Uluslararası Mühendislik Araştırma ve Geliştirme Dergisi
International Journal of Engineering Research and Development

UMAGD, (2022) 14(1), 237-245.

10.29137/umagd.1029913

Cilt/Volume:14 Sayı/Issue:1 Ocak/January 2022

Araştırma Makalesi / Research Article

A Linear Programming Model for Latency Minimization and Fault Tolerance in Software Defined Network Controller Placement

Yasin İnağ[1], Mehmet Demirci [1]

[1]*Department of Computer Engineering, Faculty of Engineering , Gazi University, 06570, Ankara, TURKEY*

***Başvuru/Received**: 29/11/2021* ***Kabul / Accepted**: 22/12/2021* ***Çevrimiçi Basım / Published Online:** 31/01/2022*

***Son Versiyon/Final Version**: 31/01/2022*

Abstract

This study proposes a mathematical model for the controller placement problem(CPP) in Software Defined Networks (SDN). For SDN networks, multiple controllers are often suitable, and the location of these controllers affects the network performance. In case the controller is at fault, the resilience and reliability of the network should be ensured. Therefore, the recommended model optimizes the number and location of the controller(s) in the network while controller(s) is at fault. The model aims to minimize the latency by considering the reliability and resilience of the network. The simulation results show that the average latency is slightly increased whilst providing network resilience. The proposed model can be used to integrate SDN into existing networks or plan a new SDN network.

Key Words

“Controller placement, SDN, capacitated, reliability”

***Sorumlu Yazar:** yasininag@gazi.edu.tr*

1. Introduction

Traditional IP networks are managed by switches which the control and networking protocols are running them. With this architecture, network management becomes complicated as it is implemented individually, and different commands are required for each device (Kreutz et al., 2014). Furthermore, traditional architectures are not used switches that allow separation of the control plane and the data plane. For this reason, dynamic and flexible solutions are not provided for requirements such as optimization, device errors, security, etc. in networks. SDN is a developed technology to provide resilience and programmability to networks and to facilitate dynamic network management (Benzekki, El Fergougui, & Elbelrhiti Elalaoui, 2016; Kreutz et al., 2014; Nunes, Mendonca, Nguyen, Obraczka, & Turletti, 2014; Xia, Wen, Foh, Niyato, & Xie, 2014).

In recent years, the use of SDN in many new-generation network technology applications such as network function virtualization (NFV) (Li & Chen, 2015), edge computing (Baktir, Ozgovde, & Ersoy, 2017), Internet of Things (Bera, Misra, & Vasilakos, 2017), mobile/wireless networks (Haque & Abu-Ghazaleh, 2016), 5G(Gupta & Jha, 2015) has gained importance. In addition, SDN is increasingly used in data center networks and Wide Area Networks (WAN), and it is used by companies such as Google, Facebook (Michel & Keller, 2017).

The main idea that makes SDN important is that it separates the data and control planes and presents a central controller. The SDN control plane provides the ability to monitor and configure the network. The implementation of a new routing protocol is easily accomplished with the controller.

Early research observed that a single controller suffices for real networks(Heller, Sherwood, & McKeown, 2012). However, in recent studies, the necessity of multiple controllers has emerged due to reasons such as scalability, security, and flexibility(T. Hu, Guo, Yi, Baker, & Lan, 2018). These studies showed the importance of SDN, but many challenges still remain. The Controller Placement Problem (CPP) is how to place controller(s) in the network. In a network with a certain number of nodes, the number and location of controllers are an open issue. When solving this problem, constraints such as latency, flexibility, reliability, energy saving, and load balancing are generally used. Delay is one of the essential criteria in determining the controller position. Delay becomes a critical metric in the developing and rapidly increasing data traffic. For this reason, in this study, we aim to provide the delay at the optimum level.

In the literature, some studies optimize the delay between the controller and the switches. Heller et al.(Heller et al., 2012) addressed the controller placement problem. They examined the effects of placement, taking into account the average delay and worst-case delay. Yao et al. (Yao, Bi, Li, & Guo, 2014) evaluated the latency and load balancing between controllers and switches. Salahi et al. (Sallahi & St-Hilaire, 2014) proposed a solution by determining the average latency time as a constraint.

In multi-controller networks, managing the network in case controller(s) is not working is another important problem. Our study aims to ensure the resilience and reliability of the network with optimum delay when controller errors occur after determining the controller location. The proposed model is called the Controller Position Problem with Capacity and Fault tolerance(CF-CPP).

In CF-CPP, the number of controllers and their positions are determined using the k-median clustering algorithm at the beginning. When determining the number of controllers, controller fault tolerance is considered a constraint. In case of a possible controller fault during network transmission, which switches should be connected to which controllers are solved by linear programming that we recommended. Minimum delay is aimed at restricting the number of controllers and the number of free ports in the controller. While ensuring the network's reliability and minimizing the average delay, the effects of the number of controllers and fault tolerance are examined.

The main contributions of the paper are the following:

- This study proposes a model for determining the controller location and possible controller failure in SDN-supported networks.
- CF-CPP is mathematically formulated and simulated in a GNS3 simulation environment. Although the primary purpose of this study is to minimize the average delay, it is to provide flexibility and dynamism to the network against the controller errors that may occur in the network.
- An algorithm with k-median clustering and linear programming is proposed to ensure the flexibility and reliability of the network.
- In case of possible controller fault in a network where controller locations are determined, the network's reliability is ensured by assigning the switches connected to this controller to other controllers in the system.

The rest of the paper is organized as follows: Section 2 examines the related works on CPP in the literature comprehensively. The details of the proposed models are explained in Section 3. Extensive simulation results are presented and discussed in Section 4. Finally, Section 5 summarizes the paper and presents the future directions to guide the interested readers.

2. Related Works

CPP is one of the important problems in SDN-supported networks and is extensively studied in the literature. While controller placing, different network improvement objectives such as minimum latency, cost, reliability, and durability, etc. are determined. As providing these purposes; the number of controllers, load balancing, controller update cost, etc. different constraints are used (Das, Sridharan, & Gurusamy, 2019; Isong, Molose, Abu-Mahfouz, & Dladlu, 2020; Killi & Rao, 2019; Lu, Zhang, Hu, Yi, & Lan, 2019; Singh & Srivastava, 2018; Wang, Zhao, Huang, & Wang, 2017).

The controller placement problem was first described in the literature by Heller et al.(Heller et al., 2012). They used a k-means clustering algorithm to determine the number and location of controllers. They offered a solution to CPP by optimizing the latency between the switches and the controller. While it was observed that increasing the number of controllers reduces latency, controller fault tolerance and capacity are not examined. In a similar study, Kobo et al.(Kobo, Abu-Mahfouz, & Hancke, 2019) proposed a solution that provides minimum latency and resilience using the k-median algorithm. The simulation results show that better performance was obtained by using local and central controllers and that the controller change has little effect on the average delay.

Guang Yao et al.(Yao et al., 2014) committed a model that considers delay and controller capability. This model was obtained by adapting the K-median (k-center) algorithm to the problem. The simulation results showed that their models reduced the number and load of controllers. Kuang et al. (Kuang, Qiu, Li, & Liu, 2018) produced a solution using the k-means algorithm to determine the controller placement. They divided large-scale networks into subnets and optimized latency by assigning a controller on the subnets. Wang et al. (Wang, Zhao, Huang, Duan, & Li, 2016) optimized the K-means algorithm for CPP and showed that the proposed model results yielded 2347 times faster with the simulation. In the literature, the k-median algorithm is widely used for CPP (Alenazi & Cetinkaya, 2020; Y.-N. Hu, Wang, Gong, Que, & Cheng, 2012; Jimenez, Cervello-Pastor, & Garcia, 2014; Xiao, Qu, Qi, Li, & Xu, 2014).

While SDN produces solutions to the problems encountered in traditional networks, it also exposes new problems. Reliability in determining the number and placement of controllers is an significant problem. One of the problems encountered in networks is the failure of one or more controllers(Chen, Chen, Xu, Yin, & Zhang, 2015; Fonseca & Mota, 2017; Singh & Srivastava, 2018). Yannan et al. (Y.-N. Hu et al., 2012) introduced reliability and resilience in determining the controller placement. According to the authors, the placement and reliability of the controller improved network performance.

In SDN enabled networks, the communication is broken with the failure of the controller. In order to avoid such a problem, the network architecture must be fault-tolerant. Failure of the controller means a brainless network. The authors proposed random insertion, brute force, and greedy algorithms for reliable controller placement on SDN. The greedy algorithm provided better results compared to random insertion and brute force algorithms. Yanan et al. (Y. Hu, Wang, Gong, Que, & Cheng, 2014) tried to solve the problem by specifying the maximum number of controllers when creating the network and connecting more than one controller to each switch. However, latency was not taken into account in this study.

3. Proposed Model

Controller placement, number of controllers, and controller crashes/non-working situations are important problems in SDN networks. In this section of the study, a model that produces solutions to the problems mentioned is introduced. The model we propose determines the controller placement and number using the k-median algorithm. Furthermore, to ensure the network's resilience in case of possible failure of one/many controllers after the network communication, a linear programming model is proposed, which ensures that the switches connected to the broken controller are assigned to other existing controllers of the system.

K-median; It is a clustering algorithm that minimizes the sum of the distances between the k-center and all other points in the cluster. The k-median is considered a revised model of the well-studied k-means approach and minimizes the distance between the center and the cluster points, and the center must be a node of the network (Kobo et al., 2019).

K-median:

Given a network graph $G(V,E)$, let the edge weights represent propagation latency, $d(v,s)$ represent the shortest path from node $v \in V$ to $s \in V$, $n = |V|$, is the number of nodes. S will be the number of controllers to choose from, S', is the number of controllers to be placed such $|S'| = K$, The average latency for placement of controllers $L_{avg}(S')$ is modeled as

$$L_{avg}(S') = \frac{1}{n}\sum_{v \in V} \min_{s \in S} d(v,s) \tag{1}$$

When determining the number of controllers with the K-median, the controller fault tolerance was taken into account. Fault tolerance is the percentage of failing of existing controllers in the system. With this ratio, it is calculated at least how many controllers there should be in the network. Experimental studies investigate the effect of this ratio on the average latency.

After the number of controllers determined by the fault tolerance, the problem of a possible controller not working was solved by linear programming. It is defined that the switches connected to the broken/non-working controller will be connected to which controller for the network's resilience. The present of the linear programming formulation for the CF-CPP is as follows:

$j,$ represents the controller number, $p,$ means the port number of the controller, i the number of the switch, ft represents the fault tolerance rate. d_{ij} is latency of switch i to j, M_j is capacity of j controller. Let x_{ij} =1 indicates there is a path between switch i to controller j, 0 otherwise. Let b_j=1, indicates controller fault, 0 otherwise.

$$Min \sum_{i \in \mathrm{i}} \sum_{J \in j} d_{ij} * x_{ij} \qquad (2)$$

$$\mathrm{m} \geq \frac{\mathrm{i}}{\mathrm{p}} \qquad \mathrm{m} \in \mathbb{Z} \qquad (3)$$

$$j \geq m(1 + ft) \qquad j \in \mathbb{Z} \qquad (4)$$

$$\sum_{j \in J} x_{ij} = 1 \qquad \forall i \in I \qquad (5)$$

$$\sum_{i \in I} x_{ij} \leq M_j(1 - b_j) \qquad \forall j \in J \qquad (6)$$

$$\sum_{j \in J} b_j = 1 \qquad (7)$$

$$x_{ij}, b_j \in \{0, 1\} \qquad (8)$$

Objective function Eq. (2) aims to minimize the average delay between switch and controller. Eq. (5) is the constraint that allows each switch to be assigned to only one controller. Eq. (6) is the constraint that prevents the remaining capacities of the selected controller from being exceeded. Eq. (7) is the constraint that ensures that only one of the controllers will fail. Eq. (8) is the integrality constraints.

The proposed model offers a new proposal to CPP by calculating the probability that the controller is a fault in SDN-supported networks. While this model provides communication between the controller and switches with minimum delay, it takes into account the controller capacity (port) and controller crash constraints. At least how many controllers there must be in the network and the effects of increasing the number of controllers on the resilience of the system and the average latency are examined with simulations.

Advantages of the proposed model are the following,

- Determining the required number of controllers that consider the possibility of a controller failure ensures the system's resilience.
- Increases communication performance as it provides minimum latency between switches and controllers.
- Provide the system's resilience by transferring the switch connected to the broken/not working controller to the existing controllers with minimum delay.

The disadvantage of the proposed model is the increase in the operation cost caused by the implemented controller fault tolerance mechanism. In the proposed model, the number of controllers is determined based on a defined controller fault tolerance. Therefore, the number of controllers tends to be greater than a network with no fault tolerance. Thus, the total operation cost increases.

4. Experimental Analysis

In this section, the effect of the CF-CPPP model, which we propose to ensure the resilience of the network and provide a solution to CPP, on the average delay and number of controllers in case of controller failure is examined. In the proposed model, the number of controllers is determined by taking into account the fault tolerance, and the average delay time is calculated. The average delay is calculated for the minimum number of controllers, and it is compared with the average delay time in case of controller failure.

4.1. Evaluation Metrics

This subsection describes the metrics utilized to assess the model discussed in this study. CPP and the proposed models are evaluated based on several metrics which are prevalent in the literature. These metrics are:

- Delay, which is one of the important metrics considered in CPP. It depends on the distance between the nodes and the response time between the controller and the switches.
- The number of controllers is determined based on the number of switches and the controller's location to improve network performance and provide QoS.
- Fault tolerance, which is the probability that there is interrupted communication between switches and controllers. It is usually caused by controller errors.

The results obtained from the simulation experiments regarding these metrics are fully presented and discussed in the next section.

4.2. Results and Discussion

CF-CPP has been tested on the GNS3 platform upon a graph with 45 nodes and 67 edges shown in Fig. 1, Nodes represent switches. It is assumed that the controllers have 6 ports. For this reason, the maximum number of switches that can be connected to the controllers is 6.

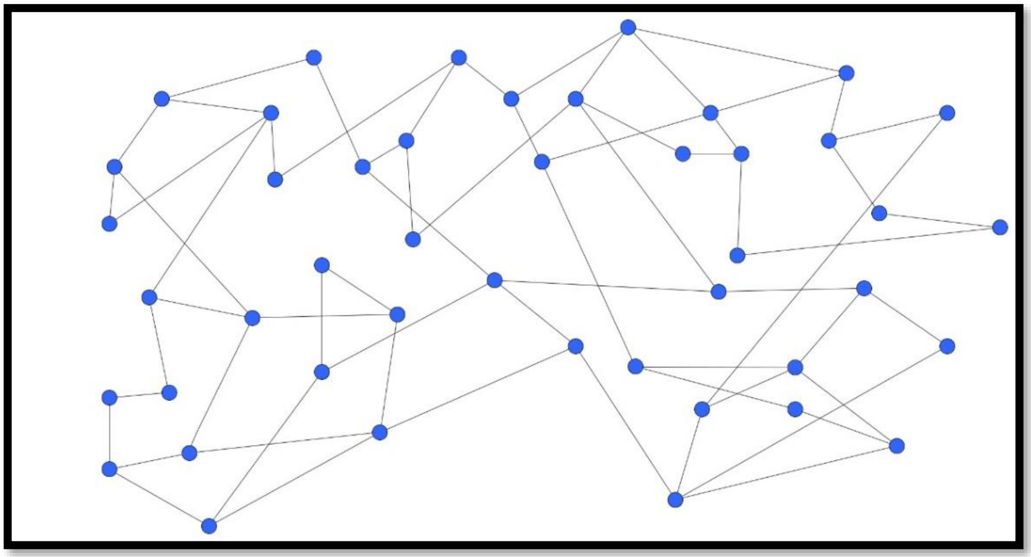

Figure 1. Graph

In K-median clustering, it is necessary to specify how many subsets the network will be divided. With Eq. (1), k-median clustering is performed in the network. With Eq. (3), the number required for each switch to be assigned to at least one controller is calculated. Then, the smallest integer k is chosen, which is greater than the ratio of the total number of switches to the controller port capacity. Controller port capacity is accepted as 6. Therefore, k=8 for our 45 switch test environment. The controller position determined by clustering with this k value and the switches connected to the controllers is shown in fig. 2.

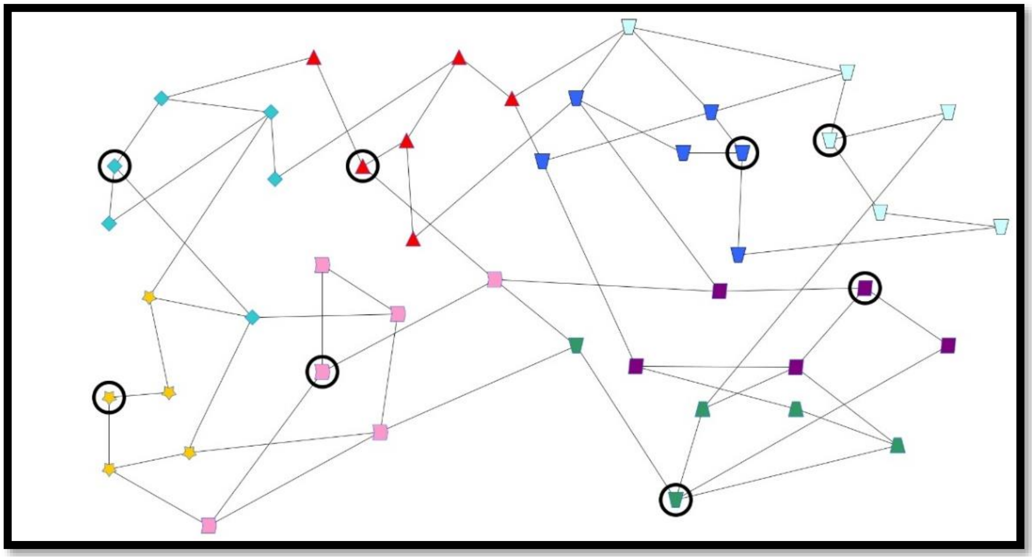

Figure 2. Optimal position selection of controller with 8-median

We add the fault tolerance ratio in CF-CPP as seen in Eq. (4) when calculating the number k. Fault tolerance is the probability of failure of controllers whose location and numbers are determined in the system. The main purpose here is to ensure that a controller that will assign each switch after the corrupted controllers. Table 1 shows the number of controllers according to the fault tolerance ratio for k = 8.

Table 1. Number of controllers according to fault tolerance ratio

Fault tolerance (%)	0	10	20	30	40
The number of controllers	8	9	10	11	12

The main purpose of CF-CPP is to ensure the resilience and reliability of the network. CF-CPP0 is the case where no controller is at fault in the proposed model. CF-CPP1 represents the situation where a controller is at fault.

CF-CPP1 is the linear programming model we recommend to ensure the continuity of the network. In case of failure of a dedicated controller, the recommended model is to assign the connected switches to the most suitable controller according to the minimum delay and controller port constraints. However, when calculating the average delay, it is not clear which assigned controller failed. For this reason, all probabilities were tried and averaged.

The (k-1) median represents the clustering again of all switches in the network when a controller fails in the network. The main purpose is to compare the reassignment by disconnecting the network's communication with the proposed model. All assigned controllers were removed from the system, and the delay was calculated and averaged.

As shown in Fig. 3, the average delay of the proposed models is compared according to the fault tolerance. As the fault tolerance rate increases, the average latency decreases. Because as the fault tolerance increases, the number of controllers assigned to the network increases. The average latency of communication in CF-CPP1 is higher than the (k-1) median. However, while the network's resilience is ensured in the proposed model, communication interruption and data loss occur when reassignment is made. While disconnecting the switches makes the system inoperable, it causes incomparable losses with instant data loss during the reassignment of the switches.

SDN provides easier network management by separating the control and data planes; however, they are susceptible to potential failures as with conventional networks. Choosing the number and location of controllers can affect the network's overall performance, as well as it is cost and resilience to potential failures. In this model, while ensuring the continuity of the network, hardware, energy, and management costs increase due to the number of controllers. In large-scale networks, determining the number of controllers by optimizing the number of controllers and costs can provide administrative and financial gains.

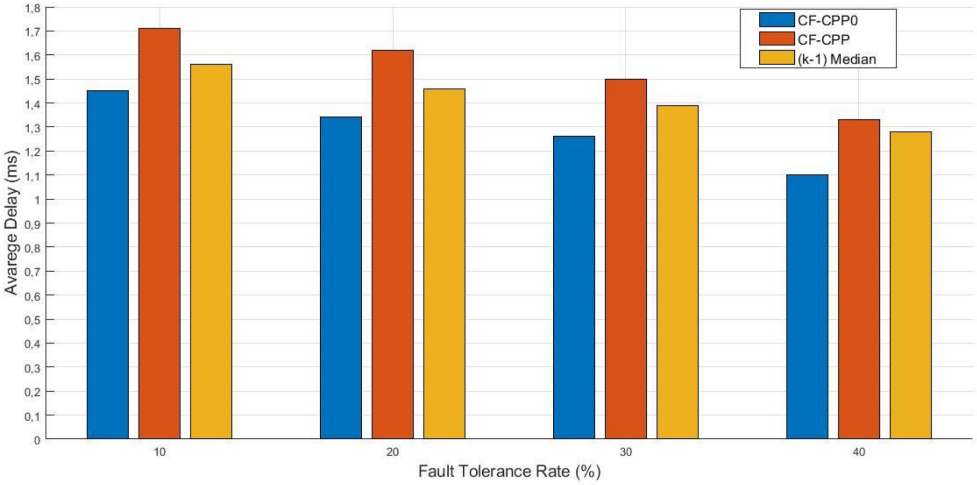

Figure 3. Relationship between average delay and fault tolerance

5. Conclusion

This study, proposes a model that solves the controller placement problem by adding port capacity and network resilience constraints and called the Controller Position Problem with Capacity and Fault tolerance(CF-CPP). The performance evaluation results show that the proposed CPP model ensures resilience in a large scale network. Although the average latency increases slightly with our proposed model, the reliability and resilience of the network increase significantly. In addition, the network's performance is improved by dynamically determining the number of controllers and fault tolerance. We intend to further improve performance by focusing on controller cost and energy consumption as our future plan.

Acknowledgment
This work is partially supported by The Scientific and Technological Research Council of Turkey (TUBITAK) under the grant number 118E212.

References

Alenazi, M. J., & Cetinkaya, E. K. (2020). Resilient placement of SDN controllers exploiting disjoint paths. *Transactions on Emerging Telecommunications Technologies, 31*(2), e3725.

Baktir, A. C., Ozgovde, A., & Ersoy, C. (2017). How can edge computing benefit from software-defined networking: A survey, use cases, and future directions. *IEEE Communications Surveys & Tutorials, 19*(4), 2359-2391.

Benzekki, K., El Fergougui, A., & Elbelrhiti Elalaoui, A. (2016). Software‐defined networking (SDN): a survey. *Security and Communication Networks, 9*(18), 5803-5833.

Bera, S., Misra, S., & Vasilakos, A. V. (2017). Software-defined networking for internet of things: A survey. *IEEE Internet of Things Journal, 4*(6), 1994-2008.

Chen, J., Chen, J., Xu, F., Yin, M., & Zhang, W. (2015). *When software defined networks meet fault tolerance: A survey*. Paper presented at the International conference on algorithms and architectures for parallel processing.

Das, T., Sridharan, V., & Gurusamy, M. (2019). A survey on controller placement in sdn. *IEEE Communications Surveys & Tutorials, 22*(1), 472-503.

Fonseca, P. C., & Mota, E. S. (2017). A survey on fault management in software-defined networks. *IEEE Communications Surveys & Tutorials, 19*(4), 2284-2321.

Gupta, A., & Jha, R. K. (2015). A survey of 5G network: Architecture and emerging technologies. *IEEE Access, 3*, 1206-1232.

Haque, I. T., & Abu-Ghazaleh, N. (2016). Wireless software defined networking: A survey and taxonomy. *IEEE Communications Surveys & Tutorials, 18*(4), 2713-2737.

Heller, B., Sherwood, R., & McKeown, N. (2012). The controller placement problem. *ACM SIGCOMM Computer Communication Review, 42*(4), 473-478.

Hu, T., Guo, Z., Yi, P., Baker, T., & Lan, J. (2018). Multi-controller based software-defined networking: A survey. *IEEE Access, 6*, 15980-15996.

Hu, Y.-N., Wang, W.-D., Gong, X.-Y., Que, X.-R., & Cheng, S.-D. (2012). On the placement of controllers in software-defined networks. *The Journal of China Universities of Posts and Telecommunications, 19*, 92-171.

Hu, Y., Wang, W., Gong, X., Que, X., & Cheng, S. (2014). On reliability-optimized controller placement for software-defined networks. *China Communications, 11*(2), 38-54.

Isong, B., Molose, R. R. S., Abu-Mahfouz, A. M., & Dladlu, N. (2020). Comprehensive review of SDN controller placement strategies. *IEEE Access, 8*, 170070-170092.

Jimenez, Y., Cervello-Pastor, C., & Garcia, A. J. (2014). *On the controller placement for designing a distributed SDN control layer.* Paper presented at the 2014 IFIP Networking Conference.

Killi, B. P. R., & Rao, S. V. (2019). Controller placement in software defined networks: A comprehensive survey. *Computer Networks, 163*, 106883.

Kobo, H. I., Abu-Mahfouz, A. M., & Hancke, G. P. (2019). Efficient controller placement and reelection mechanism in distributed control system for software defined wireless sensor networks. *Transactions on Emerging Telecommunications Technologies, 30*(6), e3588.

Kreutz, D., Ramos, F. M., Verissimo, P. E., Rothenberg, C. E., Azodolmolky, S., & Uhlig, S. (2014). Software-defined networking: A comprehensive survey. *Proceedings of the IEEE, 103*(1), 14-76.

Kuang, H., Qiu, Y., Li, R., & Liu, X. (2018). *A hierarchical K-means algorithm for controller placement in SDN-based WAN architecture.* Paper presented at the 2018 10th International Conference on Measuring Technology and Mechatronics Automation (ICMTMA).

Li, Y., & Chen, M. (2015). Software-defined network function virtualization: A survey. *IEEE Access, 3*, 2542-2553.

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.

Michel, O., & Keller, E. (2017). *SDN in wide-area networks: A survey.* Paper presented at the 2017 Fourth International Conference on Software Defined Systems (SDS).

Nunes, B. A. A., Mendonca, M., Nguyen, X.-N., Obraczka, K., & Turletti, T. (2014). A survey of software-defined networking: Past, present, and future of programmable networks. *IEEE Communications Surveys & Tutorials, 16*(3), 1617-1634.

Sallahi, A., & St-Hilaire, M. (2014). Optimal model for the controller placement problem in software defined networks. *IEEE Communications Letters, 19*(1), 30-33.

Singh, A. K., & Srivastava, S. (2018). A survey and classification of controller placement problem in SDN. *International Journal of Network Management, 28*(3), e2018.

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.* Paper presented at the 2016 IEEE International Conference on Communications (ICC).

Wang, G., Zhao, Y., Huang, J., & Wang, W. (2017). The controller placement problem in software defined networking: A survey. *IEEE Network, 31*(5), 21-27.

Xia, W., Wen, Y., Foh, C. H., Niyato, D., & Xie, H. (2014). A survey on software-defined networking. *IEEE Communications Surveys & Tutorials, 17*(1), 27-51.

Xiao, P., Qu, W., Qi, H., Li, Z., & Xu, Y. (2014). *The SDN controller placement problem for WAN*. Paper presented at the 2014 IEEE/CIC International Conference on Communications in China (ICCC).

Yao, G., Bi, J., Li, Y., & Guo, L. (2014). On the capacitated controller placement problem in software defined networks. *IEEE Communications Letters, 18*(8), 1339-1342.