IEEE Access
Multidisciplinary | Rapid Review | Open Access Journal

Received July 20, 2021, accepted September 6, 2021, date of publication September 16, 2021, date of current version September 24, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3113496

Cost-Effective Survivable Controller Placement in Software-Defined Networks

ALI ABDI SEYEDKOLAEI[1,2], SEYED AMIN HOSSEINI SENO[1], AHMAD MORADI[2], AND RAHMAT BUDIARTO[3]

[1]Department of Computer Engineering, Ferdowsi University of Mashhad, Mashhad 9177948974, Iran
[2]Department of Mathematical Sciences, University of Mazandaran, Babolsar 4741613534, Iran
[3]Faculty of Computer Science, Universitas Mercu Buana, West Jakarta 11650, Indonesia

Corresponding author: Seyed Amin Hosseini Seno (Hosseini@um.ac.ir)

ABSTRACT One of the problems raised in software-defined networks (SDN) is to determine the number and installation location of controllers so that the implementation cost reduced, and the survivability of the network against link or node failure increased. The current investigations in SDN focus on directly linking controllers to each other in the design of control plane. This approach, while incurring a considerable installation cost, does not carefully consider network survivability requirements. In this paper, we introduce integer-programming formulations to address controller placement problems and demonstrate through careful computational studies that the proposed method is capable of increasing network survivability while reducing the cost of network implementation. Also, due to the conditions of the environment implementation, the degree of survivability can be received as input parameter. The proposed method was implemented on different topologies and then was analyzed and compared to the optimal model for the controller placement problem (OMCPP) and reliable capacitated controller placement problem (RCCPP). Experiment results show that the proposed method has an improvement of 18.33% and 22.40% in terms of implementation costs compared to OMCPP and RCCPP methods.

INDEX TERMS Controller placement, mixed integer programming, network design, survivability, software-defined networks, cost.

I. INTRODUCTION

A software-defined network as a new generation of networks allows users and network administrators to manage and control many networks equipment, services, and network infrastructure. In software-defined networks, unlike traditional networks, where the data plane and the control plane are integrated, they are located separately [1]. SDN's emergence has attracted the attention of many researchers to its implementation in its communications infrastructure [2]. Configuration methods are often more straightforward and precise on these networks [3].

An SDN network provides ability to reach a programmable network [4]. The separation of the control plane from the global data plane has brought benefits such as better network management and increased network efficiency. In the SDN, the control plane for the data plane provides the data needed for the routing. The data plane transfers packets based on its routing table. Moreover, the control plane manages the data plane and the flow in the network [5], [6]. This mechanism motivates a variant of the location problem called Controller Placement Problem (CPP). In SDN networks, the control plane plays an important role in examining the challenges of SDN-based networks, so that network performance would be affected by the challenges and related issues. For example, a control plane architecture influences network latency reduction, availability, implementation cost, and network efficiency. Therefore, in our work, we focused on the design of the control plane and how it communicates to the data plane, and but did not consider the connection of the switches to each other, which is related to the data plane. Therefore, the purpose of the CPP is to focus on the design of the control plane architecture and the communication with the data plane.

In CPP, it is necessary to determine the locations for the controllers to which the switches are connected so that criteria such as cost, load balance, and delay that have desired values [7]. The CPP problem is one of the NP-hard problems due to its time complexity [8]. In this paper, we

The associate editor coordinating the review of this manuscript and approving it for publication was Tiago Cruz.

introduce integer programming formulations to address controller placement problem and demonstrate through careful computational studies that the proposed method is able to reduce the cost of network implementation, increase the degree of survivability of the network, and receive the degree of survivability as an input parameter, according to the conditions of the network implementation environment.

Survivability refers to the network's ability to perform a set of tasks assigned to a network component that can be characterized by a number of effective services during a failure time. However, reliability is the ability of the network to perform a set of tasks assigned under specific conditions for operating times [9]. Therefore, in order to stabilize the network in the event of failure, we use the survivability parameter in our work.

The innovations that we offer in this article include:

- Reducing the implementation costs by considering survivability.
- Determining the degree of survivability as an input parameter according to the conditions of the network implementation environment.
- Providing a mathematical model of the problem in the form of integer programming to solve the problem more effectively.

The rest of the paper is structured as follows. ''Literature Review'' Section explains the existing literatures on the CPP. ''The Mathematical Expression of the Problem'' describes the proposed mathematical formulation of the problem. ''Survivable Controller Placement'' examines the controller placement when a network failure occurs. ''The Simulation Results'' analyzes the findings of the simulation of the proposed formulations on experimented topologies in comparison with the OMCPP [10] and RCCPP [11] models. Finally, Conclusions from the simulation analysis are reported in ''Conclusions''.

II. LITERATURE REVIEW

In the last decade, many studies were concentrated on CPP. Researchers in [8] focused on the average delay and the worst delay in order to select the controller location. They considered the shortest path between the switches to reduce latency. Xiao *et al.* in [12] used spectral clustering to locate the controller. In this way, first the network is divided into several areas and in each area the location of the controller is determined. Hu *et al.* in [13] worked on the criterion of network reliability. In their work, choosing the right location for the controllers increased the reliability.

Researchers in [14] used the graph-parsing algorithm to reduce the possibility of disconnection between the switch and its controller. Obadia *et al.* [15] used the spanning-tree to solve the CPP problem. Their proposed method reduces the overhead of controllers. Researchers in [16] solved the CPP problem by considering the capacity limitation of controllers. Zhang *et al.* [17] investigated different topologies with considering different locations of the controllers. Santos *et al.* [18] used a tree sub graph to optimize the controller, which can be updated to increase availability. Furthermore, some limitations on latency and availability were considered in the mathematical model of the problem.

In Ali and Roh [19], the controllers were hierarchically clustered according to the obtained rankings. Mohanty *et al.* [11] used a mathematical model to locate controllers so that the network was reliable.

Sallahi and St-Hilaire [10] used the integer programming method to reduce the cost of network implementation. In this method, the costs of connecting the switches to the controllers and the controllers to each other were considered. In another study, Sallahi and St-Hilaire [20] improved their work in [10] so that the number of switches in the network can be changed; therefore, the network topology can be changed. However, the proposed method is not applicable for large-scale networks. Therefore, due to the use of solvers such as CPLEX, only small-scale networks can optimize the results in an acceptable time, and, the NP-hard problem must be optimized. In addition, according to the author in [9], approximately 10% of the problems cannot be solved in less than 30 hours.

Tanha *et al.* [21] used the delay parameters and the controller capacity to solve the controller placement problem. Singh *et al.* [22] focused on reliable controller placement. In addition, they considered reducing latency in their work. Lin *et al.* [23] proposed a new controller placement scheme to reduce costs for software-defined vehicular networks. In the proposed method, first, the minimum number of controllers is selected. Then the bee-cloning algorithm is used to determine whether the controller is on or off to transfer data based on real-time traffic flow.

Guo *et al.* [24] focused on reducing communication overhead on controllers when controllers failed. They used the concept of RetroFlow in their work, which allows active controllers to get rid of the control of offline switches, while maintaining flow programmability. In another study in [25], Guo *et al.* used programmability guardian to improve path programmability so that communication overhead was reduced.

He and Oki [26] proposed a model for allocating master and slave controllers for when multiple controllers may fail. In their proposed method, they programmed the mathematical model of the problem with three different objectives. These goals include the average-case expected latency, the worst-case expected latency, and the expected number of switches. Dou *et al.* [27] used an adaptive solution to recover offline flows called Matchmaker when controllers failed in SD-WANs. The proposed method can intelligently change the path of some offline flows to adjust the cost of controlling offline switches based on the ability of active controllers to control. Table 1 summarizes the literature on the CPP problem.

Based on research studies, most research has focused on metrics such as latency, scalability, and reliability. However, less attention has been paid to the issue of cost, which is one of the criteria in discussing the possibility of implementing a

TABLE 1. Summary of Literature Review on CPP.

Ref.	Methods	Objectives			
		Latency	Scalability	Reliability	Cost
[7]	The average latency and worst-case latency	*	-	-	-
[10]	Integer linear programming	*	-	-	*
[11]	Integer linear programming	*	-	*	-
[12]	The spectral clustering algorithm	-	*	*	-
[13]	Determining the failure rate	-	-	*	-
[14]	Graph parsing algorithm	*	-	*	-
[15]	The spanning tree topology	*	-	-	*
[16]	Using an efficient algorithm	*	-	-	-
[17]	Cluster of multiple controllers	*	*	*	-
[18]	Tree subgraph	*	-	*	-
[19]	Hierarchical clustering	*	*	*	-
[20]	Integer linear programming	*	*	-	*
[21]	Using the clique algorithm	*	*	*	-
[22]	Varna optimization method	*	*	*	-
[23]	Bee colony algorithm	*	*	-	*
[24]	RetroFlow algorithm	*	-	*	-
[25]	Programmability guardian	*	*	*	-
[26]	Integer linear programming	*	-	*	-
[27]	Matchmaker algorithm	*	*	*	-
-	Integer linear programming	*	*	*	*

network. In addition, the criterion of network survivability is an important criterion in case of network failure, which has received less attention. However, we want to keep the network stable in the event of a failure until our operation is complete. Also, in most previous studies, the network has been considered static while network dynamics are normal in real environments. Therefore, in this paper, we tried to consider the network dynamics in solving the controller placement problem to bring the problem closer to the real environment. Also, we considered in our work the different modes of failure that will occur in the network. In addition, we considered the components used during network implementation to be heterogeneous.

III. THE MATHEMATICAL EXPRESSION OF THE PROBLEM

According to Fig. 1, the network is initially assumed to be an undirected graph, G, with N nodes. Graph nodes consist of two sets of switches and possible locations to install the controller represented by the symbols S and F, respectively. In other words, $N = S \cup F$ and $S \cap F = \emptyset$.

Another symbol used in the mathematical model is E, which represents the set of connection links. E itself contains the links connecting the controller to the switch and the controller to the controller indicated by the symbols E_F and E_S, respectively. Links are considered directional when the process of controller placement and assigning switches to them is called. In this way, the switches are connected to their controllers and the controllers are connected to each other.

$$E_F = \{ab \in E | a, b \in F\} \tag{1}$$

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

where an arc from a to b is represented as ab. Other required sets are the set, P, of controller's pairs and the set of available controller types, C, as:

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

$$C = \{c_1, c_2, \ldots\} \tag{4}$$

Due to the limited capacity of the controllers, each controller supports a number of switches, and the other switches must be connected to other controllers. The symbols used in the model are shown in Table 2.

Then we define the following decision variables:

$$x_{ab} = \begin{cases} 1 & \text{If there is a connection between } a \text{ and } b, \\ 0 & \text{Otherwise,} \end{cases}$$

$$z_a^c = \begin{cases} 1 & \text{If the controller is located in location } a, \\ 0 & \text{Otherwise,} \end{cases}$$

$$g_{ab}^{pq} = \begin{cases} 1 & \text{If there is a flow between } p \text{ and } q \text{ that passes} \\ & \text{through } a \text{ and } b \\ 0 & \text{Otherwise,} \end{cases}$$

p and q represent the two nodes selected in the graph, which are considered as source and destination nodes in a graph path. p and q are members of the set F.

Assumptions considered in this paper are as follows:

- The controllers are heterogeneous.
- The packets are of the same size.
- The delay between the two nodes was considered equal to the distance between them.

Based on the proposed framework, shown in Fig. 2, each switch can be connected to more than one controller. The reason for this arrangement is that the corresponding controller may fail, and as the backup controller exits, the network can continue to operate, which is an advantage of the proposed architecture. The number of backup controllers can be dynamically determined depending on the environmental

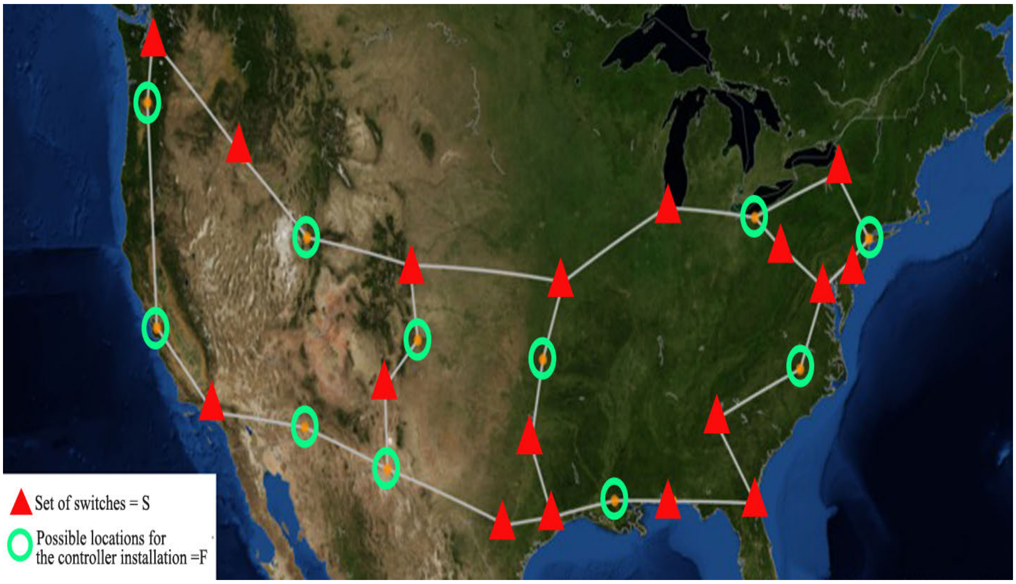


FIGURE 1. Network graph with G = (V, E).

TABLE 2. Mathematical model symbols of the problem.

Description	Symbol	Description	Symbol
Controller deployment nodes	F	Set of switches	S
Pairs of controller installation nodes	P	Set of connection links	E
		Connection cost between node a and b	ω_{ab}
Installing the controller type c in place a	$z^c{}_a$	Cost of installing the controller type c in place a	γ_a^c
The link between node a and b	x_{ab}	The flow from p to q on the edge ab	g_{ab}^{pq}
Packet processing capacity per controller type c	α^c	The amount of controller port	μ^c
Set of controllers	C	The number of packages sent	β

conditions in which the network is implemented. Another advantage of the proposed architecture is to specify different paths at the controller pair's controller levels so that other paths can be exploited if the controller fails or the controller's link at the controller levels fails. The number of communication paths can also be dynamically determined according to the network environment conditions, which benefit the proposed architecture. In Table 2, the symbols of the mathematical model of the problem are described.

Formulation (1): $\min \sum_{ab\in E} x_{ab}\omega_{ab} + \sum_{c\in C} \gamma^c \sum_{a\in F} z_a^c$

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

$$\sum_{\substack{b\,\in\, F \\ ab\,\in\, E_S}} x_{ab} = 1; \quad \forall a \in S \quad (6)$$

$$x_{ab} \le \sum_{c\in C} z_b^c; \quad \forall ab \in E_S, a \in S, b \in F \quad (7)$$

$$\sum_{c\in C} z_a^c \le 1; \quad \forall a \in F \quad (8)$$

$$\sum_{\substack{a\,<\,b \\ b\,\in\, F}} x_{ab} + \sum_{\substack{b\,\in\, S \\ ba\,\in\, E_S}} x_{ba} \le \sum_{c\in C} \mu^c * z_a^c; \quad \forall a \in F \quad (9)$$

$$\sum_{\substack{b\,\in\, S \\ ba\,\in\, E_S}} \beta_b * x_{ba} \le \sum_{c\in C} \alpha^c * z_a^c; \quad \forall a \in F \quad (10)$$

$$x_{ab} \in \{0, 1\} \quad \forall ab \in E \quad (11)$$

$$z_a^c \in \{0, 1\} \quad \forall a \in F,\ c \in C \quad (12)$$

$$g_{ab}^{pq} \in \{0, 1\} \quad \forall ab \in E_F,\quad \forall pq \in P \quad (13)$$

The objective function in this model includes the cost of the link and the cost of the controller deployment, which are calculated with expression $\sum_{ab\in E} x_{ab}\omega_{ab}$ and $\sum_{c\in C} \gamma^c \sum_{a\in F} z_a^c$, respectively [28].

Constraint (5) ensures a connected path between both nodes p and q in which the controller is installed. The reason for this constraint is that in case of communication link failure

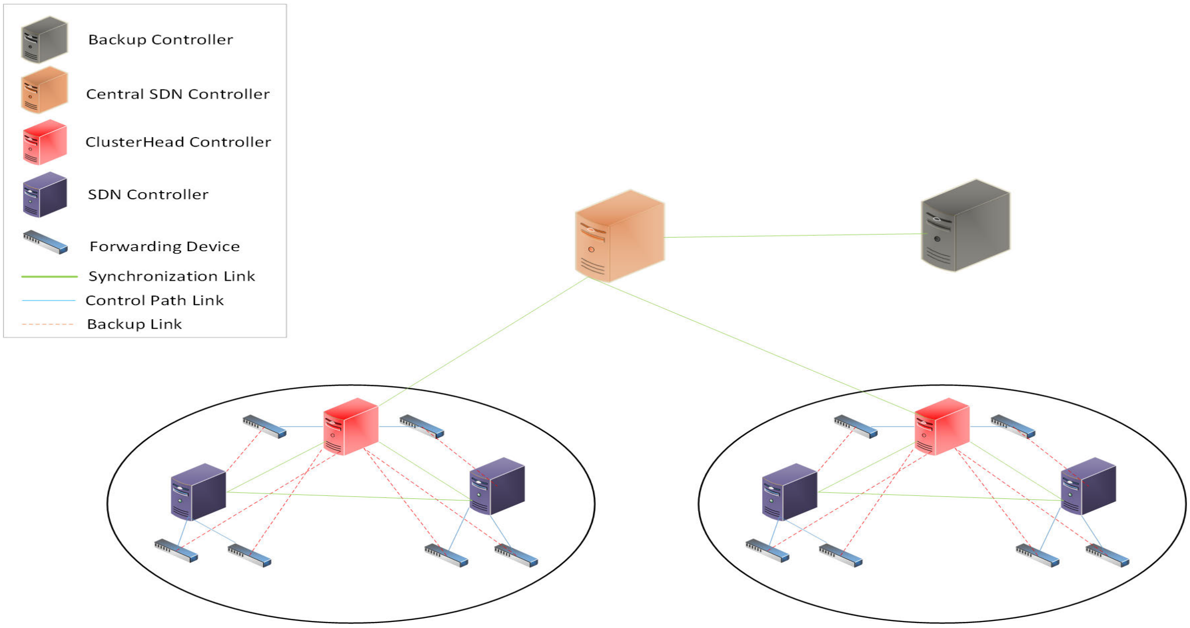


FIGURE 2. The proposed framework.

between the controllers, other paths can be used to send packets. Constraints (6) and (7) indicate the limitation of connecting each switch to only one controller and assigning the switch to locations with a controller, respectively. In constraints (6), it can be considered according to the failure conditions of the controllers that switches are connected to more controllers. Constraint (7) states that the switches cannot be connected to places without a controller. Constraint (8) states that no more than one controller can be installed in one place. Constraints (9) to (10) describe the limitations of port and controller's capacity, respectively. Constraint (9) states the number of links connected to the controller should not exceed the number of ports. Constraint (10) also states that the total number of packets sent by the switches assigned to a controller should not exceed the processing capacity of the packets by that controller. Finally, the constraints (11), (12), and (13) represent the problem decision variables. Thus, according to (5) - (13), the values that these variables take from the problem to reach a feasible solution are zero or one.

Since the controller placement problem is considered a linear programming problem, the expression $\sum_{c \in C} z_p^c * \sum_{c \in C} z_q^c$ in clause (5) causes it to be nonlinear. For this purpose, to linearize that phrase and make it easier, we will first linear the constraint (5) using a method called McCormick [28]. In this method, we will have:

$$t_{pq} = \sum_{c \in C} z_p^c * \sum_{c \in C} z_q^c \tag{14}$$

$$\begin{cases} t_{pq} \le \min(\sum_{c \in C} z_p^c, \sum_{c \in C} z_q^c) \\ t_{pq} \ge \max(0, \sum_{c \in C} z_p^c - (1 - \sum_{c \in C} z_q^c)) \end{cases} \quad \forall pq \in P \tag{15}$$

Or equivalently:

$$\begin{cases} t_{pq} \le \sum_{c \in C} z_p^c \\ t_{pq} \le \sum_{c \in C} z_q^c \\ t_{pq} \ge \sum_{c \in C} z_p^c - (1 - \sum_{c \in C} z_q^c) \end{cases} \quad \forall pq \in P \tag{16}$$

As a result, the mathematical model of the problem is expressed as follows:

Formulation (2) :

$$\min \sum_{ab \in E} x_{ab}\omega_{ab} + \sum_{c \in C} \gamma_a^c \sum_{a \in P} z_a^c \quad (5) - (16)$$

$$\sum_{ab \in E_F} g_{ab}^{pq} - \sum_{ba \in E_F} g_{ba}^{pq} = \begin{cases} t_{pq} & a == p \\ -t_{pq} & a == q \\ 0 & a \ne p, q \end{cases} \quad \forall a \in F, \ \forall pq \in P \tag{17}$$

$$t_{pq} \in \{0, 1\} \quad \forall pq \in P \tag{18}$$

t_{pq} is a binary variable. If the controller is installed in node p and node q, the value of one and otherwise the value of zero is placed in it.

IV. SURVIVABLE CONTROLLER PLACEMENT

Controller failure has an adverse effect on switches and sometimes leads to disable some controller functions [30]. Also, link failure can lead to nodes disconnecting from each other. Therefore, in the controller placement strategy, attention to network survivability is very important. Depending on the controller or link failure, different failure modes can be described as follows:

A. SWITCH TO CONTROLLER LINK FAILURE

In the case of switch to controller link failure, the connection between a switch and its corresponding controller is lost. Therefore, backup controllers should be used to send switch data. ξ_a indicates the number of backup controllers for switch a. Hence, constraint (6) is updated as constraint (6)′. Constraint (6)′ states that the switch a can use another $\xi_a - 1$ controller as a backup controller to send its requests if the main controller fails.

$$\sum_{\substack{b \in F \\ ab \in E_S}} x_{ab} = \xi_a; \quad \forall a \in S \tag{6'}$$

B. DISCONNECT BETWEEN CONTROLLERS

When the link between the controllers is lost, *disjoint paths* can be used to communicate between the controllers. Disjoint paths should have no common edge. Therefore, in the problem model, we use parameter η_{pq} to determine the number of disjoint paths between nodes p and q. Thus, constraint (17) in the mathematical model of the problem is updated as follows:

$$\sum_{ab \in E_F} g_{ab}^{pq} - \sum_{ba \in E_F} g_{ba}^{pq} = \begin{cases} \eta_{pq} * t_{pq} & a == p \\ -\eta_{pq} * t_{pq} & a == q \; \forall a \in F, \; \forall pq \in P \\ 0 & a \neq p, q \end{cases} \tag{17'}$$

Also, the constraint (19) is added to the mathematical model of the problem to ensure that there is no common path in the network graph.

$$g_{ab}^{pq} + g_{ba}^{pq} \leq x_{ab} \quad \forall pq \in P, \; \forall ab, \; ba \in E_F \tag{19}$$

C. LOSS OF CONTROLLER

When loss of controller event occurs, the switches connected to the faulty controller are connected to other backup controllers. When there is a faulty controller in the connection path between two controllers, we will use the concept of *node disjoint paths* to prevent disconnection between the controllers. Since the node disjoint paths are also edge disjoint paths, we use constraint (17)′ to solve this event.

According to the described mathematical model, the proposed method has the advantage of obtaining the required degree of survivability of the network by considering the environment parameters where the network is implemented. The proposed model will also optimally use controller ports to communicate with switches and other controllers. In the proposed model, constraints related to each controller, such as a port, capacity, and the numbers of packets sent are considered so that these constraints are different for each controller due to their heterogeneity.

V. THE SIMULATION RESULTS

This section evaluates our mixed-integer programming formulations against the existing formulations in OMCPP [10] and RCCPP [20]. OMCPP also uses mixed-integer programming to solve the problem. In this method, only the cost of implementing the network is considered. In addition, the control plane topology is considered a full mesh. In RCCPP, a network is considered as an undirected graph in which each node represents a switch or controller. This research aims at reducing the cost of implementation so that reliability is considered. Reliability has been shown to provide backup controllers to connect switches in the event of a controller failure. In order to evaluate the proposed method, we carry out experiments on Oxford, Ntelos, and Lambdanet topologies from the Internet topology Zoo [31] in comparison with OMCPP and RCCPP methods. Information on these topologies is reported in Table 3.

TABLE 3. Topology information.

Topology	|V|	|F|	|S|
Oxford	19	4	15
Lambdanet	41	11	30
Ntelos	48	8	40

These experiments were performed using a system with Intel Core-i5 processor and 8 GB of RAM. For CPLEX [32], the time limit is set to 7200 seconds. CPLEX solution parameters are given in Table 4. The results of these experiments were plotted using MATLAB software.

TABLE 4. Symbols used in the problem.

	Controller		
	#1	#2	#3
Cost per controller (γ^c)	$1200	$2500	$6500
Number of ports per controller (μ^c)	8	16	32
Processing Capacity by Controller (α^c)	2500 Byte	4000 Byte	8000 Byte
Link cost per meter		$8.25	
Packet size (β)		150 Byte	

Criteria for evaluation include network implementation cost, average latency, and load balancing rate. Therefore, in the following, we evaluate the proposed method in comparison to OMCPP and RCCPP methods with regards to the set criteria.

A. NETWORK IMPLEMENTATION COST

In this section, the cost of implementing the network for different values of the degree of network survivability, indicated by the symbol R, is reported. The degree of survivability is an important parameter that is used to validate the survivability of the network. Therefore, we use Formulation (3) to calculate the maximum degree of survivability for both locations where the controller is installed.

Formulation (3): $R = \max \pi$

$$\sum_{ij \in E_\iota} f_{ij} - \sum_{ji \in E_\iota} f_{ji} = \begin{cases} \pi & i == s \\ -\pi & i == t \quad \forall i \in N_\iota \\ 0 & i \neq s, t \end{cases} \tag{20}$$

$$f_{ij} + f_{ji} \leq 1 \quad \forall ij \in E_\iota \tag{21}$$

TABLE 5. Comparison of the proposed method with omcpp and rccpp methods.

Topology	Proposed	OMCPP	RCCPP	Imp Proposed vs. OMCPP %	Imp Proposed vs. RCCPP %
Oxford	93523	105056	106360	12.33	13.73
Lambdanet	380010	461696	491700	21.50	29.39
Ntelos	339573	411405	421320	21.15	24.07

In formulation (3), for each topology ι, The number of deployed controllers is indicated by the symbol n_ι. K_ι represents a complete graph with n_ι nodes. Thus, $K_\iota = (V_\iota, E_\iota)$. s and t represent the beginning and end nodes of the set N_ι. Since the complete graph induces a symmetric topology for each pair of controllers installed, the maximum value obtained from the execution of the above formulation is equal for each pair of controllers [33].

Indeed, we compute survivability between every pair of installed controllers with the aim of a mixed integer programming formulation. Then, the obtained survivability will be given as input to our proposed formulation. This input will provide a proper basis for comparison while different formulations are to design low-cost solutions of similar intended survivability requirements. The results of the experiments are shown in Table 5.

In Table 5, the first column shows the name of the experimented topology. The second to fourth columns are costs obtained by the proposed method and OMCPP and RCCPP methods, respectively. Finally, the fifth and sixth columns report a comparison of the proposed method with OMCPP and RCCPP methods in term of the percentage improvement in cost reduction calculated by (22) of this percentage.

$$\frac{Cost_{OMCPP\ or\ RCCPP} - Cost_{pro}}{Cost_{pro}} * 100 \tag{22}$$

Based on the results obtained in Table 5, it can be concluded that the proposed method performs better in experiments, which are on different topologies. This advantage is most evident when the size of the network increases. The reason is the proper design of the control plane. The reason is that, the proposed method designs the control plane architecture in such a way that a lower cost is imposed in terms of network connection link consumption. In addition, the survivability of the network is increased. However, the OMCPP and RCCPP methods use a full mesh topology to design the control plane. This topology provides a great cost to connect the controller to each other. According to studies and analysis of experiments results, at least 65% of the cost of network implementation is related to the cost of network equipment connection link.

The reason for the decrease in the percentage of improvement in some of the experiments topologies is related to the topology structure and how the switches and controllers are located. Therefore, in experiments, the switches are randomly placed in the network, and, the location of the controllers is very important.

Figures 3(a) to 3(c) show the results related to the cost of implementing a survivable network. In these figures, the R symbol indicates the degree of survivability. According to the results shown, the proposed method for network implementation is cost-effective considering the ability to survive in the event of failure. In addition, the proposed method offers a high degree of survivability compared to OMCPP and RCCPP methods, which are also less costly than the methods compared. However, OMCPP and RCCPP methods do not provide any flexibility in choosing the degree of survivability. In contrast, the proposed method can dynamically receive the degree of survivability from the input. Due to this advantage of the proposed method, the degree of survivability required for the network can be determined by considering the environmental conditions of the network. This has a direct impact on the cost of implementing the network. Increasing the degree of survivability causes to raise the cost of the network. Another advantage of the proposed method is the optimal use of controller ports. In contrast, OMCPP and RCCPP methods occupy a lot of controller ports due to the use of full mesh topology. Therefore, as mentioned earlier, this topology increases the cost of the network due to the direct connection of the controllers to each other. Hence, there will be no balance between the costs considered as the objective function.

Furthermore, OMCPP and RCCPP methods do not guarantee the survivability of a small network. For example, when implementing SDN network, we use only two controllers. At the control plane to connect the controllers to each other according to OMCPP and RCCPP methods, only one link is placed between the two controllers to communicate. However, when this link is disconnected, the controllers lose communication and the network becomes problematic. Therefore, it is very important to pay attention to the architecture of the control plane.

As shown in Fig. 3(a), there is no answer to any of the methods for the value of $R = 4$, because in Oxford topology, a maximum of four controllers can be installed, while $R = 4$ requires five controllers.

In Fig. 3(b) and Fig. 3(c), the cost of implementing the network by the proposed method is less than OMCPP and RCCPP methods with different values of R. For example, in Fig. 3(b), the cost obtained by the proposed method for $R = 2, 3$ is less than the costs obtained by OMCPP and RCCPP methods with the values of $R = 1$ and $R = 2$, respectively. In other words, it can be said that the proposed method designs a network in such a way that it has a lower degree of reliability and costs less than OMCPP and RCCPP methods.

Finally, it can be concluded that the proposed method is more cost-effective than the OMCPP and RCCPP, even when the degree of survivability increases compared to the OMCPP and RCCPP.

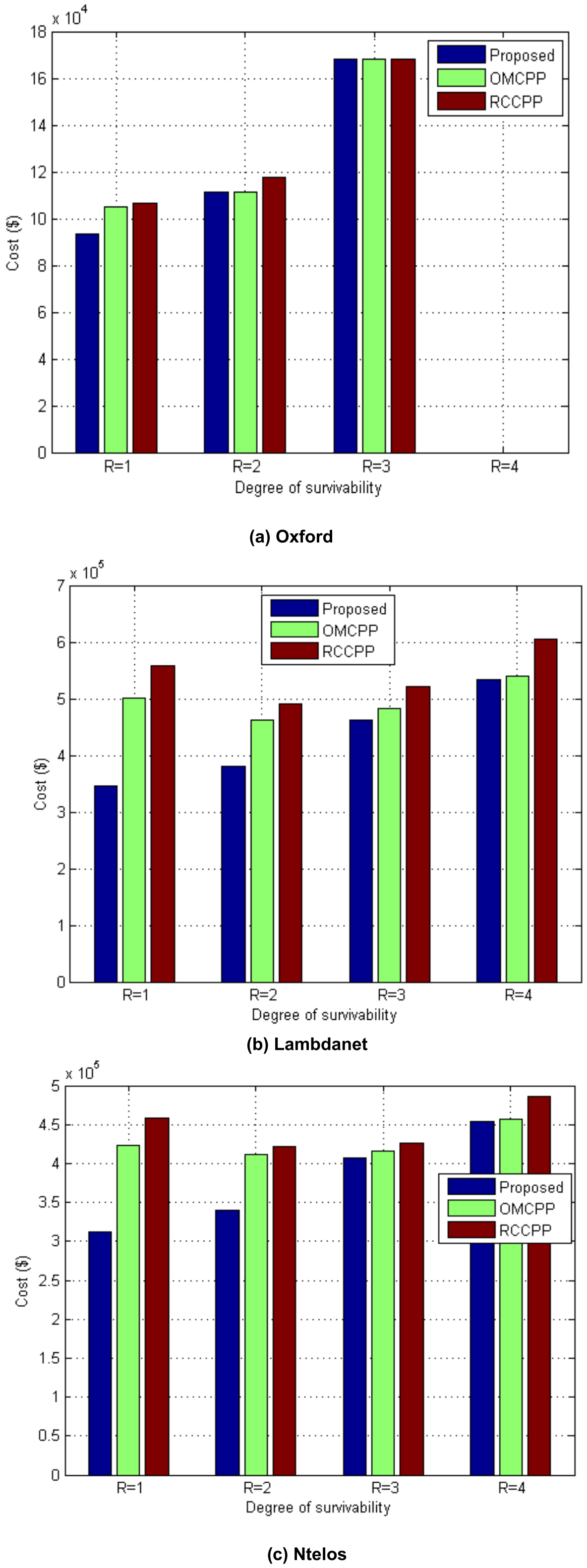


FIGURE 3. **The network implementation cost for different values of degree of survivability (R).**

B. THE AVERAGE DELAY

Calculating the average delays such as propagation, processing and transmission is considered and the delays are displayed with symbols d_{prop}, d_{proc}, and d_{tran}, respectively. In these calculations, the values of d_{proc} and d_{tran} are 0.05ms and 0.08ms, respectively. Also, d_{prop} for every 1 kilometer equals to 0.3 milliseconds. In this experiment, the average delay is calculated based on the number of controllers for the three topologies Oxford, Lambdanet, and Ntelos. The results of this experiment are shown in Figs. 4(a), 4(b) and 4(c).

According to the results shown in Fig. 4, the average delay decreases with the increasing number of controllers, because as the number of controllers increases, it becomes more likely to be located near switches. Therefore, switches can be connected to their nearest controller, which reduces the time, which takes to send data from the switch to the controller. As a result, propagation delays are reduced. Furthermore, each controller has an appropriate number of switches connected, and also has a positive effect on the load balance of the controller. As shown in Figs. 4(b) and 4(c), the propagation delay on the Ntelos topology is less than on the Lambdanet topology, though the number of nodes on the Lambdanet topology is less than on the Ntelos topology. This is due to the greater number of locations for the controller, which causes the controllers to be more scattered than each other, since the controllers need to communicate with each other to sync. The distance of most controllers has an effect on the propagation delay.

C. CONTROLLER LOAD BALANCING RATE

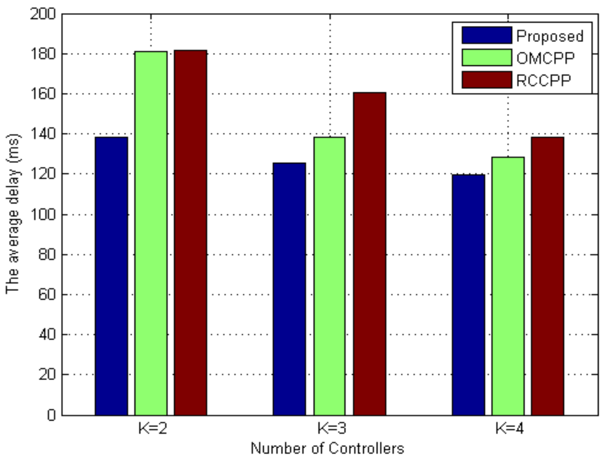
In this section, the load-balancing rate for the controllers is calculated using (23). If the load-balancing rate increases, it indicates the load is balance in each controller. One of the suitable models for analyzing network traffic is the use of the Poisson model. This distribution is suitable when independent sources of traffic are high.

$$R_{CLB} = \sum_{i=1}^{d} [\frac{\theta_i}{n}.\frac{\sum_{m=1}^{\theta_i} \beta_m}{\sum_{j=1}^{n} \beta_j}] \tag{23}$$

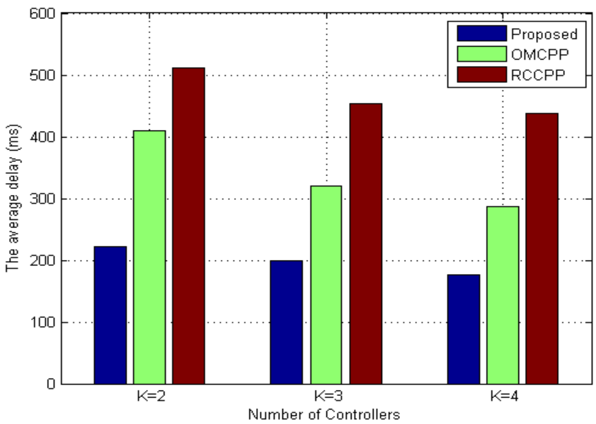
The number of switches in each domain i is indicated by the symbol θ_i. d, and n also represents the number of domains and the total number of network nodes, respectively. The value of R_{CLB} is calculated based on the number of controllers. Figures 5(a), 5(b) and 5(c) show the results of this experiment.

Based on the results of Fig. 5, it can be concluded that the optimal number of controllers affects the amount of load on the controllers. The reason is that the small number of controllers causes the controllers to overflow due to the large volume of load from the switches connected to them. Also, the large number of controllers has a significant effect on the data sent to each controller for synchronization. As a result, considering the capacity of the controllers, we must focus on maintaining the load balance of each controller.

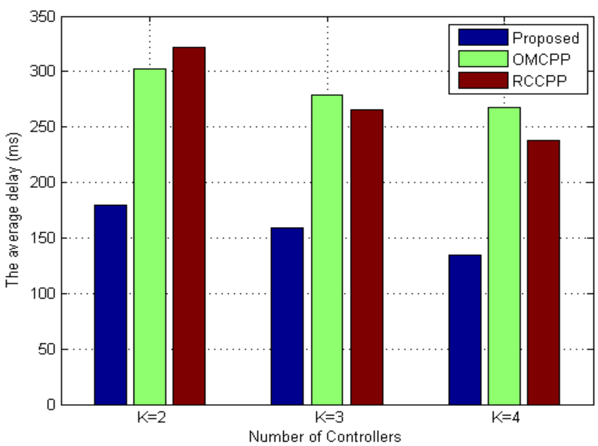
As shown in Fig. 5(a), when the network is implemented based on two controllers, the controllers are overloaded. As a result, the value of R_{CLB} is high. In contrast, increasing the



(a) Oxford



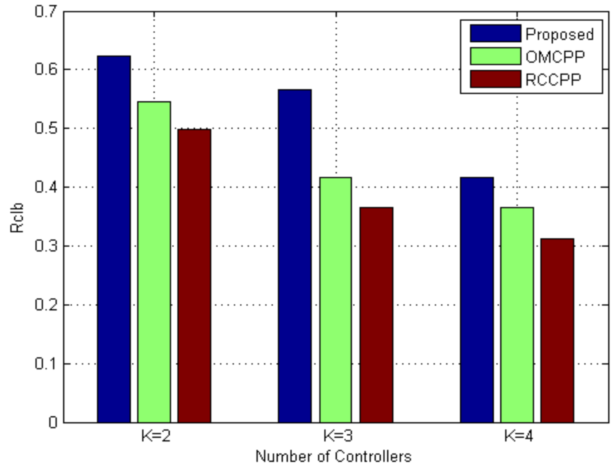
(b) Lambdanet



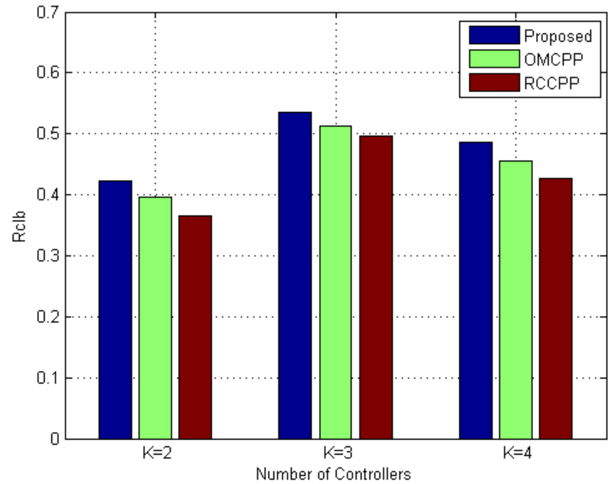
(c) Ntelos

FIGURE 4. The average delay.

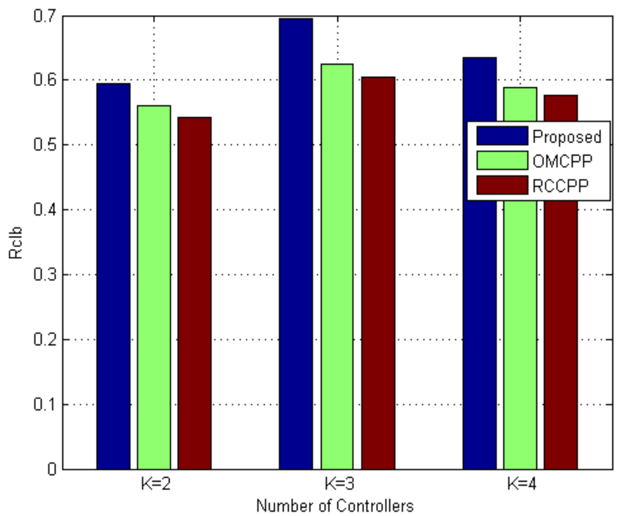
number of controllers reduces the amount of load divided into controllers. In Figs. 5(a) and 5(b), since the optimal number of controllers is 3, the value of R is high. The compared methods have a lower R_{CLB} value than the proposed method due to the lack of focus on the control plane architecture. The average R_{CLB} in OMCPP and RCCPP methods is 0.5 and 0.46, respectively. However, the R_{CLB} for the proposed method



(a) Oxford



(b) Lambdanet



(c) Ntelos

FIGURE 5. The load balancing rate.

is 0.55. As a result, it can be said that the optimal implementation of control plane, deployment of controllers in suitable places and connection of the nearest switches to them have a favorable effect on increasing the load balance rate, which is considered in the proposed method.

VI. CONCLUSION AND FUTURE WORK

Two metrics of cost and survivability as factors affecting the efficiency of the control plane and the overall efficiency of the network were examined in this paper. Researchers in the literatures less considered the two metrics. Since minimizing the cost is one of the effective factors in network implementation, it is very important to consider the cost in solving the controller placement problem, as improper location of controllers may increase the cost of network implementation. As for the survivability criterion, since networks are at risk in real environments, solutions must be considered for the network's stability in these regards. Therefore, paying attention to the survivability criterion also has a significant effect on solving the CPP.

A mathematical model of the problem was presented, taking into account the stated criteria, in which the cost was considered as an objective function and survivability as a constraint. The results of the experiment indicate the superiority of the proposed method both in terms of implementation cost and network survivability. Therefore, the proposed method has an improvement of 18.33% and 22.40% in terms of implementation cost compared to OMCPP and RCCPP.

In future work, we will try to use heuristic algorithms to achieve the optimal solution in the shortest time. Also, determining the appropriate parameters to select the appropriate location for the controllers can use multi-criteria decision-making methods.

REFERENCES

[1] O. Blial, M. Ben Mamoun, and R. Benaini, "An overview on SDN architectures with multiple controllers," *J. Comput. Netw. Commun.*, vol. 2016, Apr. 2016, Art. no. 9396525.

[2] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.

[3] W. Xia, Y. Wen, C. H. Foh, D. Niyato, and H. Xie, "A survey on software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 17, no. 1, pp. 27–51, 1st Quart., 2014.

[4] H. Selvi, S. Güner, G. Gür, and F. Alagöz, "The controller placement problem in software defined mobile networks (SDMN)," in *Proc. LTE Netw. Archit., Wiley Telecom, Softw. Defined Mobile Netw. (SDMN)*, 2015, pp. 47–129.

[5] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 4, pp. 1955–1980, 4th Quart., 2014.

[6] S. Sezer, S. Scott-Hayward, P. K. Chouhan, B. Fraser, D. Lake, J. Finnegan, N. Viljoen, M. Miller, and N. Rao, "Are we ready for SDN? Implementation challenges for software-defined networks," *IEEE Commun. Mag.*, vol. 51, no. 7, pp. 36–43, Jul. 2013.

[7] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 42, no. 4, pp. 473–478, Sep. 2012.

[8] A. K. Singh and S. Srivastava, "A survey and classification of controller placement problem in SDN," *Int. J. Netw. Manage.*, vol. 28, no. 3, p. e2018, May 2018.

[9] A. P. Snow, U. Varshney, and A. D. Malloy, "Reliability and survivability of wireless and mobile networks," *Computer*, vol. 33, no. 7, pp. 49–55, Jul. 2000.

[10] A. Sallahi and M. St-Hilaire, "Optimal model for the controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 19, no. 1, pp. 30–33, Jan. 2015.

[11] S. Mohanty, P. Priyadarshini, B. Sahoo, and S. Sethi, "A reliable capacitated controller placement in software defined networks," in *Proc. 3rd Int. Conf. Comput. Methodologies Commun. (ICCMC)*, Mar. 2019, pp. 822–827.

[12] P. Xiao, W. Qu, H. Qi, Z. Li, and Y. Xu, "The SDN controller placement problem for WAN," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Oct. 2014, pp. 220–224.

[13] Y. Hu, W. D. Wang, X. Y. Gong, X. R. Que, and S. D. Cheng, "On the placement of controllers in software-defined networks," *J. China Univ. Posts Telecommun.*, vol. 19, pp. 92–171, Oct. 2012.

[14] Y. Zhang, N. Beheshti, and M. Tatipamula, "On resilience of split-architecture networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–6.

[15] M. Obadia, M. Bouet, J.-L. Rougier, and L. Iannone, "A greedy approach for minimizing SDN control overhead," in *Proc. 1st IEEE Conf. Netw. Softw. (NetSoft)*, Apr. 2015, pp. 1–5.

[16] G. Yao, J. Bi, Y. Li, and L. Guo, "On the capacitated controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1339–1342, Aug. 2014.

[17] T. Zhang, A. Bianco, S. De Domenico, and P. Giaccone, "The role of inter-controller traffic for placement of distributed SDN controllers," 2016, *arXiv:1605.09268*. [Online]. Available: http://arxiv.org/abs/1605.09268

[18] D. Santos, T. Gomes, and D. Tipper, "SDN controller placement with availability upgrade under delay and geodiversity constraints," *IEEE Trans. Netw. Service Manag.*, vol. 18, no. 1, pp. 301–314, Mar. 2021.

[19] J. Ali and B.-H. Roh, "Quality of service improvement with optimal software-defined networking controller and control plane clustering," *Comput., Mater. Continua*, vol. 67, no. 1, pp. 849–875, 2021.

[20] A. Sallahi and M. St-Hilaire, "Expansion model for the controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 274–277, Feb. 2017.

[21] 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, Mar. 2018.

[22] A. K. Singh, S. Maurya, N. Kumar, and S. Srivastava, "Heuristic approaches for the reliable SDN controller placement problem," *Trans. Emerg. Telecommun. Technol.*, vol. 31, no. 2, p. e3761, Feb. 2020.

[23] N. Lin, Q. Zhao, L. Zhao, A. Hawbani, L. Liu, and G. Min, "A novel cost-effective controller placement scheme for software-defined vehicular networks," *IEEE Internet Things J.*, vol. 8, no. 18, pp. 14080–14093, Sep. 2021.

[24] Z. Guo, W. Feng, S. Liu, W. Jiang, and Y. Xu, "RetroFlow: Maintaining control resiliency and flow programmability for software-defined WANs," in *Proc. IEEE/ACM Int. Symp. Qual. Service (IWQoS)*, Jun. 2019, pp. 1–10.

[25] Z. Guo, S. Dou, and W. Jiang, "Improving the path programmability for software-defined WANs under multiple controller failures," in *Proc. IEEE/ACM 28th Int. Symp. Qual. Service (IWQoS)*, Jun. 2020, pp. 1–10.

[26] F. He and E. Oki, "Master and slave controller assignment with optimal priority policy against multiple failures," *IEEE Trans. Netw. Service Manage.*, early access, Mar. 8, 2021, doi: 10.1109/TNSM.2021.3064646.

[27] S. Dou, G. Miao, Z. Guo, C. Yao, W. Wu, and Y. Xia, "Matchmaker: Maintaining network programmability for software-defined WANs under multiple controller failures," *Comput. Netw.*, vol. 192, Jun. 2021, Art. no. 108045.

[28] P. M. Castro, "Tightening piecewise McCormick relaxations for bilinear problems," *Comput. Chem. Eng.*, vol. 72, pp. 300–311, Jan. 2015.

[29] S. M. Hashemi, M. Rezapour, and A. Moradi, "An effective hybrid PSO-based algorithm for planning UMTS terrestrial access networks," *Eng. Optim.*, vol. 42, no. 3, pp. 241–251, Mar. 2010.

[30] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, "On reliability-optimized controller placement for software-defined networks," *China Commun.*, vol. 11, no. 2, pp. 38–54, Feb. 2014.

[31] S. Knight, H. X. Nguyen, N. Falkner, R. Bowden, and M. Roughan, "The internet topology zoo," *IEEE J. Sel. Areas Commun.*, vol. 29, no. 9, pp. 1765–1775, Oct. 2011.

[32] A. Moradi, A. Lodi, and S. Mehdi Hashemi, "Virtual private network design over the first Chvátal closure," *RAIRO-Oper. Res.*, vol. 49, no. 3, pp. 569–588, Jul. 2015.

[33] A. Moradi, A. Lodi, and S. M. Hashemi, "On the difficulty of virtual private network instances," *Networks*, vol. 63, no. 4, pp. 327–333, Jul. 2014.

ALI ABDI SEYEDKOLAEI received the B.Sc. degree in computer engineering from the University of Allameh Mohaddes Noori, Noor, Iran, in 2007, and the M.Sc. degree in computer engineering from Shahid Beheshti University, Tehran, Iran, in 2015, where he is currently pursuing the Ph.D. degree. His research interests include software defined networking, wireless networks, sensor networks, and optimization.

SEYED AMIN HOSSEINI SENO received the B.Sc. and M.Sc. degrees in computer engineering from Ferdowsi University of Mashhad, Mashhad, Iran, in 1990 and 1998, respectively, and the Ph.D. degree from Universiti Sains Malaysia, Penang, Malaysia, in 2010. He is currently an Associate Professor with the Department of Computer Engineering, Ferdowsi University of Mashhad. His research interests include wireless and sensor networks, network protocols, QoS, and network security.

AHMAD MORADI received the B.Sc. degree in applied mathematics from the University of Mazandaran, Babolsar, Iran, in 2004, and the M.Sc. and Ph.D. degrees in computer science from Amirkabir University of Technology, Tehran, Iran, in 2006 and 2015, respectively. He is currently an Assistant Professor with the Department of Mathematical Sciences, University of Mazandaran. His research interests include design and analysis of algorithms, combinatorial optimization, and integer programming.

RAHMAT BUDIARTO received the B.Sc. degree in mathematics from Bandung Institute of Technology, Indonesia, in 1986, and the M.Eng. and D.Eng. degrees in computer science from Nagoya Institute of Technology, Japan, in 1995 and 1998, respectively. He is currently a Full Professor with the Faculty of Computer Science, Universitas Mercu Buana. His research interests include intelligent systems, brain modeling, IPv6, network security, wireless sensor networks, and MANETs.