ACCEPTED FROM OPEN CALL

The Controller Placement Problem in Software Defined Networking: A Survey

Guodong Wang, Yanxiao Zhao, Jun Huang, and Wei Wang

Abstract

Recently, a variety of solutions have been proposed to tackle the controller placement problem in SDN. The objectives include minimizing the latency between controllers and their associated switches, enhancing reliability and resilience of the network, and minimizing deployment cost and energy consumption. In this article, we first survey the state-of-the-art solutions and draw a taxonomy based on their objectives, and then propose a new approach to minimize the packet propagation latency between controllers and switches. In order to encourage future research, we also identify the ongoing research challenges and open issues relevant to this problem.

Introduction

Software defined networking (SDN) is emerging as an innovative paradigm for next-generation networking. The distinctive feature of SDN, compared with the conventional networking, lies in the fact that the control plane is decoupled from the data plane. Specifically, the control plane is formed by a set of dedicated controllers, which serve as intelligent 'brains' of the SDN, while the data plane is composed of multiple simplified packet forwarding switches. This decoupling enables the network to become directly programmable and achieve several benefits [1–3], for example, simplifying the network management, improving the efficiency of network utilization, and enabling network innovations.

The representative communication interface of SDN is OpenFlow [1], which initially assumes a single controller for the sake of simplicity. Such a protocol, however, may suffer from scalability and performance issues when the network scale continually expands. Accordingly, numerous multi-controller approaches are then proposed, and fortunately they achieve a common basic architecture with joint efforts. As illustrated in Fig. 1, the basic SDN architecture consists of three layers: data plane, control plane, and application plane. The data plane is composed of packet forwarding switches that are managed by controllers via southbound application programming interfaces (APIs). The controllers are connected to the application plane via northbound APIs to facilitate network control and network services.

With multiple controllers, one critical issue is the controller placement problem [2]. The controller placement problem typically refers to how to place controllers in an SDN-enabled network and how to allocate associated switches to those controllers so as to achieve an objective, including shortening the latency, enhancing the reliability, increasing the energy efficiency, and so on. An example is given in Fig. 1 to demonstrate the controller placement problem. The pink dashed lines illustrate a possible connectivity between switches and controllers, but it might not be optimal in terms of performance. The controller placement solutions aim to seek optimal methods to connect controllers and their associated switches in an SDN-enabled network.

The controller placement problem has attracted enormous research interest, as it heavily affects almost every aspect of SDN, ranging from state distribution options to fault tolerance to network performance [2]. This motivates us to survey and summarize the current development of those solutions from their objectives, mathematical methods, and detailed strategies. We believe that a comprehensive survey on the latest developments and open challenges in this field would be beneficial to both researchers and practitioners. Specifically, the research progress is classified into three categories according to their objectives. For each category, the representative models, objectives, and solutions to the controller placement problem are elaborated. From a comprehensive study of the literature, we learn that decreasing the packet propagation latency between controllers and switches is especially significant for wide area networks. Therefore, a new approach, named the Clustering-Based Network Partition Algorithm (CNPA), is proposed in this article to address this problem as a case study. Specifically, CNPA is able to gradually partition a network into subnetworks and place controllers in those subnetworks to shorten the propagation latency between controllers and their associated switches. Last, the unsolved challenges and research directions are discussed to shed light on future studies for the controller placement problem.

Controller Placement Approaches

In this section, we classify ongoing research efforts on the controller placement problem. Before elaborating those approaches, the general formulation of the controller placement problem is presented first. For an SDN-enabled network, the main network elements include switches, controllers and links, which connect those devices. Therefore, the network is often modeled as an undirected graph $G = (V, E)$, where V represents the set of switches or controllers, and E is the set of physical links among those switches or controllers. Specifically, E refers to the physical distance between any two devices, and the distance affects the propagation latency in a physical network. As multiple links may exist between two switches, the shortest path distance, for example, achieved by

Guodong Wang and Yanxiao Zhao are with South Dakota School of Mines and Technology.

Jun Huang is with Chongqing University of Posts and Telecommunications.

Wei Wang is with San Diego State University.

Digital Object Identifier: 10.1109/MNET.2017.1600182

 0890-8044/17/$25.00 © 2017 IEEE

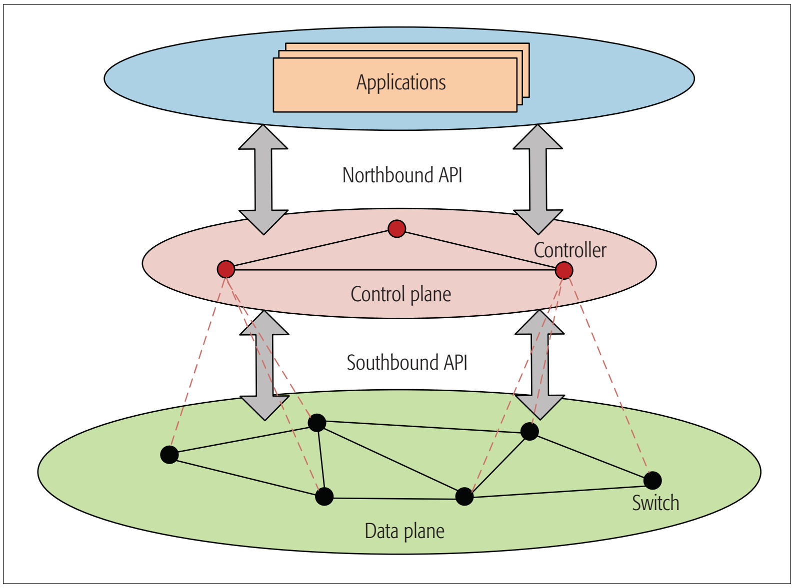

FIGURE 1. A simplified architecture for SDN.

Dijkstra's algorithm, is adopted to represent the distance between two switches. Specifically, the shortest path between switch u and v ($u, v \in V$) is denoted by $d(u, v)$ in this model. The output of the controller placement is a tuple $\{k; C; S\}$, where k is the number of controllers and $C = \{c_1, c_2, \cdots, c_k\}$ is the set of controllers, which are placed into the network; $S = \{S_1, S_2, \cdots, S_k\}$, where $S_i = \{s_{i1}, s_{i2}, \cdots\}$, denotes the allocation of switches S_i to controller c_i, which subjects to $S_i \cap S_j = \phi$ $(i, j \in k)$ and $S_1 \cup S_2 \cdots \cup S_k = S$. The studies on the controller placement problem generally exploit methods to find out the best k nodes in a network to place controllers so that a pre-defined objective function is optimized. We classify the representative existing solutions with regard to objectives and summarize them in Table 1. There are three main objectives: minimizing network latency, maximizing reliability and resilience, and minimizing deployment cost and energy consumption. In the rest of this section, representative works in each category are elaborated in detail.

Minimize Network Latency

The latency between controllers and switches is especially critical because all functions of SDN are carried out through frequent message exchanges between controllers and switches. The overall latency consists of packet transmission latency, propagation latency, switch queuing latency, and controller processing latency. The packet transmission latency equals the ratio of the packet size and the transmission rate of a link. The packet propagation latency is mainly determined by the distance between controllers and switches. The switch queuing latency is introduced by the congestion level of a link. The controller processing latency is primarily affected by loads of controllers. The above-mentioned components of latency do not weigh equally for different networks. For a wide area network that extends over a large geographical distance with large capacity links (e.g., OS3E), the packet transmission latency and switch queuing latency are negligible, while the packet propagation latency and controller processing latency dominate. This is because of the relatively small control message size in SDN and large capacity of control links, which are decoupled from the data plane. In the literature, most of the solutions to the controller placement problem are proposed under the wide area network scenario. When a wide area network is considered, research on minimizing the network latency in SDN can be classified into two subcategories, which aim to shorten the packet propagation latency and controller processing latency, respectively.

Minimizing the Propagation Latency: The controller placement problem resembles the facility location problem when propagation latency is considered. Therefore, solutions can be borrowed from the contexts of both "locating warehouses near a factory" and "optimizing locations of distribution nodes." For example, the impacts of controller placement on both the average and worst-case propagation latency are investigated in [2]. In particular, the controller placement problem is formulated as the minimum k-median problem. The average latency of a random controller placement is shown to be between 1.4 and 1.7 times larger than that of the optimal solution. Essentially, the propagation latency between controllers and switches plays the most significant role for an SDN-enabled network, and it will continually attract great interest in the SDN research community.

Minimizing the Controller Processing Latency: The controller processing latency is another significant contributor to the overall latency. This latency occurs especially when a controller's loads exceed its capacity. To the best of our knowledge, quantitative research, for example, mathematic analysis of relationships among the processing latency, the number of controllers, the concurrent sessions of switches, and the session requesting rate, has not been observed in the literature. One qualitative research study was conducted by Yao *et al.* [3], which considers both the controllers' loads and the propagation latency in their solution. Specifically, the controller placement problem is treated as a capacitated K-center problem and is solved through integer programming. Simulation results show that this strategy can reduce the number of controllers as well as the load of the busiest controllers.

Minimizing latency is always a critical objective for network planning and deployment since it significantly affects the network performance. That is why solutions are being constantly proposed to minimize the latency between controllers and their associated switches in the literature. Besides minimizing the latency between controllers and switches, the latency among controllers is also critical when multi-controllers are adopted in SDN, as it affects the consistency among multiple controllers. So far, quantitative research on the controllers' processing latency has received limited investigation. Those two directions are expected to be thoroughly studied in the future.

Maximize Reliability and Resilience

In the context of SDN, since network failures could cause disconnections between the control plane and data plane, and further disable some of the switches, it is of extreme importance to enhance the reliability and resilience in SDN. Reliability is defined as a performance metric that is inversely proportional to the "expected percentage of control path loss" [4], and the optimization target is to minimize the expected percentage of control path loss. In contrast, resilience is defined to reflect the capacity of controller placement strategies to sustain loss of connectivity upon controller or link disruptions [7]. Although these two terms are often used interchangeably in other fields, they are kept different in the existing works on SDN.

Maximizing Reliability: In the investigation of network reliability, it is usually assumed that at any time there is at most one physically failed component in an SDN-enabled network. Since the likelihood of failures in multiple components is small, this assumption is reasonable. Similar to the solutions of minimizing the latency between controllers and switches, this NP-hard problem can be transformed to a *k*-median problem, which is commonly addressed by approximation algorithms. For example, simulated annealing is adopted in [4] to increase the reliability of SDN by placing appropriate controllers in the network. While simulated annealing is a well-known technique, their contribution lies in optimizing the configurations of the algorithm, which reduces the search space and ensures rapid convergence to a near optimal placement. Another approach is to build a robust tree topology to minimize losses of management if failures happen in the network [5]. The robust control topology is composed of robust trees and controllers placed at the root of those trees. The robust tree is able to minimize the loss of management due to node or link failures. In addition, the robust trees are homogeneous and have balanced numbers of nodes on different branches, so the loads of controllers in different trees are balanced.

Maximizing Resilience: Research on network resilience can be further divided into two categories based on whether duplicate links are adopted in the network. In the first category, it is assumed that there is only one link between any two nodes. One of the representative works is Guo's solution [6], where the interdependence graph is borrowed to model the problem, and cascading failure analysis is adopted to analyze the impact of controller placement on SDN resilience. In their work, an SDN-enabled network is divided into two independent networks: the switch-switch network (SS) for data forwarding, and the controller-switch network (CS) for network control. Simulation results indicate that the expected network resilience is inversely proportional to the average path length of the network topology. Different from the single link approach, redundant link solutions are also proposed in the literature to increase the network resilience. One representative work is Survivor [7], which introduces multiple concurrent connections into the network. In order to maximize the network connectivity, Survivor first chooses controller positions that yield the highest number of node-disjoint paths between controllers and switches. Afterward, a list of backup controllers are constructed to improve survivability in case of failures. Simulation results verify that path diversity has greatly increased the survivability of SDN, and capacity awareness is also essential to handle overload during both normal and failure states.

Objectives	Solutions	Details	Methods
Minimize network latency	Heller's Solution [2]	Examine the impacts of placements on average and the worse-case propagation latency	K-center
	CCPP [3]	Reduce both the number and load of controllers	Integer programming
Maximize reliability and resilience	Hu's solution [4]	Maximize the reliability of networks	Simulated annealing
	K-critical [5]	Create robust control topology and balance load among controllers	Robust tree
	Guo's Solution [6]	Design a new resilience metric and improve the resilience of SDN	Interdependence graph and cascading failure
	Survivor [7]	Explore path diversity and improve survivability of SDN	Generic, proximity-based and residual capacity-based heuristics
	POCO [8]	Evaluate trade-off between failure free and controller failure values	Pareto-based optimal placement
Minimize deployment cost and energy consumption	Sallahi' solution [9]	Minimize the cost of installing controllers, linking controllers to switches and linking controllers together	Linear programming
	Rath's solution [10]	Minimize packet drops, delay and cost of deployment	Non-zero-sum based game theoretic
	GreCo [11]	Reduce the cost of energy consumption	Heuristic approach
	LiDy [12]	Propose a dynamic flow management algorithm to reduce energy consumption and maintenance costs	Heuristic location search and placement algorithm

TABLE 1. Classification of the controller placement solutions.

It is noteworthy that redundant links are definitely beneficial to network resilience, but they also increase the network deployment cost. Therefore, the trade-off between failure-free conditions and controller failures are evaluated in [8], which reveals that in the dominant topologies, more than 20 percent of all nodes need to work as controllers to guarantee continuous connection of all nodes to one of the controllers. Their algorithm provides no specific recommendation for a particular placement, but returns a set of Pareto-optimal placements, which enable network operators to choose an appropriate one. This strategy gives operators more flexibility in deploying an SDN-enabled network.

Minimize Deployment Cost and Energy Consumption

The overall cost of a network is also a big concern and has drawn attention as well. The overall cost mainly consists of the deployment cost and energy consumption. The deployment cost refers to the network equipment (controllers and switch-

Legacy devices in the Internet, such as switches and routers, typically run constantly regardless of the traffic load. After adopting SDN in a network, traffic can be monitored in real time and re-routed quickly. Therefore, energy consumption could be reduced through minimizing the number of active links during off-peak periods.

es) cost and their operating expenditure, which includes installing controllers into networks, linking controllers to switches, and linking those controllers. Associated efforts have been conducted to reduce deployment cost. In addition, the energy consumption of network devices at low network load still accounts for more than 90 percent of that at busy hour load. Therefore, solutions to shut down devices that are at low load to reduce energy consumption have also been proposed in the literature.

Minimizing Deployment Cost: Efforts to minimize the cost of controller deployment can be classified into two approaches, for example, static approach and dynamic approach. The static approach aims to calculate the optimal number, location, and type of controllers as well as the interconnections among switches and controllers to minimize the deployment cost. For example, an optimal model is proposed in [9] to formulate the deployment cost in an SDN-enabled network. Based on this model, a slight decrease is observed in the deployment cost by increasing the number of potential placement locations. This indicates that network operators have more options to select desired locations for controllers and thus reduce the overall cost. One shortcoming of this model is that solving this NP-hard problem is time-consuming. In particular, some of the calculations cannot be achieved in 30 hours, which means that this approach only applies to static controller placement. In contrast, dynamic solutions, which have low complexity and run in real time, are then proposed. A representative work is found in [10], which periodically calculates the overall load of the network and compares it with the controllers' capacity. In this way, the required controllers can be dynamically added or deleted according to the change of loads.

Minimizing Energy Consumption: Legacy devices in the Internet, such as switches and routers, typically run constantly regardless of the traffic load. After adopting SDN in a network, traffic can be monitored in real time and re-routed quickly. Therefore, energy consumption could be reduced through minimizing the number of active links during off-peak periods. One of the approaches to minimizing the energy consumption is GreCo [11], which aims to shut off as many links as possible while ensuring necessary connectivity between switches and controllers with a pre-defined delay bound and load balance. A heuristic algorithm is proposed to connect switches to other controllers and enable surviving links to form a connected network. Simulation results verify that GreCo is able to save up to 55 percent of energy during off-peak hours and use as few as 20 percent more links compared to the optimal solution. The other approach [12] proposes to solve this problem through two steps. The first step is to determine the locations of controllers to bound communication latencies. The second step is to determine the number of controllers per module to meet the load requirement. In addition, a dynamic flow management algorithm is proposed to activate the required number of controllers.

Some valuable research has been observed in decreasing network deployment cost and energy consumption, but it requires further efforts in the future. For example, one of the strategies adopted in the controller placement of SDN is flow management, which includes migrating traffic load to redundant links [7], dynamically adding or deleting controllers to/from a network [10], and switching off links to save energy [11]. Note that flow management inevitably results in extra cost, which is underexplored in the literature. In addition, the trade-off between network performance and energy saving also calls for in-depth evaluation.

Network Partition Algorithm for Controller Placement

As presented earlier, the latency between controllers and their associated switches is especially critical for an SDN-enabled network. Further analysis reveals that the latency is typically composed of two components: packet propagation latency and packet processing latency. In comparing those two components, decreasing the packet propagation latency is more challenging, especially for wide area networks [2]. Take the real-world network topology OS3E as an example. The round-trip propagation latency of the longest link in OS3E is 50 ms, which is unacceptable for SDN control messages. In this article, we propose to partition a network into subnetworks to shorten the packet propagation latency. CNPA is further proposed as a case study to elaborate the network partition approach.

The Cluster-Based Network Partition Algorithm

The preparatory work of the CNPA is to calculate the shortest path as well as the shortest path distance between any two nodes. Given a network topology with numerous links and nodes, the adjacent matrix of this topology is calculated first. Coordinates of those nodes are obtained from Google Map, and the distances of those links are calculated by using the "haversine" formula. The shortest path as well as the shortest path distance between any two nodes are calculated based on the coordinates and the adjacent matrix by using Dijkstra's algorithm. Note that the Euclidean distance is usually adopted in the default clustering algorithms, such as *K*-center and *K*-means. However, the Euclidean distance does not meet the requirement of partitioning a network topology, as the nodes are only connected by the given links instead of the Euclidean connection. Therefore, the shortest path distance is adopted in CNPA to partition a network into subnetworks. Accordingly, associated clustering algorithms, including *K*-means and *K*-center, which are introduced to compare with the CNPA, are also revised to adopt the shortest path distance to partition a network. In addition, there are two critical parameters in clustering algorithms: "center" and "centroid." For clarification, the initial nodes which are selected to perform clustering are referred to as the "center." The actual node that is eventually found by CNPA is referred to as the "centroid," which has the shortest distance to other nodes in the subnetwork.

Processes of CNPA are listed as follows. The first step of CNPA is to randomly select one node as the center of the network. As the center is randomly selected, CNPA will further find out the actual center (centoid) of the network in the second step. Specifically, CNPA calculates each node's sum distance to other nodes and selects the one that has the minimum sum distance as the centroid. In the third step, CNPA will find out the second center of the network. The second center is selected as the node that is furthest away from its cluster's centroid. In the fourth step, CNPA treats the centroid (c_1) and the second center (c_2) as two initial centers and allocates associated nodes to those centers. Specifically, for each node n_i, calculate its distance to those two centers and get two distances, $d_1 = d(n_i, c_1)$ and $d_2 = d(n_i, c_2)$. Compare those two distances and allocate the node to the center that is closer to it. For example, if $d_1 < d_2$, node n_i will be assigned to c_1. Once the node is assigned to one center, the centroid of this cluster will be recalculated based on the minimum sum distance described in the second step. The process is repeated until the network is eventually divided into K subnetworks. Unlike regular clustering algorithms, which randomly select all K centers at once and then optimize all of them during the following iterations, the proposed CNPA first divides the entire network into two subnetworks, then three to K subnetworks. During each partition, it is able to shorten the maximum distance between the centroid and the associated nodes, and hence the maximum distance is significantly decreased compared to the regular clustering algorithms such as K-center and K-means.

Performance Evaluation

In this subsection, we evaluate the performance of the proposed CNPA in comparison to two representative solutions, K-center and K-means. K-center is adopted widely in the literature to decrease the propagation latency between controllers and switches [2, 3], and K-means is the most popular clustering algorithm. The Internet2 OS3E, a real-world network topology from the Internet Topology Zoo [2], is chosen as the network scenario in this article due to its widespread adoption in evaluating the performance of the controller placement problem [2–4]. As a typical wide area network, the OS3E is composed of 34 nodes, which are connected by 43 links. Those nodes are distributed widely from Canada to most states in the United States. The maximum distance (shortest path distance) of this network is 3120 miles, which is from Vancouver, British Columbia, Canada, to Miami, Florida, United States. The corresponding round-trip propagation latency of this link is about 50 ms. In the following simulations, coordinates of nodes in the OS3E are obtained from Google Map, and distances of those links are calculated by using the haversine formula. The shortest path distance between any two nodes is calculated by using Dijkstra's algorithm. The associated propagation latency is calculated by

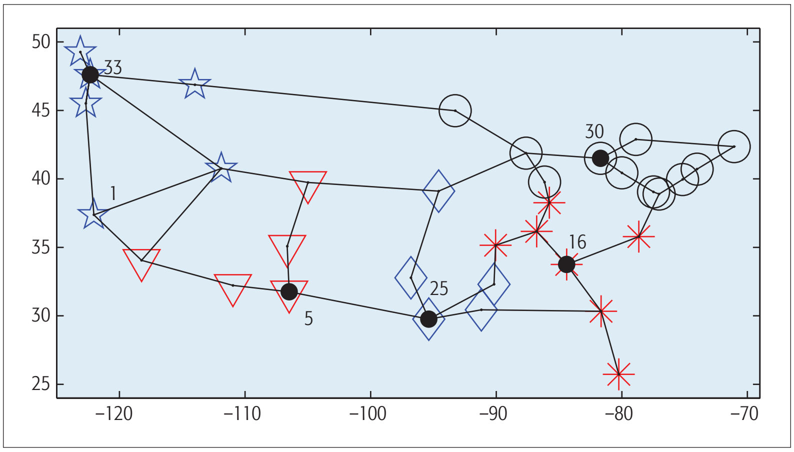

FIGURE 2. Network partition of the OS3E achieved by CNPA (K = 5).

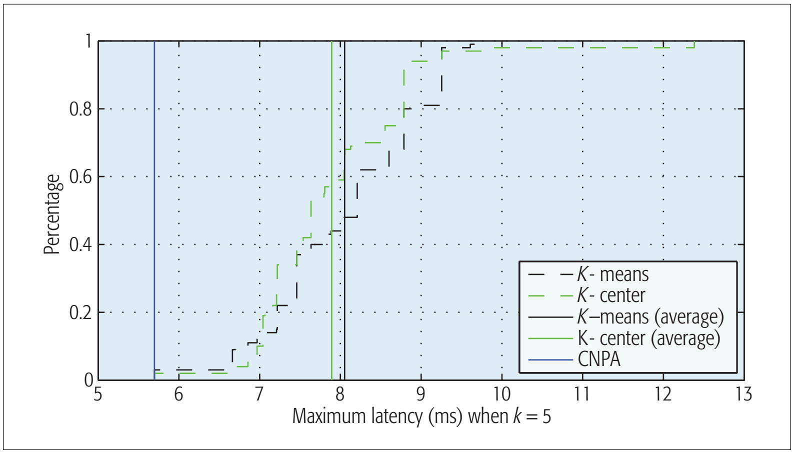

FIGURE 3. Maximum latency CDFs for all possible partitions of the OS3E network (K = 5).

$$latency = \frac{distance\,(m)}{2 \times 10^8 (m/s)}$$

More detailed explanations of those processes are given in the "preparatory work" of the CNPA above.

For clustering algorithms like K-center and K-means, the resulting partition varies for each execution because the initial centers in those algorithms are randomly selected. To evaluate their performance, we have conducted our experiment 100 times with both the K-center and K-means, and compare them with CNPA. CNPA is only executed once, as it always gets the same result for every execution. This is because CNPA starts with one entire network and has fixed initial centers, which leads to the same subnetwork partition.

In the first simulation, the Internet2 OS3E network is partitioned into five subnetworks by CNPA. Figure 2 demonstrates the network partition results achieved by CNPA. The five subnetworks are distinguished by "∘," "∗," "∇," " ★," and "◊." The centroid of each cluster, where the controller is placed, is denoted by "•." To compare the performance in terms of latency, Fig. 3 depicts the maximum latency cumulative distribution function (CDF) of all the partitions. The dashed black and green curves represent the maximum latency CDF where the network is partitioned into five subnetworks by K-means and K-center, respectively. The solid black and

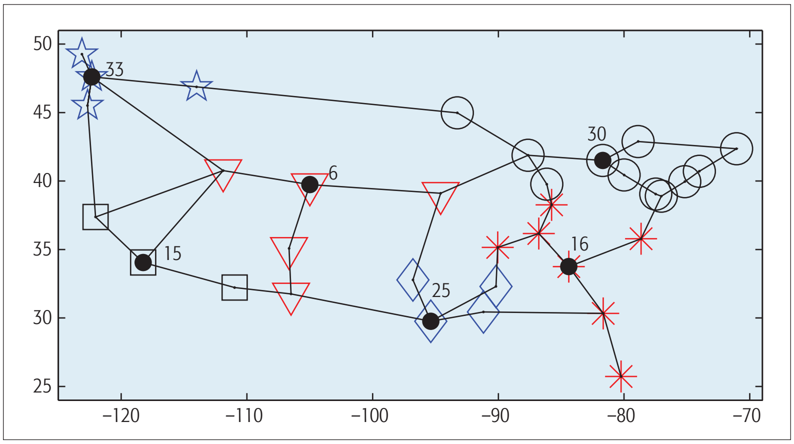

FIGURE 4. Network partition of the OS3E achieved by CNPA (K = 6).

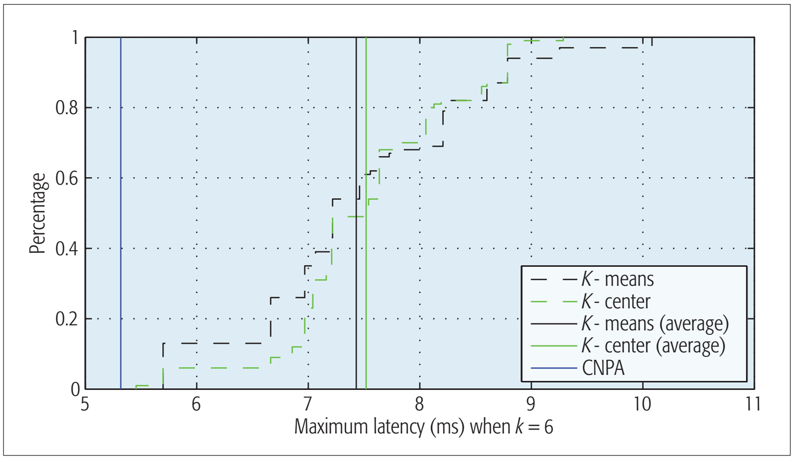

FIGURE 5. Maximum latency CDFs for all possible partitions of the OS3E network (K = 6).

green lines stand for the average value of 100 maximum latencies achieved by K-means and K-center, respectively. The solid blue line is the maximum latency CDF achieved by CNPA. It can be observed that among these 100 partitions adopted by K-means and K-center, there is a slim chance that those partitions can reach the smallest maximum latency, while CNPA can constantly obtain the smallest maximum latency.

To illustrate how a network is partitioned by CNPA and compare its performance with K-center and K-means, the Internet2 OS3E network is continually partitioned into six subnetworks by the CNPA, and the result is depicted in Fig. 4. Here, we elaborate how the partition is obtained from Fig. 2 to. Fig. 4. As depicted in Fig. 2, there are five subnetworks, and the latency between node 33 and node 1 has the biggest value in all of those subnetworks. Therefore, the subnetwork denoted by "★" is selected by CNPA as the target subnetwork, which will be further partitioned. Specifically, node 1 joins the 5 existing centroids, (nodes 5, 16, 25, 30, and 33) and serves as the new 6 initial centers, which are adopted by CNPA to continually partition the network into 6 subnetworks. The resulting six subnetworks and their corresponding centroids are highlighted in Fig. 4.

The results in terms of maximum latency CDF are plotted in Fig. 5, where we can observe that the partition deployed by CNPA has greatly decreased the maximum latency between the controller and associated switches. It is worth noting that the maximum latency achieved by CNPA is even smaller than the one achieved by both K-center and K-means, which are executed 100 times.

Conclusion

The controller placement problem has drawn considerable attention from academia since it plays a critical role in deploying SDN in wide area networks. This article has surveyed the state-of-the-art solutions to the controller placement problem with different objectives. In general, the existing work is classified into three categories, including how to shorten latency between controllers and switches, maximize network reliability and resilience, as well as minimize deployment cost and energy consumption. Despite those efforts, a number of challenges in controller placement remain unaddressed. To encourage more forthcoming research work on this subject, we identify the following issues and possible directions for future research.

Efficient Algorithm: The controller placement problem is known as NP-hard, and seeking an optimal solution for this problem is time-consuming. This indicates that it would be extremely challenging to obtain the optimal result in real time. A feasible solution is to partition the network into subnetworks and place controllers in subnetworks instead of the whole network to reduce the complexity. Although we propose a network partition algorithm in this article, more efficient algorithms need to be designed with quality of service consideration.

Multi-Objective Optimization: The controller placement problem is typically formulated as a multi-objective optimization issue, which poses new challenges in complexity and modeling. The majority of existing solutions primarily consider one or two objectives, but this is insufficient for an actual network. Future research can be conducted by analyzing trade-offs between the objectives discussed above and constructing an appropriate multi-optimization model.

Cooperation among Multiple Controllers: Multi-controller coexistence is the future trend for SDN to accommodate the fast-growing services. Benefits of multiple controllers cannot be fully achieved without fine-grained cooperation among controllers. Therefore, coordination and communication among controllers are required and deserve thorough consideration to guarantee quality of service, especially for large-scale SDN-enabled networks.

Cost Awareness: The cost concern with controller placements has emerged recently, but the equilibrium between network performance and cost reduction has not been fully evaluated. This calls for further evaluation. For example, operations like switching links on or off are favorable for saving energy, but such operations will inevitably incur cost to networks [9]. Trade-offs between those operations deserve in-depth examinations.

References

[1] J. Liu *et al.*, "Device-to-Device Communications for Enhancing Quality of Experience in Software Defined Multi-Tier LTE-A Networks," *IEEE Network*, vol. 29, no. 4, July 2015, pp. 46–52.

[2] Y.-D. Lin *et al.*, "An Extended SDN Architecture for Network Function Virtualization with a Case Study on Intrusion Prevention," *IEEE Network*, vol. 29, no. 3, May 2015, pp. 48–53.

[3] J. Liu *et al.*, "Reliability Evaluation for NFV Deployment of Future Mobile Broadband Networks," *IEEE Wireless Commun.*, vol. 23, no. 3, June 2016, pp. 90–96.
[4] N. McKeown *et al.*, "OpenFlow: Enabling Innovation in Campus Networks," *ACM SIGCOMM Computer Commun. Review*, vol. 38, no. 2, 2008, pp. 69–74.
[5] B. Heller, R. Sherwood, and N. McKeown, "The Controller Placement Problem," *Proc. 1st Wksp. Hot Topics in Software Defined Networks*, 2012, pp. 7–12.
[6] G. Yao *et al.*, "On the Capacitated Controller Placement Problem in Software Defined Networks," *IEEE Commun. Letters*, vol. 18, no. 8, 2014, pp. 1339–42.
[7] L. F. Müller *et al.*, "Survivor: An Enhanced Controller Placement Strategy for Improving SDN Survivability," *IEEE GLOBECOM*, Austin, TX, Dec. 2014, pp. 1909–15.
[8] A. Sallahi and M. St-Hilaire, "Optimal Model for the Controller Placement Problem in Software Defined Networks," *IEEE Commun. Letters*, vol. 19, no. 1, 2015, pp. 30–33.
[9] Y. Hu *et al.*, "On Reliability-Optimized Controller Placement for Software-Defined Networks," *China Commun.*, vol. 11, no. 2, 2014, pp. 38–54.
[10] Y. Jimenez, C. Cervello-Pastor, and A. J. Garcia, "On the Controller Placement for Designing a Distributed SDN Control Layer," *IFIP Networking Conf.*, 2014, pp. 1–9.
[11] M. Guo and P. Bhattacharya, "Controller Placement for Improving Resilience of Software-Defined Networks," *Int'l. Conf. Networking and Distributed Computing*, 2013, pp. 23–27.
[12] D. Hock *et al.*, "Pareto-Optimal Resilient Controller Placement in SDN-Based Core Networks," *Int'l. Teletraffic Congress*, 2013, pp. 1–9.
[13] H. K. Rath *et al.*, "Optimal Controller Placement in Software Defined Networks (SDN) Using a Non-Zero-Sum Game," *IEEE Int'l. Symp. World of Wireless, Mobile and Multimedia Networks*, 2014, pp. 1–6.
[14] A. Ruiz-Rivera, K.-W. Chin, and S. Soh, "GreCo: An Energy Aware Controller Association Algorithm for Software Defined Networks," *IEEE Commun. Letters*, vol. 19, no. 4, 2015, pp. 541–44.
[15] M. T. I. ul Huque, G. Jourjon, and V. Gramoli, "Revisiting the Controller Placement Problem," *Proc. IEEE 40th Conf. Local Computer Networks*, 2015, pp. 450–53.

Biographies

Guodong Wang (wgdaaa@gmail.com) is currently a research scientist at the South Dakota School of Mines and Technology. He received his Ph.D. degree from the Chinese Academy of Sciences in 2013. His research interests include protocol design and performance evaluation of various communication networks. His current research mainly focuses on software defined networks (SDN), IoT and smart grid networks.

Yanxiao Zhao (yanxiao.zhao@sdstmt.edu) received her Ph.D. degree from Old Dominion University, USA, in May 2012. She is currently an assistant professor in the Electrical and Computer Engineering Department at the South Dakota School of Mines and Technology, USA. Dr. Zhao's research interests include protocol design, performance evaluation and experimental implementation for various communication and networking systems. Her research is supported by a National Science Foundation (NSF) grant and Air Force grants. She was the recipient of the Best Paper Award at WASA 2009 and Chinacom 2016.

Jun Huang [M'12, SM'16] (xiaoniuadmin@gmail.com) is currently a full professor at the School of Computer Science, Chongqing University of Posts and Telecommunications. He has authored 90+ publications and several of them are in prestigious journals or conference proceedings. His research interests include SDN, IoT, and D2D/M2M. He is a senior member of the IEEE.

Wei Wang [M'10, SM'16] (wwang@mail.sdsu.edu) is an assistant professor with the Department of Computer Science, San Diego State University, USA. He received his Ph.D. degree in computer engineering from the University of Nebraska–Lincoln, USA, in 2009. His major research interests include QoE/QoS issues in wireless multimedia communications, network economics, and IoT. He won Best Paper Awards at IEEE WCNC 2008 and ANSS 2011.