclusion is made in Section IV.

Model Checking

Model checking is a method of verifying the correctness of systems [4]. It is widely used to check whether the work done by the system is as expected [5]. In general, the process of model checking is as follows:

1. Build a model for a system
2. Verify the validity of different properties

Probabilistic model checking is based on the construction and analysis of a probabilistic model. It can calculate the probability of all states that can occur. There are many different probabilistic models, including Continuous-time Markov chains (CTMCs) [6]. CTMC is commonly used to study system performances and dependability. In this paper, we aim to research the dependability of the SDN, thus we construct the model using CTMC.

PRISM is a tool for probabilistic model checking [6], [3]. More specifically, as reliability and stability both are important features for developing a networking system, PRISM can analyze them by performing probabilistic analysis of Markovian processes. In this paper, it will be used to model checking the SDN CTMC model.

SDN Model

Figure 1 indicates the architecture of a SDN, which includes three layers and two APIs. The three layers refer to application, control and data plane; and the two APIs are Northbound APIs and Southbound APIs.

- The application plane is on the top layer. It comprises of different SDN applications (SDN App) and for each SDN application, it consists of one application logic and one or more Northbound API drivers. These applications transmit their requests and desired network behaviors via Northbound APIs to the SDN controller [7].
- The control plane is in the middle layer. In this plane, there are SDN controllers, which are responsible for translating the requirements from Application layer to the instances in the Data plane. Additionally, it provides the abstract view of the network to the SDN applications [7].
- The data plane, which comprises of logic network devices (or SDN datapaths), is at the bottom layer. The capabilities of these network components are exposed through Southbound API agents.

We use PRISM to analyze dependability of the SDN networks. If the control plane of SDN fails, the whole network system will automatically shut down. Thus, it is necessary to use multiple controllers: if one fails, the remaining controllers can still work.

The improved system structures is shown in Figure 2, which includes two possible architectures (Figure 2(a) and Figure 2(b)). Both contain the following components: (1) three applications, (2) northbound API, (3) southbound API, and (4) three packet forwarding devices. The differences are: SDN in Figure 2(a) has four controllers, and SDN in Figure 2(b) has six controllers.

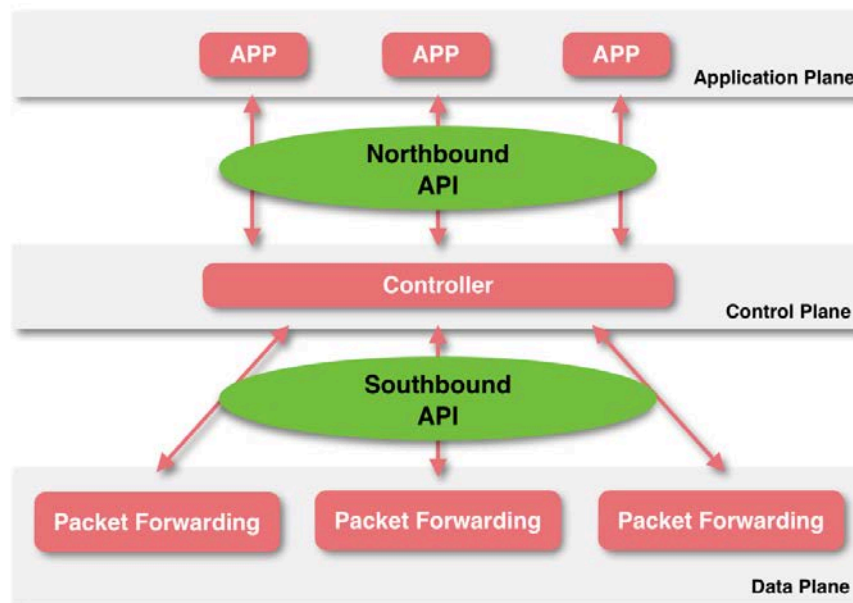
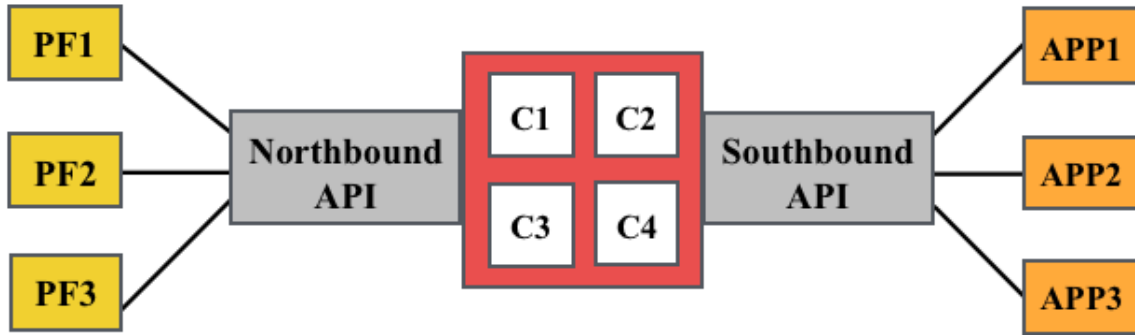
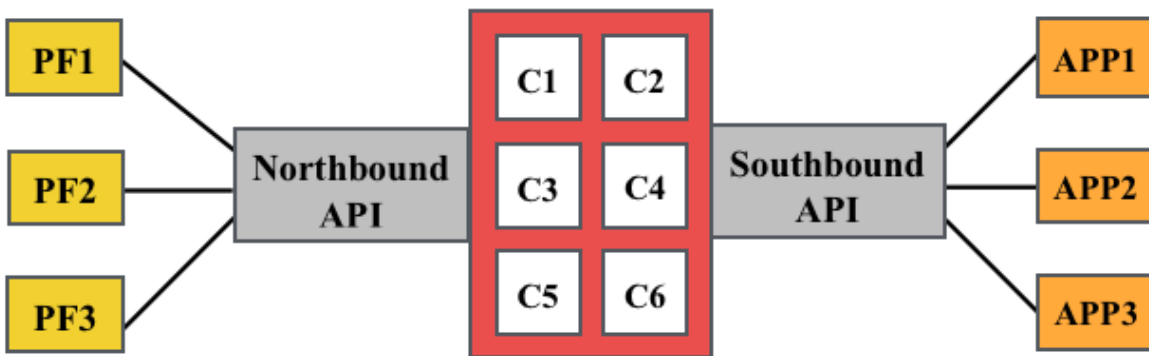


Fig.1.The structure of Software-defined Network



(a) Modified system (4 clustered controllers)



(b) Modified system (6 clustered controllers)

Fig. 2. Structure of modified systems

Result

This section presents the results of the probability of each failure type occurring first, as well as the expected time spent in three states: “up”, “danger” and “down”.

Figure 3(a) represents the probability of each failure type occurring first in 24 hours by using 4 controllers; Figure 3(b) represents the probability of each failure type occurring first in 24 hours by using 6 controllers; Figure 3(c) represents the probability of each failure type occurring first in 30 days by using 4 controllers; Figure 3(d) represents the probability of each failure type occurring first in 30 days by using 6 controllers.

- Compared with Figure 3(a), it is clear that the probability of controller failure in Figure 3(b) is smaller. It means 4 controllers are less reliable than 6 controllers in a short run.
- Figure 3(c) and Figure 3(d) reveal that 4 controllers are less reliable than 6 clustered controllers in a long run.
- This shows that adding more controllers can improve SDN reliability.

PRISM is able to compute the long-run failure probability. Table 1 shows the failure probability of each component in three SDN systems of which the controller number are 1, 4 and 6 respectively. From Table 1, it can be seen that the probability of one controller that leads to system shutting down is 0.048, while the ones of clustered 4 and 6 controllers are at E-05 and E-07 levels. This indicates the trend that with the number of controllers increasing, the probability that controller leads to system shutting down decreases.

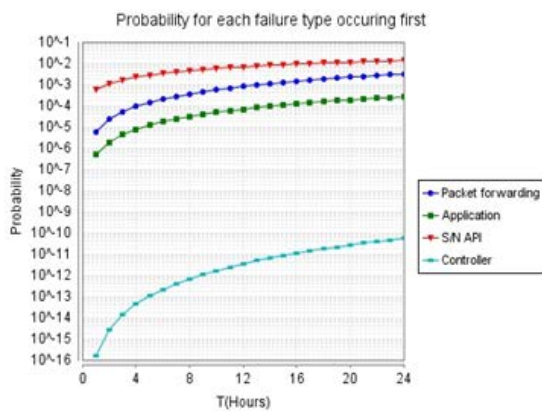
In Table 2, depending on the parameter K that ranges from 1 to 7, time spent in “up” and “danger” states until the system fails increases. We can see that if we assume that **failure rates of packet forwarding devices and APIs are much higher than the rates of controllers**, using more controllers will not increase the dependability of the whole system significantly.

Table.1. The probability that each failure type is the eventual cause of system shutdown

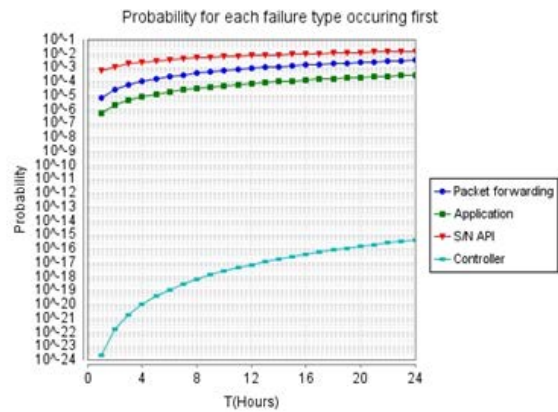
Failure type	Probability(1)	Probability(4)	Probability(6)
Packet forwarding	0.6214	0.6539	0.6539
Application	0.0877	0.0929	0.0929
S/N API	0.2425	0.2532	0.2532
Controller	0.0484	4.2053E-05	9.3085E-07

Table.2. Expected time spent in states “danger” and “up” before system shutdown occurs

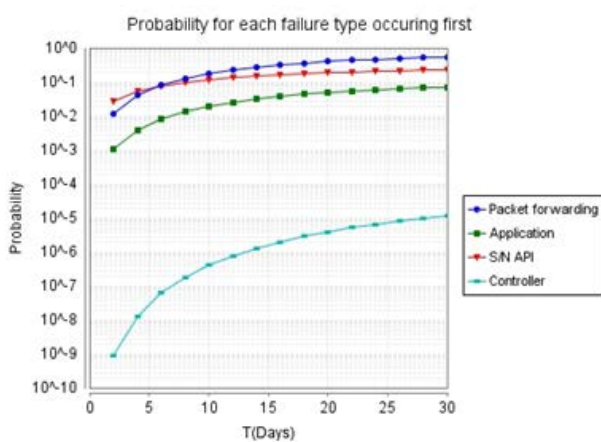
K	Expected time (1):		Expected time (4):		Expected time (6):	
	danger(hrs)	up(days)	danger(hrs)	up(days)	danger(hrs)	up(days)
1	0.236	14.323	0.244	14.796	0.244	14.796
2	0.293	17.660	0.305	18.353	0.305	18.354
3	0.318	19.100	0.331	19.899	0.331	19.900
4	0.327	19.628	0.341	20.467	0.341	20.468
5	0.330	19.809	0.344	20.663	0.344	20.664
6	0.331	19.871	0.346	20.729	0.346	20.730
7	0.332	19.891	0.346	20.751	0.346	20.752



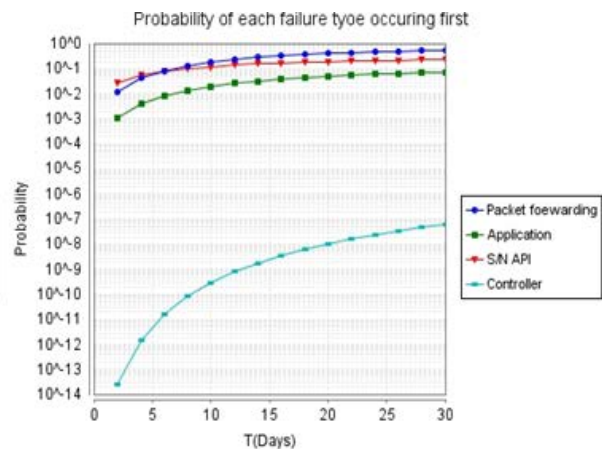
(a) First 24 hours of system operation (4 clustered controllers)



(b) First 30 days of system operation (4 clustered controllers)



(c) First 24 hours of system operation (6 clustered controllers)



(d) First 30 days of system operation (6 clustered controllers)

Fig. 3. The probability that each of the four possible failure type is the cause of system shutdown

Conclusions

This paper illustrated how the probabilistic model checker PRISM can be used to analyze the dependability of each component in SDN systems. The modified SDN systems are constructed with the continuous-time Markov chains model. The result shows the trend that the reliability of this system is improved by adding additional controllers.

Acknowledgement

This work has been supported in part by the XJTLU RDF140243.

References

- [1] 2015, May. 14. Software-Defined Networking (SDN) Definition [Online]. Available: <http://www.opennetworking.org/sdn-resources/sdn-definition>
- [2] Kwiatkowska, M. et al. 2009. "PRISM: probabilistic model checking for performance and reliability analysis" ACM SIGMETRICS Performance Evaluation Review, vol. 36, no. 4, pp. 40-45.
- [3] Kwiatkowska, M. et al. 2004a. "Probabilistic symbolic model checking with PRISM: A hybrid approach" International Journal on Software Tools for Technology Transfer, vol. 6, no. 2, pp. 128-142.
- [4] Kwiatkowska, M. et al. 2004b. "Controller Dependability Analysis by Probabilistic Model Checking" Control Engineering Practice, vol.15, pp.1427–1434.
- [5] Kauttuo, J. 2012, May. 30. State-of-the-art report on probabilistic model checking and its applicability to critical I&C systems analysis [Online]. Available: http://www.academia.edu/11811443/State-of-the-art_report_on_probabilistic_model_checking_and_its_applicability_to_critical_I_and_C_systems_analysis
- [6] Kwiatkowska, M. et al. 2002. "PRISM: Probabilistic symbolic model checker" Computer performance evaluation: modelling techniques and tools, pp. 200-204.
- [7] 2013, Dec. 12. *SDN Architecture Overview* [Online]. Available: <https://www.opennetworking.org/images/stories/downloads/sdn-resources/technical-reports/SDN-architecture-overview-1.0.pdf>