Controller selection in software defined networks using best-worst multi-criteria decision-making

Article *in* Bulletin of Electrical Engineering and Informatics · April 2020

DOI: 10.11591/eei.v9i4.2393

3 authors, including:

Esmaeil Amiri
University of Surrey
17 PUBLICATIONS **59** CITATIONS

SEE PROFILE

Mohammad Hossein Rezvani
Qazvin Islamic Azad University
61 PUBLICATIONS **280** CITATIONS

SEE PROFILE

Bulletin of Electrical Engineering and Informatics
Vol. 9, No.4, August2020, pp. 1506~1517
ISSN: 2302-9285, DOI: 10.11591/eei.v9i4.2393

Controller selection in software defined networks using best-worst multi-criteria decision-making

Esmaeil Amiri[1], Emad Alizadeh[2], Mohammad Hossein Rezvani[3]
[1,2]Department of Electrical and Computer Engineering, Isfahan University of Technology, Iran
[3]Faculty of Computer and Information Technology Engineering, Qazvin Branch, Islamic Azad University, Iran

Article Info

Article history:

Received Feb 9, 2020
Revised Apr 9, 2020
Accepted Apr 25, 2019

Keywords:

Best-worst multi-criteria
Multi-criteria optimization
Performance evaluation
SDN controller
Software-defined network

ABSTRACT

Controllers are the key component of software-defined network (SDN) architecture. Given the diversity of open SDN controllers, the following question arises for the network administrators: How can we choose the appropriate SDN controller? Different characteristics of the controllers have greatly increased the complexity of the right decision. Multi-criteria decision-making methods (MCDMs) isa family of robust mathematical tools to address complex problems regarding multiple objectives. In this paper, we study the most important features of SDN controllers. To this end, we compare the well-known SDN controllers including NOX, POX, Beacon, Floodlight, Ryu, ODL, and ONOS. Leveraging a novel MCDM technique called the best-worst multi-criteria (BWM), we find the most appropriate SDN controller. We solve an optimization problem and evaluate its performance in terms of significant criteria such as throughput and latency. Initial evaluation revealed that the ONOS and ODL have the highest throughput, while the lowest throughputs belong to the NOX, POX, and Ryu. However, final evaluation concerning all criteria confirmed the robustness of the ONOS and the ODL compared to other controllers.

Corresponding Author:

Mohammad Hossein Rezvani,
Faculty of Computer and Information Technology Engineering,
Qazvin Branch, Islamic Azad University,
Room 303, Computer Engineering Department, Islamic Azad University, Nokhbegan Blvd. Qazvin, Iran.
Email: rezvani@qiau.ac.ir

1. INTRODUCTION

Traditionally, network protocols are run inside the network devices. This architecture suffers from several problems, such as manageability, flexibility, and extensibility. To tackle the problems, the software-defined networking (SDN) concept has been proposed. In the SDN, the control plane of network devices is separated from the forwarding plane. The control plan is implemented centrally in a controller. The data plane performs packet forwarding, based on rules installed on the SDN controller. The SDN controller knows the global topology and information of the network. Therefore, it provides flexible and efficient management. The controller is programmable through applications communicating with it via the northbound application programming interfaces (APIs) [1-3].

The controller is the most significant part of the SDN architecture. Currently, there are a variety of open-source SDN controllers available for the community. The first SDN controller was NOX [4] which initially developed by Nicira [5]. Later, the NOX became the basic platform for many subsequent SDN controllers such as POX [6], ONIX [7], and Beacon [8]. Also, Floodlight [9], Ryu [10], OpenDayLight (ODL) [11], and ONOS [12] are the next generations of SDN controllers with more advanced features.

Due to the lack of standardization for SDN Networks [2], several platforms have been introduced by both the academic and industry sectors. These platforms have different features such as northbound and southbound APIs, single or hierarchical SDN controller model [13, 14], and supporting legacy network devices [15]. Given the inherent diversity of SDN controllers, the following question arises for the network administrators: How to choose the right SDN controller platform? In this paper, we suggest a new approach to answer this question.

Due to diverse and different features, the selection of an SDN controller is a multi-objective problem. The well-known multi-criteria decision-making method (MCDM) is a robust mathematical tool to address complex decision problems involving multiple objectives [16]. In many works of literature related to computer networks such as [14, 15], various MCDM methods have been introduced. The most popular MCDM methods are multi-attribute utility theory (MAUT) [17], analytic hierarchy process (AHP) [18], analytic network process (ANP) [19], VIKOR [20], and Technique for the order of prioritization by similarity to ideal solution (TOPSIS) [21, 22].

Recently, Rezaei et al. [23, 24] proposed a novel MCDM technique called the (BWM) decision-making method. Compared to previous MCDM algorithms, this technique remarkably reduces the number of required comparisons. In this paper, we use the BWM algorithm to select the best SDN controller subject to network constraints. We run the so-called method over most well-known SDN controllers, namely the NOX, the POX, the Beacon, the Floodlight, the Ryu, the ODL, and the ONOS. The major contributions of our research are as follows:

- We present a comprehensive study of both quantitative and qualitative features of the SDN controllers.
- We give a comprehensive comparison of the most well-known controllers (NOX, POX, Beacon, Floodlight, Ryu, ODL, ONOS) considering both quantitative and qualitative features.
- We analyze the best and worst features and then rank other criteria to select the best controller.
- We use the BWM which is a novel MCDM technique to select the best SDN controller.
- The proposed optimization approach leads to choosing the best SDN controller concerning user requirements.

To the best of our knowledge, no research has so far deployed multi-objective optimization based on the BWM. Our proposed MCDM-based framework succeeds to find the best selection of SDN controllers considering both quantitative and qualitative criteria together. The organization of this paper is as follows: Section 2 describes the fundamentals and concepts of the problem and significant studies related to multi-objective SDN controller selection; section 3 presents the proposed scheme consisting of the proposed decision-making method in detail; in section 4, we present the results to rank the candidate SDN controllers; Finally, section 5 concludes the study and sketches future research directions.

2. BACKGROUND AND LITERATURE REVIEW

Simply speaking, SDN is a paradigm shift for traffic management by providing programmable control. Figure 1 shows the SDN architecture. There are three different layers:

- Application layer: This layer provides service abstractions. It can be used by applications and services.
- Control layer: This layer consists of a logically centralized controller which is the key component of the SDN architecture. The controller takes requests from the application layer and manages the forwarding planes via southbound API standard protocols such as the OpenFlow, and the OpFlex [25].
- Infrastructure layer: This layer consists of simple switches. When a switch receives a new flow, it sends a request to the controller. Then, the controller uses its global knowledge to find a path for the flow and sets the required rules down to the switches.

The performance evaluation of a typical SDN controller includes scalability, reliability, efficiency, and robustness [26]. Our literature survey shows that almost all researchers use the CBench tool [27] to evaluate the performance of different SDN controllers. The CBench calculates the latency, the throughput, and the threading of the SDN controller. Zhu et al. [28] evaluated the performance of the NOX, the Beacon, and the Maestro [29] comparatively. In this paper, a version of NOX, namely NOX-MT, is developed to gain better performance. Tootoonchian et al. [30] performed the same study, but this time for the Floodlight, the Beacon, the NOX-MT, and the Maestro. In this study, the controllers are evaluated concerning the proposed architecture. Their results show that the performance of almost all SDN controllers varies based on their features such as switch partitioning, packet batching, and multi-threading.

Shalimov et al. [31], presented a new tool, named HCprobe. The HCprobe is capable to compare seven different SDN controllers, namely the NOX, POX, Floodlight, Beacon, Ryu, MUL [32], and Maestro. The authors concluded that the proposed SDN controllers have several security vulnerabilities. They also reported that the Beacon is the most effective controller based on test results. Suh et al. [33] introduced almost the same tool for performance analysis of SDN controllers with more systematic measurements. In [34], latency and throughput for ODLare calculated under different scenarios. Their results show that 1)

the throughput of ODL is notably affected by the cluster size and 2) the controller failover time is dependent on the number of switches and the failure detection threshold.

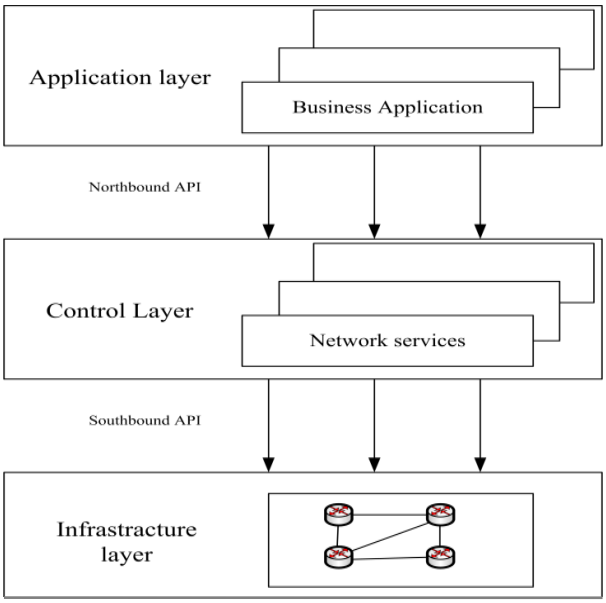


Figure 1. The SDN architecture

Salman et al. [35] analyzed the performance in terms of the latency and throughput metrics. They performed the analysis over the MUL, Beacon, Maestro, ONOS, Ryu, OpenDayLight, Floodlight, NOX, IRIS [36], MUL, Libfluid-Msg, and the POX under a varying number of switches and threads. Their results show that the MUL and the Libfluid-Msg have the best throughput performance, while the Maestro has the best latency. In [37], the authors did the same evaluation for Ryu, Floodlight, ONOS, and ODL. From the performance evaluation, ONOS showed the best throughput, while in latency mode, Ryu was the best.

In [38], authors calculate the performance of the POX and the Floodlight by analyzing network throughput and the round-trip time (RTT) using Mininet [39]. They evaluated different topologies (single, linear, tree, and user-defined) to connect the switches. Their results illustrate that the Floodlight outperforms the POX, for both evaluations. Nguyen-Ngoc et al. [40] evaluated the performance of the ONOS controller by developing several modules for the OFCProbe [41] which is itself a benchmark tool. In this paper, four parameters are evaluated including network topology discovery time, topology change detection time, reactive path provisioning time, and asynchronous message processing time.

Khondoker et al. [42] provided a set of comparison attributes such as virtualization, open-source, age and interfaces. The comparison was done using the analytic hierarchy process (AHP) which is an MCDM-based method. According to their analysis, the Ryu controller was chosen as the most suitable one. However, the assumed scale is subjective and changing the scale might lead to a different conclusion.

None of the above-mentioned researches provide a comprehensive optimization method in such a way that could result in finding the best SDN controller based on multiple candidate criteria. As mentioned earlier, in this article we are going to solve a multi-objective optimization problem. Some authors of this article have valuable experiences in solving this kind of problem with metaheuristic methods. Interested readers can refer to [43-45] for deeper study.

3. PROPOSED SCHEME

Given the diversity of SDN controllers, the following question arises for the network administrators: How can we choose a proper SDN controller? In this section, we are going to present a comprehensive and systematic method using advanced tools for decision making to answer this question. Advanced tools are used because choosing a proper SDN controller among several choices with many different characteristics can increase the complexity of the right decision. On the other hand, a large variety of different problems emerging in computer networks engineering projects can sufficiently be tackled using the MCDM methods and its related techniques to make the right decision among multiple options. As said before, there have been introduced many MCDM techniques. Without paying attention to their differences, all of them firstly requires

the definition of options and criteria, and then demand a measure (e.g. weights) for assessing the relative significance of the criteria [16].

In this paper, we use the best-worst multi-criteria decision-making method (BWM), one of the latest members of MCDM family methods. The BWM was proposed by Rezaei [24] in 2015. It can calculate the weight of each criterion concerning different criteria based on pair wise comparisons. BWM has less complexity compared to other methods. Figure 2 illustrates the procedure diagram of the proposed scheme to select the best SDN controller among multiple controllers based on the preference of the best criterion over all other criteria. In the following subsections, we are going to propose each step of this diagram in detail.

3.1. Identification of options

In this section, seven most well-known SDN controllers have been compared to each other. Below, we give the list of these controllers, their origin and characteristics:

- NOX: It was the first SDN controller developed by Nicira to be run on Linux distributions. It supports the OpenFlow protocol and is written in C language [4].
- POX: It is the next generation of the NOX controller. It is a python-based controller for fast development. Writing software for the POX controller is fast and very easy which makes it suitable for research purposes, demonstrations or experimentation [6].
- Beacon: Beacon is a fast, cross-platform, modular and Java-based OpenFlow controller that supports both event-based and threaded operations [8].
- Floodlight: The Floodlight is an enterprise-class, Apache-licensed, Java-based open-source OpenFlow Controller. It is supported by the Big Switch Company [9].
- Ryu: The Ryu is a python-based and open-source SDN controller providing an easy environment for developers to create new network management and control applications. The Ryu supports fully OpenFlow versions (1.0 to 1.5) and Nicira Extensions. All of the Ryu's code is freely available under the Apache 2.0 license [10].
- OpenDayLight (ODL): The ODL is a modular SDN controller for customizing and automating networks of any size and scale. It was designed for a commercial purpose that addresses a variety of use-cases in existing network environments. One of the main key features of the ODL is supporting clustering and distributed data store architecture [11].
- ONOS: A key feature of the ONOS is that it can run as a distributed SDN controller; allowing it to use the CPU and memory resources of multiple servers. Multiple ONOS controllers can be linked to form an ONOS cluster with each controller manages a set of switches. In addition, it provides fault tolerance when the server fails [12].

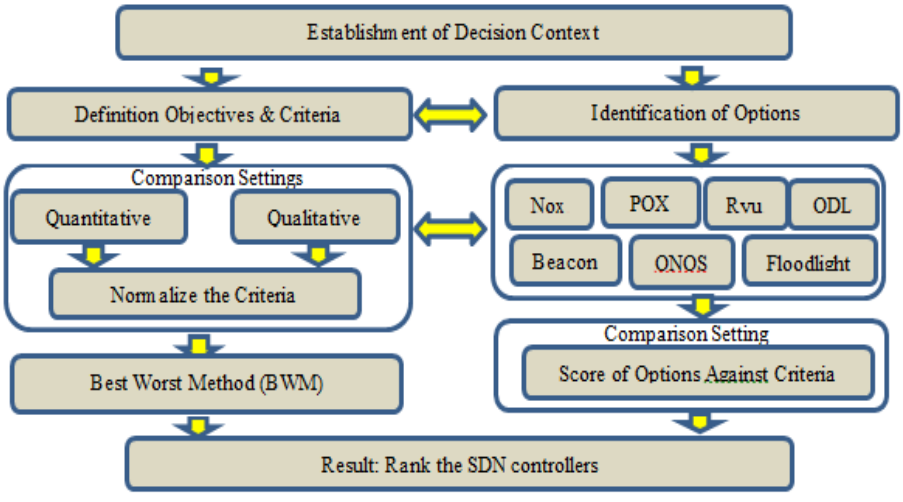


Figure 2. The procedure diagram of the proposed scheme

3.2. Definition objectives and criteria

SDN controllers have different and almost diverse features. We divide the features into two categories, including the "quantitative" and the "qualitative" ones. The qualitative features are:

- APIs: As shown in Figure 1, the SDN has three layers: application, control, and infrastructure. It can be seen that the control layer manages the forwarding devices with southbound APIs such as the OpenFlow protocol [46], and the OpFlex [25]. The controller is also programmable with northbound APIs such as REST [47].
- The OpenFlow Version: we consider the OpenFlow as a separate criterion because OpenFlow is known as a key component of the SDN architecture and it has been updating regularly in recent years. While other controllers do not support all versions, the OpenFlow protocol has different versions.
- Documentation: Most of the controllers discussed in this paper are Open-source. An open-source program allows everyone to change it based on custom-made requirements. Unfortunately, the majority of the controllers suffer from rich documentation.
- Regular update: Unfortunately, some SDN controllers do not provide regular update, which causes some problems for the developers' community.
- Programming language: Controllers have been written using different programming languages, such as C, C++, Java, and Python. We use the programming language to write the applications in the controllers. Simplicity, having a rich library for networking, and user-friendliness are the most significant features for a programming language.
- Legacy Network: Generally, it is difficult to completely replace all legacy network devices with SDN-enabled ones. We expect that several legacy network devices can be used in migration toward the SDN [48]. Thus, translating the SDN controller policy for legacy network devices, and vice-versa, is a subtle task [49]. The SDN controller can provide this mechanism by using a legacy network management protocol such as the NETCONF [50].
- Distributed Model: There are two general models for the controller, including single and distributed (logically centralized) [13]. With a single SDN controller, all switches connect to the same controller. This model suffers from two major drawbacks: the scalability and the single point of failure [51]. The distributed model is known as a good solution in which the load of the network can be shared among multiple controllers. Figure 3 shows an example of a logically centralized controller. In this figure, the controllers share their local views to form a global network view. In Figure 3, the controllers need an interface to share their information. These interfaces are called east-west-bound APIs.
- Modularity: The modularity maintains the integration and functionality of applications. High modularity allows the SDN controller to perform tasks faster in a distributed manner.
- GUI: TheGUI is a form of user interface that allows users to interact with network devices through graphical icons and visual indicators. It is an important option for network administrators to monitor the network.

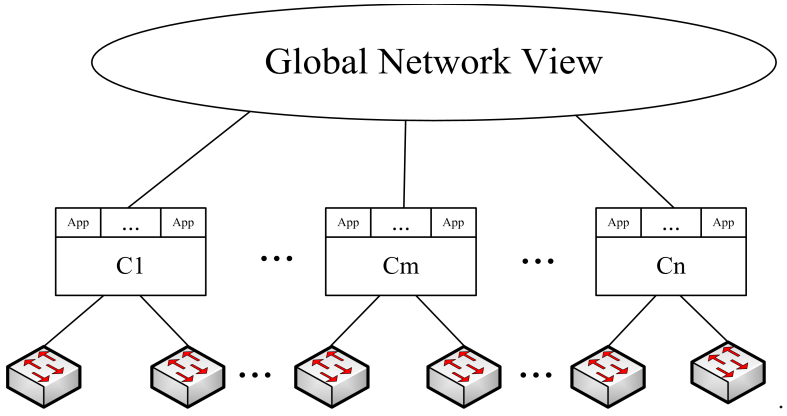


Figure 3. An example of the logical centralized controller

Table 1 presents a feature-based comparison among SDN controllers based on the above-mentioned qualitative features. Now that the qualitative features of SDN controllers were discussed, we proceed to explain the quantitative features. The quantitative features of SDN controllers are as follows:

- Throughput and latency: Controller bench marker (Cbench) is a good testing tool for OpenFlow controllers by generating packet-in events for new flows. The Cbench emulates a group of switches, which are connecting to a controller. It also sends packet-in messages and waits for flow-mods to get pushed down [27]. The Cbench is run in two modes: *throughput* and *latency*. In the throughput mode, it computes the maximum number of packets that could be handled by the controller. In the latency mode, it sends a packet-in message and waits for flow-mods to get pushed down. In this manner, the Cbench computes the total execution time.

We run our test on a system with core i7 CPU, 3.2GHz×8 and 12GB RAM. Also, all experiments were repeated 10 times for each size of the network. Table 2 shows the parameters of the Cbench command which we have used. The Cbench command is:./cbench-c localhost -p 6633 -m 10000 -l 10 -s 16 -M 1000 -t.

Table 1. A feature-based comparison among SDN controllers

	NOX	POX	Beacon	Ryu	Floodlight	ODL	ONOS
First Release	2008	2011	2010	2012	2012	2013	2014
License	GPL 3.0	Apache 2.0	GPL 2.0	Apache 2.0	Apache 2.0	EPL 1.0	Apache 2.0
Multi-Thread	No	No	Yes	Yes	Yes	Yes	Yes
Platform	Linux	Linux MAC windows	Linux MAC windows	Linux MAC windows	Linux MAC windows	Linux MAC windows	Linux MAC windows
APIs	OVSDB OFREST	OVSDBOF REST	OVSDB OF REST	OVSDB, OF, REST NETCONFO FCONFIG	OVSDBOF REST	OVSDB,OFREST,YANG NETCONFPCEP, BGPSNMP	OVSDB, OF REST, NETCONF
OF Version	1	1	1	1.0 to 1.5	1.0 to 1.3	1.0 to 1.3	1.0 to 1.3
Documentation	poor	poor	poor	Medium	Very Good	Very good	Medium
Language	C++	python	Java	Python	Java	Java	Java
Legacy	No	No	No	No	No	Yes	Yes
Network Category	Single	Single	Single	Single	Single	Distributed	Distributed
GUI	Poor	Poor	Poor	Medium	Medium	Very Good	Very Good
Regular Updating	Poor	Poor	Poor	Medium	Medium	Very Good	Very Good
Modularity	Poor	Poor	Medium	Medium	Medium	Very Good	Very Good

Table 2.parameters of the Cbench command

-c	controller hostname or IP address
-p	controller port for switches to connect to
-l	internal iterations per test
-m	duration of an internal iteration (in msec)
-M	number of MACs/hosts to emulate per switch
-S	number of switches to emulate
-t	throughput (vs. latency mode)

Figure 4 shows the throughput of the Cbench program for different scales of the network. As stated before, the throughput is defined as the number of responses per second from the controller. As is evident in the figure, the ONOS has the highest throughput, while the lowest throughputs belong to the NOX, POX, and Ryu. Similarly, Figure 5 shows the latency of different controllers. As is evident in the figure, the NOX, POX, and RYU have lower latency compared to other controllers.

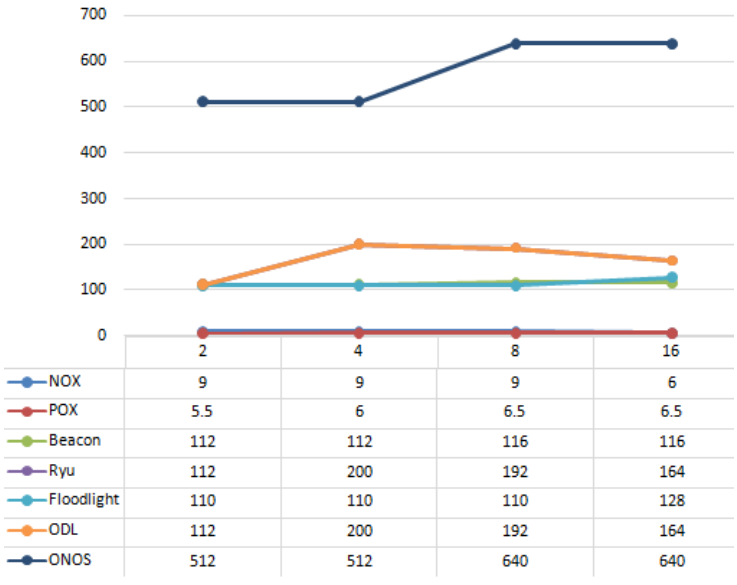
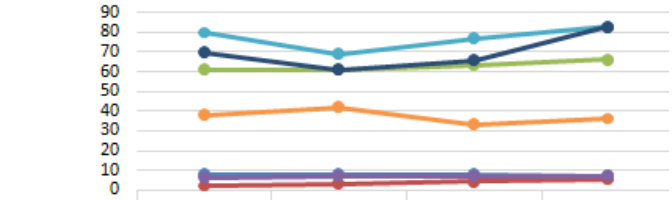


Figure 4.Thethroughput of the Cbench program for different scales of the network (thousand responses per second)



	2	4	8	16
NOX	8	8	8	7
POX	2	3	4	5
Beacon	61	61	63	66
Ryu	6	7	7	7
Floodlight	80	69	77	83
ODL	38	42	33	36
ONOS	70	61	66	83

Figure 5.The latency of different controllers (ms)

3.3. Comparison settings

In this section, first, we explain comparison settings and then we consider the above-mentioned features as the most important criteria. To compare SDN controllers, we assign a level in a range [1, 4] to each controller for each qualitative criterion as follow:

- Level 4: Very Good
- Level 3: Good
- Level 2: Medium
- Level 1: Poor

In the MCDM terminology, a matrix that shows the values of criteria is called the *"decision matrix"*. As is evident in the decision matrix of Figure 6, we augment both quantitative and qualitative criteria altogether. The level of each controller for each criterion is obtained from Figure 6.

	APIs	OF Version	Doc.	Language	Legacy Net	Category	GUI	Regular Update	Modularity	Throughput (8 switches)	Latency (8 switches)
Nox	1	1	1	2	1	1	1	1	1	9	8
POX	1	1	1	4	1	1	1	1	1	5.5	2
Beacon	1	1	1	3	1	1	1	1	1	112	61
Ryu	3	4	3	4	1	1	2	2	2	10	6
Floodlight	1	2	3	3	1	1	2	2	2	112	80
ODL	4	2	3	3	4	4	3	4	4	110	38
ONOS	3	2	3	3	4	4	3	4	4	512	70

Figure 6. A decision matrix containing the qualitative and the quantitative data

MCDM methods involve several steps among them normalization is an important one. Since the normalization concept is used to eliminate the units of each criterion, all the criteria are dimensionless [52]. To normalize the decision matrix, we should notice that there are two types of criteria in the decision matrix, including quantitative and qualitative. We use different methods for the normalization of these two types. For normalization of the qualitative criterion $x_{i,j}$ in the decision matrix we use the following relation:

$$p_{i,j} = \frac{x_{i,j}}{\sum_{k=1}^{m} x_{k,j}}, \qquad \forall j \in \{1,2,...,n\} \tag{1}$$

Since the magnitudes of changes in quantitative criteria (throughput and latency) are exponential, we use the logarithmic scale for normalization. To normalize the throughput criterion $x_{ij}\, x_{i,j}$ in the decision matrix we use the following relation:

$$p_{i,j} = \frac{\log_2 x_{i,j}}{\sum_{k=1}^{m} \log_2 x_{k,j}}, \qquad \forall j \in \{1,2,...,n\} \tag{2}$$

Similarly, to normalize the latency, we use

$$p_{i,j} = 1 - \frac{\log_2 x_{i,j}}{\sum_{k=1}^{m} \log_2 x_{k,j}}, \qquad \forall j \in \{1,2,...,n\} \tag{3}$$

In (3), note that because lower values of latency are more desirable we have used a negative sign. Finally, the normalized decision matrix of qualitative criteria is shown in Figure 7. The normalized decision matrix of throughput and latency are shown in Figures 8 and 9, respectively.

	APIs	OF Version	Doc.	Language	Legacy Net	Category	GUI	Regular Update	Modularity
Nox	0.07	0.77	0.067	0.09	0.77	0.77	0.77	0.067	0.067
POX	0.07	0.77	0.067	0.18	0.77	0.77	0.77	0.067	0.067
Beacon	0.07	0.77	0.067	0.136	0.77	0.77	0.77	0.067	0.067
Ryu	0.21	0.31	0.2	0.18	0.77	0.77	0.15	0.13	0.13
Floodlight	0.07	0.15	0.2	0.136	0.77	0.77	0.15	0.13	0.13
ODL	0.28	0.15	0.2	0.136	0.31	0.31	0.23	0.27	0.27
ONOS	0.21	0.15	0.2	0.136	0.31	0.31	0.23	0.27	0.27

Figure 7. Normalized decision matrix

	2	4	8	16
Nox	0.082	0.08	0.079	0.066
Pox	0.064	0.065	0.068	0.069
Beacon	0.177	0.173	0.172	0.175
Ryu	0.086	0.084	0.083	0.085
Floodlight	0.177	0.194	0.190	0.188
ODL	0.176	0.172	0.17	0.179

Figure 8. The normalized decision matrix for throughput criterion

	2	4	8	16
Nox	0.911	0.903	0.902	0.9
Pox	0.927	0.935	0.948	0.966
Beacon	0.81	0.808	0.807	0.803
Ryu	0.911	0.909	0.908	0.914
Floodlight	0.8	0.798	0.801	0.79
ODL	0.837	0.838	0.824	0.826

Figure 9. The normalized decision matrix for latency criterion

3.4. Best worst method (BWM)

A major problem to select the best controller is the different importance of each criterion. For example, for a controller, a regular update is much more important than its language. So we need to determine the importance weight of each criterion. In MCDM terminology, the importance of each criterion is called the *"weight of the criterion"*. There are different ways to determine the weight of each criterion such as entropy method and (BWM). In this paper we use the BWM method which includes five steps as following:

Step 1: Determine a set of decision criteria. In this step, a set of criteria $\{c_1, c_2, ..., c_n\}$ is considered. In our study, the criteria are APIs(c_1), OpenFlow Version(c_2), Documentation(c_3), Language(c_4), Legacy Net(c_5), Category(c_6), GUI(c_7), Regular Update(c_8), Modularity(c_9), Throughput(c_{10}), Latency(c_{11}).

Step 2: Determine the best and the worst criteria. The best criterion has the highest importance. In our study, we consider OpenFlow version, regular update, supporting distributed model as the best criteria, while language is the worst criterion. Note that the best and the worst criteria are chosen arbitrary and based on the network characteristics.

Step 3: Determine *Others-to-Worst* vector, like step 3 as follows:

$$\mathbf{A}_w = (a_{w1},\ a_{w2},...,a_{wn})^{\mathrm{T}} \tag{4}$$

where a_{wj} denotes the preference of the criterion c_j over the worst criterion c_w .In our study, the vector $\mathbf{A}_w$ is as follows:

$$\mathbf{A}_w = (7,7,2,1,6,7,3,8,5,4,4)^{\mathrm{T}} \tag{5}$$

Step 4: Determine the preference of the best criterion over all other criteria. The resulting *Best-to-Others* vector would be:

$$\mathbf{A}_B = (a_{B1},\ a_{B2},...,a_{Bn})^{\mathrm{T}} \tag{6}$$

where a_{Bj} denotes the preference of the best criterion c_B over criterion c_j . We should notice that for each pair a_{Bi} and a_{wi} we have $a_{Bi} = a_{wB} / a_{wi}$, which a_{wB} is the preference of *"the best over the worst"* criterion. The *Best-to-Others* vector in our study is considered as follows:

$$\mathbf{A}_B = (1.143,\ 1.143, 4, 8,\ 1.33,\ 1.143,\ 2.66,\ 1,\ 1.6,\ 2,\ 2)^{\mathrm{T}} \tag{7}$$

Step 5: Find the vector of optimal weights $(w_1^*,\ w_2^*,,...,w_n^*)$. To find the optimal weights of the criteria, the maximum absolute differences $|\ w_B / w_j - a_{Bj}\ |$ and $|\ w_j / w_w - a_{jw}\ |$, for all j , must be minimized. In other words, we should solve a min-max optimization problem as follows:

$$\min \max_j \left\{ \left| \frac{w_B}{w_j} - a_{Bj} \right|,\ \left| \frac{w_j}{w_w} - a_{jw} \right| \right\} \tag{8}$$

$$\text{s.t.} \sum_j w_j = 1, \quad w_j \geq 0 \ \text{ for all } \ j \tag{9}$$

The optimization problem in (8) and (9) is equivalent to the following problem:

$$\min\ \xi \tag{10}$$

$$\text{s.t.} \left| \frac{w_B}{w_j} - a_{Bj} \right| \leq \xi, \quad \text{for all } \ j \tag{11}$$

$$\left| \frac{w_j}{w_w} - a_{jw} \right| \leq \xi, \quad \text{for all } \ j \tag{12}$$

$$\sum_j w_j = 1, \quad w_j \geq 0 \ \text{ for all } \ j \tag{13}$$

For any value of ξ^* , multiplying the first set of the constraints in (12) by w_j and the second set of constraints by w_w , the solution space of (12) is an intersection of 4n-5 linear constraints. Therefore given values large enough for ξ, it can be concluded that the solution space is non-empty. By solving the optimization problem in (10)-(13), we obtain the optimal weights $(w_1^*,\ w_2^*,,...,w_n^*)$ and ξ^* .

Figure 10 shows the optimal weights of criteria for our study. As is evident in the figure, the best criterion (regular update) attains the most important weight compared to other criteria. Inversely, the worst criterion (language) attains the least importance weight.

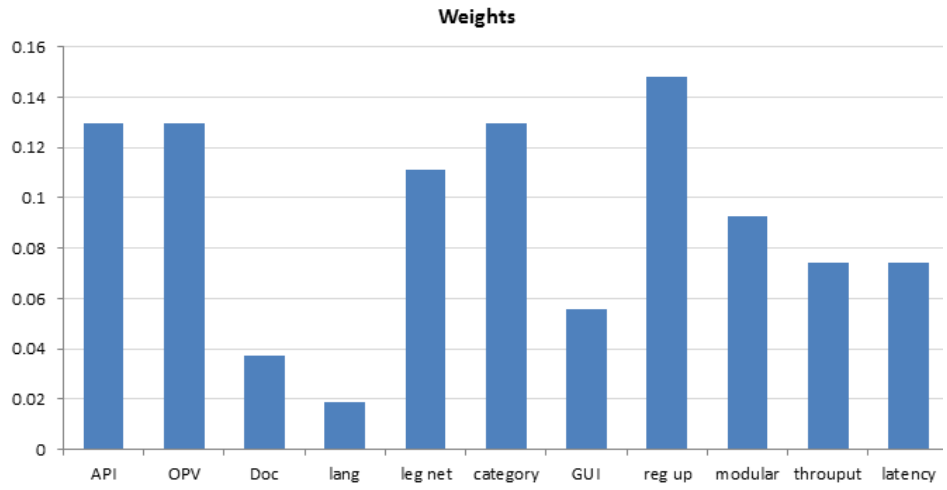


Figure 10. Optimal weights for our study

4. RESULT: RANK THE SDN CONTROLLERS

Now, we proceed to select the best controller based on the decision matrix and the weights of criteria calculated in (10)-(13). The weighted linear combination (WLC) aggregation method multiplies each raw of the decision matrix by the weights of criteria and then sums the results as follows:

$$(w_1,\ w_2,...,w_n)\begin{bmatrix} x_{1,1} & \dots & x_{1,n} \\ \dots & & \dots \\ x_{m,1} & \dots & x_{m,n} \end{bmatrix} \tag{14}$$

Finally, we rank the best SDN controller according to the results. Figure 11 shows this ranking. As is evident in the figure, the ODL and the ONOS are the best controllers based on our criteria, while the POX, the NOX, and the Beacon are less desirable. Another interesting point is that ranking does not change for different scales of the networks. It is worth mentioning that the selection of the criteria (the best and the worst criteria)is subjective in our study and changing these parameters may lead to a different conclusion. As a result, the parameters in the BWM should be considered carefully based on the specific characteristics of the network requirements.

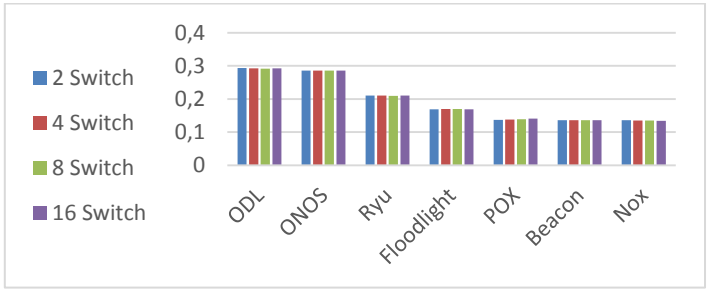


Figure 11. Feature-based comparison of the candidates SDN controllers

5. CONCLUSION

In this article, we targeted one of the most important challenges of SDN networks, namely controller selection. After reviewing previous studies along with their objectives and limitations, we divided the features into two categories, namely quantitative and qualitative. The quantitative features are throughput and latency that are calculated by the CBench tool. On the other hand, the most important qualitative features are APIs, Programming language supported by the controller, supporting legacy network, modularity, GUI, regular update, and documentation of the controller. We considered the quantitative and the qualitative features as the criteria to compare the most well-known SDN controllers. We modelled the SDN selection problem using a mathematical technique from the family of MCDM techniques, called the (BWM).It was observed that the proposed method could establish a better trade-off in terms of the above-mentioned criteria. The results

show that the ONOS and the ODL are the best controllers, while the POX, the NOX, and the Beacon are located at the bottom of the list. The optimization framework presented in this research could help the network administrators to find the most appropriate SDN controller based on the given network characteristics.

Future studies are suggested to exploit machine learning methods, especially reinforcement learning, to solve the problem. Addressing this estimation can undoubtedly lead to more accurate modelling. The machine learning algorithms are usually used when the answer to a problem (judgment label) with a dataset is known in advance. For this purpose, a classifier must be designed so that for the input features, it can determine the best class label (the type of SND controller).

REFERENCES

[1] D. Kreutz et al., "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14-76, 2014.
[2] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Communications Surveys and Tutorials*, vol. 17, no. 1. pp. 27-51, 2015.
[3] H. Farhady, H. Lee, and A. Nakao, "Software-defined networking: A survey," *Comp. Netw.*, vol. 81. pp. 79-95, 2015.
[4] "NOX controller," [Online], Available at: https://github.com/noxrepo/nox, [Accessed March 2019].
[5] "Nicira Company," [Online], Available at: https://www.vmware.com/company/acquisitions/nicira.html, [Accessed March 2019].
[6] "POX controller," [Online], Available at: https://github.com/noxrepo/pox, [Accessed March 2019].
[7] T. Koponen et al.,"Onix: A distributed control platform for large-scale production networks," *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation*,vol. 10, pp. 351-364, 2010.
[8] D. Erickson, "The Beacon OpenFlow controller,"*Proceedings of the 2013 ACM SIGCOMM Workshop on Hot Topics in Software Defined Networking*, pp. 13-18, 2013.
[9] "Floodlight controller," [Online], Available at: https://github.com/floodlight/floodlight, [Accessed March 2019].
[10] "Ryu controller," *Ryu SDN Framework Community*, 2017, [Online], Available at: https://osrg.github.io/ryu/, [Accessed March 2019].
[11] "OpenDayLight controller," *OpenDaylight Project a Series of LF Projects*, 2018, [Online], Available at: https://www.opendaylight.org/, [Accessed March 2019].
[12] "ONOS controller," *Open Networking Foundation*, 2020, [Online], Available at: https://opennetworking.org/onos/, [Accessed March 2019].
[13] Y. E. Oktian, S. G. Lee, H. J. Lee, and J. H. Lam "Distributed SDN controller system: A survey on design choice," *Comput. Networks*, vol. 121, pp. 100-111, 2017,
[14] E. Amiri, E. Alizadeh, and K. Raeisi, "An efficient hierarchical distributed SDN controller model," *2019 IEEE 5th Conference on Knowledge Based Engineering and Innovation*,pp. 553-557, 2019.
[15] E. Amiri, M. R. Hashemi, and K. R. Lejjy, "Policy-based routing in RIP-hybrid network with SDN controller," *4th National Conference on Applied Research in Electrical, Mechanical, Computer and IT Engineering*, pp. 1-8, 2018.
[16] E. K. Zavadskas, Z. Turskis, and S. Kildiene, "State of art surveys of overviews on MCDM/MADM methods," *Technological and Economic Development of Economy*, vol. 20, no. 1. pp. 165-179, 2014.
[17] H. Min, "International supplier selection: A multi-attribute utility approach," *Int. J. Phys. Distrib. Logist. Manag.*, vol. 24, no. 5, pp. 24-33, 1994.
[18] T. L. Saaty, "What is the analytic hierarchy process?," *Mathematical Models for Decision Support*, pp. 109-121, 1988.
[19] T. L. Saaty, "Fundamentals of the analytic network process-dependence and feedback in decision-making with a single network," *J. Syst. Sci. Syst. Eng.*, vol. 13, no. 2, pp. 129-157, 2004.
[20] E. Alizadeh, K. R. Lejjy, and E. Amiri, "Improving routing in vehicular Ad-hoc network with VIKOR algorithm," *9th Int. Symp. on Telecom : With Emphasis on Information and Communication Technology*, pp. 337-341, 2019.
[21] E. Amiri and R. Hooshmand, "Improved AODV based on TOPSIS and fuzzy algorithms in vehicular Ad-hoc networks," *Wirel. Pers. Commun.*, 2019.
[22] E. Amiri and R. Hooshmand, "Improving AODV with TOPSIS algorithm and fuzzy logic in VANETs," *27th Iranian Conference on Electrical Engineering*, 2019.
[23] J. Rezaei, "Best-worst multi-criteria decision-making method: Some properties and a linear model," *Omega (United Kingdom)*, vol. 64, pp. 126-130, 2016.
[24] J. Rezaei, "Best-worst multi-criteria decision-making method," *Omega (United Kingdom)*, vol. 53, pp. 49-57, 2015.
[25] M. Smith, et al., "OpFlex Control Protocol," *Internet Engineering Task Force*, pp. 1-21, 2014. .
[26] A. Dixit et al., "Towards an elastic distributed SDN controller,"*Comp. Comm. Review*,vol. 43, no. 4, pp. 7-12, 2013.
[27] R. Sherwood and K. Yap, "Cbench: n openflow controller benchmarker," 2010, [Online], Available at: https://github.com/mininet/oflops/tree/master/cbench
[28] L. Zhu et al., "SDN controllers: Benchmarking & performance evaluation," *A Version is under Review at IEEE JSAC*, pp. 1-14, 2019.
[29] Z. Cai, "Maestro: Achieving scalability and coordination in centralized network control plane," *Rice Univ. Degree Dr. Philos.*, pp. 102-105, 2011.
[30] A. Tootoonchian et al., "On controller performance in software-defined networks," *2nd {USENIX} Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services*, pp. 1-10, 2012.
[31] A. Shalimov et al., "Advanced study of SDN/OpenFlow controllers," *ACM Int. Conf. Proc. Series*, pp. 1-6, 2013.
[32] "Open Mul," Sourceforge, 2020, [Online], Available at: http://sourceforge.net/p/mul/wiki/Home/, [Accessed March 2019].
[33] D. Suh et al., "On performance of OpenDaylight clustering," *2016 IEEE NetSoft Conf. and Workshops: Software-Defined Infrastructure for Networks, Clouds, IoT and Services*, pp. 407-410, 2016.

[34] Z. K. Khattak, M. Awais, and A. Iqbal, "Performance evaluation of OpenDaylight SDN controller," *20th IEEE International Conference on Parallel and Distributed Systems*, pp. 671-676, 2014.
[35] O. Salman, I. H. Elhajj, A. Kayssi, and A. Chehab, "SDN controllers: A comparative study," *18th Mediterranean Electrotechnical Conference: Intelligent and Efficient Technologies and Services for the Citizen*, pp. 1-6, 2016.
[36] B. Lee, S. H. Park, J. Shin, and S. Yang, "IRIS: The Openflow-based recursive SDN controller," *International Conference on Advanced Communication Technology*, pp. 1227-1231, 2014.
[37] L. Mamushiane, A. Lysko, and S. Dlamini, "A comparative evaluation of the performance of popular SDN controllers," *IFIP Wireless Days*, vol. 2018, pp. 54-59, 2018.
[38] I. Z. Bholebawa and U. D. Dalal, "Performance analysis of SDN/openflow controllers: POX versus floodlight," *Wirel. Pers. Commun.*, vol. 98, no. 2, pp. 1679-1699, 2018.
[39] "MiniNet," 2018, [Online], Available at: http://mininet.org/, [Accessed March 2019].
[40] A. Nguyen-Ngoc et al, "Benchmarking the ONOS controller with OFCProbe," *IEEE 7th Int. Conference on Communications and Electronics*, pp. 367-372, 2018.
[41] M. Jarschel, C. Metter, T. Zinner, S. Gebert, and P. Tran-Gia, "OFCProbe: A platform-independent tool for OpenFlow controller analysis," *2014 IEEE 5th Int. Conf. on Communications and Electronics*, pp. 182-187, 2014.
[42] R. Khondoker, A. Zaalouk, R. Marx, and K. Bayarou, "Feature-based comparison and selection of software defined networking (SDN) controllers," *2014 World Congress on Comp. Appl. and Infor. Systems*, pp. 1-7, 2014.
[43] E. Parvizi and M. H. Rezvani, "Utilization-aware energy-efficient virtual machine placement in cloud networks using NSGA-III Meta-heuristic Approach," *Cluster Computing*, pp. 1-23, 2020.
[44] A. Mohammadi and M. H. Rezvani, "A novel optimized approach for resource reservation in cloud computing using producer–consumer theory of microeconomics," *J. Supercomput.*, vol. 75, no. 11, pp. 7391-7425, 2019.
[45] S. Tavakoli-Someh and M. H. Rezvani, "Multi-objective virtual network function placement using NSGA-II meta-heuristic approach," *J. Supercomput.*, vol. 75, no. 10, pp. 6451-6487, 2019.
[46] N. McKeown et al., "OpenFlow," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69-74, 2008.
[47] W. Zhou, L. Li, M. Luo, and W. Chou, "REST API design patterns for SDN northbound API," *2014 IEEE 28th International Conference on Advanced Information Networking and Applications Workshops*, pp. 358-365, 2014.
[48] E. Amiri and R. Javidan, "A new method for layer 2 loop prevention in software defined networks," *Telecommun. Syst.*, vol. 73, no. 1, pp. 47-57, 2020.
[49] M. Tanha, D. Sajjadi, R. Ruby, and J. Pan, "Traffic engineering enhancement by progressive migration to SDN," *IEEE Commun. Lett.*, vol. 22, no. 3, pp. 438-441, 2018.
[50] R. Enns, "NETCONF configuration protocol," *Network*, vol. RFC 4741, pp. 1-95, 2011.
[51] F. Bannour, S. Souihi, and A. Mellouk, "Distributed SDN control: Survey, taxonomy, and challenges," *IEEE Commun. Surv. Tutorials*, vol. 20, no. 1, pp. 333-354, 2018.
[52] T. M. Lakshmi and V.P. Venkatesan., "A Comparison of various normalization in techniques for order performance by similarity to ideal solution (TOPSIS)," *Int. J. Comput.Algorithm*, vol. 3, no. 3, pp. 255-259, 2014.

BIOGRAPHIES OF AUTHORS

Esmaeil Amiriwas born in 1992 in Malayer, Iran. He received a B.Sc. in electrical engineering from Razi University, Kermanshah, in 2014. He also received his M.Sc. degree in telecommunication networks engineering from Isfahan University of Technology (IUT), Isfahan, in 2017. His major research interest is high-performance computing (HPC) with topics such as cloud/fog computing, software-defined networks (SDNs), Network Function Virtualization (NFV), and internet of things (IoT). He also is interested in data center performance measurement and applications of AI Algorithms in networked systems.

Emad Alizadeh was born in 1993 in Gachsaran, Iran. He received a B.Sc. in electrical engineering (communications) from Yazd University, in 2014. He also received his M.Sc. degree in electrical engineering (telecommunication networks) from Isfahan University of Technology (IUT), Isfahan, in 2018. Since 2018, he has been pursuing his Ph.D. program in telecommunication networks at IUT. His major research interest is applied mathematics, social networks, complex networks, software-defined networks (SDNs), Network Function Virtualization (NFV), and internet of things (IoT). He also is experienced in Applied mathematics, the Internet of things, Complex and social networks.

Mohammad Hossein Rezvani is currently a faculty member of Computer and Information Technology Engineering at Qazvin Branch of Islamic Azad University (IAU), Qazvin, Iran. He received his Ph.D. degree in computer engineering from Iran University of Science and Technology (IUST)in 2011. In February 2012, Dr. Rezvani joined Iran Computer and Video Games Foundation (ICVGF) as an R&D consultant. He cooperated with an academic work-group in ICVGF to develop a new academic major, named "Computer Game Design". Dr. Rezvani was the dean of the department of *"Computer Game Development"* in ICVGF, a branch of the "University of Applied Science and Technology (UAST)" in Tehran, Iran, until 2016. His major research interest is the high-performance computing (HPC) with topics such as performance analysis, analytical modelling, optimization of computer networks, economic network modelling, convex optimization, and metaheuristic approaches. He also is interested in E-marketing behavioural measurement and assessment, and quantitative E-marketing researches.