IST-153 Workshop on CYBER RESILIENCE Resilience and Security in Software Defined Networking

Camen Mas-Machuca, *Senior Member, IEEE,* Petra Vizarreta, Raphael Durner, and Jacek Rak, *Member, IEEE*

***Abstract*—This paper gives an overview of the most important issues on resilience and security in Software Defined Networking.**

I. Introduction

SOFTWARE Defined Networking (SDN) is a recent paradigm that aims increasing network flexibility and efficiency by separating the control from the data plane. The SDN architecture is depicted in Figure 1. The data plane consists of interconnected forwarding devices, which forward packets based on their forwarding tables, which are built based on the input from the controller. The control plane is the intelligent layer that configures that path at the data plane based on the requirements from the application layer and also provides an abstract view of the data plane to the application layer. Data flows can be set based on request from the application layer, or based on new flows from connected users. In the last case, the forwarding device will contact the controller through the so-called secured channel to know how to proceed.

Although the control plane is a logically centralized entity, it can be physically distributed at different locations. In that case, forwarding devices are assigned to one (or more) controllers. Coordination among the controllers is required (e.g., federation, hierarchical).

II. Data Plane Resilience

Data plane resilience deals with the protection and restoration of data flows. Existing protection schemes for transport networks such as dedicated or share path protection, which finds link and/or node disjoint paths can be also applied to SDN networks. These schemes aim at offering 100% reliability and have been further extended in order to consider QoS/security aspects and use less resources when possible [2], [3]. The compromise between protection and restoration in terms of flow restoration time and used resources is targeted by pre-computing several disjoint paths, from which the best one is selected in case of failure. Another proposed technique by Xie et al. [4] proposes a proactive local failure recovery module running at the

C. Mas Machuca, P. Vizarreta and R. Durner are with the Chair of Communication Networks, Technical University of Munich, TUM, Germany e-mail: (see http://lkn.ei.tum.de).

J. Rak is with the Telecommunications and Informatics Department of Computer Communications, Gdansk University of Technology, Poland

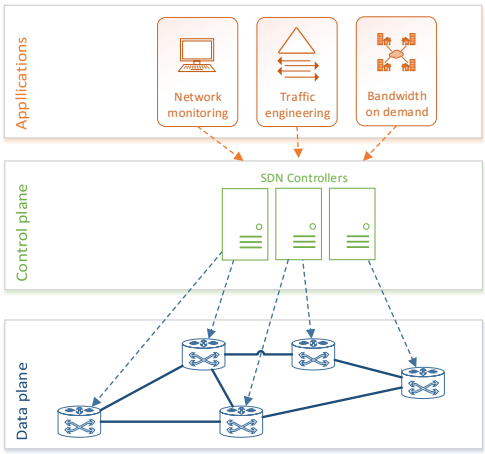

Fig. 1. Software Defined Networking architecture (figure adapted from [1]).

forwarding components able to restore flows in case of one local failure. The paper also proposes a reactive splicing module implemented at the controller, which allows to restore flows in case of multiple failures. Flow restoration is triggered by the controller and hence, it is important that the controller is available when the failure occurs. Furthermore, each controller implementation offers different approaches to address failures scenarios, which can be further extended (e.g., the POX controller offers several algorithms extended by Vaghani et al. [5]).

III. Control Plane Resilience

In SDN, the control plane of any network device is shifted to the SDN controller(s). Hence, any device has to be connected at least to one controller. The loss of connectivity between the forwarding devices and their designated controllers, as well as the failures of the controllers themselves, might seriously diminish the overall network performance. Heegaard et al. [6] presented five classes of threats to reliability in SDN, which can be summarized as follows:

- Threats affecting Control Flows
  - Connectivity loss between forwarding devices and controller(s)
  - State consistency between the controller replicas
- Threats affecting the controller

  - Controller outages
  - Controller software design
- Human error and misconfiguration of the network

Let us briefly present several representative papers addressing these threats. The "Human error and network misconfiguration" threat is not specific to SDN based networks, but has potentially have a much broader impact than in traditionally distributed legacy networks, since controller would disseminate the configuration to the entire network.

A. Control Flows

The control plane in SDN is logically centralized, but may employ multiple physically distributed SDN controllers across the network in order to improve the resilience [7]. Ross et al. [8] showed that in order to achieve 99.999% availability of the control plane, the forwarding devices have to be connected to at least two controllers for most of today's wide area networks. These control flows are referred as secure channels.

The resilience of the control plane highly depends on the number and the location of the controllers in the network. Several controller placement algorithms maximizing the control path diversity [9], and optimization of minimal cut sets have been proposed literature. Vizarreta et al. [10] compared two control path protection designs and also proposed an optimal strategy based on solution of the corresponding Integer Linear Programming (ILP) problem. It has been shown that protecting control paths can improve the control path loss up three orders of magnitude, while adding a small extra delay. However, since the problem of resilient control paths planning is NP-hard, this approach does not scale for large networks. Recent efforts have been focused on finding the efficient approximation algorithms for resilient control path design.

In order to improve the fault tolerance, controllers may deploy distributed storage system to replicate the current state of the nodes and flows under their control. Maintaining the state consistency has to find the compromise between accuracy and control traffic, as the other controllers have to be informed about any state update (e.g., new flow rule installed). Sakic et al. [11] proposed an adaptive consistency framework, where sharing the state updates can be deferred in time, depending on the application requirements, and hence balancing the trade-off between control plane latency and message overhead. It is important to provide and maintain the reliable connection between the controllers to prevent the loss of the state update messages, that could compromise the control plane reliability.

B. Controller

The SDN controller is essentially a software component running on commodity hardware which makes it susceptible to different types of failures. In [1] different failure modes of SDN controller were analyzed. The authors have shown that the failures of hardware and operating system, although less frequent than software failures, contribute more to the controller outages.

The SDN controller is required to perform large set of tasks, ranging from network state monitoring, traffic steering and enforcement of network performance policies, which requires a rather complex software. Today's production grade SDN controllers have grown to have more than 3 million lines of code [12], and software bugs are inevitable. Some software bugs, such as an error in path computation element or concurrency issues, cannot be overcome with the simple redundancy, and more sophisticated fault tolerance mechanisms are needed. The state-of-the-art literature is still missing a comprehensive study on nature and frequency of software related failures.

IV. Security in SDN

As SDN emerges from research to productive deployments, the security of SDN gains more and more importance. The most prominent SDN protocol is OpenFlow, which is descending from Ethane [13]. Ethane was developed to provide fine grained control in enterprise networks in order to improve the security. One main difference is the change in network behavior from "allow-first-restrict-later" to "restrict-first-allow-later". This approach improves security in SDNs inherently, when compared to legacy networks . Additionally with the introduction of a centralized control plane, a global network view is getting available. Using this global view, largely facilitates network verification methods like introduced for example by Kazemian et al. [14]. This is critical to ensure the isolation of multiple network zones with different security demands.

On the other hand, SDN also introduces new attack vectors. In the following, the main attack vectors are structured according to the planes introduced in Figure 1.

A. Attacks from the Data Plane

If the attacker has only access to the data plane, like every host in the network, there are some possible attack vectors: an attacker can try to overload the controller [15], the secure channel between controller and forwarding devices [16] or even the switch table [17] by injecting certain packets with high rate. Existing works that try to prevent these Denial of Service attacks use anomaly detection mechanisms and block the attacker's packets directly in the data plane [18], [19]. One main advantage of SDN is the automatic configuration of the network. One example is the automatic topology discovery, usually performed with the Link Layer Discovery Protocol (LLDP). Without any precautions, like for example authenticated LLDP Packets, an attacker can manipulate the topology view of the controller using forged packets. This can be further exploited for eavesdropping attacks [17].

B. Attacks from the Control Plane

If the attacker can get access to the control plane, by for example hijacking a forwarding device, even more serious threats are possible. An attacker could use conventional means to perform a Man-in-the-middle attack against the secure channel [20], giving him full control over the network. Additionally attacks with malformed packets in the control plane can cause failures of the controllers [21] and in consequence cause network failures. To meet these risks, authentication

and encryption of the secure channel is crucial. Unfortunately authentication is not always supported in the current SDN ecosystem [22].

C. Attacks from the Application Plane

Additional risks can turn up from the usage of malicious or malfunctioning SDN applications. This can be relieved using formal verification methods in the controller [23]. These methods can be used to enforce security rules, like for example the isolation of different network zones.

One issue for a secure operation of an SDN that remains open is the verification of the security of all components and a full bottom up trust relationship between all components and layers.

V. Conclusion

This paper has given an overview of the most important issues and some proposed solutions in order to increase the reliability and security in Software Defined Networking. As it has been mentioned, the flexibility and efficiency offered by SDN comes with some challenges (e.g., higher software failures).

Acknowledgment

This article is based upon work from COST Action CA 15127 (Resilient communication services protecting end-user applications from disaster-based failures RECODIS) supported by COST (European Cooperation in Science and Technology.

References

[1] P. Vizarreta, P. Heegaard, B. Helvik, W. Kellerer, and M. M. Carmen, "Characterization of failure dynamics in sdn controllers," in *Resilient Networks Design and Modeling (RNDM), 2017 9th International Workshop on*. IEEE, 2017.

[2] M. Furdek, N. Skorin-Kapov, and L. Wosinska, "Attack-aware dedicated path protection in optical networks," *Journal of Lightwave Technology*, vol. 34, no. 4, pp. 1050–1061, Feb 2016.

[3] J. Yallouz and A. Orda, "Tunable qos-aware network survivability," *IEEE/ACM Transactions on Networking*, vol. 25, no. 1, pp. 139–149, Feb 2017.

[4] A. Xie, X. Wang, W. Wang, and S. Lu, "Designing a disaster-resilient network with software defined networking," in *2014 IEEE 22nd International Symposium of Quality of Service (IWQoS)*, May 2014, pp. 135–140.

[5] R. Vaghani and C.-H. Lung, "A comparison of data forwarding schemes for network resiliency in software defined networking," *Procedia Computer Science*, vol. 34, pp. 680 – 685, 2014, the 9th International Conference on Future Networks and Communications (FNC'14)/The 11th International Conference on Mobile Systems and Pervasive Computing (MobiSPC'14)/Affiliated Workshops. [Online]. Available: http://www.sciencedirect.com/science/article/pii/S1877050914009521

[6] P. E. Heegaard, B. E. Helvik, and V. B. Mendiratta, "Achieving dependability in software-defined networkinga perspective," in *Reliable Networks Design and Modeling (RNDM), 2015 7th International Workshop on*. IEEE, 2015, pp. 63–70.

[7] D. Levin, A. Wundsam, B. Heller, N. Handigol, and A. Feldmann, "Logically centralized?: state distribution trade-offs in software defined networks," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 1–6.

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

[9] L. F. Müller, R. R. Oliveira, M. C. Luizelli, L. P. Gaspary, and M. P. Barcellos, "Survivor: an enhanced controller placement strategy for improving sdn survivability," in *2014 IEEE Global Communications Conference*. IEEE, 2014, pp. 1909–1915.

[10] P. Vizarreta, C. M. Machuca, and W. Kellerer, "Controller placement strategies for a resilient sdn control plane," in *Resilient Networks Design and Modeling (RNDM), 2016 8th International Workshop on*. IEEE, 2016, pp. 253–259.

[11] E. Sakic, F. Sardis, J. W. Guck, and W. Kellerer, "Towards adaptive state consistency in distributed sdn control plane," in *Conference on Communications (ICC), 2017 IEEE International*. IEEE, 2017.

[12] Linux Foundation, "Opendaylight." [Online]. Available: https://www.opendaylight.org/

[13] M. Casado, M. J. Freedman, J. Pettit, J. Luo, N. McKeown, and S. Shenker, "Ethane: Taking control of the enterprise," in *Proceedings of the 2007 Conference on Applications, Technologies, Architectures, and Protocols for Computer Communications*, ser. SIGCOMM '07. New York, NY, USA: ACM, 2007, pp. 1–12. [Online]. Available: http://doi.acm.org/10.1145/1282380.1282382

[14] P. Kazemian, M. Chan, H. Zeng, G. Varghese, N. McKeown, and S. Whyte, "Real time network policy checking using header space analysis." in *NSDI*, 2013, pp. 99–111.

[15] S. Shin and G. Gu, "Attacking software-defined networks: A first feasibility study," in *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking*. ACM, 2013, pp. 165–166.

[16] L. Schehlmann, S. Abt, and H. Baier, "Blessing or curse? revisiting security aspects of software-defined networking," in *Network and Service Management (CNSM), 2014 10th International Conference on*. IEEE, 2014, pp. 382–387.

[17] R. Klöti, V. Kotronis, and P. Smith, "OpenFlow: A security analysis," *Proceedings - International Conference on Network Protocols, ICNP*, 2013.

[18] S. M. Mousavi and M. St-Hilaire, "Early detection of DDoS attacks against SDN controllers," *2015 International Conference on Computing, Networking and Communications, ICNC 2015*, pp. 77–81, 2015.

[19] R. Durner, C. Lorenz, M. Wiedemann, and W. Kellerer, "Detecting and mitigating denial of service attacks against the data plane in software defined networks," in *IEEE Conference on Network Softwarization - Workshop on Security in NFV-SDN*, 2017.

[20] K. Benton, L. J. Camp, and C. Small, "OpenFlow Vulnerability Assessment Categories and Subject Descriptors," *Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking - HotSDN '13*, pp. 151–152, 2013.

[21] A. Shalimov, D. Zuikov, D. Zimarina, V. Pashkov, and R. Smeliansky, "Advanced study of SDN/OpenFlow controllers," *Proceedings of the 9th Central & Eastern European Software Engineering Conference in Russia on - CEE-SECR '13*, pp. 1–6, 2013. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2556610.2556621

[22] R. Durner and W. Kellerer, "The cost of security in the sdn control plane," *CoNEXT Student Workhop*, 2015.

[23] H. Hu, W. Han, G.-j. Ahn, and Z. Zhao, "FLOWGUARD," in *Proceedings of the third workshop on Hot topics in software defined networking - HotSDN '14*. New York, New York, USA: ACM Press, 2014, pp. 97–102. [Online]. Available: http://dl.acm.org/citation.cfm?doid=2620728.2620749