VOL.39 Nº3 2022
https://doi.org/10.2223\IMECTA.11675.en

Analysis of Impacting Topological Characteristics for the Allocation of Controllers in an SD-WAN Network: RNP case

Vicente Padrenosso[a], Marcelo Vasconcelos[b], André Mendes[c]
[a,b,c]Instituto Militar de Engenharia (IME)
[a]vicente@ime.eb.br, [b]vasconcelos@ime.eb.br, [c]andre.chaves@ime.eb.br

RESUMO: Encontrar uma solução viável e ótima para o problema de alocação de controladores SDN, é uma tarefa desafiadora. Nesse viés, este trabalho foi desenvolvido com o objetivo de expor as características reais de uma rede, que possam impactar na escolha do posicionamento de um controlador SDN. Ademais, são apresentados os experimentos com a ferramenta POCO (Pareto-based Optimal Controller-placement) e com os controladores ONOS e Floodlight, que serviram de comparativo para auxiliar na tomada de decisão quanto ao melhor posicionamento do controlador dentro da rede. Para isso, foi utilizada a topologia da rede Ipê, da RNP (Rede Nacional de Pesquisa), como base desta pesquisa. Em síntese, os resultados demonstraram que devem ser considerados os aspectos, não só estáticos da rede, mas também as características dinâmicas e os aspectos relacionados ao modelo e ao fim a que se destina a utilização dos controladores, fatores que podem impactar no posicionamento.

PALAVRAS-CHAVE: Controladores. Floodlight. ONOS. POCO. SDN.

ABSTRACT: Finding a viable and optimal solution to the SDN controller allocation problem is a challenging task. In this bias, this work was developed with the objective of exposing the real characteristics of a network, which may impact the choice of positioning an SDN controller. Furthermore, the experiments with the POCO tool (Pareto-based Optimal Controller-placement) and with the ONOS and Floodlight controllers are presented, which served as a comparison to assist in decision making regarding the best positioning of the controller within the network. For this, the topology of the Ipê network, from RNP (National Research Network), was used as the basis for this research. In summary, the results showed that not only the static aspects of the network should be considered, but also the dynamic characteristics and aspects related to the model and the purpose for which the controllers are used, factors that can impact the positioning.

KEYWORDS: Controllers. Floodlight. ONOS. POCO. SDN.

1. Introduction

In recent years, Software Defined Networking (SDN) has gained a lot of attention, and has been gaining more strength with current scenarios. Therefore, flexibility, programmability and scalability are some of the benefits brought by software-defined networks, compared to traditional network infrastructure.

In the work by Nunes [1], the SDN network is defined as a network paradigm, in which the forwarding hardware (data plane) is dissociated from the controller's decisions (control plane). In SDN, network intelligence is logically centralized in software-based controllers, and network devices become simple packet forwarders, which can be programmed through open interfaces compatible, for example, with the OpenFlow protocol, [2].

The controller in an SDN can be considered the "brain" of the network, as mentioned in Kreutz's work [3]. Acts as a strategic control point in the SDN network, managing flow control for switches and routers. So, as organizations deploy more SDN networks, controllers are tasked with managing the interconnection between SDN domains using open protocol interfaces such as OpenFlow.

OpenFlow, initially proposed by Stanford University, arose from the need for researchers to run experimental protocols on the academic network. It is the most well-known and accepted architecture for SDN networks, and has a well-defined set of specifications. It is an open protocol that allows a controller to manage network devices and dictate their behavior. The implementation of open standards simplifies network design and operation, allowing researchers to conduct experiments without the need for manufacturers to expose the inner workings of their products, or developers to write vendor-specific control software.

In the architecture established by the OpenFlow protocol, in addition to the controller figure, there is also the role of the OpenFlow switch. The OpenFlow switch has the guarantee of reliability in the exchange of messages with the controller through the SSL (Secure Socket Layer) protocol. Furthermore, the OpenFlow protocol interface guarantees the standardization of the messages sent by the controller to the switch, in order to define the packet forwarding behavior, according to the flow table. The flow table is made up of rules, and each rule consists of actions associated with flows. The entries in this table are updated by the controller, so through this table the switch performs packet forwarding.

Also, according to Lange [4], in the OpenFlow architecture, a logically centralized controller manages the network switches, providing them with rules to establish their behavior in packet operations. Therefore, the position of each controller in the network affects competing objectives such as inter-controller latency, switch-controller latency, and resiliency.

In this sense, an important issue during the implementation of an SDN is the positioning of the controller in the network, that is, deciding where to position an overview of the network, which allows managing resources in a simple and effective way, considering taking advantage of programmable capacity to obtain efficient responses with minimal latency between the nodes and the controller, achieving the maximum throughput between them.

The first approach on the subject was made by Brandon Heller [5], followed by other researchers who added different questions in their observations. However, most research in the area of SDN on controller placement focuses much of its efforts on network analysis without considering its real characteristics, for example, network traffic and the type of controller that will be implemented. Therefore, the different demands of a network in production and the individual load caused by different nodes connected to a controller is still not considered in detail.

Given the above, the main contributions of this work are to investigate, comparatively, which topological characteristics may impact the positioning of controllers in real SD-WAN (Software Defined - Wide Area Network) networks, through a study on the use of the exhaustive method implemented by the POCO tool (Pareto-based Optimal COntroller) [6], which consists of, in summary form, calculating the geographic distances in relation to all positions of a topology containing n nodes, in order to find the best positions for the controller within the set of all possible solutions, where n is the number of nodes in the network. Furthermore, in works related to the POCO tool, no exact information was found regarding the maximum number of supported nodes.

This tool is implemented in Matlab and available as open source software. The POCO approach exhaustively evaluates all possibilities for controller placement in a topology and calculates the latency and resiliency trade-off for each position. Thus, the algorithm does not provide any specific recommendation for a given placement, but returns a set of placements based on the Pareto-based Optimal Placement method [7], which allows network administrators to choose an appropriate placement. Thus, this strategy gives more flexibility in the deployment of an SD-WAN network.

So, to achieve the proposed objectives, the results defined by the POCO tool are compared in the case study of the Ipê network, from RNP (National Education and Research Network/ Rede Nacional de Ensino e Pesquisa), inserted in a virtualized environment, considering its real characteristics of connection and traffic, and also the analysis of active ONOS and Floodlight controllers in this topology, as will be presented in **section 4** - Applied methodology.

As a result, it is demonstrated that in studies on controller positioning, only the considerations about geographic distances and latencies defined by the POCO tool are not enough. Soon, the tests proved that the latency characteristics together with the bandwidth and traffic information are also important factors in the definition of the positioning, in addition, the tests also showed that the aspects related to the model and the use of the controllers must be considered.

After this introduction, the theoretical foundation and motivation will be presented in **section 2**, in Section 3 the related works, in **section 4** the applied methodology, in **section 5** the experiments carried out and the analysis of the results, finally, the conclusion and future work in **section 6.**

2. Motivation

According to more recent works, such as the one carried out by Tamal Das [8], which presents a survey with a deep research on the problem of positioning the SDN controller, and extensively classifies other works in various perspectives, it is possible to verify that the majority of research in the area of SDN on the placement of controllers concentrates a large part of their efforts on network analysis, considering only its static characteristics, that is, data such as calculations of geographic distances and latencies related to these distances, as also observed in Hock's work on the POCO tool [6]. Furthermore, network characteristics and the individual load caused by different nodes connected to a controller are still not considered in detail.

First, a main goal for good controller placement is to minimize latencies between nodes and controllers in the network. However, just looking at delays and static performance measures is not enough. An SDN controller is a strategic point of control on the network. Thus, deciding where to allocate the SDN controllers, which switches will be controlled by each controller, how many controllers are needed so that there is no limiting point, and which controllers have the best performance according to the placement in each scenario are important issues, in order to minimize the cost of covering the network and guarantee a good performance.

3. Related Works

Several studies have been carried out with the objective of dealing with the problem of controller positioning, other studies have worked to compare SDN controllers in isolation, focusing only on controller performance. However, few studies have helped in decision making to select a controller with the best performance, taking into account its positioning within a network with real characteristics.

The controller positioning problem in the SDN architecture was introduced by Heller [5], where an optimization regarding the latency of the nodes until the designated controller was performed. In this article, the author mentions that for WANs (Wide Area Networks), the best positioning depends, among other metrics, on latency.

In the field of the positioning problem, Heller [5] shows the network performance by varying the position of the controllers in the network. Thus, finding the location and number of controllers that will be needed is a challenging task in a network architecture such as SD-WAN.

In the work of Stanislav Lange [4], POCO (Pareto-based Optimal COntroller placement) [6] is presented, a framework for finding the optimal placement of the controller so that connectivity between the switches and the controller is maximized taking into account the capacity of the controller. First, the authors propose a brute force algorithm, but it is only valid for small networks. Then, the authors propose heuristics to solve the problem of positioning controllers in large networks. For that, they use an algorithm based on Pareto Simulated Annealing [9]. In this work, the authors focus on maximizing network resilience. Where they considered placing controllers in a dynamic SDN network where latency variations exist between controllers and their switches. However, they do not

VOL.39 Nº3 2022
https://doi.org/10.2223\IMECTA.11675.en

consider the position of controllers taking into account the dynamic allocation of the network.

Placement needs to be chosen carefully. The POCO framework, has the ability to handle small and medium-sized topologies, which provide the solution in seconds. However, for large-scale networks, the exhaustive evaluation requires a considerable amount of computational effort and memory usage. It is in this context that the search for a computational solution closer to real environments becomes necessary.

The work by Rastogi and Bais [10] performs a comparative analysis in terms of traffic capacity. The main objective of this work is to present an analysis between two controllers, called Pox and Ryu, respectively, in terms of traffic handling capacity. The Mininet emulator was used to emulate the SDN controllers environment, and thus monitor traffic performance. However, the work did not consider the real characteristics of the network, such as the capacity and occupancy of the links.

As for the comparison of controllers, the work carried out by Amin Tootoonchian [11], one of the first comparative studies of SDN controllers, considered a limited number of controllers (NOX, NOX-MT, Beacon and Maestro) focusing only on controller performance. With the advancement of technologies such controllers are already considered outdated.

Bondkovskii's work [12] makes a qualitative comparison between two open source SDN controllers, OpenDaylight and Open Network Operation System (ONOS). The study focuses on the Northbound interface of these devices.

More recent research, like the one carried out by Lusani Mamushiane [13], studies and evaluates the performance of some popular open source controllers like ONOS, Ryu, Floodlight and OpenDaylight in terms only of latency as a metric, using an OpenFlow benchmarking tool called Cbench.

Also, in the work carried out by Tamal Das [8] a survey is presented with a deep research on the problem of positioning the SDN controller, which extensively classifies the existing works in several perspectives.

In the work of Ola Salman [14] a qualitative evaluation of open source SDN controllers is performed (MUL, Beacon, Maestro, ONOS, Ryu, OpenDaylight, Floodlight, NOX, IRIS, Libfluid-based and POX). Metrics evaluated are latency and throughput performed over a variable number of switches. The obtained results suggest that MUL and Libfluid-based have the best throughput performance, while Maestro showed the best latency performance. Other related works are compared in **table 1**.

In this work, ONOS and Floodlight controllers will be compared, which will be evaluated in two positioning scenarios in an SD-WAN network, according to the methodology that will be presented in **section 4**.

Related works - Multiple Objectives

Trabalho	Positioning	Switch-Controller Latency	Latency between Controllers	Controller Capability	Resilience	Reliability	Load balancing	Switch-controller traffic	Number of controllers	Real Topology Analysis	Controller application	Public	Qualis
[6]	✓	✓	✓							✓		IEEE	A1
[4]	✓	✓	✓		✓		✓			✓		IEEE	A1
[15]	✓	✓		✓			✓					IEEE	A1
[16]	✓	✓				✓	✓					IEEE	A1
[17]	✓	✓	✓	✓	✓	✓				✓		IEEE	A1
[18]	✓	✓	✓	✓		✓				✓		IEEE	A1
[19]	✓	✓				✓	✓			✓		IEEE	A1
[20]	✓	✓	✓	✓			✓			✓		Springer	A1
[21]	✓	✓	✓		✓		✓			✓		IEEE	A1
[22]	✓	✓	✓		✓		✓			✓		IEEE	A1
[23]	✓	✓	✓		✓		✓		✓	✓		IEEE	A1
[24]	✓	✓		✓			✓			✓	✓	IEEE	A1

Tab. 1: Related works.

https://doi.org/10.2223\IMECTA.11675.en

4. Applied Methodology

This work is based on the study of the exhaustive method used in the POCO tool (Pareto-based Optimal Controller) to compute the optimal positioning of SDN controllers in the Ipê network, of RNP.

Mininet [25] was also used as an SDN network emulator used in a virtualized environment to analyze and compare the impact of positioning using ONOS and Floodlight controllers in the operation of networks enabled with OpenFlow. **Figure 1**, shows an example of the RNP topology in POCO's graphical interface.

Fig. 1 – Example of the POCO graphical interface.

The Mininet emulator was chosen mainly because it is possible to emulate real networks and also for the following reasons: it supports several SDN controllers; hosts are Linux with "real" features that can potentially run any program that runs on Linux; enables traffic analysis by capturing packets using, for example, Wireshark or TCPDUMP; allows connectivity tests with ping / fping or performance measurements with iPerf; it is possible to create links with specific bandwidth (bandwidth) and latencies, in addition to deactivating and activating the links.

There are a number of controllers available in the literature. Among the best known are NOX, POX, Floodlight, OpenDaylight (ODL), Open Network Operating System (ONOS) and RYU. However, from a practical implementation perspective, it is very difficult to determine which controller will perform best on any type of network. In the research carried out by Zhu [26], it is possible to verify a good comparative analysis on the resources of several controllers.

In this sense, the criteria for choosing the controllers analyzed in this research include: interfaces (supported protocols and versions), REST API, graphical user interface (GUI), modularity, supported operating system (OS), development and support partnerships, documentation , programming language, virtualization, application area and architecture. So, ONOS and Floodlight controllers were chosen.

The entire test environment was configured in a virtual machine (VM), created with the Ubuntu 16.04.05 LTS operating system, containing the default kernel version 4.4.0-87-generic, 8 GB RAM memory, 8 processors and 30 GB internal storage space. This VM was configured on a desktop with Windows 10 Home - 64 bits, Intel Core i7-9700 processor and 16 GB of RAM memory.

The topology chosen for the experiments was the Ipê network, from RNP (National Education and Research Network). RNP is the first Internet access network in Brazil, it integrates more than 800 teaching and research institutions in the country, it has 28 nodes located in all the states of the national territory benefiting more than 3.5 million users, it has evolved from telephone lines to very high capacity fiber optic connections, from 64 Kb/s to over 100 Gb/s. All information used in the experiments regarding the connections between the links, bandwidth and incoming and outgoing traffic between the nodes were collected from the traffic panorama of the real RNP topology, available on the institution's website, as shown in **figure *2*.**

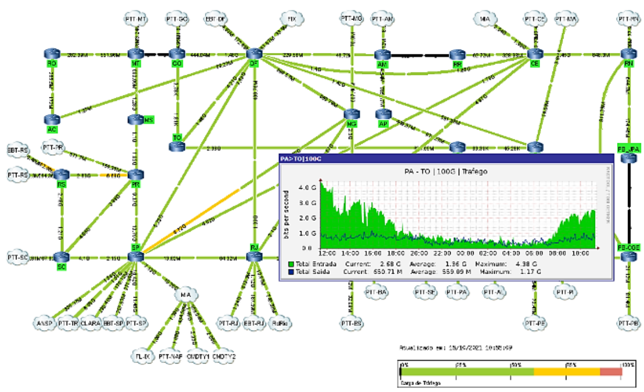

Fig. 2 – Panorama of RNP connections and traffic.

Initially, a study was carried out of the main functionalities of the POCO tool for defining the positioning of the controller in an SD-WAN network. In addition, the ONOS and Floodlight controllers were also compared, which were evaluated in two positioning scenarios, where in the first scenario only distance and latency information were considered, that is, the links were tested free of traffic, using the full capacity of the link. In the second scenario, the occupancy traffic of the directly connected links was included, according to the real information of the RNP traffic panorama. Then, all 28 possible positions for the controller in the RNP topology were evaluated, and these positions were defined as: (i) optimal position and (ii) worst position. By optimal position, it is understood the positioning in which the latency between the nodes and the controller is minimized. By worst position, it is understood the position for which there is greater latency between the nodes and the controller.

Thus, the results defined by testing the positions with the controllers were compared with the results of the positions defined by the exhaustive method of the POÇO tool, which only considered the first scenario, without traffic. These comparisons did not only consider static metrics such as, for example, distances between network nodes, but also quantitative measures, such as data transfer, jitter and packet loss in the traffic scenario.

5. Tests And Analysis Of Results

With the entire test environment configured, tests were carried out with the topologies created, integrating the controllers in scenarios without traffic (**scenario 1**) and with traffic (**scenario 2**). Thus, for each test, a host (srv) was positioned to simulate the controller in each of the 28 positions of the RNP topology and the fping command was executed to obtain latency information from the controller (srv) in that particular position, bound for all other hosts in other positions.

To obtain better statistical results and achieve greater efficiency in the comparisons, 30 samples of the tests with fping ($n = 30$) and a confidence level of 95% were considered. So, in this case, 30 tests were performed with fping for each of the topology positions, and in each test 30 ICMP packets were sent to each destination.

For the execution of the tests in **scenario 2**, where the network traffic was also considered (occupation of the links), the iPerf program was used to simulate the traffic, considering the link capacities and the real traffic values extracted from the traffic panorama of the RNP topology. After that, the test with fping was also performed, in the same way as in the scenario without traffic.

Iperf is a tool for active measurements of the maximum achievable bandwidth in IP networks, being able to perform packet injection (both TCP and UDP) to measure the performance of computer networks. It supports adjustment of various parameters related to time, buffers and protocols (TCP, UDP, SCTP with IPv4 and IPv6). For each test, it reports bandwidth, loss, and other parameters.

Thus, the results of the tests with the controllers in the scenario without traffic, considering the best and the worst position, were compared with the results obtained by the POCO tool calculations. Therefore, these results of the controllers could confirm the positions established by the POCO tool or showed new positions, considering other factors such as, for example, the capacity of the links. Tests in the traffic scenario showed the importance of considering metrics such as link capacities, data transfer capacity, jitter and packet loss, in addition to just considering the distance between nodes.

For the execution of the tests with the POCO tool, in order to compare more results, the positions and topology information of the RNP that was operational in three different periods (November 2020, March 2021 and July 2021) were used, since some changes were observed in these topologies, for example, it was noted that some links ceased to exist or their capacities were expanded, and other new links were established.

5.1 Poco Placements

As first results, the positions defined by the POCO tool for the three mentioned topologies were examined.

VOL.39 Nº3 2022
https://doi.org/10.2223\IMECTA.11675.en

For this, the failure-free scenario was considered, with only one controller (k = 1). In these results, it is possible to notice that the average latency values are identical or very close for the same positions of the tested topologies. Therefore, it is noted that the results were maintained with the exhaustive method used by the POCO tool for latency calculations based only on geographic distances. Therefore, it can be concluded that even with the changes in the topology links, but without changes in the physical positions of the nodes, the results did not show significant changes, that is, the distances and latencies were maintained, as shown in **figure 3.**

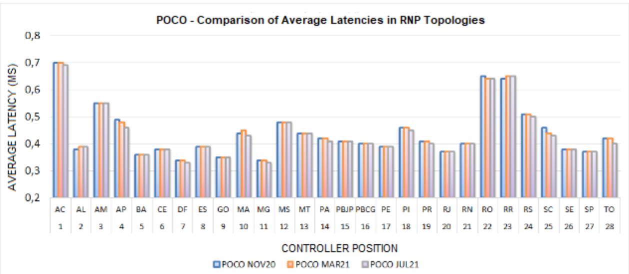

Fig. 3 – Classification of POCO positions.

So, for the three tested topologies, position 7 (DF) was defined as the best position, with the lowest average latency, and position 1 (AC) as the worst position, with the highest latency. Position definitions are represented by larger circles in **figure 4 and 5.**

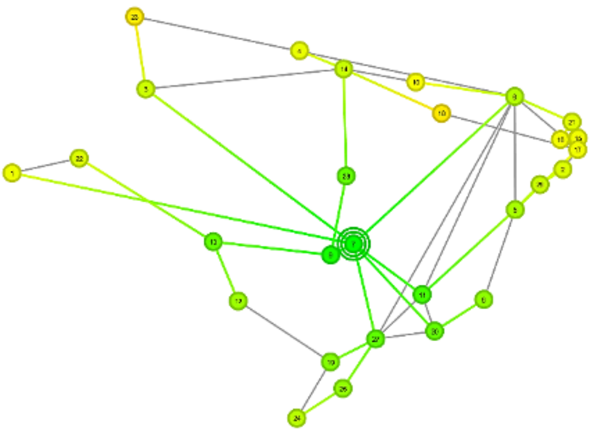

Fig. 4 – Classification of the best POCO position.

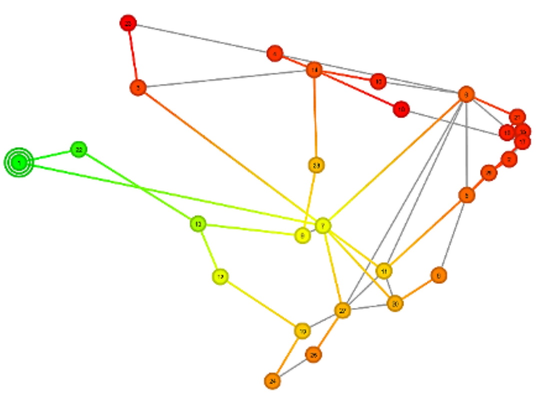

Fig. 5 – Classification of the worst POCO position.

In Fig. 4, it is also possible to see that the node defined with the lowest latency (7-DF) is also the most centralized node in the topology, therefore, the one with the shortest distance between two points, considering only the geographic distances between the nodes with established links . Also, it appears that node 7 (DF) has a good amount of redundant links, which facilitates the connection with the other nodes in the network. From another perspective, the reason that node 1 (AC) was defined as the worst position can be justified by its geographical position being the most distant in relation to the other nodes in the topology. In this way, it becomes another indication that the geographical distance between the nodes has a lot of weight in the decisions made using the POCO tool, which may not be enough for the decision of the controller's positioning.

5.2 Controller Ratings

After classifying the positions using the POCO tool, the topologies were also tested on Mininet using the ONOS and Floodlight controllers, considering the two proposed scenarios. Also, tests were performed considering critical performance metrics, such as amount of data transmitted, jitter and data loss.

5.2.1 Scenario 1 – Without Occupying The Links

In this section, the comparisons of the results of the latency tests between the POCO tool and the controllers in the first scenario, without traffic, are shown. To carry out these experiments, the topology that was operational in November 2020, when the experiments were started, was used.

To carry out the first test, already on the Mininet with the scenario and the topology ready, the links were used free of traffic, with the full capacity of the link. Then, for each test, a host (srv) was positioned to simulate the controller in each of the 28 positions and the fping command was executed to get the latency information from the controller (srv) in that particular position to all other hosts in the other positions.

To obtain better statistical results and achieve greater efficiency in the comparisons, 30 samples of the tests with fping (n = 30) and a confidence level of 95% were considered. To reach this number of samples (30 pings), tests were performed with 10, 20, 30, 40, 50 and 100 pings with the ONOS controller between two positions of the topology studied in this research. Thus, the best case was found, as shown in **figure 6**. So, in the results shown below, 30 ICMP packets were sent to each destination.

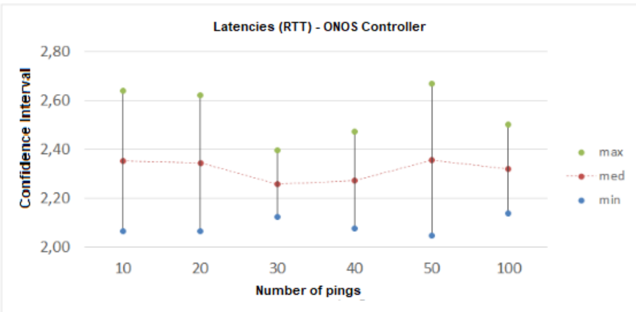

Fig. 6 – Topology Nov 20 - without occupying the links (scenario 1).

To exemplify what was previously described, follow the command that was executed from the controller host: fping -s -g 10.0.0.201 10.0.0.228 -c 30, according to parameters in **table 2**.

Tab.2: Fping parameters

-s	Prints statistics at the end of the test
-g	Generates a list of targets from an IP range, with the starting and ending IP addresses
-c	Specifies the number of packets to be sent to each destination. In this test, 30 ICMP packets were sent to each destination

The IP range (-g 10.0.0.201 10.0.0.228) means that the command was issued from the controller with IP 10.0.0.200 to all hosts in the range which varies in sequence from host 1, connected to position 1 (switch 1), with IP 10.0.0.201, to host 28, in position 28 (switch 28), with IP 10.0.0.228.

After testing the 28 positions, the results of the average latencies were compared with the results calculated by the POCO tool, as shown in the graph in **figure 7**. In this graph, it is possible to verify, already as a partial result for this scenario, that the positions defined by the POCO tool were contradicted by the tests with the controllers.

Firstly, the tests with the ONOS and Floodlight controllers carried out without occupation of the links (scenario 1), demonstrated that for both controllers position 28 (TO) was considered as the best position, contrary to the choice calculated by the POCO tool, which defined position 7 (DF) as the best position. In defining the worst position, the test results considered position 22 (RO), while the POCO tool considered position 1 (AC). These positions are also identified by markers in the graph in **figure 7**.

So, it can be observed that with the inclusion of controllers in the network, other parameters were considered, in addition to distances, such as, for example, the processing of the controllers and the network discovery processes and definition of the best paths. Thus, in the scenario without traffic, the positions defined as the lowest and highest latency contradict the POCO definitions.

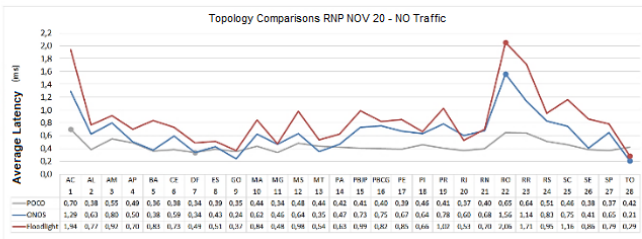

Fig. 7 – Comparison of ranking positions (scenario 1).

Another fact observed in the tests was that the ONOS controller performed better compared to

VOL.39 Nº3 2022
https://doi.org/10.2223\IMECTA.11675.en

Floodlight in terms of latency, shown in the graph in **figure** 8. In this graph, it is possible to see that the results with ONOS showed lower average latency in most of the positions tested, which means that this controller had the best use in the processing demanded by each flow received in the scenario without traffic.

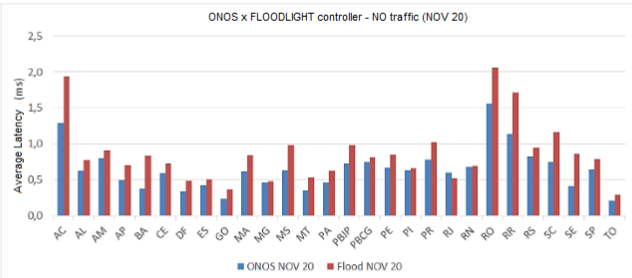

Fig. 8 – Comparison of latencies (scenario 1).

5.2.2 Scenario 2 – With Occupation Of The Links

In this scenario, experiments were developed considering the real network traffic parameters (RNP) that can significantly impact the positioning of the SDN controller. Therefore, the occupation of the links was considered, that is, the traffic on the network.

So, before running the fping tests to check the latencies, traffic was generated between each pair of directly connected nodes, according to the information obtained from the traffic overview of the real RNP network. This traffic was generated by the iPerf tool, using the UDP protocol to eliminate the possibilities of connection errors that could be caused by the TCP protocol. Also with the same UDP traffic it is possible to analyze other critical metrics, such as data transfer, jitter and packet loss, which will be presented in the next Section.

Initially, to generate traffic between two hosts using iPerf, one of the hosts must be the server, which "listens" to the traffic, and the other the client, which sends the traffic. So, as in these experiments the traffic for simulating the occupation of links was generated simultaneously between all 28 nodes, each node acted as client and server at the same time. Then, during the occupation of the network with the transfer of traffic in the environment simulated by Mininet, the fping command was once again executed, according to the parameters described in the tests of the scenario without traffic.

In this scenario, with traffic, a large computational capacity was required, mainly CPU and memory, which made tests with the complete RNP topology difficult. In this case, during the generation of traffic between the hosts, the virtualized system did not support the traffic and the controllers were not able to process all the traffic demanded during a period of time necessary for the execution of the tests with fping, therefore, the traffic was interrupted, and it was not possible to run the latency tests. Thus, as an attempt at a solution, the entire flow of packets was scaled with a decrease in bandwidth and the amount of traffic generated, however, even with these changes, it was still not possible to run the tests with a minimum of efficiency, considering the simultaneous input and output traffic in the 28 nodes of the network.

In view of this, some more recent works with possible solutions to the problem were researched. Among them, the work presented by Ahmadi [27] proposes a heuristic algorithm called Multi-Start Hybrid Non-Dominated Sorting Algorithm (or MHNSGA) to solve the controller placement problem effectively. However, in the context of multi-objective optimization, in most cases, there may not be a single solution that optimizes all considered objectives.

The author of this work argues that the results of several evaluations showed that the proposed algorithm is able to explore a large part of the search space and obtain an estimate of the Pareto optimal frontier with a high degree of accuracy. However, according to the author, compared to the exhaustive search of the POCO tool, this method may be less accurate, but requires less computational time and memory.

Thus, finding an optimal solution for controller positioning, considering traffic flow characteristics of a real network applied in a research environment with viable computational capacity, still requires further investigation of the problem.

While analyzing the flow of traffic on the network is a crucial factor in making a decision about the placement

VOL.39 Nº3 2022
https://doi.org/10.2223\IMECTA.11675.en

of SDN controllers, some other important factors must also be considered. These factors include the capabilities of the controllers and the loads on the switches.

So, still considering the characteristics of a real network in a scenario with traffic, in the next section some critical metrics will also be compared, such as data transfer, jitter and packet loss, which can influence the choice of placement and the behavior of the controller in an SDN.

5.2.3 Critical Performance Metrics

In this section, other tests will be presented, considering some critical metrics that can impact the positioning and also the operation of the controller. In this sense, the packet transfer tests will be presented in **section 5.2.4,** the tests with the jitter metric in **section 5.2.5** and the result of packet loss tests in **5.2.6.** Finally, these three metrics will be compared in **section 5.2.7,** observing the performance of ONOS and Floodlight controllers inserted in the tested topologies.

The displayed results were obtained by generating traffic on the network using the iPerf tool, in the same way as performed in the tests with **scenario 2**, described in 5.2.2. However, this traffic was reduced and generated during the total time of 60 seconds. Thus, this flow period was sufficient to evaluate these metrics without running out of computational resources. Then, the generated information was analyzed and compared, as will be presented in the next **sections**.

5.2.4 Amount Of Data Transferred

Each directly connected link was occupied with its respective maximum incoming and outgoing traffic, according to information from RNP's traffic panorama verified in the analyzed period. Then, after occupying the links and generating traffic during the total period of time analyzed (60 sec), average transfer data were collected on each of the links, that is, the average amount of data transferred (Mbps) in this test period.

The graph in **figure 9** shows the comparison of ONOS and Floodlight controllers on the amounts of data transferred (Mbps). In these graphs, it can be generally verified that the controllers had the same treatment in the processing of the traffic between the nodes, since the same amount of traffic was transferred with the two controllers and no significant differences were evidenced.

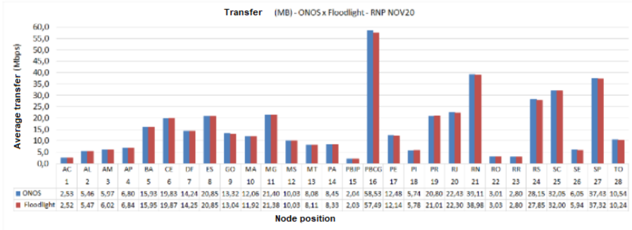

Fig. 9 – ONOS x Floodlight - Comparison of the transfer metric.

5.2.5 Jitter

In a simple and summarized way, jitter is the variation in latency, that is, this metric can be defined as the measure of variation or "fluctuation" of the time that a data packet takes to go to a destination and return.

The graphs in **figure 10** show that the ONOS controller had a slight disadvantage in the isolated comparison of the jitter metric with the Floodlight controller, as it presented a greater variation in the total delay, considering all the positions tested. Therefore, other factors must also be considered.

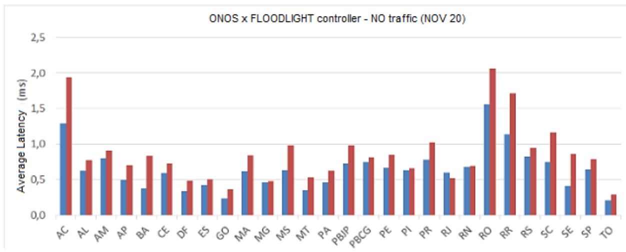

Fig. 10 – ONOS x Floodlight - Comparison of the jitter metric.

5.2.6 Package Loss

These experiments were performed in failure-free scenarios, therefore, only losses in packet transfers were considered, failures in links or devices were not simulated.

The graphs in **figure 11** show the comparison between controllers on the measured packet loss. The results show that the ONOS controller had an average packet loss lower than the loss tested with the Floodlight controller

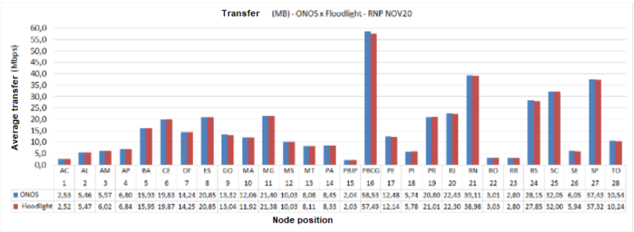

Fig. 11 – ONOS x Floodlight - Comparison of the loss metric.

5.2.7 Comparison Of Critical Metrics

In summary, in the comparison between the two controllers, the same amount of packets was transferred, with a small advantage for the ONOS controller, however, not very significant. The Floodlight controller showed less delay variation (jitter), however, there was more packet loss. **Table 3** shows the described summary, where the highest throughput, the lowest jitter and the lowest packet loss are represented.

Tab. 3 - Comparison of critical performance metrics.

Comparison of critical metrics			
	Transf	Jitter	Perda
ONOS	✓		✓
Floodlight		✓	

Therefore, with the same amount of packets transferred in both tests with the controllers, the fact that ONOS presented less packet loss can be justified by the better treatment given to the flow received at the controller, which is connected to its storage capacity. On the other hand, Floodlight presented lower jitter, which shows the low storage of data in RAM memory, in the buffer and in the cache, however, this can cause greater data loss and can reduce the controller's responsiveness.

Thus, it is also verified that, in addition to the metrics of distances, latencies and considerations of the traffic in the network, it is also important to identify the priorities of the traffic and decide which will be the most appropriate controller for this purpose.

6. Conclusion and future work

In this work, a study on the problem of positioning controllers in SDN was presented. In this context, the implementation carried out in the POCO framework was described, which, in general, presented good results considering the static measures of distance and latency, however, with the results presented in the tests including the controllers in the network, it is possible to highlight that the real characteristics of the network can significantly impact the choice of positioning of an SDN controller. Also, it is possible to observe that the choice of a certain type of controller must be thought according to the desired purpose, as its characteristics can also impact the positioning.

Finally, it should be emphasized that efficient controller placement attempts to improve the performance of metrics such as latency, traffic priorities, loss, and so on. However, the study for the controller placement problem can still comprise several different solutions. Therefore, it is expected to expand the study for the development and improvement of tools for positioning controllers, based on performance measures of real networks, and that these tools can analyze the problem of positioning controllers in the presence of data traffic.

References

[1] B. A. A. Nunes et al. "A Survey of Software-Defined Networking: Past, Present, and Future of Programmable Networks". In: IEEE Communications Surveys Tutorials 16.3 (Third 2014), pp. 1617–1634.
[2] NICK MCKEOWN ET AL. "OpenFlow: enabling innovation in campus networks". In: ACM SIG-COMM Computer Communication Review 38.2 (2008), pp. 69–74.
[3] DIEGO KREUTZ ET AL. "Software-defined networking: A comprehensive survey". In: Proceedings of the IEEE 103.1 (2015), pp. 14–76.
[4] STANISLAV LANGE ET AL. "Heuristic approaches to the controller placement problem in large scale SDN networks". In: IEEE Transactions on Network and Service Management 12.1 (2015), pp. 4–17.
[5] BRANDON HELLER, ROB SHERWOOD, AND NICK MCKEOWN. "The controller placement problem". In: ACM SIGCOMM Computer Communication Review 42.4 (2012), pp. 473-478.
[6] DAVID HOCK ET AL. "POCO-framework for Pareto-optimal resilient controller placement in SDN-based core networks". In: 2014 IEEE Network Operations and Management Symposium (NOMS). IEEE. 2014, pp. 1–2.
[7] GUODONG WANG ET AL. "The controller placement problem in software defined networking: A survey". In: IEEE Network 31.5 (2017), pp. 21–27.
[8] TAMAL DAS, VIGNESH SRIDHARAN, AND MOHAN GURUSAMY. "A survey on controller placement in SDN". In: IEEE communications surveys & tutorials 22.1 (2019), pp. 472-503.
[9] PIOTR CZYZZAK AND ADREZEJ JASZKIEWICZ. "Pareto simulated annealing - a metaheuristic technique for multiple--objective combinatorial optimization". In: Journal of Multi-Criteria Decision Analysis 7.1 (1998), pp. 34–47.
[10] ABHISHEK RASTOGI AND ABDUL BAIS. "Comparative analysis of software defined networking (SDN) controllers—In terms of traffic handling capabilities". In: 2016 19th International Multi-Topic Conference (INMIC). IEEE. 2016, pp. 1–6.
[11] AMIN TOOTOONCHIAN ET AL. "On Controller Performance in Software-Defined Networks". In: 2nd {USENIX} Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE 12). 2012, pp. 1–6.
[12] ANDREI BONDKOVSKII ET AL. "Qualitative comparison of open-source sdn controllers". In: NOMS 2016-2016 IEEE/IFIP Network Operations and Management Symposium. IEEE. 2016, pp. 889–894.
[13] LUSANI MAMUSHIANE, ALBERT LYSKO, AND SABELO DLAMINI. "A comparative evaluation of the performance of popular SDN controllers". In: 2018 Wireless Days (WD). IEEE. 2018, pp. 54–59.
[14] Ola Salman et al. "SDN controllers: A comparative study". In: 2016 18th Mediterranean Electrotechnical Conference (MELECON). IEEE. 2016, pp.1–6.
[15] Md Tanvir Ishtaique ul Huque et al. "Large-scale dynamic controller placement". In: IEEE Transactions on Network and Service Management 14.1 (2017), pp. 63–76.
[16] Kshira Sagar Sahoo et al. "Solving multi-controller placement problem in software defined network". In: 2016 International Conference on Information Technology (ICIT). IEEE. 2016, pp. 188–192.
[17] H SOFIA NANING, RENDY MUNADI, AND MUHAMMAD ZEN EFFENDY. "SDN controller placement design: For large scale production network". In: 2016 IEEE Asia Pacific Conference on Wireless and Mobile (APWiMob). IEEE. 2016, pp. 74–79.
[18] PETRA VIZARRETA, CARMEN MAS MACHUCA, AND WOLFGANG KELLERER. "Controller placement strategies for a resilient SDN control plane". In: 2016 8th International Workshop on Resilient Networks Design and Modeling (RNDM). IEEE. 2016, pp. 253–259.
[19] DIOGO MF MATTOS, OTTO CARLOS MB DUARTE, AND GUY PUJOLLE. "A resilient distributed controller for software defined networking". In: 2016 IEEE International Conference on Communications (ICC). IEEE. 2016, pp. 1–6.
[20] AHMAD JALILI, MANIJEH KESHTGARI, AND REZA AKBARI. "Optimal controller placement in large scale software defined networks based on modified NSGA-II". In: Applied Intelligence 48.9 (2018), pp. 2809–2823.
[21] DAVID HOCK et al. "POCO-PLC: Enabling dynamic pareto-optimal resilient controller placement in SDN networks". In: 2014 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS). IEEE. 2014, pp. 115–116.
[22] STANISLAV LANGE et al. "Specialized heuristics for the controller placement problem in large scale SDN networks". In: 2015 27th International Teletraffic Congress. IEEE. 2015, pp. 210–218.
[23] HU BO ET AL. "The controller placement problem for software-defined networks". In: 2016 2nd IEEE International Conference on Computer and Communications (ICCC). IEEE. 2016, pp. 2435–2439.
[24] PENG XIAO et al. "The SDN controller placement problem for WAN". In: 2014 IEEE/CIC International Conference on Communications in China (ICCC). IEEE. 2014, pp. 220–224.
[25] BOB LANTZ, BRANDON HELLER, AND NICK MCKEOWN. "A network in a laptop: rapid prototyping for software--defined networks". In: Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks. 2010, pp. 1–6.
[26] LIEHUANG ZHU et al. "SDN controllers: Benchmarking & performance evaluation". In: arXiv preprint arXiv:1902.04491 (2019).
[27] VAHID AHMADI AND MOSTAFA KHORRAMIZADEH. "An adaptive heuristic for multi-objective controller placement in software-defined networks". In: Computers & Electrical Engineering 66 (2018), pp. 204–228.