IEEE Access
Multidisciplinary : Rapid Review : Open Access Journal

Received November 22, 2018, accepted January 11, 2019, date of publication January 18, 2019, date of current version March 7, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2893283

A Survey of Controller Placement Problem in Software-Defined Networking

JIE LU, ZHEN ZHANG, TAO HU, PENG YI, AND JULONG LAN
National Digital Switching System Engineering and Technological Research and Development Center, Zhengzhou 450002, China

Corresponding author: Zhen Zhang (zhangzhen2096@163.com)

This work was supported in part by the National Key Research and Development Program of China under Grant 2016YFB0801200, Grant 2017YFB0803201, and Grant 2017YFB0803204, and in part by the Natural Science Foundation of China under Grant 61802429 and Grant 61521003.

ABSTRACT In order to simplify the management of the traditional network, software-defined networking (SDN) has been proposed as a promising paradigm shift that decouples control plane and data plane, providing programmability to configure the network. With the deployment and the applications of SDN, researchers have found that the controller placement directly affects network performance in SDN. In this paper, the state of the art of controller placement problem is surveyed from the perspective of optimization objective. First, we introduce the overview of SDN and controller placement problem. Then, we classify this paper of controller placement problem into four aspects (latency, reliability, and cost and multi-objective) depending on their objective and analyze specific algorithms in different application scenarios. Finally, we identify some relevant open issues and research challenge to deal with in the future and conclude the controller placement problem.

INDEX TERMS Software-defined networking, controller placement problem, latency, reliability, cost, multi-objective.

I. INTRODUCTION

With the continuous development of information technology, the Internet has evolved into a complex large-scale infrastructure which deeply affect people's working and living style [1]. However, the complexity and closure of traditional network makes it difficult for administrators to operate and manage, especially in dynamic and complex networks. In consequence, there is an urgent need for a new technology and method to improve existing networks.

Software-defined networking (SDN) [2]–[4] is a new promising network paradigm that propose to solve those problem, which core idea is to separate the control and data planes, and to transfer the control function of network components (such as switches and routers) to the upper control plane (controller). This decoupling allows the network to be programmed directly which could achieve many benefits [5], [6], such as simplifying network management, improving network utilization efficiency, supporting network innovation and so on.

The representative southbound interface of SDN is OpenFlow [7], which initially assumes that there is only one controller in the network for the sake of simplicity. With the expansion of the SDN network, it is hard for a single controller to meet the extensive management needs [8]. In order to improve the scalability and reliability of the network and avoid single point of failure [9], logically centralized, physically distributed multi-control network architecture emerge, such as HyperFlow [10], Kandoo [11], Onix [12] and so on. Since the location and number of controller deployments have a huge impact on network performance in a multi-controller network architecture, Controller Placement Problem (CPP) has become a hotspot in current SDN research.

For a given network, Controller Placement Problem [13] primarily considers three issues: 1)the number of controllers; 2)the position of controllers; and 3)the allocation between switch and controller, aiming to optimize objectives, such as shortening the latency [13]–[15], enhancing the reliability [16]–[18], increasing the energy efficiency [19], [20], and so on. There are some surveys of CPP with different concentrations. For example, Singh and Srivastava [21] concludes the CPP research initiatives for various application scenarios considering various factors and constraints. Zhang *et al.* [22] proposes a comprehensive literature survey on multiple controllers with design principles, architectures and placement.

Hu *et al.* [23] summarizes four main research challenges in multi-controller environment: scalability, consistency, reliability and load balancing. Wang *et al.* [24] draws a taxonomy based on their metrics, however the classification is too simple and does not analyze specific solution.

In this paper, a comprehensive survey of controller placement problem in SDN is presented. According to optimized objective, the research literature is generally divided into four main categories: latency, reliability, energy and multi-objective. In each category, we analyze the cause of this optimization goal and its impact on network performance, then introduce the representative models, results detailly for different application scenarios. The contribution of this article are summarized as follows:

- We present a detailed analysis of the objectives in CPP and summarize four optimized objectives: latency, reliability, cost and multi-objective.
- For each optimized objective, exhaustive theoretical and scenario analysis are proposed. Then, some representative and up-to-date models are introduced with solving method, results and relative merits.

The rest of this paper is organized as follows. Section II presents the overview of SDN and CPP. From section III to section VI, we discuss the current research status of different optimized objective: latency, reliability, cost and multi-objective respectively. In Section VII, we discuss the promising research directions and issues to deal with in the future and finally, the concluding remarks are presented in Section VIII.

II. OVERVIEW OF SDN AND CPP

In this section, we introduce the basic architecture of SDN firstly, then present the general formulation and optimized objective of the controller placement problem, finally indicate the difference challenges of CPP in datacenters and WANs.

A. SDN ARCHITECTURE

The SDN separates the control plane from the forwarding devices (routers and switches), the logically centralized programmable controller in the control plane can grasp the global network information, which is convenient for operators and researchers to flexibly adjust and deploy new network architecture and related technologies. Unlike traditional network devices, forwarding devices (also known as switches) in the SDN data plane provide only simple data forwarding functions. Due to the simplicity of its functions, SDN switches can quickly handle matching packets and adapt to the growing demand for network traffic.

As illustrated in Fig. 1, the basic SDN architecture consists of three layers: data plane, control plane, and application plane. The data plane has a number of network elements (switches), which provides the processing abstraction of data forwarding or streaming to the upper layer through a data-control plane interface (southbound API). The controller plane is composed of several controllers. Each controller performs fine-grained control on the network elements in the data plane according to the requirements of the upper layer application, and on the other hand, provides the abstract information model (such as resources, status, events, etc.) of the underlying network to the application layer through the application-controller plane interface (northbound API). The application plane consists of several SDN applications, which can submit the required network behavior to the controller through the interface (northbound API) in a programmable manner.

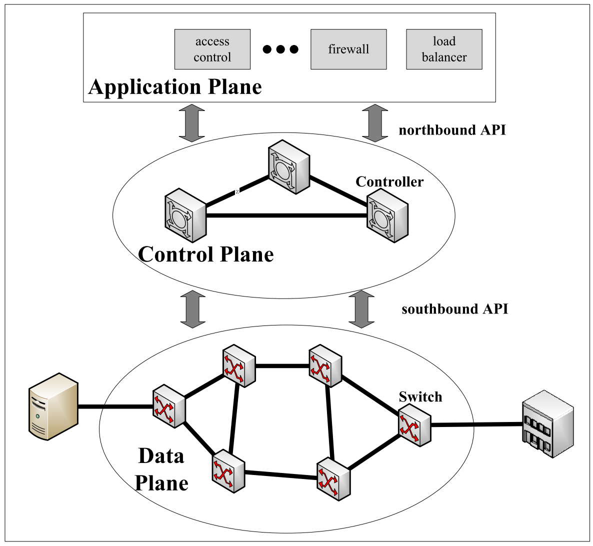


FIGURE 1. The basic architecture of SDN.

B. GENERAL FORMULATION OF CPP

In order to describe the research of the controller placement problem in the network, we firstly present a general formulation of CPP. The SDN network is often modeled as a graph $G = (V, E, S)$, where V presents the set of switches, E is the set of physical links among switches or controllers, and S denotes the set of controllers. Specifically, $n = |V|$ presents the number of nodes and $k = |S|$ refers to the number of controllers. The studies on the controller placement problem generally exploit methods to solve two questions:1) the value of K; 2)$S \rightarrow V$ mapping relation, so that a predefined objective function is optimized. In table 1, we list down the mathematical symbols used in this article.

C. THE OPTIMIZED OBJECTIVE OF CPP

Heller first analyzed the impact of controller deployment on network average latency and maximum latency. Subsequently, other objectives have been proposed, including load-balancing, reliability, energy-saving. Because of conflict between different objectives, multi-objective optimization has also been a feasible solution. Figure 2 shows the overall classification of our state-of-the- art of CPP from the view of optimized objective. The CPP is firstly classified into 4 sections: latency, reliability, cost and multi-objective.

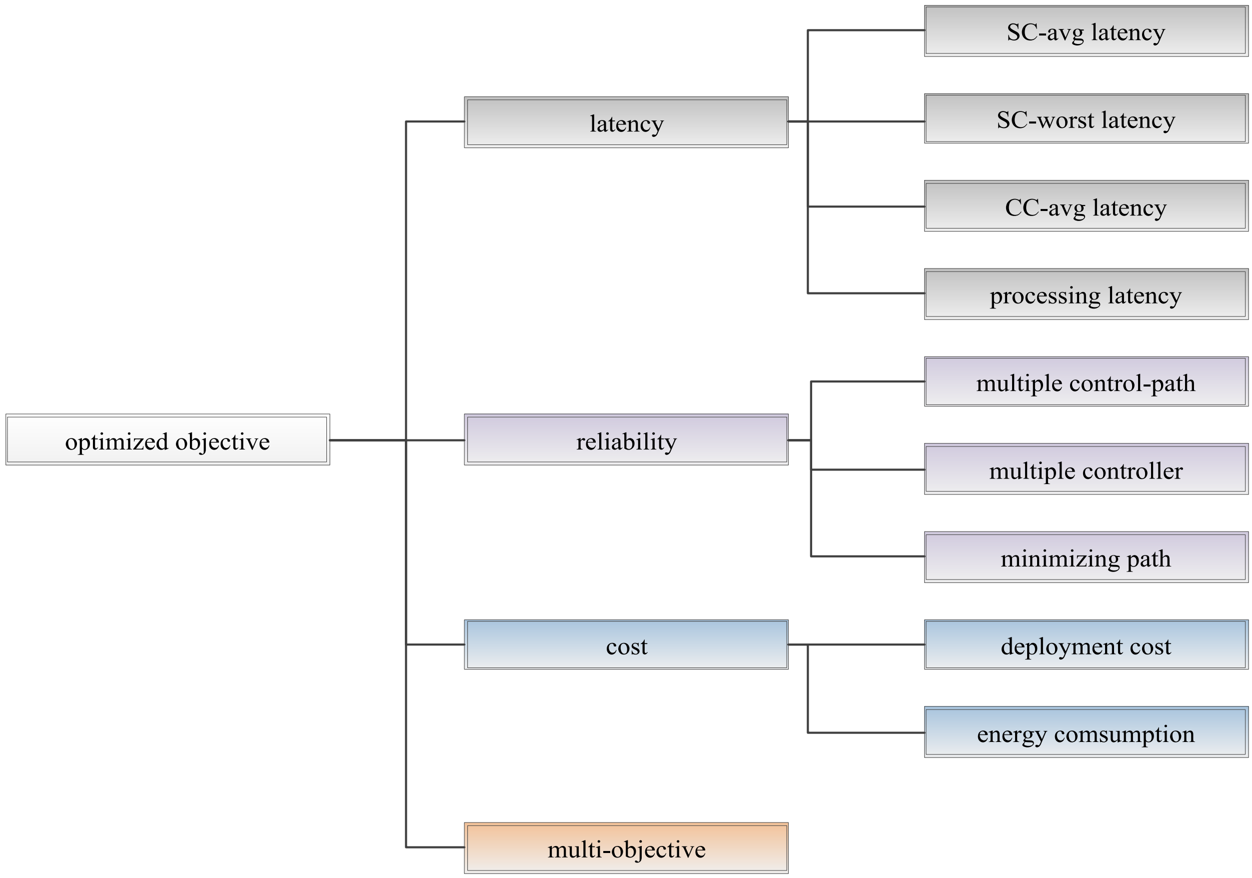


FIGURE 2. Classification of optimized objective.

TABLE 1. The mathematical symbols.

symbols	definitions
V	the set of switches
E	the set of physical links
S	the set of controllers
n	the number of nodes
k	the number of controllers
$d(v,s)$	the shortest path between $v \in V$ and $s \in S$
$l(c)$	the number of nodes assigned to controller c
p	the probability of network component failure, include node and link.

Then the latency is further classified into 4 subsections: (a)average latency between controller and switch (SC-avg latency), (b)worst latency between controller and switch (SC-worst latency), (c)average latency between controllers (CC-avg latency), and (d)processing latency. The reliability is divided into 3 subsections: (a)multiple control-path, (b) multiple controller, and (c) shortest control path. And cost is split into 2 subsections: (a) deployment cost, and (2) energy consumption.

D. THE SCENARIO OF CPP

This article focuses on CPP in wired networks including datacenters and WANs. Datacenters are the cornerstone of the big data infrastructure supporting numerous online services which contain a large number of computers, switches, and servers. The peculiarity of datacenters are high density, limited space, and small scale. In contrast, WANs interconnect multiple local area networks or datacenters over geographically dispersed locations, and its characteristics are big traffic, long distance, high link cost and large scale. Because of the different characteristics of the datacenters and WANs, the solution of CPP in these two environments is signal different. Generally speaking, in datacenter, the goal of CPP is to get optimal solution; and in the WANs, heuristic algorithm or network partition should be adopted to find the suboptimal feasible solution in a limited time by reducing the search space.

III. LATENCY

Because the control logic of the network is decoupled from simplified switches and all functions of the network are carried out through message exchanging between controllers and switches, the latency is especially critical in SDN [25]. Network latency include queuing latency, transmission latency, propagation latency, and processing latency [26]. When the network is unobstructed, the queuing latency is negligible. The transmission latency is related to the data packet and port rate, and usually a fixed value in the case of

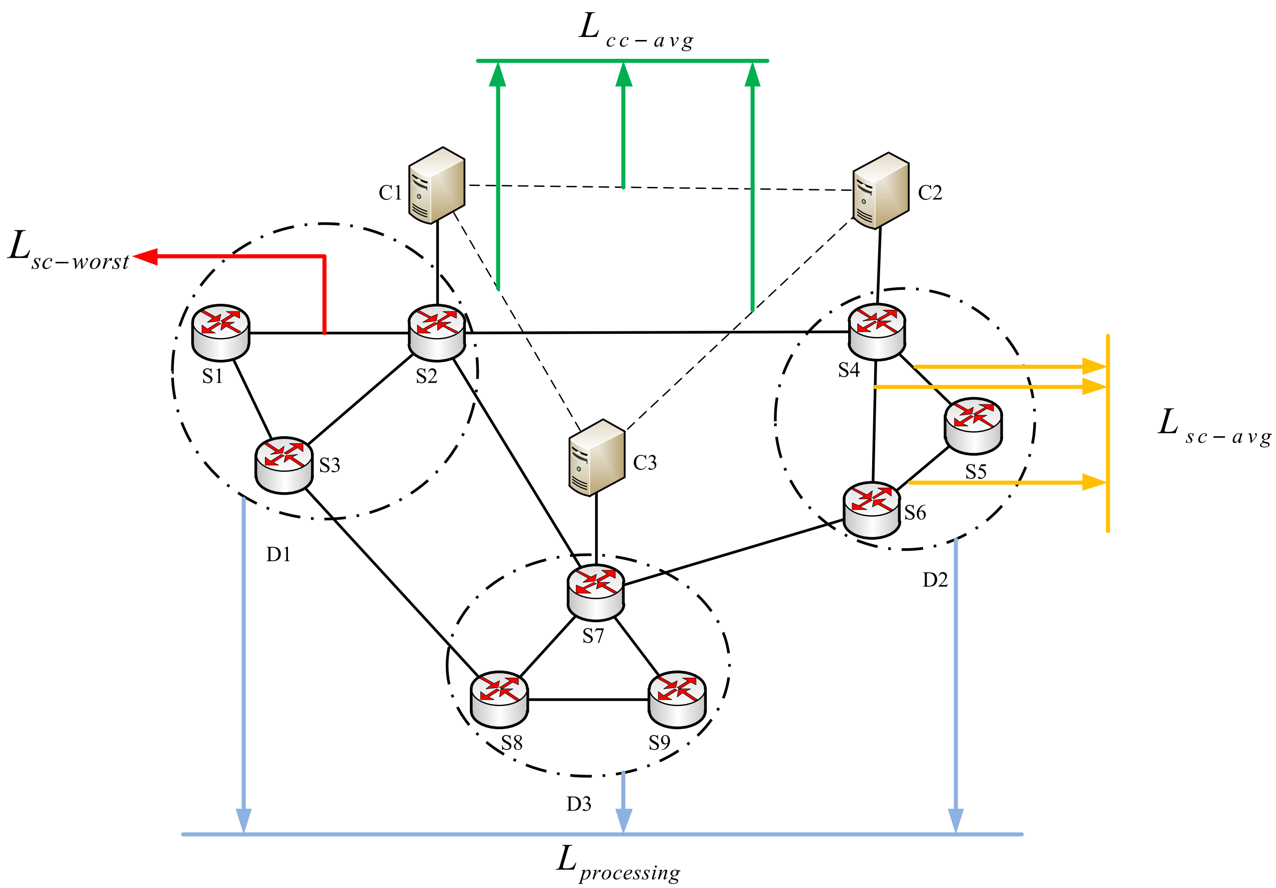


FIGURE 3. Four types of latency.

the same network device. Therefore, the CPP problem usually only considers propagation latency and processing latency [25], [27] where the propagation latency is mainly determined by the distance between two nodes and The processing latency is primarily affected by the processing capability and load of the controller.

As figure 3 show, the latency can also be analyzed from different levels of the SDN architecture. In intra-domain, there are two types of latency, the worst latency between controller and switch in D1 should be accepted and the average latency between controller and switch in D2 reflecting overall performance should be as small as possible; in inter-domain, the average latency between controllers should be considered because of communication between controllers for view consistency, and the processing latency, which would significantly increase when load exceed controller processing power, should be reduced by load balancing between controllers. After careful analysis, we categorize the latency into 4 aspects: (1) the average latency between switch and controller (SC-avg latency), (b) the worst latency between switch and controller (SC-worst latency), (3) the average latency between controllers (CC-avg latency), and (4) processing latency.

A. SC-AVG LATENCY

SC-avg latency represents the average value of the packet transmission latency between the switch and the controller, reflecting the basic performance of propagation latency in SDN. The mathematical expression is as shown in Equation 1.

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

According to the application scenario, the existing representative algorithm are divided into two categories. In datacenters, the general exhaustive algorithm is usually used to find the optimal solution, such as k-center [13] for static traffic, mixed integer programming [29] for dynamic traffic; in WANs, as the large number of switches, it cannot be solved by exhaustive way in a limited time, so heuristic algorithms [31], [32] have also been proposed to reduce the search space and ensure rapid convergence to a near optimal placement.

1) K-CENTER

Heller *et al.* [13] proposes the SC-avg latency as a measurement of CPP to reflect the overall propagation latency between the controller and the switch in the SDN network firstly. The author selects 100 actual networks from

TABLE 2. Comparison of literature of SC-AVG latency.

Year	Author	Method	Evaluation	Application Scenario	
				datacenter	WAN
2012	Heller et al. [13]	K-center	• A single controller could meet the latency needs of most mid-size networks. • Adding controllers could significantly reduce latency.	√	×
2017	He et al. [29]	Mixed integer programming	• For low flow densities, adaptive flow controller placement reduces average flow settling time by approximately 50% compared to static placement. • When the density is high, change the controller placement to ensure that the flow setting performance is negligible. •	√	×
2015	Lange et al. [31]	Pareto simulated annealing	• According to the user's requirement for accuracy, more than 20 times faster solution acceleration is possible.	×	√
2017	Liao et al. [32]	Density-based cluster	• In order to obtain a solution for controller layout, DBCP only needs to run the algorithm once. • DBCP can implement multiple applications and be easily extended.	×	√

Topology Zoo [28] to study the impact of different controller deployment strategies on average latency. The results show that in most medium-sized networks, a single controller could meet the latency requirements; at the same time, increasing the number of controllers can significantly reduce the average latency. However, the author doesn't propose theoretical analysis of algorithm.

2) MIXED INTEGER PROGRAMMING

Under dynamic traffic conditions, higher end-to-end traffic setup times may occur if there is no proper controller placement. He *et al.* [29] analyzes the controller layout of dynamic traffic flow based on the combined controller layout model: controller location and switch-to-controller assignment are optimized for the minimum average traffic settling time for different traffic conditions within the network. The linearization method is applied to transform the problem into a mixed integer programming (MIP) problem that can be optimally solved by Gurobi [30]. Two derivatives are also provided for comparison, one to optimize controller position and the other to optimize switch-to-controller assignment.

3) PARETO SIMULATED ANNEALING

In small and medium-sized networks, all deployment locations can be evaluated, but it is difficult to apply them to large-scale networks for limited computation time and resource constraints. Lange *et al.* [31] proposes the Pareto Simulated Annealing algorithm for solving controller deployment problems in large-scale networks. In the presence of time constraints in highly dynamic environments, heuristic algorithms allow for a tradeoff between time and accuracy.

4) DENSITY-BASED CLUSTERING (DBCP)

Since the heuristic algorithm may fall into a local optimal solution, Liao *et al.* [32] proposes a method called density-based controller Placement (DBCP), which uses a density-based switch clustering algorithm to divide the network into several subnets. Since switches are tightly connected in the same subnet and have fewer connections to switches in other subnets, it is sufficient to deploy one controller per subnet. In DBCP, the size of each subnet can be determined by the capacity of the deployed controller. In addition, the optimal number of controllers is obtained based on density-based clustering.

We investigate and analyze the controller placement techniques for SC-avg latency in Table 2. We compare different techniques from the aspects of year, authors, method, simulation/evaluation and application scenarios including, datacenters and Wide Area Network (WAN). The $\surd$ represents feasible and $\times$ represents not feasible. The rest tables in the paper follow the same notation.

B. SC-WORST LATENCY

SC-worst latency denotes the maximum value of the packet transmission latency between the switch and the controller, which is usually an optimal objective in a high-performance environment or a strict constraint. The mathematical expression is as shown in Equation 2.

$$L_{sc-worst} = \max_{v \in V} \min_{s \in S} d(v, s) \tag{2}$$

According to the application scenario, the existing representative algorithm are divided into two categories. In datacenters, Wang *et al.* [33] proposes optimized k-means for avoiding the impact of the random initial position. In WANs, heuristic algorithms, such as particle swarm optimization algorithm [34] and firefly algorithm [35], are proposed to speed up in limited resource and time constraints; Sahoo *et al.* [36] divides network into several subdomains to reduce the search space and adopt meta heuristic technique to solve.

1) OPTIMIZED K-MEANS

In [33], the concept of network partitioning is proposed by Wang to solve the controller placement problem. Specifically, the clustering algorithm is used to divide the network into subnets which reduce the complexity, and an optimized K-means algorithm is proposed to shorten the maximum waiting time between the centroids in the subnet and the related switches. Optimized K-means ensure that each partition can reduce the maximum delay between the centroid and other nodes, so performance is significantly improved compared to conventional K-means

2) ARTICLE SWARM OPTIMIZATION

Liao and Leung [34] presents a distributed controller layout problem to find the best solution minimizing switch-to-controller latency for wide-area software-defined networks, controller-to-controller latency, and balancing controller load. Then, the paper introduces a generic model that considers not only the layout of the controller but also the allocation of switches. In order to solve this problem with huge search space without loss of generality, authors introduce a multi-objective genetic algorithm (MOGA) based on particle swarm optimization algorithm which maintains a precomputed global best position for each single goal and selects the global best position to guide parental variation

3) COOPERATIVE GAME

Considering the importance of k-means' initialization, Killi *et al.* [35] adds cooperative game initialization to k-means algorithm and divide the network into sub-region simulations as a set of cooperative games. All of the switches are considered as players, and each switch tries to form alliances with other switches to maximize its value. Through the real network simulation of OS3E and the topology zoo, the proposed algorithm can ensure that the worst latency between the controller and the switch is close to optimal.

4) FIREFLY ALGORITHM (FFA)

Although various solutions have been proposed to solve the CPP problem of small and medium-sized networks, it still requires an alternative solution for large-scale networks like WAN which choose the number of controllers and their locations under limited resource and time constraints. Sahoo *et al.* [36] proposes two random meta heuristic techniques to find the optimal position of the controller to optimize the delay between the controller and the specified controller. Authors develop a firefly algorithm (FFA) for solving CPP, where the fitness function is used as the controller-switch latency, which needs to be minimized.

We investigate and analyze the controller placement techniques for SC-worst latency. The results are presented in table 3.

C. CC-AVG LATENCY

Communication between controllers is crucial to achieve a consistent view of the network's state, which is required for proper operation of the network application. The observation and investigation confirm that the communication overhead caused by maintaining the shared state among the controllers is very significant. The mathematical expression is as shown in Equation 3.

$$L_{cc-avg} = \frac{1}{k} \sum_{s \in S} \min_{s' \in S} d(s, s') \tag{3}$$

According to the application scenario, the existing representative algorithm are divided into two categories. In datacenters, Ksentini *et al.* [37] takes the controller as a participant in the game and optimizes the distance between controllers through game theory; Zhang *et al.* [27] transforms the CPP problem into the solution space pole by adding constraint parameters to cut the solution space of the original problem; Ishigaki gradually approaching a given target based on constraints of performance metrics. In WANs, Ishigaki *et al.* [38] considers CPP as a weighted minimum set coverage problem and propose greedy algorithm to solve it, but ignore the limitation of bandwidth, which is solved in [39]

1) BARGAINING GAME

Ksentini *et al.* [37] presents the optimal controller placement SDN networks. The proposed solution aims to integrate three key objectives: minimizing latency, minimizing communication overhead between controllers and ensuring load balancing for optimal layout and controller count. These goals are somewhat contradictory, which leads authors to rely on bargaining games to find the best solution and ensure a fair trade-off between these goals.

2) MULTIPLE DATA-OWNERSHIP MODEL

Zhang *et al.* [27] discusses Pareto-optimal controller placements considering controller-to-switch average latency and controller-to-controller latency for WAN topologies adopted in some real ISP networks. The paper formulates controller placement problem as Integer Linear Programming (ILP) problem to find the optimal placement that minimizes the reaction time perceived at the switches. Addition, authors provide new quantitative tools to optimize the planning and design of networks supporting the SDN network control

TABLE 3. Comparison of literature of SC-WORST latency.

Year	Author	Method	Evaluation	Application Scenario	
				datacenter	WAN
2016	Wang et al. [33]	Optimized k-means	• Network partition strategy significantly reduces the complexity of the controller placement. • The maximum latency can reach 2.437 times smaller than the average latency achieved by the standard k-means.	√	√
2017	Liao et al. [34]	Particle swarm optimization	• when optimizing a single objective, PSO can find a solution much closer to the real global optima • MOGA with PSO based mutation can quickly converge a pareto frontier with larger diversity towards the heuristics, but more evaluations need to be done in SDNs with various topologies and scales.	√	√
2018	Killi et al. [35]	Cooperative game	• Cooperative k-means strategy produces near-optimal solutions and outperforms the standard k-means for controller placement. • The running time of cooperative k-means is less than one second for the topology with 109 nodes.	×	√
2018	Sahoo et al. [36]	Firefly algorithm	• For communication latency, the FFA has improved performance than particle swarm optimization.	×	√

plane, especially when the network is very large and adopts an in-band control plane.

3) GREEDY ALGORITHM

Based on graph optimization theory, Ishigaki *et al.* [38] considers the communication overhead between controllers and proposes the problem of minimizing the distance between controllers. The article considers this problem as a weighted minimum set coverage problem and proves that the problem is NP-hard. After that, the author proposes three greedy algorithms with different weights.

4) SIMULATED ANNEALING

By analyzing the complexity of the controller placement and traffic routing issues, Shaoteng *et al.* [39] introduces a generic black-box optimization process formulated as a feasibility problem which implements simulated annealing for association. Unlike existing methods, the optimization process adds the extra steps needed for quantifying the consequences of deploying a control plane solution that fulfills the specified reliability and bandwidth requirements. As a powerful predictive tool, service providers and operators can use this method to fine-tune control plane deployment strategies.

We investigate and analyze the controller placement techniques for CC-avg latency in Table 4.

D. PROCESSING LATENCY

Processing latency is mainly affected by the processing power and load of the controller. When the load of the controller approaches or exceeds the processing power of the controller, the processing delay increases significantly. Therefore, the processing latency is usually optimized by balancing the load of the controllers. The mathematical expression is as shown in Equation 4.

$$L_{processing} = \min\left(\max_{s\in S} l(s) - \min_{s'\in S} l(s')\right) \tag{4}$$

According to the application scenario, the existing representative algorithm are divided into two categories. In datacenters, Yao *et al.* [40] proposes the capacitated k-center to limit the controller's load for avoiding load exceed the processing power of the controller. In WANs, network partition [25] and spanning tree [41] are used to reduce the size of the CPP and speed up.

1) CAPACITATED K-CENTER

In view of the phenomenon that the controller overload causes the controller processing latency to increase significantly, Yao *et al.* [40] considers the influence of the controller load for the first time, and propose the capacitated k-center algorithm to solve the controller deployment problem. The simulation results show that compared with the dynamic controller configuration or k-center strategy of dynamic scheduling, the new strategy can significantly reduce the number of required controllers, reduce the load of the maximum load controller, and reduce the radius.

2) MINIMUM SPANNING TREE

According to the network topology and load conditions, Jiugen *et al.* [41] model the network as a minimum span-

TABLE 4. Comparison of literature of CC-AVG latency.

Year	Author	Method	Evaluation	Application Scenario	
				Data Center	WAN
2016	Ksentini et al. [37]	Bargaining game	• The solution achieves the pareto optimal efficiency between SC-delay and CC latency in the network.	√	×
2017	Zhang et al. [27]	Integer linear programming	• An optimized choice of the controller acting as data owner can improve the reaction time by 2-4 times. • Choosing the master controller of a switch as the closest controller does not always minimize the reaction time perceived at the switches.	√	×
2017	Ishigaki et al. [38]	Greedy algorithm	• Some metrics (SC average latency) within each subnetwork can be guaranteed by selecting specific clustering methods before the placement. • Greedy algorithms using the average distance weight and constant weight are always efficient.	√	×
2018	Shaoteng et al. [39]	Simulated annealing	• Running the controller placement method regardless of control traffic leads to excessive bandwidth usage (worst cases varying between 20.1%-50.1% more) and congestion. • The approach allows near-optimal deployment under varying conditions	×	√

TABLE 5. Comparison of literature of processing latency.

Year	Author	Method	Evaluation	Application Scenario	
				datacenter	WAN
2014	Yao et al. [40]	Capacitated k-center	• The result show that capacitated k-center could significantly reduce the number of required controllers, the load on the maximum load controller, and the worst switch controller latency.	×	√
2017	Jiugen et al. [41]	Minimum spanning tree	• The algorithm can ensure that the ratio between the maximum load of the controller and the optimal load is approximately 2 the maximum latency difference of the network does not exceed 0.65 ms.	√	√
2017	Wang et al. [25]	Improved k-means	• Compared to K-means and K-center, CNPA greatly reduce the latency between controllers and their associated switches.	√	√

ning tree structure, and the network is divided into k control regions by an equalization partitioning algorithm. Finally, in each sub-domain, the controller is deployed based on the principle of minimizing the sum of all switch-to-controller distances in the area. However, this algorithm can only deal with static traffic.

3) CLUSTERING-BASED NETWORK PARTITION

In order to decrease the end-to-end latency and processing latency, Wang *et al.* [25] adopt two-steps approach to solve. First, Wang divide the network into subnets to reduce end-to-end latency between controller and switch. A Clustering-based Network Partition Algorithm (CNPA) was developed for network partitioning. Second, place multiple controllers in a subnet with a large number of switches to further reduce processing latency.

We investigate and analyze the controller placement techniques for processing latency in Table 5.

IV. RELIABILITY

Since network failures could cause communication interruptions between network components (such as controllers or switches) in SDN, which lead to severe packet loss and

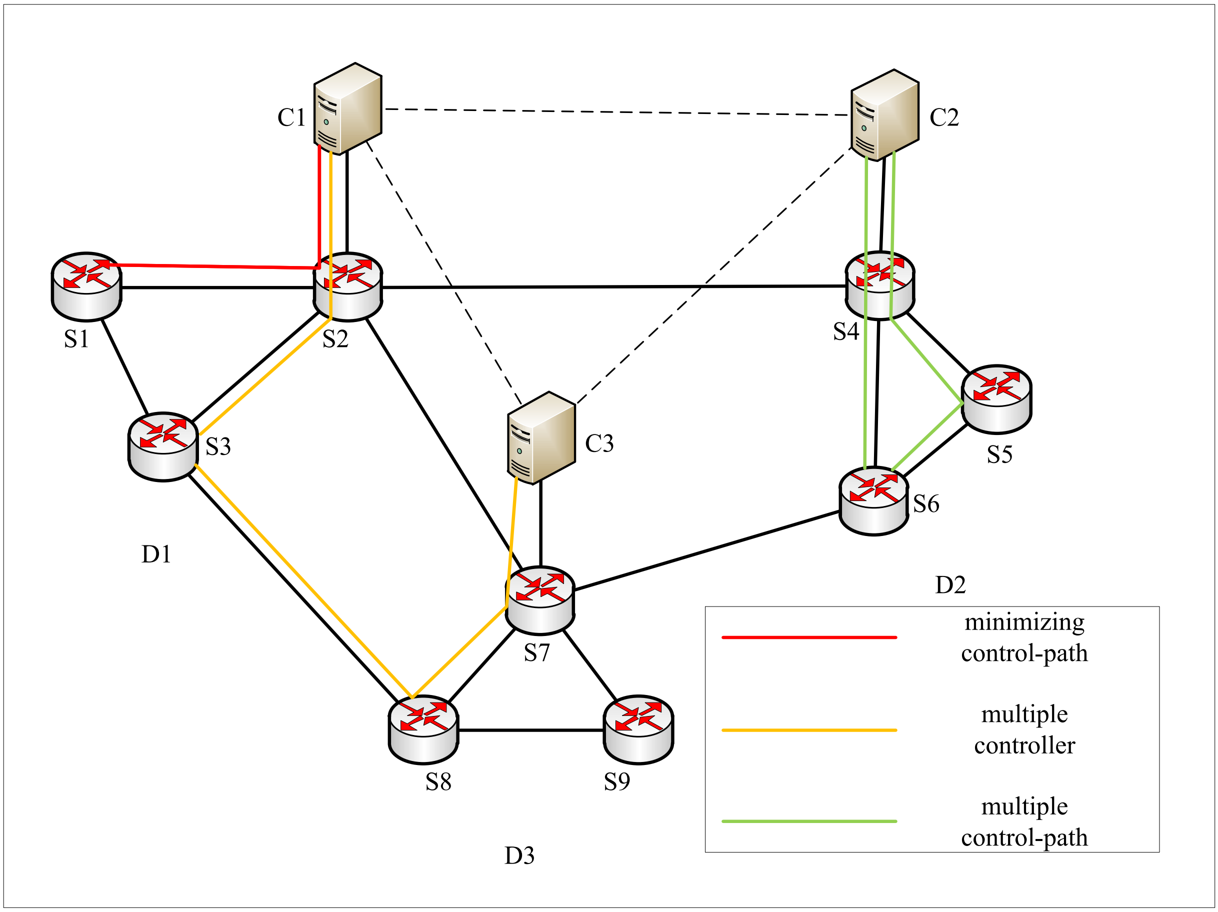


FIGURE 4. Three types of reliability.

performance degradation, it is of great importance to consider the reliability of SDN network when deploying controllers [42], [43]. The management and control SDN messages are transmitted through the control path, so the reliability of the control path directly affect the reliability of the SDN.

A complete control path consists of nodes, links, and controllers. For controller failure, switches have to be connected to multiple different controllers to avoid single point of failure. And for node failures and link failures, the first approach considers that switches connected to a controller over two disjoint control paths and the second method attempts to reduce the length of the control path which consist of fewer network elements.

After analysis, we categorize the method of improving reliability into 3 aspects: (1) multiple control-path, (2) multiple controller, and (3) minimizing control-path, which are shown in figure 4.

A. MULTIPLE CONTROL-PATH

Multiple control-path means at least two disjoint paths connecting switch and controller, which protect the control path against single link and node failures by switching to an alternate path.

p denotes the probability of failure. $P(s, v)$ is the probability that the control path is available. $mcp(v, s)$ is the number of disjoint paths between node v and controller c. The mathematical expression is as shown in Equation 5,6.

$$P(v, s) = \prod_{t \in d(v,s)} (1 - p_t) \tag{5}$$

$$R = \max \left(\sum_{s \in S} \sum_{v \in V} \left(1 - \prod_{P(v,s) \in mcp(v,s)} (1 - P(v, s)) \right) \right) \tag{6}$$

According to the application scenario, the existing representative algorithm are divided into two categories. In datacenters, Müller *et al.* [44] proposes Survivor to improving reliability by path diversity, however, path diversity makes traffic significantly increase, and Muller does not consider the capacity of the link; to address this problem, Vizaretta *et al.* [45] propose a link-capacity aware controller deployment method. in WANs, for reducing the computation complexity, a reliability factor is defined to replace the network average reliability in [46] while network partition is used in [47] to ensure rapid convergence to a near optimal placement.

TABLE 6. Comparison of literature of multiple control-path.

Year	Author	Method	Evaluation	Application Scenario	
				datacenter	WAN
2014	Muller et al. [44]	Integer Linear Programming	• the path diversity of the network (connectivity-awareness) reduces the probability of connectivity loss in around 66% for single link failures. • Adding capacity-awareness to the solution can decrease controller overloaded on both normal and failover scenarios.	√	×
2016	Vizaretta et al. [45]	Mixed Integer Linear Programming	• The path length increase is less than 2% for the model considering disjoint paths for most topologies.	√	×
2016	Liu et al. [46]	clustering algorithm	• Clustering based global optimization algorithm outperform the random algorithm and greedy algorithm based local optimization. • The performance of clustering algorithm wouldn't be affected by the number of controllers with multi-paths	×	√
2018	Moazzenі et al. [47]	Bully Algorithm	• The result show that the solution has a 51.82% improvement in failure recovery time compared to other research in the literature	×	√

1) SURVIVOR
Müller *et al.* [44] proposes an enhanced controller placement strategy (Survivor), which deal with three main features (connectivity, capacity, and recovery), to improve SDN reliability. First, connectivity between switch and controller is enhanced by explicitly increasing path diversity. Second, controller overload is avoided proactively by adding capacity-awareness mechanism in the controller placement. Third, failover mechanisms are improved by the method of composing a backup list.

2) MIXED INTEGER PROGRAMMING
In order to avoid the control plane being affected by single link and node failure, Vizaretta *et al.* [45] adopts two strategies to solve the problem of reliable controller placement, and provides seamless failover by utilizing the principle of flexible routing. The first method assumes that the switch must be connected to the controller through two disjoint control paths. The second method assumes that the switch must connect to two different controller replicas through two disjoint paths. Both methods find the minimum length of work and backup control paths for fast and efficient failover.

3) CLUSTERING ALGORITHM
For improving the network average reliability, Liu *et al.* [46] proposes the clustering based global optimization algorithm for shortest path and the local optimization algorithm based on greedy algorithm for multi-paths. For reducing the computation complexity, a Reliability Factor is defined to replace the network average reliability in multi-paths.

4) BULLY ALGORITHM
By adding network partitioning strategy, Moazzeni *et al.* [47] presents Reliable Distributed SDN to improve the reliability of SDNs with distributed controllers. A new formula for calculating the reliability of each subnet based on load is proposed, and the number and degree of nodes and the packet loss rate of the link are considered. The coordinator detects any nonactive controller and will decide which other controller is more appropriate to take over the subnetwork through the detection phase which consider reliability and distance between the failed subnet and the assigned new controller.

We investigate and analyze the controller placement techniques for multiple control-path. The results are presented in table 6.

B. MULTIPLE CONTROLLER
A controller, as a (logically) centralized control entity which performs fine-grained control on data plane and provide abstract view to application plane, is the Achilles' heel of SDN reliability since its failure would affect the normal operation of the entire network. Researchers have found that a switch connect to multiple controller could avoid single points of failure and obviously improve reliability.

The mathematical expression is as shown in Equation 7.

$$R = \max \sum_{v \in V} \left(1 - \prod_{s \in S} (1 - P(v, s))\right) \tag{7}$$

According to the application scenario, the existing representative algorithm are divided into two categories. In datacenters, Sridham *et al.* [48] proposes a switch-controller mapping scheme which assign flow requests to different controllers. In WANs, Li *et al.* [49] examine how to improve security through the Byzantine mechanism; and Ros and Ruiz [50], pays attention to how to obtain high reliability by adding backup controllers, but ignore capacity of controller; then Killi and Rao [51] proposes a controller deployment strategy taking capacity and failure of controller into account.

1) LINEAR PROGRAMMING

Sridharan [48] proposes an efficient switch-controller mapping scheme for distributed controller architecture in SDN. This scheme maps switches to multiple controllers and distributes flow setup requests between them to minimize flow setup time and meets elastic constraints which require a specified portion of the setup request on each switch to be unaffected by the controller.

2) GREEDY ALGORITHM

Li *et al.* [49] presents a secure SDN architecture, in which each switch is controlled by multiple controllers in cloud using Byzantine mechanism. This paper study a controller assignment problem to minimize the number of employed controllers while satisfying the security requirement of each switch, in terms of the required number of associated controllers and the maximum latency among them. The author introduces an efficient algorithm to solve the controller assignment with a good result ratio with the optimal assignment.

3) HEURISTIC ALGORITHM

Considering synchronous traffic between controllers is a major source of overhead in multi-controller deployments, and each controller connection consumes switch resources, Ros and Ruiz [50] introduces the Fault Tolerant Controller Placement to obtain high reliability with fewer controller. Then author present a heuristic algorithm to solve this problem with 124 publicly available topologies.

4) SIMULATED ANNEALING

Killi and Rao [51] proposes a controller layout strategy that not only considers the reliability and capacity of the controller, but also considers the controller's failure, which avoid the drastic increase in management intervention, delay, and disconnection. It. The goal of strategy is to minimize the maximum latency for all switches, from the switch to the nearest controller with sufficient capacity and the delay from the first reference controller to its closest controller, which is a mixed integer linear programming (MILP). In addition, authors propose a simulated annealing algorithm that effectively solves the problem of large-scale networks.

We investigate and analyze the controller placement techniques for multiple controller. The results are presented in table 7.

C. MINIMIZING CONTROL-PATH

Usually, the breakdown of the underlying physical components results in failure of network units (e.g., link and node), and these failures are fixed and independent of each other [16]. As a consequence, we could reduce the physical units contained in the control path to improve reliability. The mathematical expression is as shown in Equation 8.

$$R = \max \sum_{v \in V} \sum_{s \in S} P(v, s) \tag{8}$$

According to the application scenario, the existing representative algorithm are divided into two categories. In datacenters, Hu *et al.* [52] obtains the optimal solution by gradually optimization based on constraints of performance metrics; Jimenez *et al.* [53] points out choosing the shortest path cannot achieve global optimality, and build a deployment mechanism depends on the physical network characteristics and the network. In WANs, from the view of distance, Zhong *et al.* [54] considers the CPP problem as the minimum set coverage; considering resilience and Qos requirement, Tanha *et al.* [55] proposes a clique-based solution to reduce calculation time

1) GREEDY ALGORITHM

Hu *et al.* [52] presents expected percentage of control path loss to measure the reliability of the SDN control network. After formulating the reliability-aware controller placement problem, the paper proves its NP-hardness which cannot calculate the optimal solution of the problem in polynomial time. This paper adopts greedy algorithm with different parameters to solve this problem using real topologies.

2) K-CRITICAL

Since poor controller selection can seriously affect the robustness of the control network and the use of more than optimal controllers may be inefficient and costly, Jimenez *et al.* [53] devises a new reliability deployment mechanism that the optimal number of controllers depends on the physical network characteristics and the network. The authors demonstrate that choosing the shortest path between controller nodes is not the best choice for improving control layer load and robustness. Then, the authors propose k-critical to construct a robust control layer that responds to network interference to select the controller.

3) MIN-COVER

In order to ensure the reliability of the control network and meet the required propagation latency, Zhong *et al.* [54] presents a min-cover based controller placement approach. Two metrics for control path reliability are proposed,

TABLE 7. Comparison of literature of multiple controllers.

Year	Author	Method	Evaluation	Application Scenario	
				Data Center	WAN
2016	Sridharam et al. [48]	Linear programming	• The method reduces flow setup time and provides resilience in the event of a single controller failure. • In dynamic traffic conditions, the proposed method is at least three times more reliable than the single mapping method.	√	×
2014	Li et al. [49]	Greedy algorithm	• The byzantine structure has little impact on the network latency, provide better security than general distributed controller.	×	√
2016	Ros et al. [50]	Heuristic algorithm	• For getting five nines reliability, each node is required to connect to just 2 controllers. • for 75% of the topology, eight controllers or less are sufficient	×	√
2017	Killi et al. [51]	Simulated annealing	• The proposed method performs better than CCP[41] in case of failures • The solution is able to achieve near optimal solutions in less than half of the time required by the optimized formulations.	×	√

and the definitions of neighborhood and minimum coverage are given.

4) CLIQUE-BASED APPROACH

Considering resilience and Qos requirement, Tanha *et al.* [55] proposes a clique-based solution to solve this problem that solution takes both the switch-controller/inter-controller latency requirements and the capacity of the controllers into account to meet the traffic load of switches. In addition, author proposed a polynomial-time algorithm to solve this NP-hard problem, in a view of the structure of the problem using cliques by finding the maximal clique to reduce the search space.

We investigate and analyze the controller placement techniques for minimizing control-path. The results are presented in table 8.

V. COST

Considering the two aspects of economy and environmental protection, the cost consumption of the network should no longer be ignored with the rapid development of the Internet [56]. The cost of the SDN mainly includes the previous network construction and the later operation and maintenance costs. So, we classify the existing solution of cost into two aspects: (a) deployment cost, and (b) energy consumption.

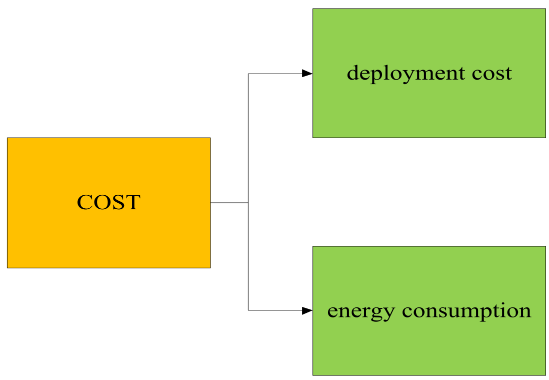


FIGURE 5. Two types of cost.

A. DEPLOYMENT COST

Deployment cost refer to the cost of network devices (controllers and switches) and their operational expenses, including installing the controller into the network, linking the controller to the switch, and linking these controllers. The mathematical expression is as shown in Equation 9.

$$C = \min(C_s + C_l + C_t) \tag{9}$$

where C_s means the cost of controllers, C_l means the cost of connecting switches and controllers, and C_t means the cost of controller interconnection.

TABLE 8. Comparison of literature of minimizing control-path.

Year	Author	Method	Evaluation	Application Scenario	
				Data Center	WAN
2014	Hu et al. [52]	Greedy algorithm	• Placing too many or too few controllers reduces reliability • The corresponding latencies when optimizing for reliability is sufficient to meet existing response-time requirements	√	×
2014	Jimenez et al. [53]	K-Critical	• From the result of building tree, the shortest trees have the lowest data loss, • Five controllers are enough for the sparse networks.	√	×
2016	Zhong et al. [54]	Min-cover	• For min-cover based approach, the total amount of disconnected switches is stable, and is less than 40 when the failure probability increase. • The approach has steadily good performance in networks of different scales and connectivity.	×	√
2018	Tanha et al. [55]	Clique-based approach	• The average controller utilization is around 80–90% against single or two controller failures	×	√

According to the application scenario, the existing representative algorithm are divided into two categories. In datacenter, Sallahi and St-Hilaire [20] adopts inter linear programming for CPP, however this method is only suitable for pure SDN environments; then Sallahi and St-Hilaire [57] adds incremental adjusting to original method for hybrid network. In WANs, in order to speed up the solution, Mostafa's method [58] choose the type and number of controllers firstly, and then determine the specific location.

1) INTER LINEAR PROGRAMMING

Given a set of switches that must be managed by controller, Sallahi and St-Hilaire [20] presents a mathematical model for the controller placement problem which simultaneously determines the optimal number, location, and type of controller(s) as well as the interconnections between all the network elements. The goal of the model is to minimize the cost of the network while considering different constraints. However, the proposed model could be only used by various enterprises and cloud-based networks

2) LINEAR PROGRAMMING

Considering that previous work could not meet large-scale network requirements, Sallahi and St-Hilaire [57] use the mathematical model proposed in [20] as a starting point and modify it so that expansion scenarios can be considered. which means adding switch and controller to existing network. The model is designed to minimize the cost of this reorganization while ensuring that the solution found is feasible

3) HEURISTIC ALGORITHM

In [58], a framework is introduced which investigates the controller placement model with more realistic assumptions in two phases. Phase I determines the required number of controllers to purchase and deploy in the network, considering various controller types, costs and the structure of underlying topologies. Phase II tries to solve the controller location-allocation problem regarding two critical objective function, called the fair load distribution and inter-controller latency. Two powerful algorithms which inherit interesting properties such as greedy mechanisms, heuristic attributes and randomized procedures are proposed to solve efficiently two phases' problem.

We investigate and analyze the controller placement techniques for deployment cost. The results are presented in table 9.

B. ENERGY CONSUMPTION

The energy consumption of network devices under low network load still accounts for more than 90% of the load during busy hours. And a reasonable controller deployment solution could shut down links and controllers as much as possible when the network is idle. The mathematical expression is as

TABLE 9. Comparison of literature of deployment cost.

Year	Author	Method	Simulation	Application Scenario	
				datacenter	WAN
2015	Sallahi et al. [20]	Inter linear programming	• The results show that only small size problems can be optimized within a reasonable time. • Approximately 10% of the problems cannot be solved within the time limit of 30 hours.	√	×
2017	Sallahi et al. [57]	Linear programming	• Adding an extra controller usually requires 5 additional links. • The computational time is 4.60 ms in the proposed method.	√	√
2018	Mostafa et al. [58]	Heuristic algorithm	• When the upper limit of latency is 12s, at least 35 controllers are needed, and when the upper limit of latency is 7s, the number of controllers is increased to 40.	×	√

shown in Equation 10.

$$C = \sum_{v \in V} w(v) f(v, s) \tag{10}$$

where $w(v)$ is number of packets. And $f(v, s)$ means the overhead of communication between v and s.

According to the application scenario, the existing representative algorithm are divided into two categories. In datacenters, Ruiz-Rivera *et al.* [19] reduces the number of active links to save energy, however rerouting of control paths could cause controller overload; so Hu *et al.* [59] proposes a binary integer program to solve CPP which considering control-path latency and controller's load. In WANs, Fernández-Fernández [60] uses heuristic algorithm to get initial value and optimize them by greedy algorithm for reducing compute complexity.

1) BINARY INTEGER PROGRAMMING

Different like, Ruiz-Rivera *et al.* [19] consider the problem of reducing the number of active links subject to the following constraints: latency between switches and controllers, link utilization and controller load. Considering the Maximum Link Utilization (MLU) when routing demands, the author develop a Binary Integer Program (BIP) to derive the optimal solution for this problem.

2) GENETIC ALGORITHM

Considering that the position of the controller also affects the energy consumption of the network, Hu *et al.* [59] study the controller placement problem from the perspective of energy consumption. This energy-aware problem is modeled as a binary integer program (BIP) that the energy consumption of the network is minimized under the constraints of control path delay and controller load.

3) GREEDY ALGORITHM

For large network topologies, Fernández-Fernández [60] propose a novel energy-aware mechanism that reduce the time complexity of approach and allow for real-time distribution of traffic demands. In this mechanism, two solution modules are conceived using knowledge of network topology and traffic engineering techniques, to reduce overall power consumption. In addition, the proposed greedy algorithms are able to converge much faster and handle larger network sizes

We investigate and analyze the controller placement techniques for energy consumption. The results are presented in table 10.

VI. MULTI-OBJECTS

In the actual application scenario of SDN, multiple performance metrics are usually selected to solve CPP at the same time, so the CPP problem can also be used as a multi-objective optimization problem [61]–[63]. Since multiple metrics cannot achieve the best synchronously, such as conflicts between energy savings and network performance, the difficulty of multi-objective optimization is how to make a reasonable trade-off between multiple performance metrics. The mathematical expression is as shown in Equation 11.

$$M = \max[L, R, C] \tag{11}$$

where L means latency, R donates reliability, and C represents cost.

According to the application scenario, the existing representative algorithm are divided into two categories. Borcoci *et al.* [61] examined how to solve the CPP problem using multi-criteria optimization algorithms, however there are no specific experimental analysis; Ksentini [37] proposed Bargaining Game, which quantify different metrics into tradable units, to solve the multi-objective optimization problem

TABLE 10. Comparison of literature of energy consumption.

Year	Author	Method	Simulation	Application Scenario	
				datacenter	WAN
2015	Ruiz-Rivera et al. [19]	Binary integer program	• Energy savings of up to 55% during off peak times	√	×
2017	Hu et al. [59]	genetic algorithm	• the energy consumption of the heuristic algorithm is close to optimal solution. • If all links have the same energy consumption, the genetic algorithm uses no more than 4% of the additional links.	√	×
2017	Fernández-Fernández et al. [60]	Greedy algorithm	• Energy-aware methods can save about 50% to 80% of energy	×	√

efficiently. In WANs, in order to improve the solving speed, Ahmadi and Khorramizadeh [62] adopts genetic algorithm which enjoy a greedy algorithm to generate a high-quality initial population and Santos *et al.* [63] use Adaptive Bacterial Foraging Optimization which customize the chemotaxis steps of ABFO to perform three variations in each iteration time

A. MULTI-CRITERIA DECISION ALGORITHMS

By analyzing some solutions for propagation latency, load and failure, Borcoci *et al.* [61] demonstrates how multi-criteria optimization algorithms can be applied to controller placement problem. The goal is to achieve overall controller placement optimization by applying a multi-standard decision algorithm (MCDA), where the input of the MCDA is a set of candidates (candidate means an instance of controller placement). However, the author only proposed that no specific experimental analysis was given.

B. BARGAINING GAME

Considering three critical objectives of controller placement: (i) the latency and communication overhead between switches and controllers; (ii) the latency and communication overhead between controllers; (iii) the guarantee of load balancing between controllers, Ksentini *et al.* [37] proposes Bargaining Game, which quantify different metrics into tradable units, to solve the multi-objective optimization problem efficiently.

C. MULTI-START HYBRID NON-DOMINATED SORTING GENETIC ALGORITHM (MHNSGA)

Ahmadi and Khorramizadeh [62] presents a heuristic algorithm called Multi-Start Hybrid Non-dominated Sorting Genetic Algorithm (MHNSGA) to solve the multi-objective controller placement problem effectively. The presented algorithm requires reasonable memory resource and enjoys a greedy heuristic to generate a high-quality initial population, smart mechanisms to encourage the diversification and intensification, and a new fast Pareto finder. Moreover, a new variant of the problem is developed in which the capacities of controllers and loads of switches are added as constraints.

D. ADAPTIVE BACTERIAL FORAGING OPTIMIZATION

Zhang *et al.* [63] uses Adaptive Bacterial Foraging Optimization (ABFO) algorithm to solve multi-objective optimization controller placement. The authors customize the chemotaxis steps of ABFO to perform three variations in each iteration time, namely, changes in controller position, changes in the mapping relationship between the controller and switch, and change of the proportion of routing requests from switches processed by the controllers.

We investigate and analyze the controller placement techniques for multi-objects. The results are presented in table 11.

VII. FUTURE WORK

With the gradual deployment and application of SDN in the network, CPP has become a realistic and meaningful research hotspot. The existing research focuses on performance metrics such as latency, reliability, cost, and multi-objective. 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. Although a series of solutions have been proposed for this problem, there are still many key issues to be solved. Next, we will discuss the future research priorities and development trends.

A. EFFICIENT ALGORITHM

The controller placement problem is a variant of facility location, which is known as NP-hard, and solving this problem is very time-consuming. The current main solution is to reduce the search space by network partitioning or find a feasible result by heuristic algorithm. However, it is still a challenging problem considering compute resource limitation and large-scale network. In the future, the research on efficient algorithms mainly focuses on online search algorithms, and the goal is to find approximate optimal solutions in a short time.

TABLE 11. Comparison of literature of multi-objects.

Year	Author	Late-ncy	Relia bility	Cost	Method	Simulation	Application Scenario datacenter	WAN
2015	Borcoci et al. [61]	√	√	×	Multi-criteria optimization algorithms	• Only proposed the theoretical framework without specific experiments.	√	√
2016	Ksentini et al. [37]	√	√	×	Bargaining Game	• Bargaining game ensure a better trade-off than a single-target solution.	√	√
2018	Ahmadi et al. [62]	√	√	×	Multi-start hybrid non-dominated sorting genetic algorithm (MHNSGA)	• The average deviation of the MHNSGA and the original Pareto optimal set is 0.8%. • With more than 14 million search spaces, MHNSGA is 20 times faster than the POCO framework.	×	√
2018	Zhang et al. [63]	√	√	√	Adaptive bacterial foraging optimization algorithm (ABFO)	• The reliability probability of heuristic [50] is about 0.199% higher than PSA [30] and 0.015% higher than ABFO. • ABFO's load balancing factor is about 96.908% lower than the heuristic, respectively, and the average is lower than 85.931% of PSA. • ABFO's worst-case latency is about 36.114% lower than the heuristic, and the average is 15.274% lower than the PSA.	×	√

B. COST-AWARENESS

Recently, some scholars have begun to pay attention to the cost of controller deployment, but the trade-off between network performance (latency, reliability, etc.) and cost has not been fully evaluated. For example, closing a link at low traffic conditions can save energy, but this operation will inevitably bring some costs. Therefore, in-depth research is needed on the cost issue.

C. VIRTUALIZED CPP

Traditionally, CPP problems have been limited to optimally assigning switches to static physical controllers and placing these controllers on a network topology. However, with the development of NFV technology, virtual controllers can significantly increase the dynamics of network control (Virtualized control plane placement problem: provisioning the control paths and architectures). In this case, it is also important to supply the controller's own requirements as a virtual entity. Future research will need to consider how to configure virtual control charts and additional virtualization support functions such as databases to form a complete distributed virtual control plane.

D. ATTACK-AWARENESS

Due to the rising risk of large human-made security attacks, improving the preparedness of networks to such attacks is becoming a key issue. By studying the importance of nodes in different network topologies, Santos *et al.* [64] look for the optimal controller deployment location to improve the robustness of the network under targeted attacks, but the current research is important to define nodes. The lack of reasonable basis, experimental verification data homogenization and other issues. In the future, the CPP should be studied from the actual network and combined with the actual attack mode to guide the controller deployment of the real network scenario.

VIII. CONCLUSION

Controller placement problem is a critical hot topic in SDN. Efficient controller placement tries to improve performance metrics, such as latency, reliability, cost and so on. Therefore, the classification of CPP can intuitively understand the concerns of different solutions. To the best of our knowledge, this is the first state-of-the-art review on CPP from the perspective of optimized objective. Meanwhile, we also consider the corresponding solution with methods, simulation and application scenario. Further, we give some promising research problems of controller placement problem in the future.

REFERENCES

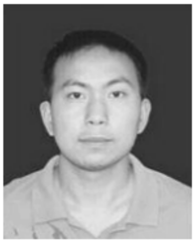
[1] H.-C. Wang and H.-S. Doong, "Validation in Internet survey research: Reviews and future suggestions," in *Proc. 40th Annu. Hawaii Int. Conf. Syst. Sci. (HICSS)*, Jan. 2007, p. 243.

[2] D. Kreutz, F. Ramos, P. E. Veríssimo, C. E. Rothenberg, S. Azodolmolky, and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proc. IEEE*, vol. 103, no. 1, pp. 14–76, Jan. 2015.
[3] B. A. A. Nunes, M. Mendonca, X.-N. Nguyen, K. Obraczka, and T. Turletti, "A survey of software-defined networking: Past, present, and future of programmable networks," *IEEE Commun. Surveys Tuts.*, vol. 16, no. 3, pp. 1617–1634, 3rd Quart., 2014.
[4] N. McKeown, "Software-defined networking," *INFOCOM Keynote Talk*, vol. 17, no. 2, pp. 30–32, 2009.
[5] Y.-D. Lin, P.-C. Lin, C.-H. Yeh, Y.-C. Wang, and Y.-C. Lai, "An extended SDN architecture for network function virtualization with a case study on intrusion prevention," *IEEE Netw.*, vol. 29, no. 3, pp. 48–53, May 2015.
[6] J. Liu, Z. Jiang, N. Kato, O. Akashi, and A. Takahara, "Reliability evaluation for NFV deployment of future mobile broadband networks," *IEEE Wireless Commun.*, vol. 23, no. 3, pp. 90–96, Jun. 2016.
[7] N. McKeown *et al.*, "OpenFlow: Enabling innovation in campus networks," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 38, no. 2, pp. 69–74, Apr. 2008.
[8] Z. Guo *et al.*, "Improving the performance of load balancing in software-defined networks through load variance-based synchronization," *Comput. Netw.*, vol. 68, pp. 95–109, Aug. 2014.
[9] Z. Guo *et al.*, "STAR: Preventing flow-table overflow in software-defined networks," *Comput. Netw.*, vol. 125, pp. 15–25, Oct. 2017.
[10] A. Tootoonchian and Y. Ganjali, "HyperFlow: A distributed control plane for openflow," in *Proc. Internet Netw. Manage. Conf. Res. Enterprise Netw.*, 2010, p. 3.
[11] S. H. Yeganeh and Y. Ganjali, "Kandoo: A framework for efficient and scalable offloading of control applications," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, 2012, pp. 19–24.
[12] T. Koponen *et al.*, "Onix: A distributed control platform for large-scale production networks," in *Proc. OSDI*, 2010, pp. 1–6.
[13] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proc. 1st Workshop Hot Topics Softw. Defined Netw.*, 2012, pp. 7–12.
[14] P. Xiao, W. Qu, H. Qi, Z. Li, and Y. Xu, "The SDN controller placement problem for WAN," in *Proc. IEEE/CIC Int. Conf. Commun. China (ICCC)*, Oct. 2014, pp. 220–224.
[15] H. Aoki and N. Shinomiya, "Controller placement problem to enhance performance in multi-domain SDN networks," in *Proc. ICN*, 2016, p. 2016.
[16] Y. Hu, W. Wendong, X. Gong, X. Que, and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *Proc. IFIP/IEEE Int. Symp. Integr. Netw. Manage. (IM)*, May 2013, pp. 672–675.
[17] Y. Zhang, N. Beheshti, and M. Tatipamula, "On resilience of split-architecture networks," in *Proc. IEEE Global Telecommun. Conf. (GLOBECOM)*, Dec. 2011, pp. 1–6.
[18] Y.-N. Hu, W.-D. Wang, X.-Y. Gong, X.-R. Que, and S.-D. Cheng, "On the placement of controllers in software-defined networks," *J. China Univ. Posts Telecommun.*, vol. 19, pp. 92–171, Oct. 2012.
[19] A. Ruiz-Rivera, K. W. Chin, and S. Soh, "GreCo: An energy aware controller association algorithm for software defined networks," *IEEE Commun. Lett.*, vol. 19, no. 4, pp. 541–544, Apr. 2015.
[20] A. Sallahi and M. St-Hilaire, "Optimal model for the controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 19, no. 1, pp. 30–33, Jan. 2015.
[21] A. K. Singh and S. Srivastava, "A survey and classification of controller placement problem in SDN," *Int. J. Netw. Manage.*, vol. 28, no. 3, p. e2018, 2018.
[22] Y. Zhang, L. Cui, W. Wang, and Y. Zhang, "A survey on software defined networking with multiple controllers," *J. Netw. Comput. Appl.*, vol. 103, pp. 101–118, Feb. 2018.
[23] T. Hu, Z. Guo, P. Yi, T. Baker, and J. Lan, "Multi-controller based software-defined networking: A survey," *IEEE Access*, vol. 6, pp. 15980–15996, 2018.
[24] G. Wang, Y. Zhao, J. Huang, and W. Wang, "The controller placement problem in software defined networking: A survey," *IEEE Netw.*, vol. 31, no. 5, pp. 21–27, Sep./Oct. 2017.
[25] G. Wang, Y. Zhao, J. Huang, and Y. Wu, "An effective approach to controller placement in software defined wide area networks," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 1, pp. 344–355, Mar. 2018.
[26] J. F. Kurose, *Computer Networking: A Top-Down Approach Featuring the Internet*. London, U.K.: Pearson, 2005.
[27] T. Zhang, P. Giaccone, A. Bianco, and S. De Domenico, "The role of the inter-controller consensus in the placement of distributed SDN controllers," *Comput. Commun.*, vol. 113, pp. 1–13, Nov. 2017.
[28] 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.
[29] M. He, A. Basta, A. Blenk, and W. Kellerer, "Modeling flow setup time for controller placement in SDN: Evaluation for dynamic flows," in *Proc. IEEE Int. Conf. Commun. (ICC)*, Jul. 2017, pp. 1–7.
[30] Gurobi Optimization. (2015). *Gurobi Optimizer Reference Manual*. [Online]. Available: http://www.gurobi.com
[31] S. Lange *et al.*, "Heuristic approaches to the controller placement problem in large scale SDN networks," *IEEE Trans. Netw. Service Manag.*, vol. 12, no. 1, pp. 4–17, Mar. 2015.
[32] 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. Netw.*, vol. 112, pp. 24–35, Jan. 2017.
[33] 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 *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
[34] L. Liao and V. C. Leung, "Genetic algorithms with particle swarm optimization based mutation for distributed controller placement in SDNs," in *Proc. IEEE Conf. Netw. Function Virtualization Softw. Defined Netw. (NFV-SDN)*, May 2017, pp. 1–6.
[35] B. P. R. Killi, E. A. Reddy, and S. V. Rao, "Cooperative game theory based network partitioning for controller placement in SDN," in *Proc. 10th Int. Conf. Commun. Syst. Netw. (COMSNETS)*, Jan. 2018, pp. 105–112.
[36] K. S. Sahoo, S. Sahoo, A. Sarkar, B. Sahoo, and R. Dash, "On the placement of controllers for designing a wide area software defined networks," in *Proc. IEEE Region 10 Conf. (TENCON)*, Nov. 2017, pp. 3123–3128.
[37] A. Ksentini, M. Bagaa, T. Taleb, and I. Balasingham, "On using bargaining game for optimal placement of SDN controllers," in *Proc. IEEE Int. Conf. Commun. (ICC)*, May 2016, pp. 1–6.
[38] G. Ishigaki, R. Gour, A. Yousefpour, N. Shinomiya, and J. P. Jue, "Cluster Leader Election Problem for Distributed Controller Placement in SDN," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2017, pp. 1–6.
[39] L. Shaoteng, R. Steinert, and D. Kostic, "Flexible distributed control plane deployment," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Jun. 2018, pp. 1–7.
[40] G. Yao, J. Bi, Y. Li, and L. Guo, "On the capacitated controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 18, no. 8, pp. 1339–1342, Aug. 2014.
[41] S. Jiugen, Z. Wei, J. Kunying, and X. Ying, "Multi-controller deployment algorithm based on load balance in software defined network," *J. Electron. Inf. Technol.*, vol. 40, no. 2, pp. 455–461, 2018.
[42] G. Yao, J. Bi, and L. Guo, "On the cascading failures of multi-controllers in software defined networks," in *Proc. 21st IEEE Int. Conf. Netw. Protocols (ICNP)*, Oct. 2013, pp. 1–2.
[43] S. Scott-Hayward, "Design and deployment of secure, robust, and resilient SDN Controllers," in *Proc. 1st IEEE Conf. Netw. Softw. (NetSoft)*, Apr. 2015, pp. 1–5.
[44] L. F. Müller, R. R. Oliveira, M. C. Luizelli, L. P. Gaspary, and M. P. Barcellos, "Survivor: An enhanced controller placement strategy for improving SDN survivability," in *Proc. IEEE Global Commun. Conf. (GLOBECOM)*, Dec. 2014, pp. 1909–1915.
[45] P. Vizarreta, C. M. Machuca, and W. Kellerer, "Controller placement strategies for a resilient SDN control plane," in *Proc. 8th Int. Workshop Resilient Netw. Design Modeling (RNDM)*, Sep. 2016, pp. 253–259.
[46] J. Liu, J. Liu, and R. Xie, "Reliability-based controller placement algorithm in software defined networking," *Comput. Sci. Inf. Syst.*, vol. 13, no. 2, pp. 547–560, 2016.
[47] S. Moazzeni, M. R. Khayyambashi, N. Movahhedinia, and F. Callegati, "On reliability improvement of software-defined networks," *Comput. Netw.*, vol. 133, pp. 195–211, Mar. 2018.
[48] V. Sridharan, M. Gurusamy, and T. Truong-Huu, "On multiple controller mapping in software defined networks with resilience constraints," *IEEE Commun. Lett.*, vol. 21, no. 8, pp. 1763–1766, Aug. 2017.
[49] H. Li, P. Li, S. Guo, and A. Nayak, "Byzantine-resilient secure software-defined networks with multiple controllers in cloud," *IEEE Trans. Cloud Comput.*, vol. 2, no. 4, pp. 436–447, Oct. 2014.
[50] F. J. Ros and P. M. Ruiz, "On reliable controller placements in software-defined networks," *Comput. Commun.*, vol. 77, pp. 41–51, Mar. 2016.
[51] B. P. R. Killi and S. V. Rao, "Capacitated next controller placement in software defined networks," *IEEE Trans. Netw. Service Manage.*, vol. 14, no. 3, pp. 514–527, Sep. 2017.

[52] Y. Hu, W. Wang, X. Gong, X. Que, and S. Cheng, "On reliability-optimized controller placement for software-defined networks," *China Commun.*, vol. 11, no. 2, pp. 38–54, Feb. 2014.
[53] Y. Jimenez, C. Cervello-Pastor, and A. J. Garcia, "On the controller placement for designing a distributed SDN control layer," in *Proc. Netw. Conf. (IFIP)*, 2014, pp. 1–9.
[54] Q. Zhong, Y. Wang, W. Li, and X. Qiu, "A min-cover based controller placement approach to build reliable control network in SDN," in *Proc. IEEE/IFIP Netw. Oper. Manage. Symp. (NOMS)*, Apr. 2016, pp. 481–487.
[55] M. Tanha, D. Sajjadi, R. Ruby, and J. Pan, "Capacity-aware and Delay-guaranteed Resilient Controller Placement for Software-Defined WANs," *IEEE Trans. Netw. Service Manage.*, vol. 15, no. 3, pp. 991–1005, Sep. 2018.
[56] M. Gupta and S. Singh, "Greening of the Internet," in *Proc. Conf. Appl., Technol., Archit., Protocols Comput. Commun.*, 2003, pp. 19–26.
[57] A. Sallahi and M. St-Hilaire, "Expansion model for the controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 21, no. 2, pp. 274–277, Feb. 2017.
[58] M. Khorramizadeh and V. Ahmadi, "Capacity and load-aware software-defined network controller placement in heterogeneous environments," *Comput. Commun.*, vol. 129, pp. 226–247, Sep. 2018.
[59] Y. Hu, T. Luo, N. C. Beaulieu, and C. Deng, "The energy-aware controller placement problem in software defined networks," *IEEE Commun. Lett.*, vol. 21, no. 4, pp. 741–744, Apr. 2017.
[60] A. Fernández-Fernández, C. Cervelló-Pastor, and L. Ochoa-Aday, "Energy efficiency and network performance: A reality check in SDN-based 5G systems," *Energies*, vol. 10, no. 12, p. 2132, 2017.
[61] E. Borcoci, R. Badea, S. G. Obreja, and M. Vochin, "On multi-controller placement optimization in software defined networking-based wans," in *Proc. ICN*, 2015, p. 273.
[62] V. Ahmadi and M. Khorramizadeh, "An adaptive heuristic for multi-objective controller placement in software-defined networks," *Comput. Elect. Eng.*, vol. 66, pp. 204–228, Feb. 2018.
[63] 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.
[64] D. Santos, A. de Sousa, and C. M. Machuca, "Robust SDN controller placement to malicious node attacks," in *Proc. 21st Conf. Innov. Clouds, Internet Netw. Workshops (ICIN)*, Feb. 2018, pp. 1–8.

JIE LU received the bachelor's degree from Zhejiang University, in 2016. He is currently pursuing the master's degree with the National Digital Switching System Engineering and Technological Research and Development Center, Zhengzhou, China. His research interests include software-defined networking, network management, and network security.

ZHEN ZHANG is currently a Lecturer with the National Digital Switching System Engineering and Technological Research and Development Center. His research interests include the aspects of network measurement and network management.



TAO HU received the bachelor's and master's degrees from Xi'an Jiaotong University, in 2015. He is currently pursuing the Ph.D. degree with the National Digital Switching System Engineering and Technological Research and Development Center, Zhengzhou, China. His research interests include software-defined networking, DDos, and network security.

PENG YI is currently a Professor with the National Digital Switching System Engineering and Technological Research and Development Center. His research interests include smart grids and wireless networks.

JULONG LAN is currently a Professor with the National Digital Switching System Engineering and Technological Research and Development Center. His research interests include big data, network security, and wireless networks.