

Evaluating Scalability, Resiliency, and Load Balancing in Software-Defined Networking [†]

Pablo Barbecho Bautista ^{1,*} , Jaume Comellas ²  and Luis Urquiza-Aguir ³ 

¹ Departamento de Ingeniería Eléctrica, Electrónica y Telecomunicaciones, Universidad de Cuenca, Ave. 12 de Abril y Agustín Cueva, Cuenca 010101, Ecuador

² Optical Communications Group (GCO), Universitat Politècnica de Catalunya (UPC), Jordi Girona 1-3, 08034 Barcelona, Spain; jaume.comellas@upc.edu

³ Departamento de Electrónica, Telecomunicaciones y Redes de Información, Escuela Politécnica Nacional (EPN), Ladrón de Guevara E11-253, Quito 170525, Ecuador; luis.urquiza@epn.edu.ec

* Correspondence: pablo.barbecho@ucuenca.edu.ec

[†] Presented at the XXXI Conference on Electrical and Electronic Engineering, Quito, Ecuador, 29 November–1 December 2023.

Abstract: With emerging technologies like cloud computing and big data, managing traditional networks has become more demanding. Software-defined networking (SDN) promises faster implementation, flexibility, and simplified network management. However, due to SDN's centralized nature, it encounters limitations. SDN controllers should have enough processing power to deal with a high amount of flow. In addition, a single point of failure may affect the network's resiliency. For these issues, multi-instance implementation enables distributed control. However, this solution implies an intrinsic controller-to-controller synchronization channel. In this article, we propose different failure scenarios in both the data and control planes to provide network administrators with a clear view of the constraints of network reliability, load balancing, and scalability in SDN environments. The simulation results show that, regarding resiliency, SDN networks require half the time compared to traditional networks in order to recover from a link failure. Regarding load-balancing capabilities, load balancing is not guaranteed with the reactive forwarding approach (on-demand flow installation). Lastly, the SDN multi-instance solution impacts the network performance by between 1% and 21% compared to the single-instance case.



Citation: Bautista, P.B.; Comellas, J.; Urquiza-Aguir, L. Evaluating Scalability, Resiliency, and Load Balancing in Software-Defined Networking. *Eng. Proc.* **2023**, *47*, 16. <https://doi.org/10.3390/engproc2023047016>

Academic Editor: Jackeline Abad

Published: 4 December 2023



Copyright: © 2023 by the authors. Licensee MDPI, Basel, Switzerland. This article is an open access article distributed under the terms and conditions of the Creative Commons Attribution (CC BY) license (<https://creativecommons.org/licenses/by/4.0/>).

Keywords: software-defined networking; network evaluation; ONOS; ODL

1. Introduction

Software-defined networking (SDN) has emerged as an architecture that centralizes a network's intelligence, breaking the control plane out of the switch and delegating this control to a central control element. In contrast to legacy networks' distributed nature, the SDN architecture centralizes network management in one device: the SDN Controller.

OpenDayLight (ODL) and the Open Network Operating System (ONOS) are the leading open-source controllers. This work starts by evaluating ODL and ONOS controllers to provide a reference point for the following testing scenarios. This work assesses how the SDN controller deals with different failure scenarios in their data and control planes.

First, we evaluate the data plane network resiliency regarding path computation and link failover recovery time. For this, we use the Intents Subsystem implemented in the ONOS. Using intent on computer networking is a new concept used in SDN. The objective is to translate high-level policies into the network configuration [1]. Then, load-balancing capabilities are evaluated regarding the network's reaction time. Here, directives are configured in the ONOS controller, considering proactive and reactive forwarding modes in charge of distributing flows between alternative paths. The Dijkstra algorithm is used to perform the routing process.

Later, at the control plane, it is critical for the controller to present enough processing power, ensuring continuity of service despite failures. Further, a single-server instance results in a single point of failure. One of the mechanisms used to manage SDN scalability issues is the multi-instance controller. The latter implies an intrinsic controller-to-controller synchronization channel. In this regard, this work evaluates the impact of this solution by increasing the number of switches and controllers (multi-instance scenario) and assessing the network's throughput (flows/s).

The main contributions of this work are as follows:

- Producing comprehensive failure scenarios in control and data planes. The proposed scenarios are intended to be easily reproduced to assess future approaches in SDN.
- Providing network administrators with a clear view of SDN reliability, load-balancing, scalability issues, and strategies to handle burdens.
- Providing ODL and ONOS open-source controller evaluation.

The remainder of this paper is structured as follows. Section 2 summarizes the related works. Section 3 introduces evaluated network features. Then, the system model is described in Section 4, including the performance evaluation of ONOS and ODL controllers. Then, the simulated scenarios and results are discussed in Section 5. Finally, conclusions are presented in Section 6.

2. Literature Review

In [2], the authors explore machine learning techniques applied in SDN. The authors optimize delay and connectivity for SD-WAN environments using a multi-agent deep reinforcement learning algorithm. Load balancing at the data plane is not considered. In this sense, SDN controllers, with a global view of network resources, may produce optimal solutions compared to traditional networks, where the load balancing is uniquely based on local information. In this regard, in this work, we present a straightforward load-balancing scenario based on the Intents Subsystem of the ONOS. Further, in [3], the authors include a load balancing evaluation of traffic among Customer Premise Edges (CPEs) in a distributed control scenario for SD-WANs.

In [4], the authors evaluate the robustness of ODL and ONOS SDN controllers, including distributed control plane failure scenarios. Nevertheless, the performance degradation due to controllers' east-west communication is not considered. Therefore, we include an evaluation of a multi-instance system, analyzing the impact of the intrinsic controller synchronization channel in distributed control solutions with an incremental number of switches at the data plane.

Finally, Table 1 summarizes the differences between other works compared to the present work.

Table 1. Overview of related works.

Author(s)	Main Objective	Scalability	Resiliency	Load Balancing
[2]	Optimization of delay and connectivity for SD-WAN scenarios.	Evaluated in terms of the number of switches.	Does not address directly.	Does not address.
[3]	Optimization of load balancing in distributed control scenarios based on deep reinforcement learning.	Evaluated considering a multi-instance scenario and CPEs.	Does not address.	Evaluated in terms of migrated flows between controllers.
[4]	Evaluate ODL and ONOS controllers robustness.	Evaluated in terms of recovery time.	Evaluated in terms of forwarded packets.	Does not address.
The present work	Evaluate SDN scalability, resiliency and load balancing.	Evaluated considering a multi-instance scenario and incremental switches.	Evaluated in terms of packet losses and recovery time.	Evaluated considering proactive and reactive forwarding in terms of the round trip time.

3. SDN Features

3.1. Network Resiliency

OpenFlow-based controllers have a centralized view of the whole network and orchestrate all the elements. Therefore, solutions with optimal path computation should be included in new network operation systems, providing minimal convergence times, network resiliency, and efficient use of the available resources. This work evaluates an online path computation solution for a resilient SDN network based on the ONOS.

3.2. Load Balancing

The multiple spanning tree protocol (MSTP) is commonly used in traditional networks to implement a trunk load-balancing solution in a loop-free topology with alternative paths. Later in this work, we implement an SDN deterministic load balancing scenario based on OpenFlow and the proactive forwarding mechanism for the ONOS controller.

3.3. Scalability

In SDN, abstracting the lower-level functionalities allows the control plane to manage the overall network through the controller. However, the paradigm of centralized control results in scalability issues. The latter means that the SDN controllers must have enough processing power to deal with the high amount of flow and be aware of the number of devices a single controller can handle.

4. System Model Description

In Figure 1, the system model is presented. First, an Ubuntu Server, version 20.04 LTS, 64 bits, 2 GB of RAM, and two CPUs is used for each controller (ONOS, ODL). Furthermore, two machines with Ubuntu 18.04, 64 bits, 1 GB RAM, and two CPUs are used for the Carleton Benchmark Suite (Cbench) and Mininet. The former refers to a benchmarking framework used for evaluating OpenFlow Controllers. The latter allows the instantiation of Open vSwitches and virtual hosts.

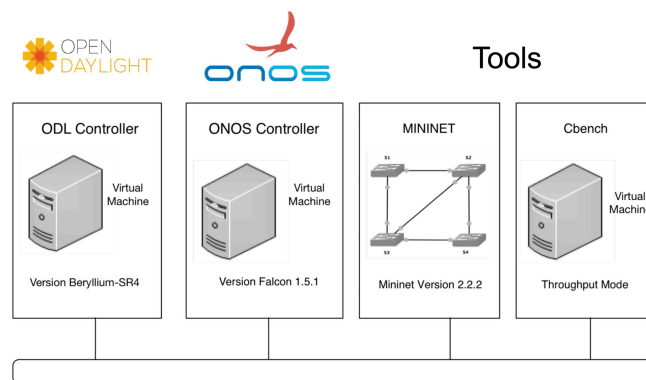


Figure 1. System model configuration includes (i) two OpenFlow SDN controllers, (ii) one Mininet instance used to implement the network topology (Open vSwitches and hosts), and (iii) Cbench used to evaluate the SDN controllers.

The pre-eminent open-source controller deployments are ODL and the ONOS. It is crucial to compare ODL with the emerging open-source ONOS controller to provide a point of reference between controllers and select one controller for the failure scenarios.

Figure 2a presents the process to evaluate SDN controllers (ONOS and ODL). First, the topology is installed using the Cbench tool, including an Open Vswitch and 1000 virtual hosts. Then, the switch creates as many packets as the switch's buffer allows, without waiting for a reply; while the buffer is not full, each packet is queued in the controller and generated outgoing packets (*Flow_Mod*) are counted. We want to measure how many packets can be handled/processed by each controller so that the controller's throughput (flows/s) is calculated.

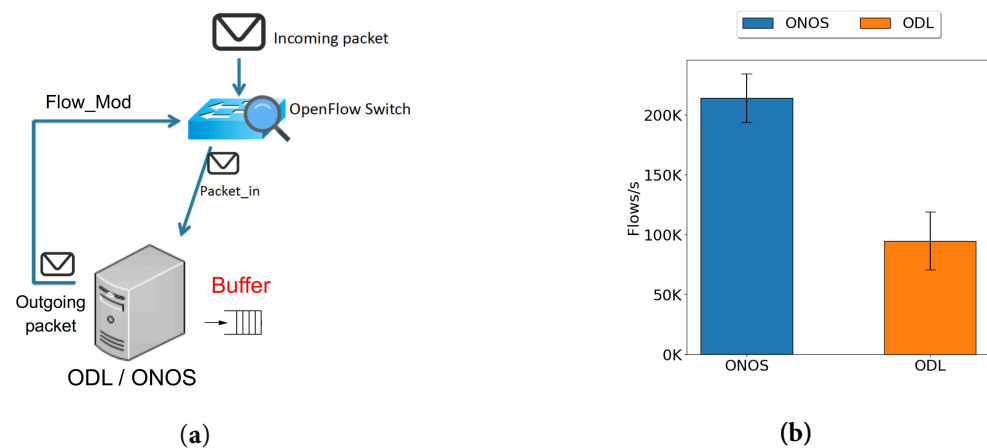


Figure 2. (a) Cbench tool process used to evaluate the performance of SDN controllers. The topology includes an Open Vswitch and 1000 virtual hosts. (b) SDN controller's throughput (flows/s). Results are presented with a 95% confidence interval (CI) obtained from ten simulation rounds with independent seeds.

Simulation results are presented in Figure 2b. Here, the ONOS controller outperforms ODL in 119,402 flows/s, which is a considerable difference. ODL shows roughly half of the capacity when compared to the ONOS. To increase confidence in the results, we conducted ten test rounds for both ONOS and ODL controllers, discarding the first and last values by the warm-up and cool-down effect. The measurement was performed under steady-state conditions without network changes during the test.

ONOS software results in a more reliable network operating system with better performance when compared to ODL. Therefore, it is suitable for evaluating the following experiments. The following section describes configurations of the different test scenarios using the ONOS as the central SDN controller.

5. Results

5.1. SDN Resiliency Scenario

Figure 3a shows the topology to evaluate network resiliency. Note that several routes are enabled to communicate H1 and H2. The scenario uses the reactive forwarding (RF) mechanism. The RF installs flow entries on-demand after a sender starts to transmit the initial packets.

The testing scenario simulates random link failure to evaluate the SDN network resiliency. In addition, to compare results, the same network topology shown in Figure 3a is implemented using legacy switches. The Spanning Tree Protocol (STP) is also configured as a typical implementation for reliable networks at L2, with alternative loop-free paths.

Figure 3b shows the results of SDN and legacy network simulations regarding the network's recovery time and packet losses. We notice that an SDN network with an OpenFlow RF mechanism responds faster in the case of a networking event like a link failure. It requires 2 s for recovery, with only one packet lost. Conversely, in traditional networks, STP requires 34 s for link recovery with 33 packet losses. The latter is due to the transition from blocking to forwarding mode to prevent temporal loop conditions. Furthermore, if Rapid-STP is used, it requires 4 s to recover from the link failure, which doubles the time necessary compared to SDN.

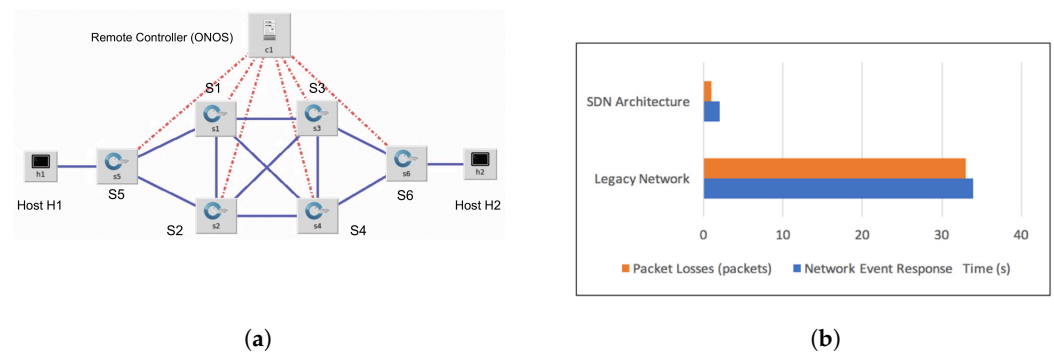


Figure 3. (a) Network configuration for resiliency evaluation. The topology comprises six OpenFlow vSwitches (based on Mininet), a single remote controller (ONOS Controller), and two virtual hosts (H1, H2). (b) Network resiliency measured in terms of recovery time and packet losses for legacy and SDN networks.

5.2. SDN Load-Balancing Scenario

The network topology to evaluate the load balancing scenario is presented in Figure 4a. The idea consists of distributing traffic flows between H2 and H1 and H3 and H1 according to directives configured in the ONOS controller. Two scenarios are evaluated: (i) proactive forwarding (PF), which guarantees load balancing, and (ii) reactive forwarding (RF), where load balancing is not guaranteed.

First, the PF approach provides a deterministic load-balancing scenario. It sets end-to-end connectivity between H3 and H1, which goes through PATH 1, S6–S4–S2–S5, in Figure 4a. Traffic H2–H1 goes through PATH 2, S3–S1–S5, in Figure 4a. Second, with the RF approach, flow entries for H2–H1 and H3–H1 are automatically installed after the first packet in the network (on-demand flow installation). The optimal path computation is based on the shortest path algorithm (Dijkstra), where the number of hops is the weight metric. Note that resulting paths may use the same links, which means a congested route and non-efficient use of resources (i.e., load balancing is not guaranteed).

In Figure 4b, we use the round trip time (RTT) to assess the network's reaction time. The RF requires, on average, 18s more to establish communication between H2 and H1 and 42 s for H3 and H1. Although the PF approach outperforms RF in both cases, by considering scalability, a non-scalable solution occurs due to the limited memory at the switch tables where flows should be installed permanently.

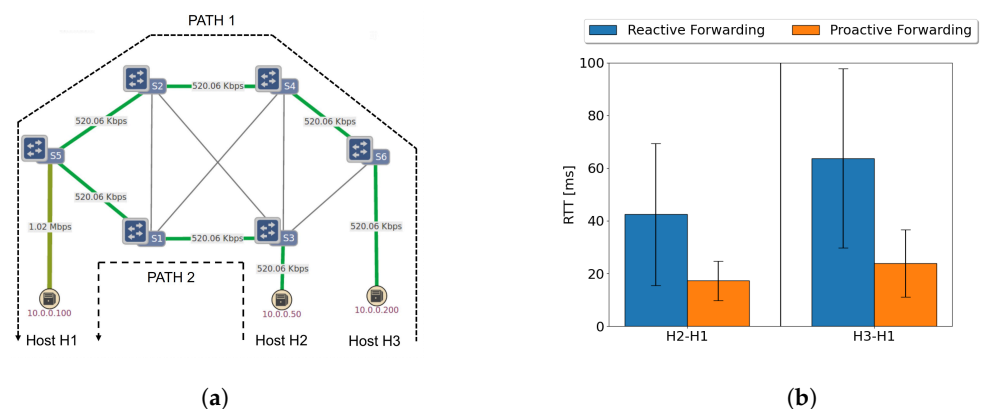


Figure 4. (a) The load-balancing scenario includes three virtual hosts (H1, H2, H3), six OpenFlow vSwitches (based on Mininet), and a remote SDN controller (ONOS) implementing proactive and reactive forwarding. (b) RTT results for proactive forwarding with load balancing and reactive forwarding without guaranteed load balancing.

5.3. SDN Scalability Scenario

One of the mechanisms used to manage SDN scalability issues comprises implementing multi-server instances (i.e., distributed control). This implies an intrinsic controller of the controller channel, in charge of controller synchronization. Note that a distributed control design also increases the system's resilience at the control plane level.

Figure 5b shows the scenario to evaluate the scalability of the multi-instance approach in SDN. Here, service management is distributed between controllers in the ONOS Cluster; see the blue and brown boxes in Figure 5b. As a point of reference, the performance of a single instance (one single ONOS controller) is evaluated and compared to the multi-instance (two ONOS controllers) implementation.

Figure 5b shows the results for single and multi-instance scenarios to evaluate this solution for SDN scalability issues. We can see the multi-instance case has a significant impact on network performance. When considering a different number of switches [1–64], the SDN performance degradation begins to be noticeable when eight switches are installed in the network. Degradation in the performance ranges from 1% to 21% compared to the single-instance case. This is due to the synchronization process between controllers, resulting in a degradation in the whole system performance.

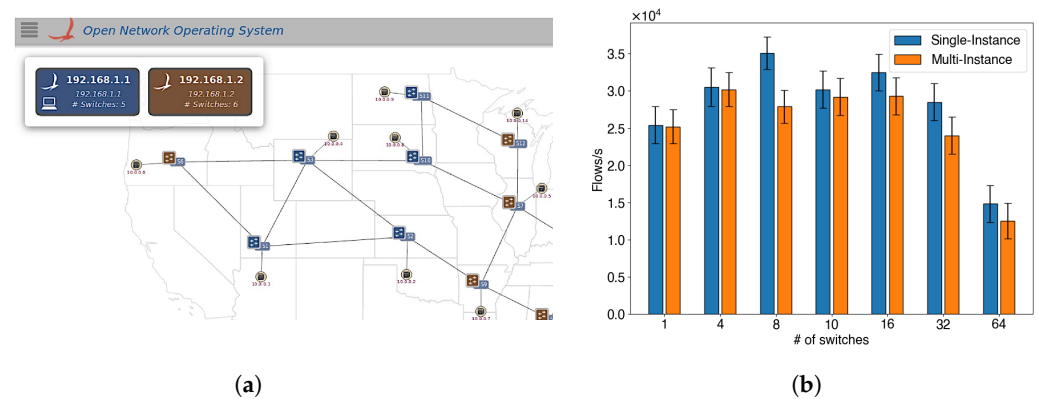


Figure 5. (a) Distributed control design to evaluate scalability. Topology is composed of 1 to 64 switches (based on Mininet) and two controllers (ONOS) configured in a cluster (multi-instance). (b) Performance of single and multiple instance implementation.

6. Conclusions

This work evaluates different failure scenarios in both data and control planes. The simulation results show that the ONOS controller handles double the throughput compared to the ODL controller. Regarding the network's resiliency, at the data plane, SDN requires half the time compared to traditional networks. The SDN load-balancing feature can be implemented by setting deterministic directives in the controller. However, this leads to scalability issues at the data plane due to the limited memory in switches. Lastly, the SDN multi-instance solution impacts the network performance by between 1% and 21% compared to the single-instance case. In this regard, the multi-instance scenario improves resiliency and scalability, although the controllers' synchronization channel limits the maximum number of cluster instances.

Author Contributions: Conceptualization, P.B.B. and J.C.; methodology, P.B.B. and L.U.-A.; software, P.B.B.; validation, P.B.B. and L.U.-A.; formal analysis, P.B.B. and L.U.-A.; investigation, P.B.B. and J.C.; data curation, P.B.B.; writing—original draft preparation, P.B.B. and J.C.; writing—review and editing, P.B.B., J.C. and L.U.-A. All authors have read and agreed to the published version of the manuscript.

Funding: This research received no external funding.

Institutional Review Board Statement: Not applicable.

Informed Consent Statement: Not applicable.

Data Availability Statement: Data are contained within the article.

Conflicts of Interest: The authors declare no conflict of interest.

References

1. Park, K.; Sung, S.; Kim, H.; Il Jung, J. Technology trends and challenges in SDN and service assurance for end-to-end network slicing. *Comput. Netw.* **2023**, *234*, 109908. [\[CrossRef\]](#)
2. Ouamri, M.A.; Azni, M.; Singh, D.; Almughalles, W.; Muthanna, M.S.A. Request delay and survivability optimization for software defined-wide area networking (SD-WAN) using multi-agent deep reinforcement learning. *Trans. Emerg. Telecommun. Technol.* **2023**, *34*, e4776. [\[CrossRef\]](#)
3. Ouamri, M.A.; Barb, G.; Singh, D.; Alexa, F. Load Balancing Optimization in Software-Defined Wide Area Networking (SD-WAN) using Deep Reinforcement Learning. In Proceedings of the 2022 International Symposium on Electronics and Telecommunications (ISETC), Timisoara, Romania, 10–11 November 2022; pp. 1–6. [\[CrossRef\]](#)
4. Ruchel, L.V.; Turchetti, R.C.; de Camargo, E.T. Evaluation of the robustness of SDN controllers ONOS and ODL. *Comput. Netw.* **2022**, *219*, 109403. [\[CrossRef\]](#)

Disclaimer/Publisher's Note: The statements, opinions and data contained in all publications are solely those of the individual author(s) and contributor(s) and not of MDPI and/or the editor(s). MDPI and/or the editor(s) disclaim responsibility for any injury to people or property resulting from any ideas, methods, instructions or products referred to in the content.