IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received February 25, 2019, accepted March 15, 2019, date of publication March 21, 2019, date of current version April 8, 2019.

Digital Object Identifier 10.1109/ACCESS.2019.2906683

SDN-Based Load Balancing Scheme for Multi-Controller Deployment

GUOYAN LI[1], XINQIANG WANG[2], AND ZHIGANG ZHANG[1]
[1]School of Computer and Information Engineering, Tianjin Chengjian University, Tianjin 300384, China
[2]School of Software and Communication, Tianjin Sino-German University of Applied Sciences, Tianjin 300350, China

Corresponding authors: Guoyan Li (ligy@tcu.edu.cn) and Xinqiang Wang (waffchiang@163.com)

This work was supported in part by the Project of Natural Science Foundation of Tianjin under Grant 17JCQNJC00500, and in part by the Project of Tianjin Science and Technology Correspondent under Grant 17JCTPJC50600 and Grant 18JCTPJC61800.

ABSTRACT Software-defined networking (SDN) separates the control plane from the data forwarding plane and realizes the flexible management of the network resources. With the explosive growth of network traffic and scale, multi-controllers need to be deployed to improve the scalability and reliability of the control plane. However, unreasonable subdomain partitioning of SDN controllers may cause the unbalanced distribution of controller loads and reduces the communication performance of the network. Therefore, in this paper, a dynamic multi-controller deployment scheme based on load balancing is proposed. We transform the flow requests into a queuing model and consider the traffic propagation delay and the capacity of controllers as two main factors affecting the deployment of the multi-controllers. In the initial static network, a modified affinity propagation algorithm (PSOAP) based on particle swarm optimization is proposed to solve the problem of clustering performance being affected by the initial values of the bias parameters and convergence coefficients, getting the reasonable network planning. With the dynamic traffic network, switches in different sub-domains are reassigned by breadth-first search (BFS) algorithm to achieve controller load balancing. The extensive evaluations demonstrate that the scheme can provide better stable, accurate, and load balancing multi-controller deployment when compared with affinity propagation (AP) and genetic algorithms.

INDEX TERMS Affinity propagation algorithm, load balancing, queuing model, software-defined networking.

I. INTRODUCTION

Software Definition Networking (SDN) is a novel network architecture which separates control plane from data forwarding plane. It can control the network traffic by application program interfaces and open program interfaces [1]. It provides a new solution for researching new network applications and future Internet technologies.

The traditional SDN implementation relying on a logically centralized controller has several limitations related to scalability and performance. With the explosive growth of internet traffic and scale, particularly when network devices are widely distributed across regions, the network scale that a single controller can support is limited. In order to avoid the problem of low network performance and single point failure caused by overburdened single controller, multiple controllers are usually deployed in network to realize the distributed control management. Multi-controller divides the control plane into several sub-domains, each controller only needs to manage the switches in its own domain. So it can alleviate the deficiencies of the control plane in reliability, scalability and versatility.

For a given network topology, the problem of controller deployment needs further research, such as how to determine the number of controllers and which switches are managed by each controller. The mapping relationship between controller and switch determines the performance of the network [2]. When the network traffic changes abruptly, the load of the controllers may be in an unbalanced state which reduces the processing capacity of the high-load controller for data flow requests, while the low-load controller resources can't be fully utilized. Hence, it is important to set the optimal number and location of controllers according to network topology.

At present, some scholars have studied the problem of multi-controller deployment in SDN. In order to minimize the propagation latency between switch and associated

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

controller, Zhao *et al.* [3] proposed a modified clustering algorithm based on affinity propagation. Compared with the traditional clustering algorithms, it can achieve better network performance. Heller *et al.* [4] first proposed load balancing algorithm for controllers. It considers the average delay and the worst delay between the switch and the controller, and solves the deployment problem between controller and switch by optimizing model. Sallahi and St-Hilaire [5] proposed a complete model for the multi-controller deployment problem based on the deployment cost, but there is no algorithm in the literature. Ishigaki and Shinomiya [6] proposed a node calculation index of the pressure center, and the controller deployment algorithm based on the center was given in the paper. In view of the robustness of the controller cooperative control, Jimenez *et al.* [7] proposed a method of minimum number of controllers and deployment planning based on K-Critical algorithm. Xiao *et al.* [8] introduced a k-means algorithm to deploy SDN controllers. The algorithm is run over only one area at the beginning and then iterated by increasing the number of partitions. Wang *et al.* [9] proposed load balancing method based on greed method.

Some researchers have studied the capacity of the controller. For example, a multi-controller deployment algorithm based on capacity of the controller was proposed in [10]. Particle swarm optimization (PSO) was introduced to solve the controller deployment problem of an SDN in [11]. The optimization objective is to minimize the propagation latency between the controller and switch. The capacity limitation of the controller is also considered. Liu *et al.* [12] proposed a network clustering PSO algorithm, which takes into account the load of controllers and propagation latency.

In terms of reliable SDN deployment, Beheshti and Zhang [13] studied the resiliency in SDN, and regarded the resiliency of the connection between the controller and the switch as a performance evaluation index. On this basis, the controller deployment problem of network resiliency optimization was defined, and the corresponding heuristic algorithm was proposed to search the optimal placement of the controller. In order to improve SDN survivability, Muller *et al.* [14] proposed a controller placement strategy which considered the path diversity, capacity of the controller, and failover mechanisms at network design time. Yan-Nan *et al.* [15] and Hu *et al.* [16], [17] studied the problem of multi-controller deployment to maximize the reliability in SDN, and proposed a measurement and deployment algorithm of SDN reliability. Guo and Bhattacharya [18] presented the measurement of SDN reliability and gave the deployment of controller based on closeness center. Some researchers have considered multiple optimization objectives as references for deployment. For example, considering various network failures, a Pareto-based optimal controller deployment framework POCO is proposed in [19]. Then, a dynamic deployment method based on the Pareto optimal controller was proposed [20] and a heuristic algorithm based on Pareto simulated annealing was proposed [21]. Ahmadi *et al.* [22] proposed a heuristic algorithm called hybrid NSGA-II, which can get faster computation times and need much less memory to perform. Jalili *et al.* [23] considered the latency between nodes and load balancing as important metrics, NSGA-II was introduced to solve the multi-objective model of control placement problem. Then a dynamic switch migration based on multi-objective optimization was proposed in [24]. In the process of multi-objective optimization, the individuals were selected by using the fitness function for crossover and mutation, and then a rapid non-dominated sorting method was used to elite strategy in population.

According to current research, most existing SDN multi-controller deployment schemes are based on the transmission delay or reliability of the switch to controller, and the controller deployment problem is transformed into the optimization model and solved by the optimization algorithm.

The following problems exist in the current study:

- Regarding the issue of multi-controller deployment, the current solution is mainly based on static deployment, which cannot meet the requirements of dynamic flow in SDN. In order to minimize the communication delay between the switch and the controller, the communication delay between the various controller domains is ignored. At present, the problem of controller deployment is solved if the number of controllers is as few as possible under the condition of average and worst delay. The network communication between different controller domains also increases with an increase in the number of controllers.
- The main factors currently considered in controller deployment are delay and stability. Controller capacity and load balancing between controllers are also very important factors.
- Current research mainly focuses on solving the controller deployment problem under a given number of controllers, and obtaining the mapping relationship between controller and switch. But the most suitable number of controllers is not easy to require in a distributed SDN. We can get the optimal number of controllers only by traversing all the candidate numbers, which is no easy to implement in a large-scale network [25].

In this paper, we propose a dynamic load balancing multi-controller deployment scheme based on the traffic propagation delay and capacity of controllers according to the network topology and actual network demand. Additionally, an improved affinity propagation algorithm (PSOAP) and a control-domain adjustment algorithm (CDAA) are proposed to solve the problem of dynamic controller deployment.

The main contributions of this research can be summarized as following:

- We propose a load balancing controller deployment model based on the intra-domain and inter-domain communication cost, and transform the traffic requests into a queuing model. Simultaneously, we consider traffic propagation delay and the capacity of controllers as two main factors in the multi-controller deployment problem.

- With the initial static network, considering of affinity propagation (AP) algorithm does not need to set the number of controllers in advance, we introduce it to solve the controller deployment problem. Then, aiming at the problem that the bias parameters and convergence coefficients in affinity propagation algorithm have limitations to the result of clustering, this paper puts forward an affinity propagation algorithm which based on particle swarm optimization (PSOAP). By taking the two parameters in algorithm as a particle, then adjust it intelligently by particle swarm optimization (PSO) algorithm. The algorithm improves the clustering effect and convergence accuracy, and achieves balanced controller deployment.
- With the dynamic traffic network, in order to ensure load balancing of controller, switches in different subdomains are reassigned by Breadth First Search (BFS) algorithm.

The rest of the paper can be organized as follows: Section II presents the model and formulation. The PSOAP algorithm and the adjustment algorithm proposed in this paper are explained in Section III. In Section IV, multiple scenarios experiments and analysis are carried out. The paper is summarized in the last section.

II. MODEL AND FORMULATION

In this section, we illustrate the problem of controller load imbalance in the process of multi-controller deployment in a distributed network, and set up the corresponding mathematical model, including a series of performance parameters that affect the deployment of controllers.

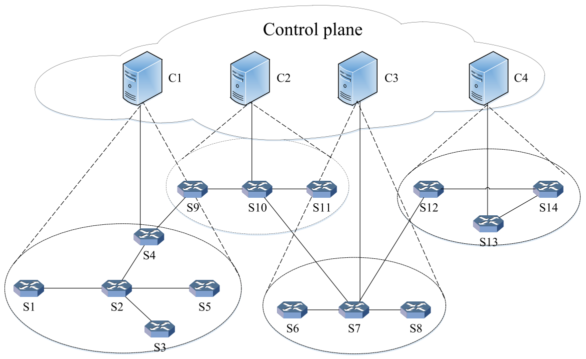

FIGURE 1. Multi-controller deployment in SDN.

A. PROBLEM DESCRIPTION

Fig. 1 shows the multi-controller deploying architecture for a distributed network. The whole network includes 14 switches (S1-S14) and four controllers (C1-C4). Differences in the number and deployment position of the controllers lead to an unbalanced state between controllers and increase the deployment cost in the distributed network. Therefore, given an SDN, the problem solved in this paper is to obtain the reasonable number of controllers and the mapping relationships between switch and controller.

The communication cost of multi-controller deployment includes the intra-domain communication cost and inter-domain communication cost. The smaller the number of controllers, the smaller the inter-domain communication cost and the greater the intra-domain communication cost, and vice versa. To ensure network performance, the propagation delay between the switch and controller should not exceed a tolerable threshold. Simultaneously, because of the limitation of the controller's processing capacity and bandwidth, the switching requests that a single controller can deal with are limited. The capacity of controller to process requests should be considered in the allocation of switches. Thus, multi-controller deployment in the SDN that is limited by traffic propagation delay and controller capacity can be described as follows: Given an SDN, for which the topology of its switches and links is known, our ultimate goal is to divide the network into reasonable control domains and identify each domain's switches in a manner that minimizes the overall intra-domain and inter-domain communication costs when the average latency of a switch to a controller does not exceed a tolerable threshold, and the single controller managing the switch request does not exceed the controller's processing capacity. Additionally, the deployment of each controller should be balanced as much as possible.

B. NETWORK MODELING

We build the multi-controller SDN model based on graph theory. The in-band communication is usually used as major communication mode in the SDN. In this paper, the controller network topology is represented by undirected graph $G = (V, E)$, where V and E are the set of nodes and links respectively. M represents the number of controllers in the network, C is the controller set and $C = \{C_1, \cdots, C_M\}$. N represents the number of switches, S is switch set and $S = \{S_1, \cdots, S_N\}$. So, the $|V| = M + N$. λ_i^t is the request rate of the i^{th} switch in slot time t. The d_{ij} represents the shortest distance between the i^{th} switch and j^{th} controller. x_{ij}^t is a binary variable, and $x_{ij}^t = 1$ indicates that the i^{th} switch is successfully connected to the j^{th} controller in time slot t. The deployment between a switch and controller can be defined as an $N \times M$ binary matrix Ω. To meet the dynamic constraints, a switch can only choose one controller as the master controller. As shown in Eq.(1):

$$x_{ij}^t = \begin{cases} 1 & i^{th} \textit{switch is connected to } j^{th} \textit{ controller} \\ 0 & \textit{others.} \end{cases} \tag{1}$$

The controller processing capacity is A and it is determined by factors such as CPU, bandwidth, and memory, $A = \{A_1, \cdots, A_M\}$. L_M is redundancy factor of each controller which ranges from 0 to 1. It means that the controller has sufficient capacity to perform state synchronization. The flow requests that the j^{th} controller needs to process in time t are the sum of the flow requests for all switches connected to it, which is represented as

$$\Phi_j^t = \sum_{i=1}^{N} \lambda_i^t x_{ij}^t. \tag{2}$$

where λ_i^t is different for each switch, it represents the flow request rate of the i^{th} switch in time slot t. In the OpenFlow network, each access switch in the network will generate flow set-up requests. Therefore, during the service time of flow requests in controller, there will be several "packet-in" requests originating from switches to controller. The flow requests between switches and controller will converge at the ingress of controller, and form the queue. According to the characteristics of OpenFlow switch sending requests to controller, we can reasonably assume that the arrival instants of new flow set-up requests constitute a Poisson process with rate Φ_j^t, and the processing time of flow requests are independent, identically random variables with negative exponential distribution, mean $\frac{1}{A_j}$. Then, based on queuing theory, the transmission and processing of traffic are described by an M/M/1 queue [26], and the flow requests are aggregated on the connected switch in the form of queuing.

With Little's law [27], we can get the average sojourn time of the j^{th} controller:

$$\omega_j^t = \frac{1}{A_j - \Phi_j^t}. \tag{3}$$

Given that the time of computing single source route is subject to the network size [28], the average response time of the j^{th} controller can be calculated as below:

$$\Delta t_j = |V|^2 \cdot \omega_j^t, \tag{4}$$

where $|V|$ represents the number of nodes in SDN, which indicates that the processing time is affected by the network size. Therefore, the average controller response time in time slot t between the switch and controller is

$$D^t = \frac{\sum_{j=1}^{M} \Phi_j^t \cdot \Delta t_j}{\sum_{j=1}^{M} \Phi_j^t}. \tag{5}$$

The main notation is summarized in Table 1.

According to the definition of notations in the table above, we calculate the network parameters, including intra-domain communication cost and inter-domain communication cost.

1) INTRA-DOMAIN COMMUNICATION COST

When a flow table, for example, a new flow table, needs to be installed on demand, the switch needs to send packets to the controller, which calculates the forwarding path and installs the corresponding flow label to the switch of the forwarding path. The switch then forwards packets according to the flow table. In this process, for controller C_j, the total flow request path delay in the time domain mainly includes the following: the switch sends packet-in to the subordinate controller and the controller calculates the forwarding path and switch installation flow table. The two communication costs between the switch and controller in the OpenFlow network can be expressed as

$$D_{req} = 2v_r \sum_{j \in M} \sum_{i \in N} (\left\lceil \frac{\lambda_i^t}{v} \right\rceil d_{ij} x_{ij}^t) \tag{6}$$

where v_r is the average rate of polling switches, it is related to the average number of links connected to switches, and v is the unit rate, $v = 1$Kb/s.

TABLE 1. Notations.

Notation	Definition
V	Network node
E	Network link
M	Number of controllers
N	Number of switches
$C=\{C_1,\ldots,C_M\}$	Controller set
$S=\{S_1,\ldots,S_N\}$.	Switch set
λ_i^t	Request rate of switch S_i in time slot t (Kb/s)
A_M	Processing capacity of controller C_M (Kb/s)
L_M	Redundancy factor of controller C_M
d_{ij}	Hop between switch S_i and controller C_j
x_{ij}^t	Binary variable x_{ij}=1 indicates S_i connects to C_j in time slot t, otherwise x_{ij}=0
Ω	Binary matrix of connections between switches and controllers

2) INTER-DOMAIN COMMUNICATION COST

In the multi-controller environment, information synchronization between controllers is required so that each controller can maintain global network status information. The state synchronization cost mainly refers to the communication cost caused by the interactive state information between the controllers, as shown in Eq.(7). And v_s is the average transmission rate of the state information of the controller. But v_s is less than v_r, because the SDN controller does not synchronize all the information in the subdomain.

$$D_{syn} = v_s \sum_{j \in M} \sum_{k \in M} d_{jk}. \tag{7}$$

Based on the above definition and analysis, the network communication cost (total) represents the cost of communication between the switches and controllers, and between the controllers at a certain time. For a given network topology, the intra-domain communication cost decreases with the increase of the number of controllers, while the inter-domain communication cost increases. Our ultimate goal is to divide the network into reasonable control domains and identify each domain's switches in a manner that minimizes the overall intra-domain and inter-domain communication costs provided by

$$min\ Total = \gamma D_{req} + (1-\gamma)D_{syn} \tag{8}$$

$$s.t. \sum_{j\in M} x_{ij}^t = 1, \quad \forall i \in N; \tag{9}$$

$$\Phi(t)_j \le L_j A_j, \quad \forall j \in M; \tag{10}$$

$$D(t) \le \delta; \tag{11}$$

$$x_{ij}^t \in \{0,1\}, \quad \forall i \in N,\ \forall j \in M. \tag{12}$$

where $\gamma \epsilon$ [0,1] is a weighting factor used to control the trade-off between the intra-domain and inter-domain communication costs. Constraint (9) ensures that the switch can only choose one controller as the master controller. Constraint (10) ensures that the load of the controller is within the normal range and does not exceed the processing capacity of the controller. Constraint (11) shows that the average delay (response time) from the switch to the controller is less than δ. Constraint (12) ensures that each switch can be connected to the main controller.

The problem of multi-controller deployment is NP-hard problem. Because of its complexity, it is impossible to solve this problem in time, particularly for distributed networks. Thus, we propose an improved clustering algorithm that can efficiently determine near-optimal solutions.

III. ALGORITHM DESIGN

In summary, the dynamic load balancing scheme proposed in this paper includes two-phase problem. In this section, we first propose PSOAP algorithm to solve the static deployment problem of multi-controller. Then, we propose a control-domain adjustment algorithm (CDAA) based on Breadth First Search (BFS) when the network traffic changes dynamically.

A. SOLUTION FOR INITIAL STATIC SITUATION

Traditional clustering algorithms, such as the *k-means* algorithm, critically depend on human intervention, different initial partitions or values of k that affect the objective function. Thus, in this section, we propose PSOAP to solve the controller placement problem, with no need to initialize the number of controllers in the network.

AP algorithm is a very effective partition-based clustering algorithm, which regards all data points as potential clustering centers [29]. It takes the measures of similarity between pairs of data points as input data and obtains the optimal solution by transferring information between data points in the iteration process [30]. It has several advantages, including efficiency, insensitivity to initialization, and ability to determine exemplars with fewer errors compared with k-means, fuzzy C-means, and other clustering algorithms. In this paper, we propose an improved AP algorithm to solve the controller deployment problem in an SDN. Specifically, we adopt the communication cost $(\lambda(t)_i d(i,j) x_{ij})$ instead of the Euclidean distance in the similarity measurement between two switches. By considering the preference parameters and damping parameters in the algorithm as particles, we adjust them intelligently using the PSO algorithm and improve the effect of clustering. PSOAP can learn the optimal number of controllers adaptively, but can also optimize the deployment relationship between a switch and controller, which ensures the reasonable distribution of controllers in the network.

In order to minimizes the object function expressed in (8), the similarity $S(i,j)$ between two nodes is defined as the communication between them, we adopts communication cost $(\lambda(t)_i d(i,j) x_{ij})$ instead of the Euclidean distance which usually used in standard AP:

$$S(i,j) = \begin{cases} \lambda(t)_i d_{ij} x_{ij} & i \ne j \\ P & i = j. \end{cases} \tag{13}$$

When $i \ne j$, $S(i,j)$ indicates the possibility of sample point j as the representative point of sample point i. Diagonal element $S(j,j)$ is preference parameter $P(j)$. The initial value of $P(j)$ typically takes the same value, which is the minimum or mean value of all non-diagonal elements in the similarity matrix. Its initial size has a great influence on the final clustering number. The larger the value of P, the more clusters are generated, and vice versa.

Unlike k-means, AP does not need to know the number of clustering centers in advance. The responsibility value $R(i,j)$ and availability value $A(i,j)$ are two important information in AP algorithm. Criterion value $C(i,j) = R(i,j) + A(i,j)$, which represents the possibility of candidate representative point j as cluster center. Therefore, AP clustering operates on four matrices: similarity matrix S, responsibility matrix R, availability matrix A, and criterion matrix C. To determine appropriate cluster centers, these matrices are updated iteratively as follows:

$$R(i,j) = S(i,j) - \max_{j' s.t. j' \ne j} \{A(i,j') + S(i,j')\}, \tag{14}$$

$$R_{new}(i,j) = \lambda R_{old}(i,j) + (1-\lambda) \times R(i,j), \tag{15}$$

$$A(i,j) = \min\{0, R(j,j) + \sum_{i' s.t. i' \notin \{i,j\}} \max\{0, R(i',j)\}\}, \tag{16}$$

$$A(j,j) = \sum_{i' s.t. i' \notin \{i,j\}} \max\{0, R(i',j)\}, \tag{17}$$

$$A_{new}(i,j) = \lambda \times A_{old}(i,j) + (1-\lambda) \times A(i,j). \tag{18}$$

Equation (14) computes responsibility matrix R. Responsibility value $R(i,j)$ reflects the suitability of switch j as cluster center of switch i. Taking into account other potential cluster centers j' for switch j. The larger the value of $R(i,j)$, the more suitable the cluster center. Equations (16) and (17) are used to update the off-diagonal and diagonal elements of availability matrix A, respectively. Availability value $A(i,j)$ reflects the accumulated evidence on how appropriate it would be for switch i to choose switch j as its cluster center, taking into account the support from switch i. Switch j is more well suited to the cluster center of switch i when availability value $A(i,j)$ is large The availabilities and responsibilities are combined to identify cluster centers, which forms criterion matrix C.

The initial availability values $A(i,j)$ and responsibility values $R(i,j)$ are set to zero, which means that there is no clustering relationship between the data at the beginning, and they are updated in the form of (14)-(18). It is very important to avoid the oscillation of AP algorithm in the update process. The Damping factor λ can improve convergence speed when AP clustering can't converge due to oscillation. The algorithm terminates when the maximum number of iterations T_m is exceeded or the amount of information changes is below a fixed threshold T_c [3]. Finally, after convergence, assignment vector Q is computed as

$$Q_i = \arg\max_j(c(i,j)). \tag{19}$$

Intuitively, $Q_i = i$ means that switch i represents an cluster center which is deployment position of the controller. Otherwise, switch i chooses switch Q_i as its cluster center, that is, switch i should connect to controller Q_i.

Considering the influence of the preference element and damping factor on the clustering results, the preference element and damping factor are used as the position coordinates of particles in PSO. In this paper, we propose a modified AP algorithm based on PSO (i.e., PSOAP). PSOAP takes the preference element and damping factor as the position coordinates of the particles in the PSO algorithm. The coordinates and speed of each particle are initialized, to select the different preference elements and the damping factors. Then, according to (20)-(21), the position and direction of particles are constantly updated. In the process of updating, the position of particles is used as the value of the preference element and the damping factor of the AP algorithm to cluster:

$$\begin{aligned} V_{id} &= \omega V_{id} + \eta_1 rand()(P_{id} - X_{id}) \\ &\quad + \eta_2 rand()(P_{gd} - X_{id}), \qquad (20) \\ X_{id} &= X_{id} + V_{id}, \qquad (21) \end{aligned}$$

where V_{id} represents the speed of i particles in the d dimension, P_{id} is the best location that the particle has experienced, P_{gd} is the best location for the group, ω is the inertia weight, and η_1 and η_2 are important parameters to regulate P_{id} and P_{gd}. In (20), we compare the current particle position with the individual optimal solution, compare the current particle position with the group optimal solution, obtain a group optimal and individual optimal development trend, and then determine the new speed direction according to this trend and the original initial velocity direction. Equation (21) that is what we previously obtained, and the new position of a particle is produced at a certain distance from the upward movement. The value of the odd dimension of X_{nd} is a new preference element, and the value of the even dimension is the damping factor, as follows:

$$P = X_{nd}, \text{ when } d \text{ is an odd function;} \tag{22}$$
$$\lambda = X_{nd}, \text{ when } d \text{ is an even function.} \tag{23}$$

The conditions for the end of the iteration are as follows: when the number of iterations exceeds the maximum or when the cluster center does not change for several consecutive times. Algorithm 1 describes our algorithm.

Algorithm 1 PSOAP

```
Input: SDN network topology $G = (V, E)$
    The request rate of switch $i$, $\lambda_i^t$
    The controller processing capacity $A_M$
    The redundancy facto $rL_M$
Output: The deployment matrix $\Omega$
1: All nodes are traversed, the get $\lambda_i^t, A_M, L_M$.
2: **if** $\Phi_M \le L_M A_M$ and $D^t \le \delta$
3: According to (10), the similarity matrix S is established, and the initialization information matrix is initialized.
4:   **repeat**: Update the parameters of the AP algorithm according to the (22) and (20);
5:     **repeat**: Update information matrix according to (14) - (16);
6:       **until**: If the end condition is reached, it will end, otherwise, Step5 will continue.
7: Calculate the fitness of each particle;
8: Compare each particle, compare its fitness with the fitness of the best position it has experienced, and update it;
9: Compare each particle, compare its fitness with the fitness of the best position experienced by the group, and update it;
10: Update the velocity of each particle according to the (19);
11: Update the next position of particle movement according to the (21);
12: **until**: Cluster centers remain unchanged for several consecutive times (min *TOTAL*) or the number of iterations exceeds the maximum number; otherwise Step4 is executed.
13:**end if**
14: Get $x_{ij}$ and the deployment matrix $\Omega$.
```

B. SOLUTION FOR DYNAMIC CHANGES SITUATION

When network traffic changes, in order to balance the load of each control-domain and reduce the total propagation delay in the domain, it is necessary to readjust the mapping relationship of inter-domain switches to optimize the load and delay indicators. In this paper, the adjustment of the switch mapping relationship is realized by the Breadth First Search (BFS) algorithm. The process of the algorithm is shown in Algorithm 2.

Firstly, a set of intermediate switches TNS is established to mark the boundary switches with the furthest distance and largest traffic from the selected controller in any control-domain. When the load of a control domain exceeds the upper limit of capacity (A_jL_j), the switch with the farthest distance and moderate flow from the controller is selected by breadth-first traversal to be moved out of the switch set of the control domain and then added to the TNS. In the slot time t, the

Algorithm 2 Control-domain Adjustment Algorithm (CDAA)

```
Input: SDN network topology $G = (V, E)$, $T$, $t$
    The request rate of switch $i$, $\lambda_i^t$
    The controller processing capacity $A_M$
    The redundancy factor $L_M$
    Switch set for each control-domain IS
    Intermediate Switch Set TNS
    Adjusted controller set LCon, switch set LS
    $\{LS, C, TNS\} = \phi$
Output: LCon, LS
1: for all $t$ with $0 \le t \le T$ do
2: for each $j \epsilon M$ do
3: TNS$\leftarrow$\{furthest from ICon$_j \cap$max($\lambda_i^t$), $i \epsilon$ IS$_j$\};
4: end for
5: for each $j \epsilon M$ do
6: if $\Phi_j^t \ge L_j A_j$ do
7: Add nodes in IS$_j$ furthest from ICon$_j$ to TNS by BFS based on (8) $\sim$ (12);
8: end if
9: end for
10: Compute function $Total = \gamma D_{req} + (1 - \gamma) D_{syn}$
11: if Total$^t$ < Total$^{t-1}$ do
12: LS$\leftarrow$IS;
13: end if
14: end for
15: Output LCon, LS;
```

minimization of *Total* is repeated several times, updating switch mapping relations with the aim of Minimizing Propagation delay and controller capacity. Finally, get the adjusted set of controllers and switches.

When network traffic changes, the scheme detects whether the load of each control-domain exceeds the processing capacity of the controller at any time slot t. Otherwise, the load is pre-adjusted until the control-domain load satisfies the efficiency interval. If the *Total* value of the pre-adjusted network is smaller than that of the slot time $t - 1$, the adjustment will be accepted. Otherwise, the next round of adjustment will be made in the slot time $t+1$, and the position of the controllers in each sub-domain will be adjusted at the time T to reduce the *Total* value further. The size of slot t can be dynamically adjusted according to network traffic to change the number of adjustments in time T, so CDAA can reduce the propagation delay between the controller and the switch while realizing the load balancing of the controller.

IV. SIMULATION AND EVALUATION

In this section, we introduce the setting of the experimental environment, make simulation experiments and analysis from three scenarios.

A. SIMULATION ENVIRONMENT SETTING

The experimental environment and parameter settings in the simulation are explained below.

1) TOPOLOGIES

In order to make the experiment more representative, we choose OS3E topology model to simulate the experiment [31]. The OS3E topology consists of 34 nodes and 42 links. All nodes in the network have the ability to deploy controller and switch, and they are independent of each other.

2) SELECTION OF CONTROLLER

The basic configuration for the experiment is Intel Core 3.4 GHZ CPU 8 GB RAM. The algorithm is implemented in Java and the results were analyzed using MATLAB. In terms of controller selection, we adopt the OpenDaylight [32] controller and write the appropriate application modules on the application layer.

3) SETTING OF PARAMETERS

The request rate of the switch obey Poisson distribution, and it range from 150 Kb/s to 550 Kb/s [12]. According to the partial measurement results of [33], the processing capacity A_M was set to 15M. To create differences between different controllers, the redundancy factor of controller L_M is arbitrarily selected between 0.9 and 1. The average rate of polling a switch is $v_r = 10$Kb/s, and the transmission rate of the state synchronization information of the controller is $v_s = 1$Kb/s.

The weight of the factor γ was set to 0.8 in our defined objective function. The coefficient is used to adjust the relative importance of intra-domain and inter-domain communication costs in the controller deployment process. In general, the communication cost of intra-domain is higher than that of inter-domain. In the study of controller placement, we should set high weight for the minimization of intra-domain communication delay.

B. SIMULATION ANALYSIS

In [3], the AP algorithm was introduced to solve the problem of control plane deployment. The simulation results demonstrated that the proposed AP algorithm could provide more stable and accurate controller deployment. To verify the performance of PSOAP, we set up a series of simulation experiments, and PSOAP was compared with the AP and Genetic Algorithm (GA) algorithms. A heuristic algorithm was used to solve the multi-controller deployment problem in [22] and [23], and obtained a good performance.

1) NETWORK PLANNING

In this experiment, we validate the network planning effect of different algorithms under the same conditions. Subdomain planning is an important indicator to determine controller load balancing. In an SDN, the more balanced the number of switches managed by the controller, the more reasonable the subdomain planning and the better the controller load balancing performance. Both PSOAP and the AP algorithm can solve the controller placement problem without initializing the number of controllers, but not the GA algorithm. Thus, simulation experiments are carried out for different number of controller k to select the optimal number of controllers for

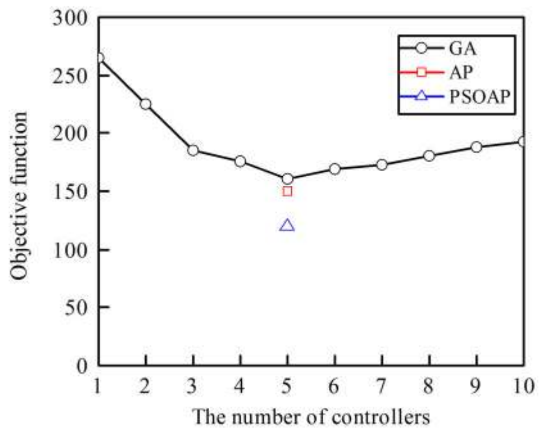

FIGURE 2. Objective function for the three algorithms.

GA algorithm. Then, we directly applied PSOAP and the AP algorithm to the OS3E topology.

Fig. 2 shows the impact of the number of controllers on the objective function using PSOAP, AP and GA algorithms on OS3E topology. In the OS3E network, the number of switches is a constant value and the setting of the topology is refered to [4]. We note that we set the T_m parameter of three algorithms to 300 iterations. In Fig. 2, through the experiment of GA algorithm, we can see that the number of controllers has a significant effect on the value of objective function. As depicted in Fig. 2, the value of the objective function decreases with the number of controllers from 5 to 10. This is simply because, as the number of controllers increases, the number of switches managed by the controller decreases. Hence, the switch-to-controller latency decreases. Moreover, we argue that the numbers of controllers needed to better handle traffic requests over OS3E topology are both five. As observed, the value ensured the optimal value of the objective function for the three algorithms, and PSOAP obtains the lowest value of the objective function.

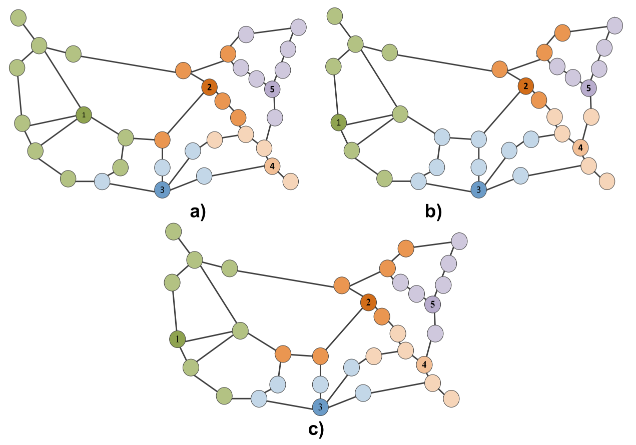

FIGURE 3. Deployment results on Internet OS3E. (a) GA. (b) AP. (c) PSOAP.

The deployment results are shown in Fig. 3, in which the three algorithms divide the OS3E topology into five domains accurately, which are distinguished by different colors. The numeric symbols indicate the location of all controllers. From a macro-topological point of view, Fig. 3 clearly shows that PSOAP can achieve a reasonable controller selection and network division, both the GA and AP algorithms caused an unbalanced deployment of switches, which would seriously affect the stability of the network.

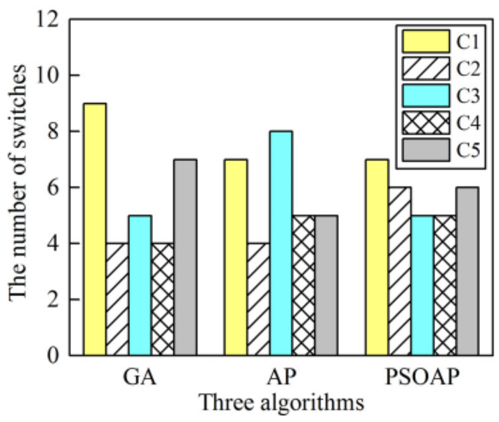

FIGURE 4. Analysis of the deployment results in the network.

To compare the subdomain planning effects of different algorithms more clearly, we summarize and analyze the experimental data in Fig. 3. The number of switches managed by each controller in the three algorithms is shown in Fig. 4. Under the same controller deployment conditions, the number of switches managed by each controller in the GA has the largest difference, in which the number of switches managed by controller C1 is twice that of controllers C2 and C4. This is because the GA uses crossover and mutation strategy to make the whole population evolve continuously and search for the optimal solution, but the algorithm is easy to fall into minimum value. When the distance between nodes is large, it is easy for the clustering operation to fall into the local optimum, resulting in the regional aggregation of switches and a poor subdomain planning effect. Although the AP algorithm does not need to set the number of controllers, the algorithm also aggregates nodes according to the distance between switches, and the effect of subdomain planning is poor. In Fig. 4, after normalizing the experimental data, PSOAP has a node equalization rate of 0.87, which has an obvious advantage over the GA (0.66) and the AP (the node equalization rate is 0.74).

We can see from the Fig. 3 and Fig. 4 the PSOAP performs well in subdomain partitioning and controller load balancing. In order to compare the performance of the three algorithms more clearly, the convergence iteration diagrams of the three algorithms are given in Fig. 5.

In Fig. 5, GA, AP and PSO-AP algorithms are iterated 165, 130 and 145 respectively to obtain the optimal solution. The execution time of PSOAP algorithm is not the best of the three algorithms, but it can obtain the optimal solution.

2) PROCESSING DELAY AND CONTROL TRAFFIC

The results of controller deployment not only affect the division of network domains, but also have a great impact on the

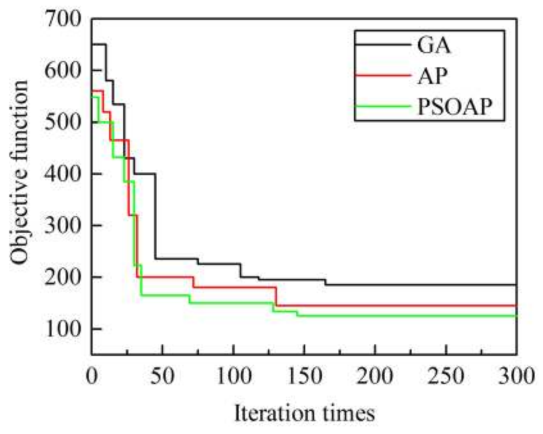

FIGURE 5. Performance analysis of three algorithms.

processing delay and control traffic of the controller. On the basis of the first network planning experiment, we conduct the second experiment and compare the delay and traffic of the three algorithms. To eliminate the random error of the experiment, all three algorithms were run 20 times under the same experimental conditions, and we took the average value as the experimental result, as shown in Fig. 6 and Fig. 7.

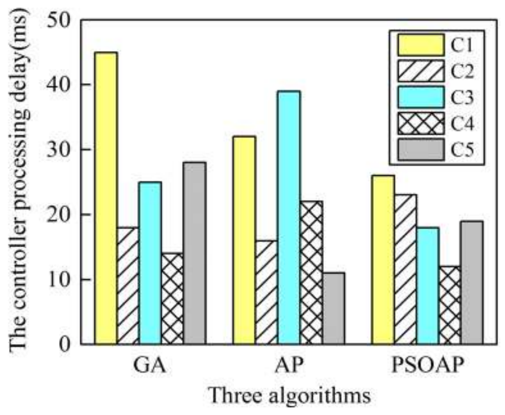

FIGURE 6. Comparison of the controller processing delay.

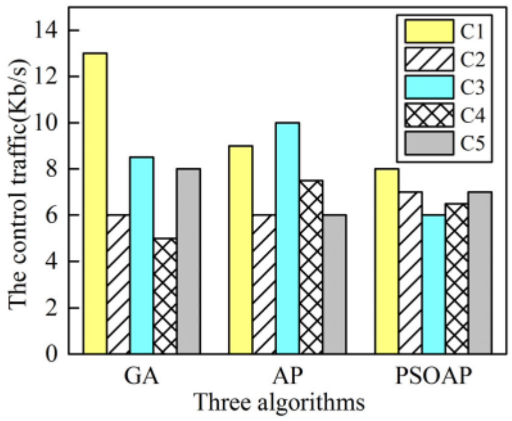

FIGURE 7. Comparison of the control traffic.

In Fig. 7, the AP algorithm optimizes the control flow in the subdomain from the Viewpoint of Controller, but it can't solve the problem of the substantial difference in processing delays. Compared with the AP and GA algorithms, PSOAP not only take into account the number of hops but also consider the request rate of the switch in the process of controller deployment. Therefore, PSOAP effectively reduces the fluctuation of the processing delay for each controller and achieves a balanced distribution of control traffic.

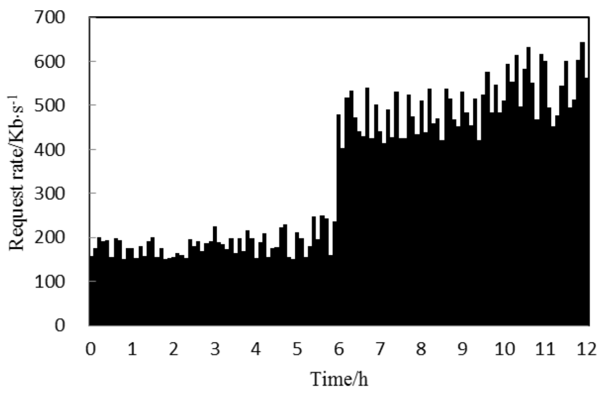

FIGURE 8. Switch flow request rate in the OS3E network.

3) LOAD BALANCING UNDER THE DYNAMIC TRAFFIC

The third experiment compared the load balancing rate and network response time of the three algorithms under different flow requests. The proposed PSOAP algorithm is deployed in the Internet OS3E network topology, we used a traffic generator to generate continuous flow requests in the switch, as shown in Fig. 8. The entire process is divided into two periods. In the first period (0-6 h), all switches have a flow request rate of less than 280 kb/s and the traffic distribution is self-similar. In the second period (7-12 h), a large number of traffic requests are generated in the switch to simulate traffic bursts. Using this traffic model, we verify the rationality and feasibility of the scheme proposed in this paper. In this experiment, the $T = 60$ min, the time slot $t = 5$ min. We run each simulation for 20 times.

Fig. 9 shows that the controller load balancing rate varied with the change of the request rate of the switch flow. When the traffic request rate of all switches is low (0-6 h), the load balancing rate of the three algorithms can keep stable. The load balancing rate of CDAA is the highest, followed by the GA algorithm, and the AP is the lowest. The GA and AP are highly sensitive to traffic variations, the load balancing rate decreases when the traffic requests of switches suddenly increase (7-12 h). CDAA effectively managed the scenario of burst flow requests in switches, and maintained the controller load balancing rate at a higher level. Compared with the other two algorithms, the CDAA improves the controller load balancing rate by at least 26.5%.

Fig.10 plots the comparison of CDAA, AP and GA in terms of average response time in OS3E. When the switch traffic request rate was low, the response time of the three algorithms is small and the network state is relatively stable. It shows

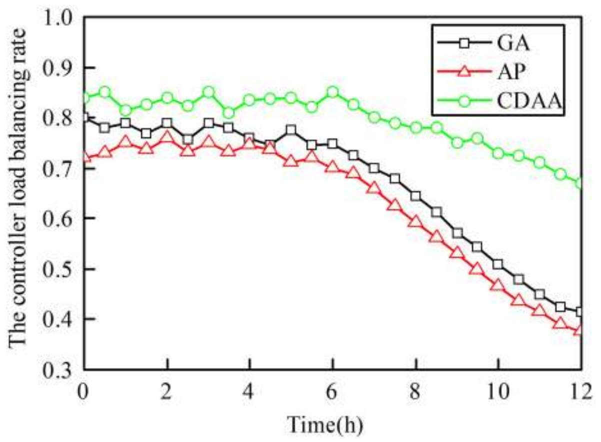

FIGURE 9. Load balancing rate.

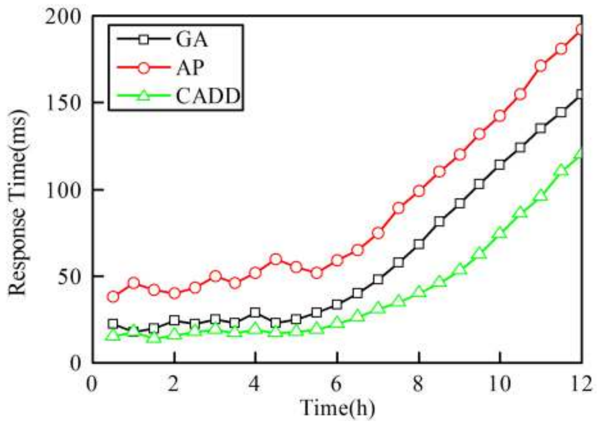

FIGURE 10. Response time under different traffic loads in OS3E.

that the response time of CDAA increases slightly with the increase of load, while the AP and GA increases sharply when traffic bursts occurred in the network. We can see from the Fig.10: (1) As the total traffic request rate increases, response time also increases since the computing resource on controllers is limited. (2) Request dynamics may cause a sudden increase of response time for AP. In the extreme case, the response time of SM is 2x that of CDAA. It demonstrates that static controller assignment results in severe load imbalance. In our simulations, this serious phenomenon takes about 70% of the time in all runs. CDAA reduces response time by 50% on average compared with AP which is a static deployment algorithm. (3) CDAA outperforms GA and reduces the response time by 25% on average. CDAA considers the intra-domain and inter-domain communication costs in the model, the adjustment of the switch mapping relationship is realized by the Breadth First Search (BFS), and a better controller deployment scheme was obtained.

V. CONCLUSION

In this paper, we discussed the problem of multi-controller placement in a distributed SDN. Taking intra-domain and inter-domain communication costs as optimization objectives, PSOAP is proposed to solve the problem of controller placement in initial static state and CDAA is proposed to solve adjustment of control-domain problem under dynamic traffic. The simulation demonstrated that the scheme can obtain the optimal number of controllers adaptively according to the network topology, assign switches to the most appropriate controllers, and achieve the balanced deployment of multi-controller architecture in a distributed SDN.

REFERENCES

[1] S. Zhang, J. Lan, P. Sun, and Y. Jiang "Online load balancing for distributed control plane in software-defined data center network," *IEEE Access*, vol. 6, pp. 18184–18191, 2018.

[2] T. Hu, P. Yi, Z. Guo, J. Lan, and J. Zhang, "Bidirectional matching strategy for multi-controller deployment in distributed software defined networking," *IEEE Access*, vol. 6, pp. 14946–14953, 2018.

[3] J. Zhao, H. Qu, J. Zhao, Z. Luan, and Y. Guo, "Towards controller placement problem for software-defined network using affinity propagation," *Electron. Lett.*, vol. 53, no. 14, pp. 928–929, 2017.

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

[5] 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.

[6] G. Ishigaki and N. Shinomiya, "Controller placement algorithm to alleviate burdens on communication nodes," in *Proc. Int. Conf. Comput., Netw. Commun.*, Feb. 2016, pp. 1–5.

[7] Y. Jimenez, C. Cervelló-Pastor, and A. J. Garcia, "On the controller placement for designing a distributed SDN control layer," in *Proc. Netw. Conf.*, Jun. 2014, pp. 1–9.

[8] 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*Oct. 2015, pp. 220–224.

[9] C. Wang, B. Hu, S. Chen, D. Li, and B. Liu, "A switch migration-based decision-making scheme for balancing load in SDN," *IEEE Access*, vol. 5, pp. 4537–4544, 2017.

[10] 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.

[11] C. Gao, H. Wang, F. Zhu, L. Zhai, and S. Yi, "A particle swarm optimization algorithm for controller placement problem in software defined network," in *Proc. Int. Conf. Algorithms Architectures Parallel Process.* Cham, Switzerland: Springer, 2015, pp. 44–54.

[12] S. Liu, H. Wang, S. Yi, and F. Zhu, "NCPSO: A solution of the controller placement problem in software defined networks," in *Proc. Int. Conf. Algorithms Archit. Parallel Process.* Cham, Switzerland: Springer, 2015, pp. 213–225.

[13] N. Beheshti and Y. Zhang, "Fast failover for control traffic in software-defined networks," in *Proc. IEEE Global Commun. Conf.*, Dec. 2013, pp. 2665–2670.

[14] 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.*, Dec. 2014, pp. 1909–1915.

[15] 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 Universities Posts Telecommun.*, vol. 19, pp. 92–97, Oct. 2012.

[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.*, May 2013, pp. 672–675.

[17] 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.

[18] M. Guo and P. Bhattacharya, "Controller placement for improving resilience of software-defined networks," in *Proc. 4th Int. Conf. Netw. Distrib. Comput. (ICNDC)*, Dec. 2013, pp. 23–27.

[19] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner, and P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Proc. 25th Int. Teletraffic Congr. (ITC)*, Sep. 2013, pp. 1–9.

[20] D. Hock, M. Hartmann, S. Gebert, T. Zinner, and P. Tran-Gia, "POCO-PLC: Enabling dynamic Pareto-optimal resilient controller placement in SDN networks," in *Proc. IEEE Conf. Comput. Commun. Workshops*, Apr./May 2014, pp. 115–116.

[21] S. Lange *et al.*, "Heuristic approaches to the controller placement problem in large scale SDN networks," *IEEE Trans. Netw. Service Manage.*, vol. 12, no. 1, pp. 4–17, Mar. 2015.
[22] V. Ahmadi, A. Jalili, S. M. Khorramizadeh, and M. Keshtgari, "A hybrid NSGA-II for solving multiobjective controller placement in SDN," in *Proc. 2nd Int. Conf. Knowl.-Based Eng. Innov.*, Nov. 2015, pp. 663–669.
[23] A. Jalili, V. Ahmadi, M. Keshtgari, and M. Kazemi, "Controller placement in software-defined wan using multi objective genetic algorithm," in *Proc. 2nd Int. Conf. Knowl.-Based Eng. Innov. (KBEI)*, Nov. 2015, pp. 656–662.
[24] B. Liu, Y. Shun, and Y. Fu, "Load balancing scheme based on multi-objective optimization for software defined network," *J. Omput. Appl.*, vol. 37, no. 6, pp. 1555–1559, 2017.
[25] 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.
[26] K. Atefi, S. Yahya, A. Rezaei, and A. Erfanian, "Traffic behavior of Local Area Network based on M/M/1 queuing model using poisson and exponential distribution," in *Proc. IEEE Region 10 Symp. (TENSYMP)*, May 2016, pp. 19–23.
[27] D. P. Bertsekas, R. G. Gallager, and P. Humblet, *Data Networks*. Upper Saddle River, NJ, USA: Prentice-Hall, 1992.
[28] E. W. Dijkstra, "A note on two problems in connexion with graphs," *Numerische Mathematik*, vol. 1, no. 1, pp. 269–271, 1959.
[29] B. J. Frey and D. Dueck, "Clustering by passing messages between data points," *Science*, vol. 315, no. 5814, pp. 972–976, 2007.
[30] L.-C. Wang and S. H. Cheng, "Data-driven resource management for ultra-dense small cells: An affinity propagation clustering approach," *IEEE Trans. Netw. Sci. Eng.*, to be published.
[31] *Internet2 Open Science, Scholarship and Services Exchange*. [Online]. Available: http://www.internet2.edu/-network/ose/
[32] Z. K. Khattak, M. Awais, and A. Iqbal, "Performance evaluation of OpenDaylight SDN controller," in *Proc. 20th IEEE Int. Conf. Parallel Distrib. Syst. (ICPADS)*, Hsinchu, Taiwan, Dec. 2014, pp. 671–676.
[33] G. Cheng, H. Chen, H. Hu, and J. Lan, "Dynamic switch migration towards a scalable SDN control plane," *Int. J. Commun. Syst.*, vol. 29, no. 9, pp. 1482–1499, 2016.

GUOYAN LI received the B.S. degree in computer science and technology from Hebei Normal University, China, in 2006, and the M.S. degree and the Ph.D. degree in computer application from the Hebei University of Technology, Tianjin, China, in 2009 and 2013, respectively. Her research interests include the future Internet architecture, network function virtualization, and software-defined networking.

XINQIANG WANG received the M.S. degree from Nankai University, Tianjin, China, in 2010. He is currently pursuing the Ph.D. degree with the School of Software, Tianjin University, Tianjin. He is also a Professor with the Tianjin Sino-German University of Applied Sciences. His research interests include network optimization, the IoT, and information processing.

ZHIGANG ZHANG received the M.S. degree in software engineering major from Nankai University, Tianjin, China, in 2005, and the Ph.D. degree in computer application from the Tianjin University of Technology, Tianjin, in 2017. He is currently a Professor with Tianjin Chengjian University, Tianjin. His research interests include the Internet architecture, network function virtualization, and network optimization.