The Journal of Supercomputing
https://doi.org/10.1007/s11227-021-03815-3

Multi-criteria decision-making for controller placement in software-defined wide-area networks

Ali Abdi Seyedkolaei[1] · Seyed Amin Hosseini Seno[1]

Accepted: 13 April 2021

Abstract

Software-defined networks have many benefits such as more control over the control plane and reduced operating costs through separating the control plane from the data plane in network equipment. One of the most critical problems in software-defined networks is a controller placement problem, which significantly influences its overall performance. The purpose of this problem is to determine the number of controllers required and how to assign switches. This paper attempts to solve this problem, aiming to reduce the network's operational cost and to improve their survivability and load balancing. Hence, we have tried to divide the network into several subdomains using segmentation. Then, we used multi-criteria decision-making methods to solve the controller placement in each subdomain. For this purpose, we considered criteria such as reliability rate, cost, delay, and processing capacity. To assign switches to controllers, we used the proposed mathematical model to minimize the objective function while observing the defined constraints. Furthermore, according to the proposed architecture, we used distance to the central controller, adjacent neighborhood, and controller memory to select the cluster head controller for each subdomain to communicate with the central controller. Finally, we performed experiments on topologies Uran, SwitchL3, and Sinet of the Internet Topology Zoo to evaluate the proposed method. In these experiments, we compared the results of the proposed method with our work-related methods, including Controller Placement Analytic Hierarchy Process (CPAHP) and Controller Placement Modified Density Peak (CPMDP), in terms of criteria such as cost, survivability, number of controllers, connection failure probability, average delay, and controller load-balancing rate. The results show that the proposed method outperforms CPMDP and CPAHP. Thus, the proposed method has a 24.97% and 19.76% improvement in reducing network implementation costs than CPMDP and CPAHP, respectively.

✉ Seyed Amin Hosseini Seno
hosseini@um.ac.ir

Extended author information available on the last page of the article

Published online: 26 April 2021

Keywords Software-defined network · Wide-area network · Controller placement · Cost · Multi-criteria decision-making · Reliability

1 Introduction

Traditional networks are not cost-effective due to their lack of flexibility and are not suitable for meeting the needs of the current Internet [1]. Therefore, the emergence of a new generation of proposed networks meets the needs and has high flexibility. Software-defined Network (SDN) allows you to access a programmable network by separating the control plane from the data plane to improve network performance [2]. This separation provides benefits such as simple network management, improved network performance, and network innovation. The control plane provides the information needed to route the network. The data plane is also responsible for transferring packets from the input port to the output port based on the information listed in its routing table. In the SDN network, the control plane is placed on a server or program called the controller. The data plane also remains in the switch or router as the forwarding. The control plane is responsible for data plane management. The emergence of SDN has attracted the attention of a large number of universities and industries to implement it in its communication infrastructure [3]. In these networks, configuration methods are often more straightforward, and more accurate, allowing for higher utilization of physical infrastructure [4].

In SDN networks, the controller is often responsible for disseminating any flow in the network, which is done by allocating input flow to switches [5, 6]. This has given the controller a pivotal role, as it can be used to provide complete knowledge of the network to flow management optimization and support of user requirements [7].

Using a controller in the SDN network has its disadvantage. In particular, the network traffic is overstated in most known areas, and the controller may overflow. Therefore, the use of multiple controllers defines a problem called the controller placement, in which the number and location of controllers, as well as how switches connected to controllers, are essential. In terms of computational complexity, this is part of the class of NP-hard problems [8, 9]. The controller placement has a significant impact on network reliability, cost, and latency [6]. The failure of the controllers affects the connections and even causes some instructions in the controller not to be executed. Therefore, the network reliability must be considered for network stability. When a load greater than the capacity of the controller is imposed on it, it causes failure because the controller does not have enough resources to handle the requests received from the switches. Sometimes, the controller failure can be in the form of a cascade, in which too much load is applied to other controllers. Therefore, a load balancing between controllers is very important. Also, a link failure may cause some switches to disconnect from the controller.

While most research on SDN applications has been conducted in data centers, there has recently been a growing focus on using SDN in wide-area networks.

Software-defined wide-area networks (SD-WAN) are expensive and difficult to manage. SD-WAN networks mainly face challenges such as link failure and disrupted connectivity between the control and data planes. In contrast, data centers are located in a secure and controlled environment due to the high degree of parallel connections. Therefore, failure resiliency, cost, latency, and scalability are the challenges to be considered when designing SD-WAN networks. Therefore, these challenges can be addressed in a proper design of the control plane architecture, proper location of the controllers, and optimal allocation of switches [10].

Therefore, in this paper, we have attempted to divide the network into several subdomains using the segmentation. Then, we used multi-criteria decision-making methods to solve the controller placement in each subdomain. For this purpose, we considered criteria such as reliability rate, cost, delay, and processing capacity. To assign switches to controllers, we used the proposed mathematical model to minimize the objective function while observing the defined constraints. Furthermore, according to the proposed architecture in Fig. 1, we used criteria such as the distance to the central controller, the proximity of the neighborhood, and the amount of memory to select the cluster head controller in each subdomain to communicate with the central controller. The advantage of the proposed algorithm is that indicators and criteria can simultaneously be used, and its outcome determines the priorities of the options, which are explained quantitatively. The run-time of the proposed algorithm is proper, and its solutions are comparable to the experimental methods. Besides, qualitative criteria can be easily quantified, and decisions can be made despite qualitative and quantitative criteria. Also, it is possible to see the effect of the criteria coefficient on the ranking of options numerically. Another advantage is that a considerable number of criteria can be considered to determine the best option.

Finally, we performed experiments on topologies Uran, SwitchL3, and Sinet of the Internet Topology Zoo to evaluate the proposed method. In these experiments, we compared the results of the proposed method with our work-related methods, including CPAHP [11] and CPMDP [21], in terms of criteria such as cost,

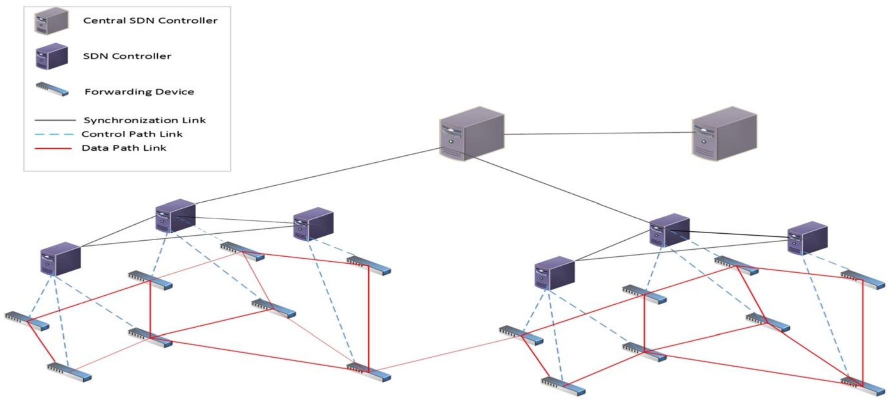


Fig. 1 Proposed architecture

survivability, number of controllers, connection failure probability, average delay, and controller load-balancing rate. The results show that the proposed method outperforms CPMDP and CPAHP. The main contributions of this paper are as follows:

- We introduce a mixed-integer nonlinear programming formulation of the survivable controller placement problem to impose a general connected topology among controllers. Then, we reduce the formulation to a mixed-integer linear program to be solved more efficiently.
- We show how to incorporate user-defined survivability requirements into our mixed-integer programming formulation.
- We demonstrate that our formulation can design networks of much less installation cost through careful computational studies while accepting a general connected topology among controllers and user-defined survivability parameters.
- Using the multi-criteria decision-making method to solve the controller placement problem and considering network dynamics and different network failure states.
- Load balancing and reducing average delay with optimal allocation of switches and solving the controller placement problem by considering heterogeneous controllers.

In the following, the paper is organized as follows: Sect. 2 describes the work that was done in the past on the controller placement problem. Section 3 describes the proposed method. Section 4 comparatively analyzes the proposed method and examines its performance in comparison with CPMDP and CPAHP methods.

2 Related works

Sminesh et al. [11] used the modified density peak (Modified-DP) clustering algorithms to the controller placement in SD-WAN. They used the inter-criteria correlation to determine the weights of each metric. Muller et al. [12] used a strategy called a survivor. In this strategy, criteria such as path diversity, the controller's capacity, and network design failure mechanisms are considered. This strategy significantly reduces network connectivity by examining various paths and preventing capacitors from increasing their load by determining the controllers' capacity. However, using this strategy for environments with multiple controllers is considered as its advantages.

Hu et al. [13, 14] used the concept of a control path failure, and the purpose of this study is to minimize the percentage of failure of the control path. In other words, how much the data sent between the controller and the switches is less corrupted, and the current location where the controller is located is the best for the controller. Although the proposed solution increases network reliability, the proposed solution is not appropriate for network scalability when the network size grows. Moazzeni et al. [15] used a distributed controller architecture to improve fault tolerance. They, in their proposed solution, with the help of a network segmentation strategy, for each subnet, the main controller and a slave controller were considered for the time of failure of the main controller. The reliability rate for selecting the coordinator controller in each subnet is obtained to determine which of the controllers is selected as an alternative controller in case of failure of the main controller.

Killi et al. [16] used an optimization model to deploy multiple controllers and assign switches to achieve maximum resilience. Also, they provided models to reduce latency. Lin et al. [17] focused on the controller traffic balancing to find the controller's best location. In this study, in addition to solving the traffic-balancing problem, the delayed time is considered. However, the problem of using only one controller has been overlooked. Researchers in [18] use the Garter Snake optimization method, a meta-heuristic algorithm that solves new iterations and temperate mating conditions. The algorithm calculates the minimum delays at the appropriate time.

Li et al. [19] used a method of dynamic deployment of controllers. In their proposed method, the flow is transferred to a queue, and then, based on two factors, the delay, and the controller capacity, several controllers are deployed. Jalili et al. [20] used the modified Non-Dominated Sorting Genetic-II algorithm (NSGA-II) on large-scale networks. The proposed algorithm has a faster computation time to solve the problem. Also, the amount of memory consumed has improved compared to the NSGA-II algorithm. In other research, Jalili et al. [21] used to delay, hop count, and link usage as assignment switches criteria. In their work, they analyzed the impact of these criteria on the quality of service. Tanha et al. [22] proposed an algorithm in which the latency and the capacity to determine the controllers' location are considered.

In most of the research conducted, the criterion of reducing the cost of the network is not considered, so Sallahi et al. [23] to reduce the cost. They argued that they should meet certain constraints. Their proposed method considers various constraints to minimize network costs. Sallahi et al., in other research in [24], developed the proposed method in [23] for the case where the network topology would change. This research's main objective is to minimize the cost of redesigning the topology to ensure that the solution found is appropriate for the changed topology. In implementing the proposed method, limitations such as controller capacity, latency, traffic, and other requirements for achieving a feasible solution have been considered. However, the proposed method used in [23] and [24] is only for small networks.

Khorramizadeh et al. [25] modeled the placement problem as a location-allocation model and expressed the proposed solution in two phases. In the first phase, they focused on determining the number of controllers needed while reducing costs. In the second phase, the location-allocation problem balances the controller load and reduces the delay between the controllers. Research in [26] introduces a parameter optimization algorithm and model, which solves the controller placement problem with the help of optimized parameters. The researchers use heuristic algorithms, including bat optimization algorithm, firefly, Verna-based optimization algorithm, and particle swarm optimization algorithm. Ali et al. [27] ranked the SDN controllers based on their supporting characteristics using the network analysis process. The highly rated controllers form a hierarchical cluster. The researchers considered network's cost and survivability to solve the problem through an iterated local search algorithm because the network is dynamic [28]. Besides, network failure events were taken into account.

According to previous studies focused on insufficient attention to several criteria simultaneously in solving the controller placement problem, we were encouraged to use metrics such as reliability, cost, delay, processing capacity, distance to the central controller, neighborhood proximity, and amount of memory in two phases in our research, simultaneously. Also, to closely resemble our work with the real network environment, we consider network elements to be heterogeneous. Table 1 summarizes the existing research on controller placement in SDN networks regarding objective aspects of latency, scalability, reliability, cost, and dynamic network. The asterisk in Table 1 indicates the consideration of these objectives in each related work.

3 The proposed method

In this work, we have considered the SDN network as a graph consisting of the locations of the controllers and switches. The switches must be connected to the controllers, and the controller installation locations are selected as candidates. Graph nodes may form clusters so that there would be several controllers in each cluster, which is the cluster head managed by a central controller. The connection of switches and controllers is in-band or out-band. The controllers, switches, and links have limitations, including resource constraints, bandwidth constraints, and the amount of data sent, respectively. Each controller processes the load sent by the switches according to its resources. We use the backup controller when the central controller fails. Besides, for link failure, we use disjoint paths. The proposed method is as follows.

At first, we divided the network into several subdomains using the segmentation method, and then specified the number and location of controllers for each subdomain so that switches could be assigned to their controllers. Therefore, one or more clusters were created in each subdomain. In the following, we created a control plane topology by connecting the controllers in each subdomain. Also, in each subdomain, we selected the best controller among all the clusters as the cluster head controller to communicate with the central controller. Therefore, we used the Technique for Order of Preference by Similarity to Ideal Solution (TOPSIS) [29] method in two phases for each subdomain. In the first phase, to select the controller's installation location, for each controller, the criteria include reliability rate, cost, delay, and processing capacity considered. Then, we used the proposed mathematical model to assign the switches to the controllers. The reliability rate criterion indicates the probability of failure of the controller in the event of a failure, and the method of calculating it described in Sect. 3.3 below. The cost includes the cost of installing the controller. Delay is also calculated based on the processing delay.

In the second phase, we selected the distance to the central controller, the proximity of the neighborhood, and the amount of memory to select the cluster head

Table 1 Comparison related works

Ref.	Methods	Objectives				
		Latency	Scalability	Reliability	Cost	Dynamic network
[11]	Using Modified- Dynamic Programming (DP) algorithm	*	*	–	–	–
[12]	Use survival strategy	–	–	*	–	–
[13, 14]	computation of failure rate of the control path	–	–	*	–	–
[15]	Network partitioning and coordinator controller	–	–	*	–	–
[16]	Using the optimization model	*	–	*	*	–
[17]	Focus on the controller load	*	–	–	–	–
[18]	evolutionary algorithm	*	–	–	–	–
[19]	Modified Particle Swarm Optimization (PSO) algorithm and breadth-first search algorithm	*	–	*	–	*
[20]	Using the modified no dominated sorting genetic algorithm II (NSGA-II) algorithm	*	*	–	–	–
[21]	Determine the parameters along with the hierarchical method for allocating the switch to the controller	*	*	–	–	–
[22]	Algorithm for locating and assigning switches to the controller with parameters of the switch to controller latency, controller to controller latency and capacity of controller	*	*	*	–	–
[23]	Integer linear programming	*	–	–	*	–
[24]	Integer linear programming	*	–	–	*	*
[25]	Integer linear programming	*	–	–	*	–
[26]	Parameter optimization in the heuristic algorithm	*	*	–	–	–
[27]	hierarchical clustering	*	*	*	–	–
[28]	Using iterated local search algorithm and dynamic switch allocation algorithm	*	*	*	*	*
–	Using multi-criteria decision-making	*	*	*	*	*

controller in each subdomain. The neighborhood proximity criterion indicates the number of controllers directly adjacent to each other via the connection link. After determining the cluster head in each subdomain, we connected them to the central controller. Finally, we connected the central controller to a backup central controller to use the backup controller if the central controller fails.

3.1 Multi-criteria decision-making model

In recent decades, researchers have focused on multiple-criteria decision-making (MCDM) models for making a complex decision. In these decisions, several measurement criteria may be used. For example, suppose that in a maritime transport problem, we intend to minimize shipping costs, maximize profitability and consider safety and actual factors, simultaneously, taking into account their degree of importance. This is a complex problem that cannot be easily explored by previous techniques so that multi-criteria decision-making models would help solve the problem. Also, the decision can be considered in another way, even when we have to consider different criteria. These decision-making models were divided into two main categories: Multiple-Objective Decision-Making Models (MODM) and Multiple-Attribute Decision-Making Models (MADM).

Multi-objective models are used for design, and multi-attribute models are used to select the best option. MADM is often used in cases where a specific problem is faced with several different attributes, including quantitative and qualitative attributes such as cost, degree of importance, capacity, and lifetime, simultaneously. The problem is to consider all of these attributes simultaneously and find the option in which the sum of these attributes is maximized. Most of the problems we face in the environment are multiple-attribute problems. Hence, given that in the placement problem, the goal is to choose the proper location to install the controllers and choose the proper controller to allocate the switches, we use MADM models. MADM models consist of several methods. One of these methods is Topsis. The Topsis method ranks options. In this method, two concepts of "ideal and anti-ideal solutions" and "similarity index" have been used. As its name implies, the ideal solution is the best in any way that is not generally available in practice, and we try to approach it. The distance of that option from the ideal solution and anti-ideal solution is to measure the similarity index of the option to the ideal solution and anti-ideal solution. Then, the options are evaluated and ranked based on the distance ratio from the anti-ideal solution to the total distance from the ideal and anti-ideal solutions.

3.1.1 The steps of the TOPSIS method

The following steps performed to implement the TOPSIS method [29]:

- *Forming a decision matrix*: The first step in this method is to form a decision matrix. The decision matrix of this method includes a set of criteria and options.

A matrix in which the criteria are placed in columns and the options are in a row. Each matrix cell is for evaluating each option according to each criterion. Once the decision matrix has formed, we must complete it with expert feedback. This process is done by the Likert scale or clock or real numbers. When the criterion is low for cost or production rate, we set the real number for each option. But in cases where the criterion is qualitative, and a quantitative number does not make sense for it, we use the spectrum 1 to 9 or the spectrum 1 to 5.

- *Normalizing the decision matrix*: Normalization in the TOPSIS method is done using the soft method. In this way, each matrix element is divided by the square root of the sum of squares of the elements of that criterion column. In this step, the decision matrix becomes a dimensionless matrix.
- *Weighting decision matrix*: In this step, we need to multiply the weight of the criteria obtained by other normal matrix methods to obtain the weighted matrix. (The Topsis method alone cannot calculate the weight of the criteria, so other methods such as AHP entropy should be used to calculate the weight of the criteria).
- *Obtaining ideal and anti-ideal solutions*: In this step, the type of criteria must be specified. The criteria are either ideal or anti-ideal. Ideal criteria are criteria that are increasing them to improve the system, such as the quality of a product and ideal solution is equal to the largest element of the criterion column. Anti-ideal is equal to the smallest element of the cell, and so are anti-ideal criteria in reverse.
- *Measuring the degree of distance*: In this step, we calculate each option's distance from its ideal and anti-ideals.
- *Calculating a similarity index*: The similarity index indicates the score of each option. Whatever the value is closer to 1 shows the superiority of that option.
- *Ranking options*: In this step, the options are sorted based on rating, in descending order.

3.2 The mathematical model of the problem

The network graph consists of nodes V and edges E. V contains switch nodes S and controller nodes P. In other words, $V = S \cap P$ and $S \cap P = \phi$.

Set E also contains two sets of E_P and E_S such that:

$$E_P = \{\text{ab} \in E \,|\, a, b \in P\} \tag{1}$$

$$E_S = \{\text{ab} \in E \,|\, a \in S,\ b \in P\} \tag{2}$$

Other sets include O, and C. O represents ordered pairs of possible locations to install the controller. C indicates the type of controller. In mathematical form:

$$O = \{(a, b)\ :\ a \in P,\ b \in P,\ a < b\} \tag{3}$$

$$C = \{{}^c1, {}^c2, \ldots\} \tag{4}$$

Table 2 Symbols used in the model

Description	Symbol	Description	Symbol
Set of possible locations for controllers	P	Set of switches	S
A set of pairs of possible controller installation locations	O	Set of the switch to controller and controller to controller edges	E
Installing the controller type c in place a	z_a^c	Connection cost between node a and b	ω_{ab}
The link between node a and b	x_{ab}	Cost of installing the controller type c in place a	γ_a^c
Number of available controller type c	δ^c	The flow from p to q on the edge ab	g_{ab}^{pq}
Packet processing capacity per controller type c	α^c	The number of ports per controller type c	μ^c
Set of controllers	C	Number of packets sent to controller by switch s	β^s

Due to the controller port's limitations, one controller and the other switches can connect several switches. Also, each controller can communicate with other controllers. Table 2 shows the symbols used in the model.

Decision variables:

$$x_{ab} = \begin{cases} 1 \text{ If edge}(a, b)\text{is selected,} \\ 0 \text{ Otherwise,} \end{cases}$$

$$z_a^c = \begin{cases} 1 \text{ If a controller of type } c \text{ is placed in node } a, \\ 0 \text{ Otherwise,} \end{cases}$$

$$g_{ab}^{pq} = \begin{cases} 1 \text{ If a unit flow from location } p \text{ to location } q \text{ passes edge}(a, b) \\ 0 \text{ Otherwise,} \end{cases}$$

Finally, the mathematical model of the problem:

$$\text{Formulation (1)}: \quad \min \sum_{ab \in E} x_{ab}\,\omega_{ab} + \sum_{c \in C} \gamma_a^c \sum_{a \in P} z_a^c$$

$$\sum_{ab \in E_P} g_{ab}^{pq} - \sum_{ba \in E_P} g_{ba}^{pq} = \begin{cases} \sum_{c \in C} (z_p^c * z_q^c) & a == p \\ -\sum_{c \in C} (z_p^c * z_q^c) & a == q \\ 0 & a \neq p, q \end{cases} \quad \forall a \in P, \forall \text{pq} \in O \tag{5}$$

$$\sum_{\substack{b \in P \\ ab \in E_S}} x_{ab} = 1; \quad \forall a \in S \tag{6}$$

$$x_{ab} \le \sum_{c \in C} z_b^c; \ \forall \text{ ab} \in E_S, a \in S, b \in P \tag{7}$$

$$\sum_{c \in C} z_a^c \le 1; \ \forall a \in P \tag{8}$$

$$\sum_{a \in P} z_a^c \le \delta^c; \ \forall c \in C \tag{9}$$

$$\sum_{\substack{a < b \\ b \in P}} x_{ab} + \sum_{\substack{b \in S \\ ba \in E_S}} x_{ba} \le \sum_{c \in C} \mu^c * z_a^c; \qquad \forall a \in P \tag{10}$$

$$\sum_{\substack{b \in S \\ \mathrm{ba} \in E_S}} \beta^s * x_{\mathrm{ba}} \leq \sum_{c \in C} \alpha^c * z_a^c; \quad \forall a \in P \tag{11}$$

$$x_{\mathrm{ab}} \in \{0, 1\} \quad \forall \mathrm{ab} \in E \tag{12}$$

$$z_a^c \in \{0, 1\} \quad \forall a \in P, \ c \in C \tag{13}$$

$$g_{\mathrm{ab}}^{\mathrm{pq}} \in \{0, 1\} \ \forall \mathrm{ab} \in E_P, \forall \mathrm{pq} \in O \tag{14}$$

The objective function represents the minimum costs of connecting network components (switch and controller) and deploying the controller. Constraint (5) indicates the communication between the two controllers. Constraints (6) and (7) indicate that each switch is connected to a controller. Constraint (8) indicates the deployment of only one controller in each location. Constraints (9) to (11) express the controller's limitations, including the available number, port, and processing capacity. Other explanations of the mathematical model of the problem are described in paper [28].

3.3 Computation of reliability rate

To reduce the controller's failure rate, we reduce the sum of failure points of the controller. Using the reliability indicator (*R*) prevents controllers' selection with a high probability of failure [30]. The calculation of the probability of failure is as follows: A model for selecting the reliability index criterion is introduced through the directional graph, in which the nodes identify the random variables. Equation (15) explains a joint probability assignment based on random variables.

$$P(Y_1, Y_2, \ldots.. Y_M) = \prod_{i=1}^{M} P(Y_i | \,\mathrm{pa}(Y_i)) \tag{15}$$

P (Y_i | pa (Y*i*)) represents Conditional Probability Distributions (CPDs) [31].

Different reasons could have an impact on controller malfunctions. For example, the controller's failure as a result of environmental influences, malfunction

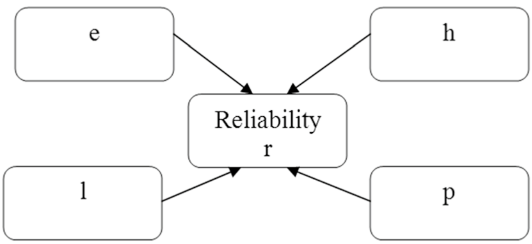


Fig. 2 Reliability selection criteria model [30]

of links, malfunctions of the controller parts, and power outage of the controller. Figure 2 shows the criteria for determining reliability.

- $P(h)$: The probability of failure due to malfunctioning of the physical components within the controller
- $P(e)$: Probability of controller failure due to environmental risks
- $P(l)$: The probability of failure due to improper connection of the link or communication channel
- $P(p)$: The probability of malfunctions due to power outage

$P(r)$ = the probability of controller failure due to the occurrence of any of the above four cases. Using Eq. (15), the probability of failure of the controller computed as follows:

$$P(r) = \prod_{i=1}^{M} P(r \mid \mathrm{pa}(r_i)) \tag{16}$$

In Eq. (16), $\mathrm{pa}(r_i)$ expresses the conditional attributes among the controllers. Now, by substituting the attributes in Eq. (16), we get:

$$P(r) = \prod_{i=1}^{M} P(r \mid h, e, l, p) \tag{17}$$

E, *l*, *h*, *p*, and *r,* respectively, indicate the probability of environmental, link, hardware, power failure, and the installed controller's reliability index. The probability of controller failure due to each failure causes (*h*, *e*, *1*, or *p*) is shown in Fig. 2.

- *h*—The controller's malfunction is mainly due to software malfunction or software unavailability malfunction of internal units.
- *e*—Environmental conditions usually are not persistent and always unforeseeable.
- *l*—Path malfunctioning which is due to link failure.
- *p*—Power outages to occur by a sudden power outage in the city or power supply burnout and electrical connections failure due to inappropriate deployment of the device.

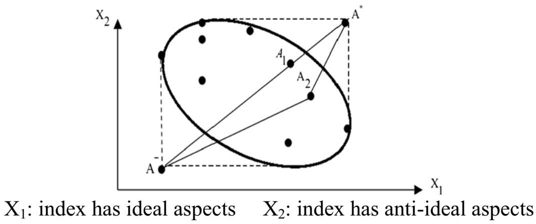


Fig. 3 Ideal solution and anti-ideal solution [29]

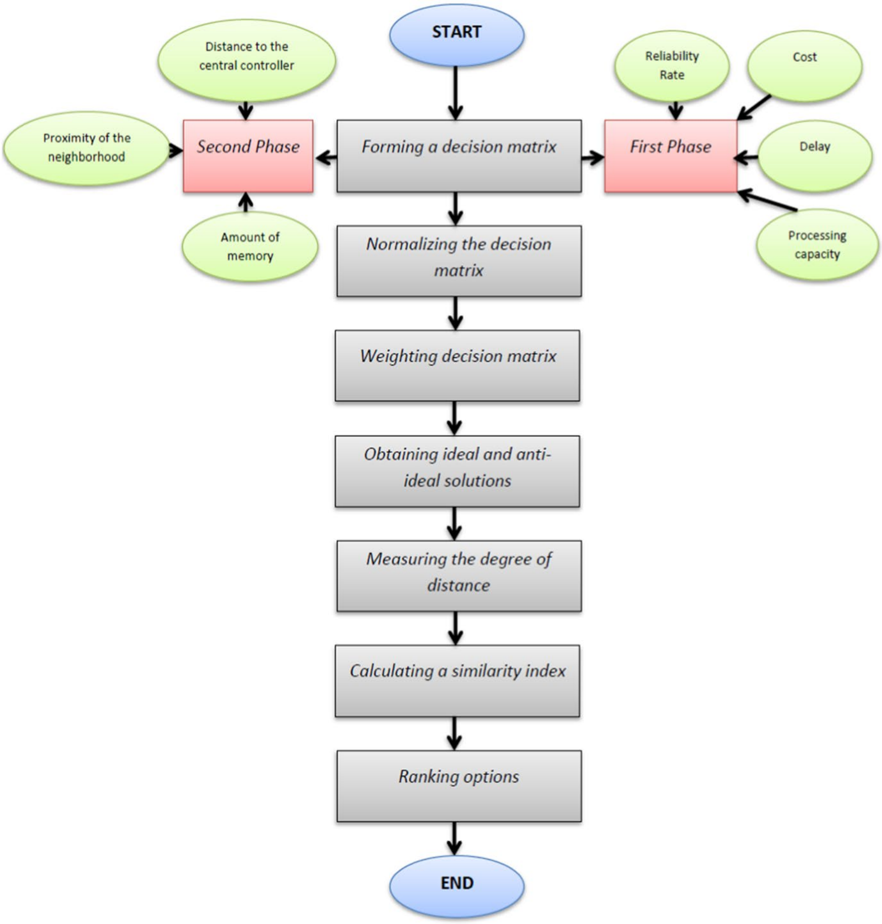


Fig. 4 Flowchart of the proposed method algorithm

3.4 The proposed method algorithm

The Topsis method's logic is that the selected option and its alternative should be the lowest distance with the ideal solution and the highest distance with the anti-ideal solution. According to Fig. 3, A^* and A^- represent the ideal and anti-ideal solutions, respectively. Compared with the A_2, the A_1 is located at the lowest distance from the ideal and highest distance from the anti-ideal.

The proposed algorithm is listed below and represented using the flowchart as given in Fig. 4.

(1) *Forming a decision matrix:* This matrix is calculated with *m* options and *n* indicators. The indicator with desirable benefit is called the *ideal index,* and the indicator with undesirable benefit is considered the *anti-ideal index*. In this

$$D_1 = \begin{array}{c} \\ A_1 \\ A_2 \\ . \\ . \\ A_m \end{array} \begin{array}{c} \begin{matrix} X_1 & X_2 & X_3 & X_4 \end{matrix} \\ \begin{bmatrix} X_{11} & X_{12} & X_{13} & X_{14} \\ X\ 21 & X_{22} & X_{23} & X_{24} \\ . & . & . & . \\ . & . & . & . \\ X_{m1} & X_{m2} & X_{m3} & X_{m4} \end{bmatrix} \end{array}$$

Fig. 5 Decision matrix for the first phase in the proposed method

$$D_2 = \begin{array}{c} \\ A_1 \\ A_2 \\ . \\ . \\ A_m \end{array} \begin{array}{c} \begin{matrix} X_1 & X_2 & X_3 \end{matrix} \\ \begin{bmatrix} X_{11} & X_{12} & X_{13} \\ X_{21} & X_{22} & X_{23} \\ . & . & . \\ . & . & . \\ X_{m1} & X_{m2} & X_{m3} \end{bmatrix} \end{array}$$

Fig. 6 Decision matrix for the second phase in the proposed method

algorithm, in the first phase, m has 10 to 30, which indicates the number of possible locations for the controller to be installed, and n equals 4, which indicates the number of indicators. In the first phase, these indicators include reliability rate, cost, delay, and processing capacity. The ideal index is the reliability rate and processing capacity, and the anti-ideal index is the cost and delay. In the second phase, to select the cluster head controller, m has values of 1 to 10, which indicates the number of controllers in each subdomain, and n equals 3, which indicates the number of indicators. In the second phase, these indicators include the distance to the central controller, the neighborhood's proximity, and the amount of memory. The ideal index is the proximity of the neighborhood and the amount of memory, and the anti-ideal index includes the distance to the central controller. Figures 5 and 6 of the decision matrices show the proposed method.

In matrices D_1 and D_2, A_i represents the ith location and the ith controller, respectively. X_{ij} is the operation of option i in relation to indicator j.

(2) *Normalizing the decision matrix*: The scales are removed from the decision matrix. That is, the values are divided by the vector size corresponding to that indicator. Thus, the r_{ij} in the matrix is obtained through Eq. (18).

$$r_{\mathrm{ij}} = \frac{X_{\mathrm{ij}}}{\sqrt{\sum_{i=1}^{m} X_{\mathrm{ij}}^2}} \tag{18}$$

In this equation, r_{ij} represents the normalized value of the X_{ij} element, and m is the number of controller installation locations in the first phase or the number of controllers in the second phase.

(3) *Weighting decision matrix*: The decision-maker determines the weight for each indicator. In this algorithm, weights for the reliability index, cost, delay, and processing capacity in the first phase are 0.3, 0.3, 0.2, and 0.2, respectively. The reason for determining these weights is because, as stated in [5], network failure causes a disconnection between network components, as a result of which packets are lost, and network performance is reduced. The network cost should not be overlooked at the time of implementation according to the set budget. Besides, cost and survivability are considered as the objective functions of the problem. Delay is very important because, in the SDN network, exchanging messages between controllers and switches performs all network operations. Also, the assignment of switches to controllers is done, taking into account the controllers' processing capacity. Hence, it is important to determine the processing capacity of the controller as one criterion. Also, weights for the distance to the central controller index, such as the neighborhood's proximity, and the amount of memory in the second phase are 0.4, 0.4, and 0.2, respectively. These weights are determined because the controller that is close to the central controller or in the vicinity of many controllers is in good condition in terms of criteria such as latency, connection cost and the possibility of failure. Also, a large amount of memory allows the controllers to receive more packets sent for processing.

$$W = (w_1, w_2, \ldots, w_j, \ldots, w_n)$$

$$\sum_{j=1}^{n} w_j = 1 \tag{19}$$

In Eq. (19), w_j represents the weight of criteria.

(4) *Obtaining ideal and anti-ideal solutions*: A^- and A^* are the two virtual options in Eqs. (20) and (21).

Ideal solution:

$$A^* = \{(\max\ v_{ij} \mid j \in J), (\min\ v_{ij} \mid j \in J') \mid i = 1, 2, \ldots, m\} = \{v_1^*, v_2^*, \ldots, v_j^*, \ldots, v_n^*\} \tag{20}$$

Anti-ideal Solution:

$$A^- = \{(\min\ v_{ij} \mid j \in J), (\max\ v_{ij} \mid j \in J') \mid i = 1, 2, \ldots, m\} = \{v_1^-, v_2^-, \ldots, v_j^-, \ldots, v_n^-\} \tag{21}$$

jj $\rightarrow J = \{j = 1, 2, 3, \ldots, n\}$ The ideal index.
jj $\rightarrow J' = \{j = 1, 2, 3, \ldots, n\}$ The anti-ideal index.

Table 3 Information of experimented topologies

Topology	|N|	|P|	|S|
Uran	24	10	14
SwitchL3	42	12	30
Sinet	74	14	60

Table 4 Problem-solving parameters

Parameter	Value		
	Controller 1	Controller 2	Controller 3
The cost of each controller	1200$	2500$	6500$
The number of ports per controller	8	16	32
The processing capacity of each controller	2000 KB	4000 KB	8000 KB
Link cost per meter	8.25$		
Packet Size-Static	250 KB		
Packet size-Dynamic	100–400 Byte		

(5) *Measuring the degree of distance*: Using Eqs. (22) and (23) is measured the distance of option i from the ideal and anti-ideal options.

$$S_{i*} = \sqrt{\sum_{j=1}^{n} \left(v_{\text{ij}} - v_j^*\right)^2} \quad i = 1, 2, 3, \ldots, m \tag{22}$$

$$S_{i-} = \sqrt{\sum_{j=1}^{n} \left(v_{\text{ij}} - v_j^-\right)^2} \quad i = 1, 2, 3, \ldots, m \tag{23}$$

(6) *Calculating a similarity index*: The similarity index is measured through Eq. (24).

$$C_{i*} = \frac{s_{i-}}{s_{i*} + s_{i-}} \quad 0 < C_{i*} < 1 \tag{24}$$

In this equation, C_{i*} indicates the similarity index of the option to the ideal solution. If $A_i = A^*$, therefore, $C_{i*} = 1$; also, if $A_i = A^-$, then, $C_{i*} = 0$.

(7) *Ranking options*: Finally, the options are sorted in the descending order.

Table 5 Symbols used in the experiments

Description	Symbol	Description	Symbol
Link disruption probability	P_{LD}	Link congestion probability	P_{LC}
Connection failure probability	P_{CF}	Link disruption between *i* and *j*	e_{ij}
Number of subdomains	k	Probability of controller failure in subdomain *i*	$P_i(r)$
Distance between *i* and *j*	d_{ij}	Connection link between *i* and *j*	y_{ij}
Probability of link failure per unit length	p_{ul}	Traffic direction between *i* and *j*	α_{ij}
link bandwidth between *i* and *j*	β_{ij}	Flow request rate of the switch *i*	δ_i
Propagation delay	d_{prop}	Processing delay	d_{proc}
Transmission delay	d_{tran}	Controller load-balancing rate	R_{CLB}
Number of switches in the domain *j*	ω_j		

4 The experiments and computational analysis

In this section, we performed experiments on the proposed method in comparison with CPMDP [11] and CPAHP [21] methods based on the criteria stated in [32]. The CPMDP method uses clustering to the deployment of controllers. The CPAHP method uses three criteria: delay, hop count, and link usage to allocate switches to controllers. These experiments performed on three topologies named Uran, SwitchL3, and Sinet from the Internet Topology Zoo [33]. Table 3 shows the information of these topologies.

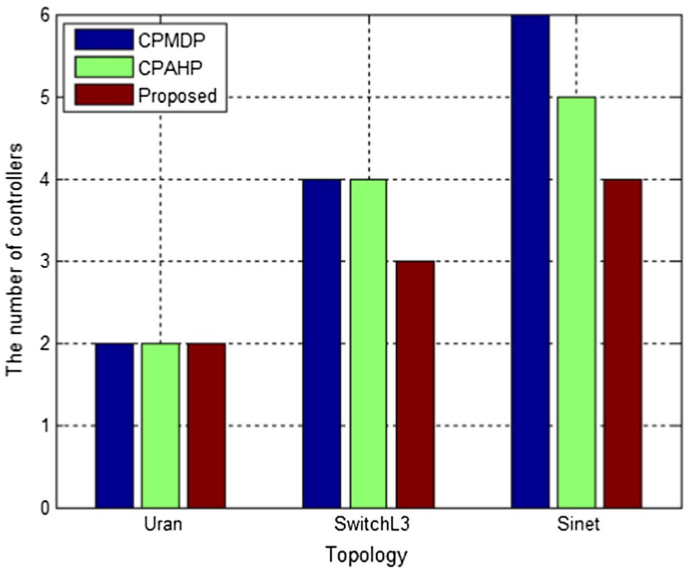


Fig. 7 Number of installed controllers in different topologies

Calculations of this section have performed on an Intel Core i5 processor under the Windows operating system with 8 GB of main memory. The proposed and comparable methods have been implemented with the help of MATLAB software. The parameters and symbols used in these experiments are described in Tables 4 and 5.

4.1 The amount of required controller

According to the three topologies Uran, SwitchL3, and Sinet, we obtain the number of required controllers for each method compared. As shown in Fig. 7, as the size of the network increases, the number of controls also gradually increases. As shown in Fig. 7, for the CPMDP method, this increase in the number of controllers is visible. For the CPAHP method, increasing the number of controllers is less than CPMDP method. Compared to the two methods described, the proposed method selects the most appropriate nodes according to the defined criteria such as reliability rate, cost, delay, and processing capacity. Therefore, regardless of the size of the network, the proposed method can deploy controllers more optimally and reduce the number of controllers.

4.2 Connection failure probability

To compute the connection failure probability, we consider the probability of the controller failure and the probability of the link failure. Link failure probability includes link disruption probability (P_{LD}) and link congestion probability (P_{LC}). In Eq. (25), this calculation is performed:

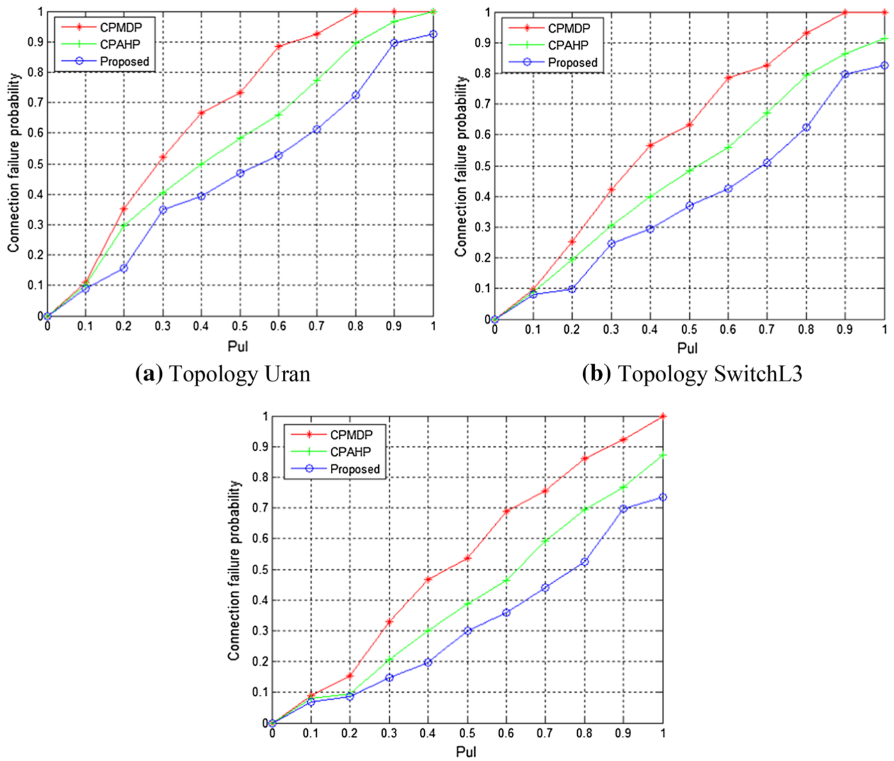
$$P_{\mathrm{CF}} = \frac{1}{k}\sum_{i=1}^{k}\left[P_i(r) + \sum_{j=1, j\neq i}^{n}\left(P_{\mathrm{LD}}(e_{\mathrm{ij}}) + P_{\mathrm{LC}}(e_{\mathrm{ij}})\right)\right] \tag{25}$$

P_i (r) is obtained based on Eq. (17). P_{LD} (e_{ij}) and P_{LC} (e_{ij}) are calculated using Eqs. (26) and (27). The variable k indicates the number of subdomains.

$$P_{\mathrm{LD}}(e_{\mathrm{ij}}) = \begin{cases} \dfrac{d_{\mathrm{ij}}.p_{\mathrm{ul}}}{y_{\mathrm{ij}}} & y_{\mathrm{ij}=1} \\ \infty & y_{\mathrm{ij}=0} \end{cases} \tag{26}$$

In this equation, P_{LD} (e_{ij}) indicates the probability of link disruption between i and j. d_{ij} and y_{ij} indicate the distance between i and j, and the connection link between these two nodes, respectively. p_{ul} also indicates the probability of link failure per unit length. In these experiments, the value of p_{ul} is obtained for every 1 km.

$$P_{\mathrm{LC}}(e_{\mathrm{ij}}) = \begin{cases} \dfrac{\alpha_{\mathrm{ij}}.\delta_i + \alpha_{\mathrm{ji}}.\delta_j}{y_{\mathrm{ij}}.\beta_{\mathrm{ij}}} & y_{\mathrm{ij}=1} \\ \infty & y_{\mathrm{ij}=0} \end{cases} \tag{27}$$



(a) Topology Uran

(b) Topology SwitchL3

(c) Topology Sinet

Fig. 8 Probability of connection failure

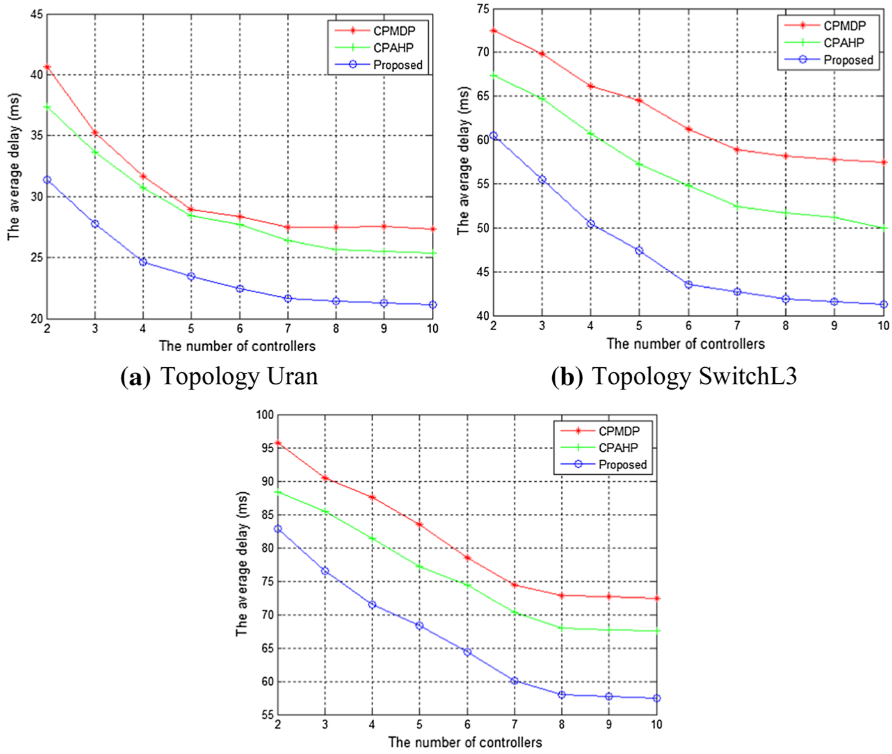
In this equation, the P_{LC} (e_{ij}) indicates the probability of link congestion between i and j. α_{ij} and β_{ij} indicate the traffic direction and link bandwidth, respectively. If the direction of traffic is from i to j, then $\alpha_{ij}=1$; otherwise, $\alpha_{ij}=0$. δ_i shows the flow request rate of the switch i.

Now, we examine the probability of connection failure for the three topologies Uran, SwitchL3, and Sinet. The result of this experiment is shown in Fig. 8.

For this reason, in case of failure in the communication link, the probability of connection failure is maximized. CPMDP and CPAHP methods consider the shortest path between nodes, thus preventing some connection disconnections. Given that in the proposed method, one of the criteria for selecting a controller is the reliability rate, so it can reduce the probability of connection failure.

4.3 The average delay

This delay consists of the propagation d_{prop}, the processing d_{proc}, and the transmission d_{tran} delays. In this experiment, the values of $d_{proc}=0.01$ ms and



(a) Topology Uran **(b)** Topology SwitchL3 **(c)** Topology Sinet

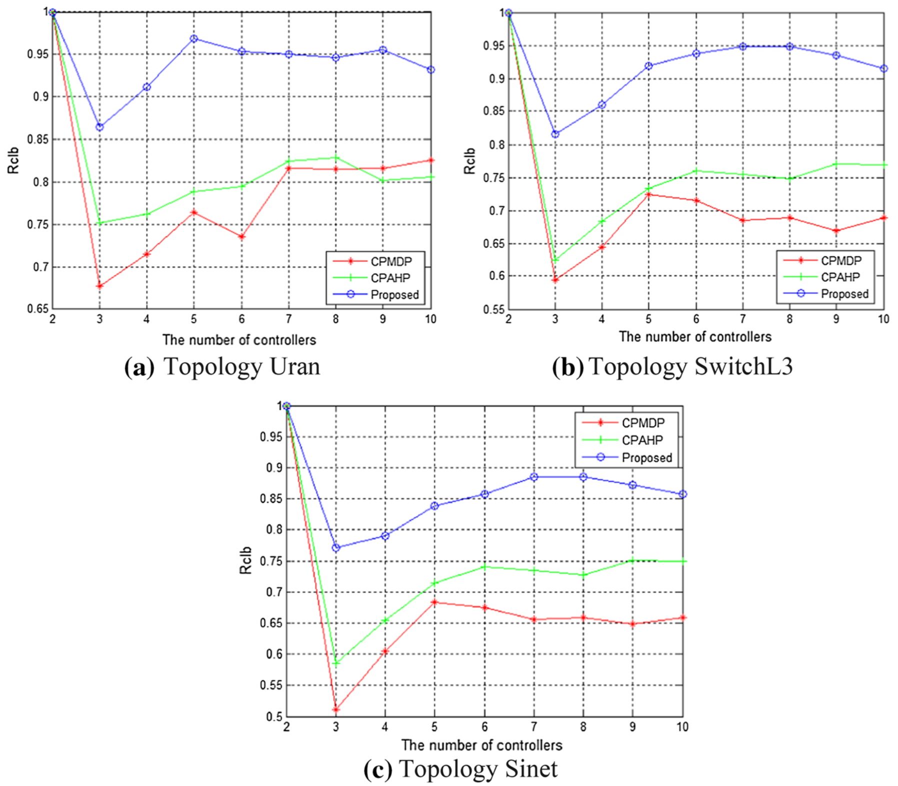
Fig. 9 Average delay

$d_{tran}=0.05$ ms are considered. Also, the value of d_{prop} increases by 0.1 ms for every 1 km. Accordingly, by changing the number of controllers, we calculate the average delay of each method. The result of this experiment is shown in Fig. 9.

As shown in Fig. 9, by increasing the number of controllers, the average latency of all three methods decreases. The proposed method performs better in reducing latency compared to the other two methods. This is due to the choice of reliable locations for the controller, the balance in the distribution of switches, and the consideration of the delay criterion when allocating the switch to the controllers. Therefore, the average delay has a significant decrease. When the number of controllers reaches higher than 7, the delay is relatively stable.

4.4 Controller load-balancing rate

We use Eq. (28) to calculate the controller load-balancing rate. A higher load-balancing rate indicates better performance of controllers in balancing load. One of the most widely used traffic models is the Poisson model. Thus, the Poisson distribution



(a) Topology Uran **(b)** Topology SwitchL3 **(c)** Topology Sinet

Fig. 10 Load-balancing rate

is used to analyze network traffic. This distribution is suitable for situations where inputs are obtained from a large number of independent sources.

$$R_{\mathrm{CLB}} = \sum_{j=1}^{k} \left[\frac{\omega_j}{n} \cdot \frac{\sum_{m=1}^{\omega_j} \delta_m}{\sum_{i=1}^{n} \delta_i} \right] \tag{28}$$

ω_j represents the number of switches in the domain j. In this experiment, the value of R_{CLB} for each method compared to the number of different controllers is calculated. The result of this experiment is shown in Fig. 10.

When the number of controllers is 2, the controllers are overloaded, so the R_{CLB} value is high. By increasing the number of controllers, the CPMDP method has a low and unstable R_{CLB} value. The CPAHP method, because of the use of link utilization metric to control traffic, the R_{CLB} value is maintained at an average of about 0.77. The proposed method tries to balance the allocation of switches to each controller by considering criteria such as reliability rate and the delay. Therefore, the R_{CLB} value in the proposed method is averaging above 0.90, which increases compared to the other two methods in experimented topologies.

Table 6 Comparison of the proposed method with CPMDP and CPAHP methods

Topology name	|P|	|S|	Proposed	CPMDP	CPAHP	Imp proposed vs. CPMDP %	Imp proposed vs. CPAHP %
Uran	10	14	330,405.4	370,910	364,191.6	12.26	10.23
SwitchL3	12	30	152,434.9	196,920	189,630	29.18	24.40
Sinet	14	60	887,489	1,184,600	1,106,300	33.48	24.66

4.5 Implementation cost

In this experiment, the results related to the best solution to the cost criterion are reported in Table 6. These costs include the total cost of connecting the switch to the controller and the controller to the controller and the controller's cost of deployment. In Table 6, the first column shows the name of the topology being experimented. The second and third columns indicate the number of possible places to install the controller and the number of switches, respectively. The fourth to sixth columns are the best solution obtained by the proposed, CPMDP, and CPAHP methods. The seventh and eighth columns also show the percentage of improvement of the proposed method compared to CPMDP and CPAHP methods, which are calculated using Eq. (29).

$$\mathrm{Imp} = \frac{\mathrm{Cost}t_{\mathrm{CPMDP}\ or\ \mathrm{CPAHP}} - \mathrm{Cost}t_{\mathrm{pro}}}{\mathrm{Cost}t_{\mathrm{pro}}} * 100 \tag{29}$$

In Eq. (29), $\mathrm{Cost}t_{\mathrm{CPMDP}\ or\ \mathrm{CPAHP}}$ and $\mathrm{Cost}t_{\mathrm{pro}}$ functions give the best cost found by the CPMDP or CPAHP methods and the proposed method, respectively.

Table 6 indicates that it can be concluded that the proposed method works better in locating the controllers than CPMDP and CPAHP methods. This superiority is the proper design of the control plane architecture using criteria such as reliability rate, cost, latency, and processing capacity. In contrast, the compared methods use a full-mesh topology to connect the controllers. This topology imposes a high cost on the network to connect the controllers to the network. Also, it occupies the controller ports for direct communication with each other.

In the following, the results related to the cost of implementing the network are reported when the survivability has different degrees. This is due to the cost of implementing the network for different conditions of the network environment. Because a low degree of survivability is sufficient to implement the network in a secure environment, a high degree of survivability is required for more insecure environments. Figure 11 shows the cost results for different R values indicating the degree of survivability, which is calculated based on the formulation (4) stated in the paper [28].

Figure 11a shows that the proposed method for $R=1,2,3$ is less expensive than CPMDP and CPAHP methods. Also, in Fig. 11b, the proposed method cost with $R=2$ is lower than CPMDP and CPAHP with $R=1$. i.e., the cost of the proposed method for

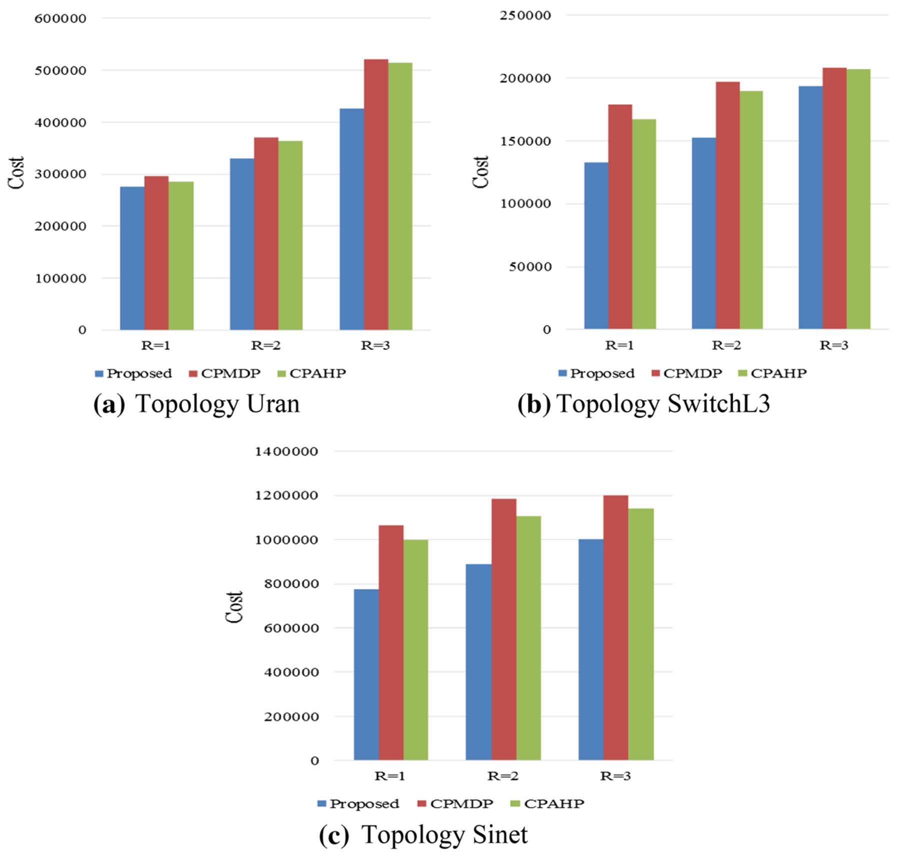


Fig. 11 Comparison of implementation cost based on different values of *R*

R=2 is 152434.86, while the cost of CPMDP and CPAHP for *R*=1 is 179164.99 and 167,350, respectively. This means that the proposed method, compared to the CPMDP and CPAHP, can design a network that increases survivability while reducing cost. Finally, it can be concluded that the proposed method is more cost-effective for large-scale networks than the CPMDP and CPAHP, even when the degree of survivability increases compared to the CPMDP and CPAHP.

5 Conclusion

One of the most critical problems in SDN networks, especially SDN-WAN networks, is the controller's location. In this problem, the goal is to determine the controllers' proper location and the optimal allocation of switches. Therefore, the control plane architecture design has a significant impact on this type of network's efficiency. Also, two essential criteria that are less considered in this type of network are cost and survivability. Therefore, one of the main concerns of this type of network providers is to reduce the cost of their implementation. In case of network failure, the network continues operating, so network survivability is essential. This paper used the multi-criteria decision method to solve the controller placement problem and optimally designed the control plane architecture. We considered criteria such as reliability rate, cost, delay, and processing capacity.

Furthermore, according to the proposed architecture, we used distance to the central controller, neighborhood proximity, and memory size criteria to select the cluster head controller for each subdomain to communicate with the central controller. Then, we present a mathematical model in which the objective functions are to reduce costs and increase network survivability. Thus, this reduction in cost and increase in survivability is made according to the considered criteria. Finally, to evaluate our method, we performed experiments on three topologies named Uran, SwitchL3, and Sinet from the Internet Topology Zoo. We compared our method's results with CPMDP and CPAHP methods in terms of some criteria, such as the number of controllers, connection failure probability, controller load balancing, average delay, and implementation cost. The state of the network was also examined in terms of dynamics and events that cause network failure. The results showed that the proposed method has a better performance in reducing costs and increasing the network's survivability. Thus, the proposed method has a 24.97% and 19.76% improvement in reducing network implementation costs than CPMDP and CPAHP, respectively. Therefore, the summary of these results indicates that the proposed method, using a suitable mathematical model close to the actual network conditions, can solve the controller placement problem. One of the drawbacks we face in solving this problem is the time required to run the algorithm, which can be improved using heuristic algorithms. Also, the lack of robust hardware equipment to perform experiments on large-scale topologies is one of the shortcomings of this research.

We will try to solve the controller placement problem with traffic forecasting based on machine learning in future work. We also plan to expand our work in the future by addressing the dynamic controller placement to meet the needs of 5G networks, such as reliable low-latency communications.

References

1. Blial O, Ben Mamoun M, Benaini R (2016) An overview on SDN architectures with multiple controllers. J Comput Netw Commun 2016:1–8
2. Selvi H, Güner S, Gür G, Alagöz F (2015) The controller placement problem in software defined mobile networks (SDMN). In: Liyanage M, Gurtov A, Ylianttila M (eds) Software defined mobile networks (SDMN): beyond LTE network architecture. Wiley, Chichester, pp 129–147
3. Nunes BAA, Mendonca M, Nguyen XN, Obraczka K, Turletti T (2014) A survey of software-defined networking: Past, present, and future of programmable networks. IEEE Commun Surv Tutor 16(3):1617–1634
4. Xia W, Wen Y, Foh CH, Niyato D, Xie H (2015) A survey on software-defined networking. IEEE Commun Surv Tutor 17(1):27–51
5. Lu J, Zhang Z, Hu T, Yi P, Lan J (2019) A survey of controller placement problem in software-defined networking. IEEE Access 7:24290–24307
6. Jammal M, Singh T, Shami A, Asal R, Li Y (2014) Software defined networking: state of the art and research challenges. Comput Netw 72:74–98

7. Jarraya Y, Madi T, Debbabi M (2014) A survey and a layered taxonomy of software-defined networking. IEEE Commun Surv Tutor 16(4):1955–1980
8. Xie J, Guo D, Hu Z, Qu T, Lv P (2015) Control plane of software defined networks: a survey. Comput Commun 67:1–10
9. Wang G, Zhao Y, Huang J, Duan Q, Li J (2016) A K-means-based network partition algorithm for controller placement in software defined network. In 2016 IEEE International Conference on Communications (ICC) (pp. 1–6). IEEE.
10. Michel O, Keller E (2017) SDN in wide-area networks: A survey. In 2017 Fourth International Conference on Software Defined Systems (SDS) (pp. 37–42). IEEE
11. Sminesh CN, Kanaga EGM, Roy A (2019) Optimal multi-controller placement strategy in SD-WAN using modified density peak clustering. IET Commun 13(20):3509–3518
12. Müller LF, Oliveira RR, Luizelli MC, Gaspary LP, Barcellos MP (2014) Survivor: an enhanced controller placement strategy for improving SDN survivability. In Global Communications Conference (GLOBECOM), 2014 IEEE (pp. 1909–1915). IEEE
13. Hu YN, Wang WD, Gong XY, Que XR, Cheng SD (2012) On the placement of controllers in software-defined networks. J China Univ Posts Telecommun 19:92–171
14. Hu Y, Wendong W, Gong X, Que X, Shiduan C (2013) Reliability-aware controller placement for software-defined networks. In Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on (pp. 672–675). IEEE
15. Moazzeni S, Khayyambashi MR, Movahhedinia N, Callegati F (2018) On reliability improvement of software-defined networks. Comput Netw 133:195–211
16. Killi BPR, Rao SV (2019) Towards improving resilience of controller placement with minimum backup capacity in software defined networks. Comput Netw 149:102–114
17. Lin SC, Wang P, Luo M (2016) Control traffic balancing in software defined networks. Comput Netw 106:260–271
18. Torkamani-Azar S, Jahanshahi M (2020) A new GSO based method for SDN controller placement. Comput Commun 163:91–108
19. Li G, Wang X, Zhang Z (2019) SDN-based load balancing scheme for multi-controller deployment. IEEE Access 7:39612–39622
20. Jalili A, Keshtgari M, Akbari R (2018) Optimal controller placement in large scale software defined networks based on modified NSGA-II. Appl Intell 48(9):2809–2823
21. Jalili A, Keshtgari M, Akbari R, Javidan R (2018) Multi criteria analysis of controller placement problem in software defined networks. Computer Communications
22. Tanha M, Sajjadi D, Ruby R, Pan J (2018) Capacity-aware and delay-guaranteed resilient controller placement for software-defined WANs. IEEE Trans Netw Serv Manag 15:991–1005
23. Sallahi A, St-Hilaire M (2015) Optimal model for the controller placement problem in software defined networks. IEEE Commun Lett 19(1):30–33
24. Sallahi A, St-Hilaire M (2017) Expansion model for the controller placement problem in software defined networks. IEEE Commun Lett 21(2):274–277
25. Khorramizadeh M, Ahmadi V (2018) Capacity and load-aware software-defined network controller placement in heterogeneous environments. Comput Commun 129:226–247
26. Li Y, Guan S, Zhang C, Sun W (2020) Parameter optimization model of heuristic algorithms for controller placement problem in large-scale SDN. IEEE Access 8:151668–151680
27. Ali J, Roh BH (2021) Quality of service improvement with optimal software-defined networking controller and control plane clustering. CMC-COMPUT MATER CONTINUA 67(1):849–875
28. Abdi Seyedkolaei A, Hosseini Seno SA, Moradi A (2021) Dynamic controller placement in software-defined networks for reducing costs and improving survivability. Trans Emerg Telecommun Technol 32(1):1–17
29. Shelebaf A, Tabatabaei S (2020) A novel method for clustering in WSNs via TOPSIS multi-criteria decision-making algorithm. Wirel Pers Commun 112:985–1001
30. Murugaanandam S, Ganapathy V (2019) Reliability-based cluster head selection methodology using fuzzy logic for performance improvement in WSNs. IEEE Access 7:87357–87368
31. Hassas Yeganeh S, Ganjali Y (2012) Kandoo: a framework for efficient and scalable offloading of control applications. In Proceedings of the first workshop on hot topics in software defined networks (pp. 19–24). ACM
32. Hu T, Yi P, Zhang J, Lan J (2018) Reliable and load balance-aware multi-controller deployment in SDN. China Commun 15(11):184–198

33. Knight S, Nguyen HX, Falkner N, Bowden R, Roughan M (2011) The internet topology zoo. IEEE J Sel Areas Commun 29(9):1765–1775

Publisher's Note Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Authors and Affiliations

Ali Abdi Seyedkolaei[1] · **Seyed Amin Hosseini Seno**[1]

Ali Abdi Seyedkolaei
a.abdi@mail.um.ac.ir

[1] Faculty of Engineering, Ferdowsi University of Mashhad, Mashhad, Iran