Multi-Controller Placement Problem in Software-Defined Networking: A Survey

Maher Saab
InterNetWorks Research Lab, School of Computing, Universiti Utara Malaysia, Malaysia
Department of Information Technology, Technical College of Management, Baghdad, Iraq
m87alsaab@gmail.com

Shahrudin Awang Nor
InterNetWorks Research Laboratory, School of Computing, Universiti Utara Malaysia, Malaysia
Changlun, Malaysia
shah@uum.edu.my

Yousef Fazea
InterNetWorks Research Laboratory, School of Computing, Universiti Utara Malaysia, Malaysia
Changlun, Malaysia
yosiffz@uum.edu.my

Mohamed Elshaikh
School of Computer and Communication Engineering Universiti Malaysia Perlis (UniMAP)
Perlis, Malaysia

***Abstract*— Software-defined networking (SDN) is able to control and monitor the network through softwarization. This new emerging paradigm makes the process of placing controllers decisive to its scalability and performance. Many valuable studies have found that the Controller Placement Problem (CPP) in the infancy stage of the research is a significant aspect of the WAN SDN with respect to splitting strategy. However, these studies have overlooked the network size of domains and network balance splitting. This article critically reviews the state-of-the-art solutions and systematically classifies the multi-controller placement strategies in WAN SDN.**

***Keywords*— Software-defined networking, WAN SDN, Controller Placement Problem, Latency, Balanced Partitioning**

I. Introduction

The wide applicability of Software-defined networking (SDN) [1] [2] technology has made it a better choice over all other architectures of the Future Internet, such as Information-centric networking (ICN) and Named Data Networking (NDN). The main idea behind SDN is the separation of the control plane from the data planes, and migration of the control function of network devices to one external entity, the so-called SDN controller [3]. This emerging technology gives the network the ability to be directly programmed which offers another dimension, thereby improving the network utilization efficiency [4] and simplifying the network management [5]. The SDN architecture consists of three planes: data plane, control plane, and application plane as shown in Fig. 1. First, the data plane majorly comprises of network equipment like routers and switches which become simple forwarding devices that transmit packets by a decision made from the control plane. Second, the control plane is also known as the network intelligence, which runs a network operating system. It acts like an overpass between the application plane and the data plane. The communication between the data plane and the control plane is achieved through the Application Programming Interface (API) such as OpenFlow protocol [6]. Third, the application plane consists of the end-user business applications which use the SDN communications and network and is located on the top of the control plane.

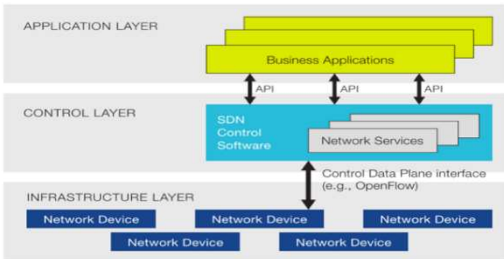

Fig. 10. Simplified View of the Software-Defined Network Architecture [7]

When the SDN appeared, the main initiative was to have a single controller that manages the whole network. However, due to day-to-day increasing network requests, the centralized single controller has encountered some problems and challenges more obviously in the Wide Area Network (WAN) [8] [9]. In addition, if the centralized single controller fails, the switches will lose their ability to transmit the arrived new packets, which damage the entire network. To overcome those issues, multiple controllers is proposed as a new controller approach which helped solve the problem of a single controller [10]. But this approach, however, poses a set of challenges missed. One of the challenges faced by multiple SDN controllers is how to deal with the scalability problem [11]. As a result, the researchers have introduced the concept of placement multiple controller in WAN SDN, and this problem is known as Controller Placement Problem (CPP) [12] that has become an interesting topic in the area of SDN research. In addition, finding the controllers' locations in the WAN SDN to solve the scalability is still a prominent challenge. Further, some open and challenging questions arise in the process of the placement controllers in the WAN SDN [13] [14]. Such questions are: i) What are the minimum numbers of controllers required to handle the entire network? ii) Where in the network will the controller(s) be placed, and, iii) How many networking devices can a controller be attached to them? Moreover, when there are multiple controllers in WAN, the desired outcomes present challenges for the optimization objectives. Hence, the primary focus of the discussion is on multi-controller placement techniques in relation to optimization objectives, such as latency [15], balanced partition [16] or a combination of these objectives.

This study elucidates the various controller placement techniques in WAN SDN. Each solution focuses on specific metrics and objectives can be envisaged when selecting the best controller placement in a WAN.

VII. Materials and methods

The multi-controller placement can be categorized based on the view of propagation latency and load balancing optimized objectives. This drives us to broaden the focus to survey and summarize the current development of those solutions from their objectives, techniques, and detailed strategies. After the analysis, the research progress is classified into two categories according to their objectives and summarize them in Table 1. First, the controller placement concentrates on determining the controllers' locations for improving the scalability. Second, the graph partition focuses on splitting the whole network into small sub-domains.

VIII. Multi-Controller Placement

The basic idea behind the multi-controller network architecture such as Opendaylight [17] and HyperFlow [18] is to monitor the whole network and eliminate the limitation of a single controller failure. In scalability [11] [19] which majorly causes constraints with WAN, the implementation of multi-controller network architecture in WAN can solve the scalability issue introduced by the single controller principle [18] [20] [21]. If the controllers are randomly distributed in WAN, it might lead to assigning unbalanced load on controllers, as well as it might lead to reducing the overall performance of WAN SDN.

a. Controller Placement Problem based on Propagation Latency

The locations of the controllers and the minimum number of controllers that can be placed in a network have been in existence since 2012. The past years have witnessed extensive studies and proposed solutions for the CPP by numerous researchers and have been proven to be non-deterministic polynomial-time hard (NP-hard) problems.

In order to review the solutions that describe the current research status of CPP in WAN, the general formula of CPP is provided firstly. For a network with specific nodes and links, the network elements are considered an undirected graph $G = (V, E, S)$, whereby V refers to the set of switches, E represents the physical links that linked those switches, and S refers to the set of controllers. Clearly, $n = |V|$ denotes the number of switches and $k = |S|$ denotes to the number of controllers. Generally, the studies on the CPP have utilized techniques to solve two primary issues: 1) finding the k's value; 2) $S \rightarrow V$ mapping relation, wherein the pre-selected objective function is optimized.

Latency is a particularly crucial performance metric in measuring the networking performance as it impacts the quality of service in rising real-time applications. The time needed for arrival and managing each switch within the network is referred to as latency. The fundamental principle beyond SDN is the decoupled network's control logic from forwarded devices and all of the functions of the network are executed through a secure channel by message exchanging between the controllers and the switches. When a switch transmits the package, it first checks the identical flow entry of its own flow table. If the switch cannot find the identical flow entry, it will request from its controller a new flow table. Based on the process of transmitting data, the controllers' locations are mainly determined by the propagation latency between the switches and its controller. Accordingly, the propagation latency is a large part of the latency and it is undoubtedly an important design metric in long-propagation-delay WAN.

In the following section, the switches to controller propagation latency will be reviewed in detail. After a careful analysis, there are two main metrics in propagation latency used for solving the problems CPP: (1) the switch to controller average latency (SC-avg latency), (2) the switch to controller worst latency (SC-worst latency) as discussed below.

i. Switch to controller average latency (SC-avg latency).

SC-avg latency indicates the average value of the switch to forward the packet to the controller, which reflects the essential efficiency of propagation latency in SDN. Equation 1 shows the mathematical representation of SC-avg latency.

$$L_{sc-avg}\,(S_i) = \frac{1}{n} \sum_{v \in V} \min_{s \in S} d(v, s) \qquad (1)$$

ii. Worst Latency between Switch and Controller (SC-worst latency).

SC-worst latency indicates the greatest value of the packet transference latency among the switch and the controller and represents an essential metric in a high-performance environment. Equation 2 shows the mathematical expression of SC-worst latency.

$$L_{SC-worst}\,(S_i) = \max_{(v \in V)} \min_{(s \in S_i)} d(v, s) \qquad (2)$$

The first use for the SC-avg latency as a metric for CPP to minimize the maximum propagation latency between the switch to the controller was by Heller et al. [12]. By using the K-center algorithm, Heller et al. compared different methods of placement with various numbers of controllers in order to reduce the average latency from switches to controllers. They selected 100 real networks from the Internet Topology Zoo [23] and Internet2 [24] to conduct their study. Unsurprisingly, their results showed that in nearly all topologies with medium-sized, a single controller is adequate to achieve “existing reaction-time requirements”, meanwhile if the number of controllers increased, the average latency will be reduced significantly. However, Heller et al. did not find a theoretical analysis for their proposed algorithm.

Under the polynomial approximation approach, Li et al. [25] initialized the study of CPP in WAN SDNs by taking the latency and controllers installation cost into consideration. The first effort is to develop the approximation algorithms such as the average latency or the maximum latency is bounded. They proposed the Bounded Maximum Latency Placement (BMLP) and Bounded Average Latency Placement (BALP) in order to find both lower and upper boundaries for identifying the minimal number of controllers. The second one is Balanced Cost Latency Placement (BCLP) by addressing both the latency and the cost of installing and maintaining the controllers. They introduced a trade-off function to minimize both the latency and the cost of the controllers. However, they simulated their algorithm on real network topologies and the results confirmed their theoretical analyses.

Zhang, Wang, and Huang [26] formulated a Multi-Objective Optimization Controller Placement (MOCP) problem and established the mathematical model to address the cooperation among multiple CPP to achieve the network reliability, controllers load balancing and reducing the propagation delay among switches and controllers. The authors used the Adaptive Bacterial Foraging Optimization (ABFO) algorithm and redefined its heuristic rules and bioinspired computing to find a solution to MOCP problem. They simulated their algorithm in MATLAB and the results showed the improvement of the multiple controllers' performance by minimizing the control path worst-case latency between the switch to its associated controller and other optimized objectives.

Geographically, with WAN SDN, the time-lag between the controller and its associated nodes is always increasing, thus leading to a poor network performance or QoS. In their study, Alowa and Fevens [28] proposed a technique called High Degree with Independent Dominating Set (HDIDS) to address the CPP to reduce the worst and average response times among the controllers and its switches. The algorithm is composed of twofold. (1) the node with the highest degree was selected as an available controller, (2) dividing the network into subdomains, assigning one controller for each subdomain, and utilizing the technique of independent dominating set to guarantee that the controllers are distributed with minimal response times. The experimental results of the proposed algorithm revealed a better performance from the perspective of reducing the maximal response time among controllers and its associated switches, as well as minimizing the average response time.

TABLE I. CLASSIFICATION OF THE CONTROLLER PLACEMENT APPROACHES

Objective	Technique	Details	Evaluation
Controller Placement Problem based on Propagation Latency	K-center [12]	Study the impact of propagation delay on multi-controller placement.	Distributing multi-controllers among the network may minimize the propagation delay effectively.
	Polynomial Approximation Algorithm [25]	Investigate WAN SDN from latency and controller's installation cost.	The approximation algorithm Balanced Cost-Latency Placement has a high performance in balancing the network latency and controllers cost.
	MOCP [26]	Balance the load between controllers and reduce the worst-case latency.	The proposed solution showed the improvement of the multiple controllers' performance by minimizing switch-controller worst latency.
	HDIDS [28]	To reduce the controller-to-switch worst and average response times.	The algorithm offers a better performance to reduce the maximal response time.
Graph Partition based on Propagation Latency	K*-means algorithm [31]	Introduced graph partitioning technique to reduce latency and balance the load.	The proposed solution produced more balance compared to the optimized K-means algorithm.
	SAPKM [33]	Defining two weighted cost functions to address the load balancing in SDWAN.	The proposed solution showed SAPKM's efficacy in minimizing the average load by considering the topology structure and the distribution of the flow traffic.
	DBCP [15]	Proposed speed-to-response algorithm that can be easily implemented.	DBCP can run the algorithm once to get a solution for the controller layout, and it can implement multiple applications and be easily extended.
	Cooperative Game [35]	Split the network into balanced subdomains and reduce the maximum delay.	Cooperative K-means outperforms the standard K-means and produces near-optimal solutions.

b. *Graph Partition based on Propagation Latency*

Studies announced in early efforts that the controller's reaction time relies on the latency of propagation between the controller and its related switches. The controller reaction time depends heavily on the latency between switch-to-controller and the controller load, as laid down in recent claims. Tootoonchian et al. [29] tested the controller performance and found that there are a limited number of service applications handled by the controller. Once the controller is overloaded, and a big propagation latency occurs, the network partitioning technique is introduced as a solution for CCP. The objective of partitioning WAN is to allocate nodes into small domains having strong internal connections with weak connections between domains. On the other hand, different network factors greatly improve the network manageability, scalability, and privacy, because fewer network nodes will be served by each domain controller. Hence, the domain partitioning problem can be classified into two groups, including balanced partitioning in which domains of the partition are approximately equal in size and unbalanced partitioning in which domains of the partition are different in size as discussed below.

The number of the subdomains is given by K, namely Ni (I = 1, 2,.., K). Therefore, the subdomains denote as:

$$Gi = (Vi, Ei, Si) \tag{3}$$

Which subject to:

$$\bigcup_{i=1}^{k} V_i = V \,;\ \bigcup_{i=1}^{k} E_i = E,\ \bigcup_{i=1}^{k} S_i = S \tag{4}$$

$$G_i \cap G_j = \varnothing \quad \forall\, i,j \in \{1,2,\ldots,3\}\ i \neq j \tag{5}$$

$$G_i \text{ is a connected sub graph } \quad \forall i \in K \qquad (6)$$

Equation (4) implies that the set of the subnetworks should cover the original network. Equation (5) indicates that one link or node can only be assigned to one subnetwork. In addition, equation (6) implies that all the nodes in one sub-network are linked via links.

The problem of network partitions that is similar to the problem of clustering and solutions may be taken from clustering algorithm contexts. Hence, different techniques are used for network partitioning such as spectral clustering, multilevel partitioning, local search, and density-based clustering through which networks can be partitioned [30].

By partitioning the large networks into unbalanced subdomains, the switches will be distributed asymmetrically, whereby the more switches are connected to a controller, the heavier will be the load on that controller as shown in Fig.s 2 and 3 [23]. These Fig.s illustrate the switches assignment to controllers based on the propagation latency that is used for the assignment process. Based on the Fig.s, an imbalanced distribution of switches among controller 2 and 3 can be noticed. In the former case, the partitioning of networks and an imbalanced distribution of switches among controllers may make an infeasible placement.

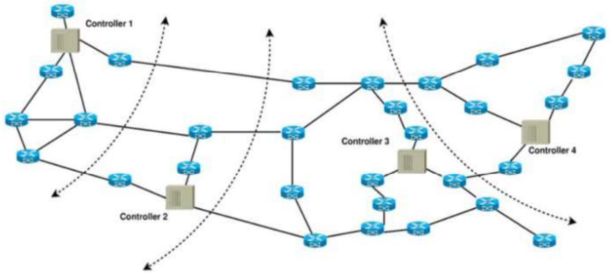

Fig. 11. Splitting depending on average-case latency

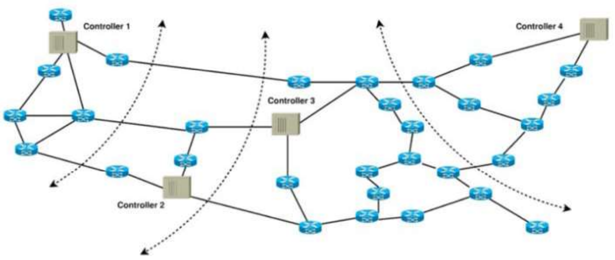

Fig. 12. Splitting depending on worst-case latency

Kuang et al. [31] introduced the network partition approach to solve the CPP. The authors partitioned an WAN into multi-domains by the algorithm based on K*-means suggested by Qi et al. [32]. That is, an WAN is partitioned into multi-domains to reduce the maximum latency and achieve the load balance for each controller. The K*-means algorithm increases the initial clustering points in order to obtain small clusters and each cluster contains one or more initial centers. Therefore, the more are K initial centers, the more will obtain the K clusters. Hence, the simulation results demonstrate that the proposed partition technique obtained more balanced results compared to the optimized K-means algorithm [27].

Yang et al. [33], integrated the centroid-based clustering algorithm [34] and proposed a hybridized efficient algorithm known as Simulated Annealing Partition-based K-Means (SAPKM) algorithm. The whole network is split into several subdomains, whereby each domain is managed by one controller. The proposed model, however, is formulated as a mixed-integer linear program (MILP) and mathematically formulated objective metrics including controller load reduction, propagation delay, and network reliability. In their study, the authors defined two different cost functions from the point of view of topology structure and the distribution of traffic flow to estimate the load balancing efficiency. The experimental results have shown SAPKM's efficiency by reducing the load balancing and the average load and minimizing the propagation latency, as well as improving the performance of network reliability. Unlike previous efforts on CPP which decided in advance the number of controllers required in a large-scale network, a study by Liao et al. [15] presents a technique known as Density-Based Controller Placement (DBCP), that utilizes dense-based clustering to determine the number of controllers. The authors simplified the problem of placing multiple controllers as the problem of placing a single controller by analyzing the topology structures and partitioning the topology into several domains. Since the different domains have fewer connections and within the domains have tight connections, it is adequate to place one controller in each domain. In their study, they assessed DBCP's performance on 262 publicly available topologies in the network. As a result, DBCP delivers a better performance with regard to the average propagation latency and other optimized goals.

A study by Killi et al. [35] proposed a network partition algorithm by making use of K-means algorithm with a cooperative game theory initialization. The network graph is divided into domains as a set of cooperative games, whereby the nodes are regarded as players of the game and each node attempts to maximize its value by building alliances with other nodes. The authors applied their study on the Internet2 OS3E topology. As a result, the analysis indicates that the solutions of the proposed algorithms are nearly optimal regarding the worst-case propagation delay between the controller and its associated nodes. In addition to that, the proposed algorithm produces balanced domains.

The previous studies showed that CPP in the infancy stage of the research is a significant factor of WAN SDN. The effective positioning of the controllers within the large-scale networks is intended to increase the network performance by distributing the load among controllers, minimizing the propagation latency, and maximizing the reliability, etc. Hence, a future research effort towards this end could provide a genuinely dependable solution for the problem of SDN controller positioning and open opportunities for academicians, businesses, and suppliers. However, studies on the controller utilization are insufficient; therefore, it needs further studying and examining.

IX. CONCLUSIONS

The basic idea behind the multi-controller network architecture is to monitor the entire network to eradicate the single controller failure limitations. The implementation of multi-controller network architecture in WAN can solve the scalability issue introduced by the single controller principle. Therefore, several literary works have addressed the problem of multi-controller placement with different optimization objectives. The multi-controller scalability can be evaluated in terms of controller placement and graph partition. On the one hand, the controller placement concentrates on determining the appropriate locations of controllers to increase the scalability. On the other hand, the graph partition in WAN SDN illustrates partitioning the whole network into small SDN subdomains and distributes the load among controllers. Consequently, this study systematically classified the current controller placement techniques that have the potential to increase the scalability and network performance. Therefore, this survey focuses on the controller placement approaches to two main factors, namely propagation latency and balanced partition as a significant optimized objective.

REFERENCES

[9] D. Kreutz, F. M. V. Ramos, P. Esteves Verissimo, C. Esteve Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-Defined Networking: A Comprehensive Survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.

[10] A. Prajapati, A. Sakadasariya, and J. Patel, "Software defined network: Future of networking," in Proceedings of the 2nd International Conference on Inventive Systems and Control, ICISC 2018, 2018, pp. 1351–1354.

[11] M. Saab, S. A. Nor, and Y. Fazea, "Software defined networking and open flow technologies: Challenges and future directions of programmable networks," *J. Adv. Res. Dyn. Control Syst.*, vol. 12, no. 2, pp. 809–817, 2020.

[12] X. Yao, H. Wang, C. Gao, and S. Yi, "Maximizing network utilization for SDN based on particle swarm optimization," in *Proceedings - 2016 IEEE 28th International Conference on Tools with Artificial Intelligence, ICTAI 2016*, 2016, pp. 921–925.

[13] T. Truong, Q. Fu, and C. Lorier, "FlowMap: Improving network management with SDN," in *Proceedings of the NOMS 2016 - 2016 IEEE/IFIP Network Operations and Management Symposium*, 2016, pp. 821–824.

[14] N. McKeown *et al.*, "OpenFlow: enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, p. 69, Mar. 2008.

[15] ONF, "Software-defined networking: The new norm for networks," *ONF White Pap.*, vol. 2, pp. 2–6, 2012.

[16] O. Michel and E. Keller, "SDN in wide-area networks: A survey," in *2017 4th International Conference on Software Defined Systems, SDS 2017*, 2017, pp. 37–42.

[17] Y. E. Oktian, S. G. Lee, H. J. Lee, and J. H. Lam, "Distributed SDN controller system: A survey on design choice," *Comput. Networks*, vol. 121, pp. 100–111, Jul. 2017.

[18] Y. Zhang, L. Cui, W. Wang, and Y. Zhang, "A survey on software defined networking with multiple controllers," *Journal of Network and Computer Applications*, vol. 103. Academic Press, pp. 101–118, 01-Feb-2018.

[19] M. Karakus and A. Durresi, "A survey: Control plane scalability issues and approaches in Software-Defined Networking (SDN)," *Comput. Networks*, vol. 112, pp. 279–293, Jan. 2017.

[20] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proceedings of the first workshop on Hot topics in software defined networks - HotSDN '12*, 2012, p. 7.

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

[22] N. Perrot and T. Reynaud, "Optimal placement of controllers in a resilient SDN architecture," in *Proceedings of the 2016 12th International Conference on the Design of Reliable Communication Networks, DRCN 2016*, 2016, pp. 145–151.

[23] J. Liao, H. Sun, J. Wang, Q. Qi, K. Li, and T. Li, "Density cluster based approach for controller placement problem in large-scale software defined networkings," *Comput. Networks*, vol. 112, pp. 24–35, Jan. 2017.

[24] P. Xiao, Z. Y. Li, S. Guo, H. Qi, W. Y. Qu, and H. S. Yu, "A K self-adaptive SDN controller placement for wide area networks," *Front. Inf. Technol. Electron. Eng.*, vol. 17, no. 7, pp. 620–633, Jul. 2016.

[25] J. Medved, R. Varga, A. Tkacik, and K. Gray, "OpenDaylight: Towards a model-driven SDN controller architecture," in *Proceeding of IEEE International Symposium on a World of Wireless, Mobile and Multimedia Networks 2014, WoWMoM 2014*, 2014, pp. 1–6.

[26] A. Tootoonchian and Y. Ganjali, "Hyperflow: a distributed control plane for openflow," *Proc. 2010 internet Netw. …*, pp. 3–3, 2010.

[27] A. Abuarqoub, "A Review of the Control Plane Scalability Approaches in Software Defined Networking," *Futur. Internet*, vol. 12, no. 3, p. 49, Mar. 2020.

[28] T. Koponen *et al.*, "Onix: A distributed control platform for large-scale production networks," in *Proceedings of the 9th USENIX Symposium on Operating Systems Design and Implementation, OSDI 2010*, 2010, pp. 1–6.

[29] S. Hassas Yeganeh and Y. Ganjali, "Kandoo: a framework for efficient and scalable offloading of control applications," in *Proceedings of the first workshop on Hot topics in software defined networks - HotSDN '12*, 2012, p. 19.

[30] J. H. A. Joseph Leung, Laurie Kelly, "Handbook of Scheduling: Algorithms, Models, and Performance Analysis - Google Books," *Book Handbook of Scheduling: Algorithms, Models, and Performance Analysis*, 2004. [Online]. Available: https://books.google.com.my/books?hl=en&lr=&id=MAY1ZstmGPkC&oi=fnd&pg=PP1&dq=J.+Y.+Leung,+Handbook+of+Scheduling:+Algorithms,+Models,+and+Performance+Analysis.+CRC+Press,+2004&ots=FCXc6I-7r9&sig=aNkE4MU12tCnLYku_9dTbfY7Kh0&redir_esc=y#v=onepage&q=J.%2520Y. [Accessed: 04-Sep-2018].

[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] "Internet2." [Online]. Available: https://www.internet2.edu/products-services/advanced-networking/layer-2-services/. [Accessed: 11-Feb-2020].

[33] T. Li, Z. Gu, X. Lin, S. Li, and Q. Tan, "Approximation algorithms for controller placement problems in software defined networks," in *Proceedings - 2018 IEEE 3rd International Conference on Data Science in Cyberspace, DSC 2018*, 2018, pp. 250–257.

[34] B. Zhang, X. Wang, and M. Huang, "Multi-objective optimization controller placement problem in internet-oriented software defined network," *Comput. Commun.*, vol. 123, pp. 24–35, Jun. 2018.

[35] G. Wang, Y. Zhao, J. Huang, Q. Duan, and J. Li, "A K-means-based network partition algorithm for controller placement in software defined network," in *2016 IEEE International Conference on Communications, ICC 2016*, 2016, pp. 1–6.

[36] A. Alowa and T. Fevens, "Combined Degree-Based with Independent Dominating Set Approach for Controller Placement Problem in Software Defined Networks," in *Proceedings of the 2019 22nd Conference on Innovation in Clouds, Internet and Networks and Workshops, ICIN 2019*, 2019, pp. 269–276.

[37] A. Tootoonchian, S. Gorbunov, Y. Ganjali, M. Casado, and R. Sherwood, "On Controller Performance in Software-Defined Networks," in *nd {USENIX} Workshop on Hot Topics in Management of Internet, Cloud, and Enterprise Networks and Services (Hot-ICE 12)*, 2012.

[38] M. Guo and P. Bhattacharya, "Controller Placement for Improving Resilience of Software-Defined Networks," in *2013 Fourth International Conference on Networking and Distributed Computing*, 2013, pp. 23–27.

[39] H. Kuang, Y. Qiu, R. Li, and X. Liu, "A hierarchical K-means algorithm for controller placement in SDN-Based WAN architecture," in *Proceedings - 10th International Conference on Measuring Technology and Mechatronics Automation, ICMTMA 2018*, 2018, vol. 2018-Janua, pp. 263–267.

[40] J. Qi, Y. Yu, L. Wang, and J. Liu, "K∗-means: An effective and efficient k-means clustering algorithm," in Proceedings - 2016 IEEE

International Conferences on Big Data and Cloud Computing, BDCloud 2016, Social Computing and Networking, SocialCom 2016 and Sustainable Computing and Communications, SustainCom 2016, 2016, pp. 242–249.

[41] K. Yang, D. Guo, B. Zhang, and B. Zhao, "Multi-Controller Placement for Load Balancing in SDWAN," *IEEE Access*, vol. 7, pp. 167278–167289, 2019.

[42] G. Wang, Y. Zhao, J. Huang, and Y. Wu, "An Effective Approach to Controller Placement in Software Defined Wide Area Networks," in *IEEE Transactions on Network and Service Management*, 2018, vol. 15, no. 1, pp. 344–355.

[43] B. P. R. Killi, E. A. Reddy, and S. V. Rao, "Cooperative game theory based network partitioning for controller placement in SDN," in *2018 10th International Conference on Communication Systems and Networks, COMSNETS 2018*, 2018, vol. 2018-Janua, pp. 105–112.