

RESEARCH

Open Access



# A controller deployment scheme in 5G-IoT network based on SDN

Xin Cui<sup>1,2\*</sup>, Xiaohui Gao<sup>3</sup>, Yan Ma<sup>2</sup> and Weihai Wang<sup>4</sup>

\*Correspondence:

cxsd2007@163.com

<sup>1</sup> School of Computer Science and Technology, Shandong University of Technology, Zibo, China  
Full list of author information is available at the end of the article

## Abstract

Due to the wide spread of 5G networks, network users have higher requirements for communication delays. Software-defined network is a new paradigm of decoupling the control logic from packet forwarding devices. We can reduce communication latency in the network by optimizing the location of the controller and improve the communication performance of the network. In this paper, the controller deployment problem of multi-network area is studied in order to reduce the average communication latency. In order to solve this problem, we proposed an optimized DPC algorithm. Specifically, on the basis of DPC algorithm we quoted the idea of triangular stability from the BeeDPC algorithm, and introduced SC measurement indicators, the degree of separation between sub-network regions and the degree of aggregation within the sub-network region, so as to divide the network region more reasonably. At the same time, we have break out of the constraints in the DPC algorithm and introduced closeness centrality to get a more reasonable placement scheme and reduced the limitations of the DPC algorithm. Simulation results have shown that the optimized DPC algorithm can effectively reduce the average delay between the control layer and the data layer, improve the network performance, and enhance network stability and reliability.

**Keywords:** Software-defined network, Controller placement, Optimized DPC algorithm, Average communication latency, Network partition

## 1 Introduction

What is SDN network? SDN separates the control layer from the data layer of network equipment when compared with the traditional network, and each controller administers a range of switches, which makes the network more automatic and the elastic control of network traffic becomes a reality. Meanwhile, SDN network is mainly composed of three-layer architecture, application layer, control layer and data layer. Each layer has its own unique functions, so that the operability of SDN network can be significantly better than the traditional network. At the moment, it provides a better stage which we can develop the core network.

At the same time, SDN starts to develop comprehensively in the Internet of Things (IoT) due to the advantages of SDN. With the continuous growth of mobile data services in mobile network, more stringent requirements are put forward for low latency

and high performance in mobile network. The exponential growth of mobile communications has significantly improved the quality of human interactions regardless distance [1–5]. Therefore, we need some new technology which can better deal with the challenges of 5G, including security, availability, reliability and so on. The scheme is carried out in document [6] provides users with a secure and reliable IoT environment, which well preserves users' privacy. Literature [7, 8] came up with a new architecture of the Internet of Things, such as SDN/NFV (Network Functions Virtualization). These architectures are built on SDN to achieve high reliability and availability of the network.

We know that the scheme of controller placement will have an impact on the reliability, communication latency and stability of the network in some degree. The distribution of controllers affects the overall performance of the network directly, such as network communication delay is an important index. In the literature [9–14], the question of SDN controller placement has been studied specifically, Killi et al. studied the controller layout in SDN, and proposed the communication delay between switch, and controller was one of the optimization goals. At the same time, Alowa and Fevens proposed that the key to the problem lies in minimizing the communication delay between the switch and the controller when studying the controller layout in SDN and so on. Heller et al. [15] first studied the influence of controller placement on average delay and maximum delay in the literature, and proposed the communication latency as the main consideration when we make a scheme of the development of controllers. Yao et al. [16] attached great importance to the significance of the communication delay in the traffic load between the controller and the switch.

As we know, the location problem of the controller in SDN can be simplified as the coverage matching problem [17–19] or the dominant set problem [20]. Therefore, in the literature [21], Julili et al. further studied the randomization mechanism of greedy population. Literature [22] further studies the work in the literature [23] on the basis of paper [24] and proposes a controller distribution method for multi-objective perception. In the literature [25–27] emphasizes the two disadvantages of the physical centralized SDN control plane (scalability and reliability) and compares two multi-objective controller layout methods that deal with performance and reliability indexes simultaneously.

Therefore, the research on the layout of multiple controllers in SDN networks has become popular, such as DevoFlow [28], Onix [29], HyperFlow [30] and Kandoo [31].

We also found that some clustering methods can be used for controller distribution problems. These algorithms can often calculate the reasonable controller layout scheme efficiently.

In the literature [32], Alex Rodriguez et al. put forward a new way of thinking about clustering which name is DPC algorithm. In this algorithm, two concepts of local density and relative distance are proposed for clustering. By calculating these two indexes, regional division and cluster center judgment are performed. However, this algorithm is too simple in the judgment of noise points and does not reasonably calculate the category of noise points. At the same time, after the noise point is introduced, the local density of the sub-cluster will change correspondingly, and the center point will also change accordingly, but the algorithm does not take this into account.

For the purpose of solving the problem of controller deployment which is related to SDN networks, this article comes up with a multi-controller deployment algorithm

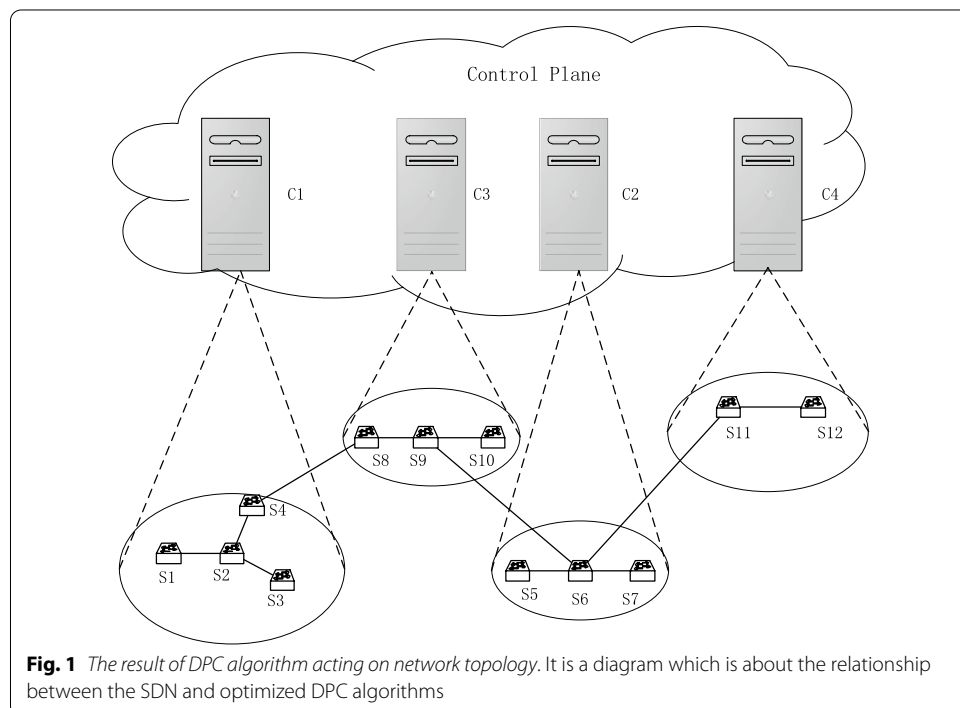
which in the light of the DPC algorithm and the network partition—optimized DPC algorithm, and the optimized DPC algorithm is a clustering algorithm essentially. The SDN network region is divided into the several network partitions according to the quantity of controllers.

Next, we calculate the center of each subnetwork region and use this point as the deployment point for the controller. We borrowed the idea of triangle stability which is from BeeDPC algorithm and introduced the SC evaluation index [33] on the basis of DPC algorithm. The purpose of doing these is to reduce the impact of the truncation distance in the DPC algorithm. At the moment, we introduce the closeness centrality to determine the best controller placement scheme for each sub-network area. In the performance evaluation, for comparison purposes, K-means algorithm, optimized K-means algorithm, optimal placement algorithm and optimized DPC algorithm are applied to the same topology (OS3E Internet2 Topology) for comparison. Experimental results show that the algorithm in this paper produces the best performance. And now, we will give a diagram which is about the relationship between the SDN and Optimized DPC algorithms.

In Fig. 1, the optimized DPC algorithm is used to divide the network area and optimize the method of judging the classification of noise points. Next, we further introduce related measurement indicators to recalculate the clustering center to make the clustering effect better.

According to the above, our main contributions in this paper are as follows:

- 1 We introduce triangular stability to calculate the attribute categories and regional division of noise points based on the DPC algorithm.



- 2 After the initial attribution of all data points, we completed the second regional division according to SC evaluation indicators.
- 3 Determine the center point of each subregion by calculating the proximity centrality of each subregion node.

In the remainder of this article, the II part introduces related work which describes the achievements of predecessors in this direction and a brief introduction to the work we have completed. In the third part, we use the corresponding mathematical algorithm to express the problem and give the relevant index to estimate the implementation efficiency of the algorithm we proposed—average communication latency. For this problem, we use the optimized DPC algorithm to build the relevant mathematical model and explain how to use the mathematical model to solve this problem. The fourth part gives the simulation results of the improved DPC algorithm. The fifth part introduces the performance analysis of the optimized algorithm when compared with other related algorithms. The sixth part gives the cost–benefit analysis of deploying the controller on the Internet2 OS3E topology diagram. The seventh part gives some discussions about the research article which include the implications of the findings and the highlight limitations of the study. The seventh part gives a summary and generalization of the article. Finally, the eighth part gives the list of abbreviation which is introduced in the article.

## 2 Related work

In the introduction part of the literature, we reviewed the technical concepts and research trends of network area division in SDN, and we also learned about the DPC algorithm and related concepts of the improved DPC algorithm we proposed. In order to produce better contrast effect, the following three sections will introduce four algorithms. The traditional K-means algorithm, The optimized K-means algorithm, The optimal placement algorithm and the optimized DPC algorithm.

### 2.1 The traditional K-means algorithm

In the literature [34–36], traditional K-means algorithm is mainly used to calculate network sub-regions and regional center points. However, we know that the k-means algorithm conducts clustering with K points in the space as the center. Meanwhile, the remaining data points close to them are classified by iterative method, and the values of each clustering center are updated in this process, so as to obtain the best clustering result. However, K-means algorithm has obvious disadvantages. Because K-means requires the artificial designation of initial clustering centers, the problem is that different initial clustering centers have a huge impact on the final results. Meanwhile, the final result obtained by K-means algorithm can only be guaranteed to be locally optimal. Most importantly, the K-means algorithm cannot properly handle noise points.

### 2.2 The Optimized K-means algorithm

The paper [37] makes some improvements on K-means algorithm. This algorithm is still K-means algorithm in essence, but at the same time of partitioning molecular regions, the longest distance in each subnet is cut, and the vertex of the longest distance is used as the starting point of the next subnet segmentation, so as to shorten the delay in the

network. This algorithm is also limited by the selection of the initial clustering center, but compared with the traditional K-means algorithm, it can better obtain the clustering effect.

### 2.3 The optimized placement algorithm

Literature [38] proposed a kind of optimal placement algorithm, the core of which is the greedy algorithm. The initial input of the algorithm is  $K$  controllers and topology  $G(V, E)$ ,  $V$  is node set and  $E$  is link set. Optimal placement algorithm is output through several iterations. During an iteration, the input data for the current iteration is the output of the previous iteration. In many cases, the final output is not the best, but it can be accepted by people. At the same time, the computational scale of the algorithm will increase exponentially with the increase in the data scale.

### 2.4 The optimized DPC algorithm

We come up with an optimized DPC algorithm which is aimed at dealing with the question of poor aggregation in low density areas and the division of noise points. This algorithm not only simply judges the distance between the noise point and the center of mass of each network region to judge the noise point to the attribution, but also introduces the SC measurement indicator and drawing on the idea of triangular stability between points and points, which reduces the impact of truncation distance, so as to more reasonably divide the network region. Next, in order to give a more reasonable deployment scheme, we calculate the tight centrality of all nodes in each network region, and find the optimal development scheme of controllers, so as to reduce the average communication delay of the whole network.

## 3 Methods

### 3.1 The evaluation index

There are three indicators for evaluating SDN network performance: the communication latency between the data layer and the control layer, the information synchronization latency at the control layer, and the controller load balancing. In this paper, we chiefly consider the communication latency between the control and data layers.

We know that the Euclidean distance needs to be used in the clustering algorithm, and there is no direct Euclidean distance between the switches in the SDN network. Therefore, we can imagine the SDN network as a topology and apply  $G(V, E)$  to describe the SDN Network Topology. This topology diagram is a weighted undirected graph where  $V$  represents a switch and  $E$  represents the physical link (the communication latency between switches and controllers).  $K$  indicates the quantity of controllers to be placed. At the same time, the amount of network partitions is worked out based on the magnitude of  $K$ , which represents the quantity of subnets divided by this algorithm.

By definition, the average latency is the quotient of the sum of the shortest latency from all SDN switches in the entire software-defined network to its controller and the number of switches. The average latency reflects the SDN network transmission, and the value of the overall latency is also an important performance indicator for evaluating the quality of an SDN network. The formula is as follows:

$$T_{\text{avg}} = \frac{1}{N} \sum_{v \in V} \min_{k \in K} d(v, k)$$

In this formula,  $d(v, k)$  represents the propagation communication delay between switch  $v$  and controller  $k$ ,  $v$  represents switch set,  $k$  represents controller set, and  $N$  is equal to the number of switches.

### 3.2 Algorithm optimization

This algorithm which we put forward is uniform to the cluster clustering problem. The optimized DPC algorithm is based on the DPC algorithm and optimizes the classification method of cluster-like data points to solve the problems in the DPC algorithm, such as: (1) truncation distance greatly affects the clustering results. (2) The point category judgment between adjacent clusters (noise point) is unreasonable. (3) The clustering results are not ideal at low density. Consequently, this article performs clustering in the light of the DPC algorithm clustering principle and makes related improvements, thus giving a more reasonable set of cluster points and data points, so as that we have ability to deal with these questions more reasonable. Thereby, the average communication latency can be reduced. So as to better introduce the optimized DPC algorithm, we introduce the DPC algorithm firstly.

The DPC algorithm is in the light of two basic assumptions:

**Assumption 1** The local density of the cluster center is greater than the density of the surrounding nodes.

**Assumption 2** The distance among the cluster focuses is comparatively far.

In order to find the cluster center which satisfies these two conditions, the DPC algorithm introduces the concepts of relative distance and local density.

The DPC algorithm works out the relevant information (relative distance and local density) of the data points. Next, it will generate the corresponding decision diagram where the abscissa is the relative distance and the ordinate is the local density. According to the data of the decision map which we generate, we set the points which have higher local density and relative distance as the cluster center, and then, we mark the points which have lower local density but higher relative distance as noise points. Next, what we need to do is allocate the remaining points on the line. The principle of allocation is to allocate every remaining point to the cluster center with its nearest neighbor and a local density greater than it. The DPC algorithm is in the light of fast searching and finds peaks in density which is a clustering algorithm essentially. Generally, this algorithm can be applied to the network partition. The main steps are as follows:

---

**Algorithm 1:** The standard DPC algorithm

**Step 1:** Compute the shortest path between the points

**Step 2:** Set the truncation distance and generate the decision graph which is including relative distance and local density

**Step 3:** Select the corresponding points from the decision graph according to the quantity of network regions. These points should meet the conditions which have high relative distance and local density

**Step 4:** After we finished **step1 – step3**, some points are sorted, and some noise points may be generated. Next, we allocate the noise points in the light of the distance between points and the cluster center

---

And now, we will give a detailed explanation of these four parts.

*Step 1* The distance between each data point will be entered as an algorithm in the form of a distance matrix.

*Step 2* Generate decision diagram. We compute the local density and the relative distance of every data point. Next, we generate the decision diagram which horizontal axis is local density and vertical axis is relative distance.

*Step 3* Select the cluster center in the midst of the data points. Using the relevant data of the decision diagram we generated in step 2, the point with higher local density and higher relative distance is marked as the cluster center, and the point which has higher relative distance but lower local density is marked as the noise point.

*Step 4* Complete the allocation of the remaining points. The principle of allocation is to allocate every remaining point to the nearest cluster center.

Nevertheless, the DPC algorithm has undeniable defects: firstly, this algorithm introduces the parameter of truncation distance in computing the local density, and the setting and selection of this parameter are artificially performed in this process. The selection of the truncation distance directly affects the simulation results. Secondly, the selection of the cluster center point is completed through human-computer interaction, which increases the instability and the nondeterminacy of the simulation consequence. Once a point is misjudged, this will have a direct effect. There are noisy points in the DPC algorithm, and there are a number of uncertain points. The network topology graph is used to divide the network area to ensure that all nodes must be included. So as to decrease the influence of the truncation distance and ensure the generation of better cluster centers and better network partitions, we refer to relevant ideas in the BeeDPC algorithm and introduce the two indicators of cluster-to-cluster separation and cluster aggregation. What we have done can not only more accurately judge the clustering of data points, but also improve the sensitivity of the DPC algorithm to the data points between clusters, and enhance the stability and reliability of the results.

The parameters used in this mathematical model are summarized in Table 1.

The optimized DPC algorithm includes two network region partitions and one confirmation of the center of the sub-network region. The first clustering is a preliminary clustering of all data points which can classify all data points. And the second clustering is to perform secondary clustering on the switch nodes in dispute after classification to complete the network partition. Finally, we will confirm the location of the center point of each network area. The essence of the first clustering is to confirm the number of network regions. We quote the relevant parameters in the decision graph and generate the local density  $\rho_i$  and the relative distance  $\delta_i$  in the light of the DPC algorithm. Next, we introduce the parameter  $\lambda_i$  ( $\lambda_i = \rho_i \cdot \delta_i$ ), and arrange  $\lambda_i$  from small to large. The number of zones depends on the number of controllers  $K$  we deploy. The optimized DPC algorithm is as follows:

**Table 1** List of notation

Parameters	Meaning
$\rho_i$	Local density
$\delta_i$	relative distance
$d_{ij}$	Euclidean distance between switch node $s_i$ and switch node $s_j$
$d_c$	Truncation distance
$hp_i$	Relatively high density point
$c_j$	$l$ th subnet area
Sep	Sub-network regional separation
Comp	Degree of aggregation within the sub-network area
$v_i$	Controller nodes
$c$ or $k$	The number of controllers
$u_{ij}$	The degree to which the switch $s_j$ belongs to the controller node $v_i$
$s_i$	Switch nodes
$m$	Weight index

**Algorithm 2:** The optimized DPC network partition algorithm

**Step 1:** determine the shortest path among points on the authority of the Dijkstra algorithm, and generate the distance matrix

**Step 2:** Take the distance matrix as input and determine the data information of every point which contains relative distance and local density by DPC algorithm

**Step 3:** Classify the switch nodes according to the DPC algorithm for the first time

**Step 4:** Complete the second region division of switch nodes according to the idea of triangular stability, and record the disputed nodes

**Step 5:** Complete the judgement of the disputed area attribution according to the SC evaluation indicators

**Step 6:** Compute the center points of each network region according to comparing the closeness centrality of each switch node

Next, we will combine the questions in the article to give a detailed plan for each step.

### 3.2.1 Generate the decision graph by first clustering

The first step is to generate the corresponding decision graph for initial classification. The network topology diagram composed of switch nodes in an SDN network is an undirected graph, and there is no directly connected Euclidean distance. Therefore, we determine the shortest path between points by applying the Dijkstra algorithm and store these data in a matrix to replace the Euclidean distance. Because the SDN switches are randomly distributed in the region, we will two-dimensionally make the ground surface into a property with two attributes of horizontal coordinate  $x$  and vertical coordinate  $y$ , and set  $n$  switches as data points in two-dimensional space. set  $S = \{s_1, s_2, \dots, s_n\}$ .

This algorithm supplies two approaches to calculate the local density of switch nodes. We know that there are two methods to cope with this question which name is kernel distance method and cutoff distance method. For the calculation of local density, we can choose each one of these methods. The formulas for the cutoff and nuclear distance methods are as follows:

$$\rho_i = \sum_{i \neq j} \chi(d_{ij} - d_c), \chi(x) = \begin{cases} 1 & \text{if } d_{ij} - d_c < 0 \\ 0 & \text{others} \end{cases}$$



or

$$\rho_i = \sum_{i \neq j} e^{\left(\frac{d_{ij}}{d_c}\right)^2}$$

The principle of  $d_c$  is as follow: we sort the distances incrementally, and find the appropriate value in the first 1–2% of the sorted distance and use that as the value of the truncation distance  $d_c$ .

Next, sort the local density  $\rho_i$  of all switch nodes in descending order:

$$\rho_{q1} \geq \rho_{q2} \geq \rho_{q3} \cdots \geq \rho_{qn}$$

$\rho_{q1}$  is the point of maximum local density, and its relative distance is the maximum value (maximum distance) of the entire switch node set to that point. For other points, the relative distance of these points is the shortest distance between this point and other points with greater local density value.

$$\delta_i = \begin{cases} \max_{i \neq j}(\delta_j) & \text{if } \rho_i = \max(\rho_n) \\ \min_{j: \rho_j > \rho_n} & \text{others} \end{cases}$$

We generate a decision diagram based on the relative distance  $\delta_i$  and the local density  $\rho_i$  we obtained, where  $\rho_i$  is the horizontal axis and  $\delta_i$  is the vertical axis. Next, we introduce the parameter  $\lambda_i$ .

$$\lambda_i = \rho_i \cdot \delta_i$$

The first step is to sort  $\lambda_i$  in descending order. Next, we select the data point corresponding to  $\lambda_i$  according to the number of controllers. This switch node is used as the controller node which is assigned at the first time.

### 3.2.2 The first partition of the network and the classification of switch nodes

The purpose of this step is to classify and judge other switch nodes except the controller node. First, all switch nodes are allocated according to the allocation principle of the DPC algorithm, which means that each remaining node is allocated to the sub-network area where the nearest switch node whose local density is greater than him is located. Next, reclassify other switch nodes (including noise points) and label these nodes, which means finding the nearest high density value point  $hp_1$  to the next closest high density value point  $hp_2$ . If  $hp_1$  and  $hp_2$  belong to the same subnet area, then the node is part of subnet  $c_i$ . Otherwise, the ownership of the node is considered to be different, which indicates that it can belong to both  $hp_1$  and  $hp_2$ .

### 3.2.3 Secondary classification of controversial switch nodes

After all the switch nodes have completed the secondary classification, the objectionable nodes will be judged. After the first clustering, the nodes include classified switch nodes and disputed switch nodes. The classified nodes include the controller nodes and the switch nodes to which the sub-network area belongs. Next, we perform secondary clustering on the switch nodes that are in dispute. Since the switch nodes in dispute may

belong to the subnet  $c_i$  or  $c_j$ , we introduce here the two levels of inter-cluster separation and intra-cluster aggregation.

The concept is to indicate the degree of separation between network partitions and the degree of aggregation within the network partition, rather than simply judging the distance from an undefined switch node to a controller node in two network regions. The degree of separation between clusters represents the separation effect between two clusters. The higher the degree of separation between clusters, the better the separation effect of clusters is. The degree of clustering within a cluster represents the clustering effect of a class cluster. The lower the degree of clustering within a cluster, the better the clustering effect is. At the same time, we introduce the SC metric to represent the inter-cluster separation Sep and the intra-cluster aggregation degree Comp. The SC metric is defined as:

$$SC = \frac{Sep}{Comp}$$

Sep indicates the degree of separation between sub-network regions. The formula is as follows:

$$Sep = \frac{1}{c} \sum_{i=1}^c \sum_{j=1}^n \mu_{ij}^m \cdot \left| s_i - \frac{1}{c} \sum_{i=1}^c v_i \right|^2 \times \min_{i \neq i'} |v_i - v_{i'}|$$

where  $\max \mu_{ij} < \alpha \left( \alpha \in \left[ \frac{1}{c}, 0.6 \right] \right)$  and  $m$  generally ranging from 1.5 to 2.5. The degree of separation between sub-network regions indicates the degree of separation between different sub-network regions. A larger value indicates that the separation between different sub-network regions is more obvious.

Comp represents the degree of aggregation within the sub-network area, and the specific definition formula is as follows:

$$Comp = \sqrt{1 + \frac{1}{c} \sum_{j=1}^c \frac{1}{\lambda_i} \sum_{j=1}^{n_{e(i)}} \mu_{ij}^m |x_i - v_i|^2}, \lambda_i = \sum_{j=1}^{n_{e(i)}} \mu_{ij}$$

where  $\max \mu_{ij} < \alpha \left( \alpha \in \left[ \frac{1}{c}, 0.6 \right] \right)$  and  $m$  generally ranging from 1.5 to 2.5. The degree of aggregation within a sub-network area indicates the degree of compactness within a sub-network area. The smaller the value, the more compact the sub-network area.

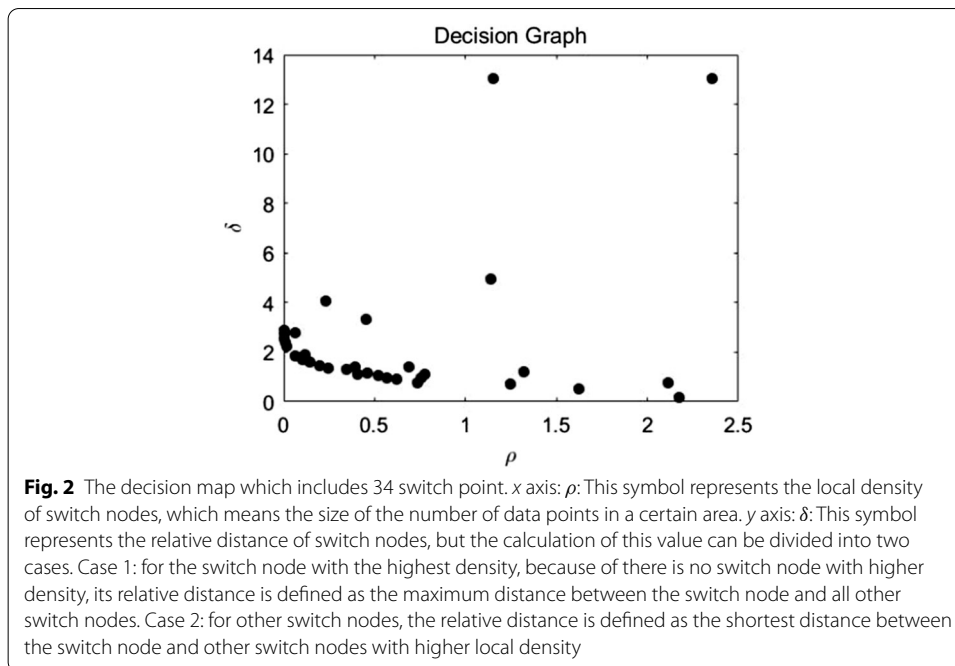
Therefore, the larger the SC, the better the network area division effect is. Calculate the SC metric for each uncertain switch node and select the sub-network area to which the SC metric is the largest.

### 3.2.4 Calculate the SC measurements for all disputed points until the partitioning of all switch points is complete

Repeat the evaluation until the clustering of all switch nodes is completed.

### 3.2.5 Introduce the closeness centrality index and confirm the final position of the controller

The node of the controller where we arrange is the center of the network in each network partition. Next, we calculate the closeness centrality of each node for the sake of



guaranteeing the stability and dependability of the network, and the position of the controller node in the corresponding network is obtained through the calculation results.

In this section, closeness centrality indicates how difficult it is for the node to reach other nodes. Its value is the inverse of the average distance from all other nodes, and the formula is defined as:

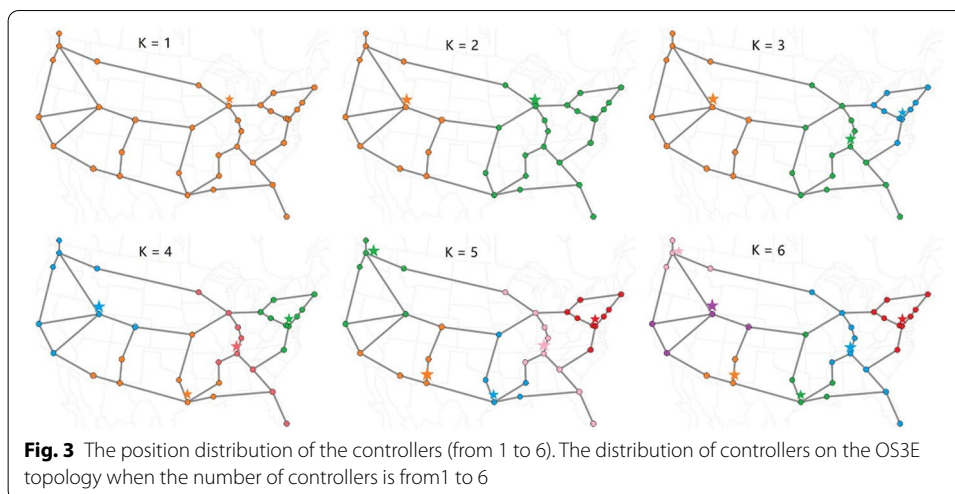
$$C_v = \frac{N}{\sum_{i \neq j} d_{is}}$$

Among them,  $N$  indicates the amount of switches that the network partition includes. Meanwhile,  $d_{is}$  represents the minimum communication latency betwixt the controller  $v_j$  and the switch  $s_i$ .

The greater centrality of a point, the shorter the communication latency betwixt the controller and switches is. Therefore, on the basis of step 1–step 5, we calculate the closeness centrality of each node in each network partition, and set the one which has the highest closeness centrality as the final deployment location of the controller.

#### 4 Simulation on the OS3E topology

In this section, so as to verify the correctness and stability of the algorithm we perform simulation experiments on the Internet2 network topology. In this SDN network topology diagram, each city is regarded as a switch node, and the distance between nodes is calculated at the same time. Meanwhile, we assume that light travels at  $v$  in a vacuum, but it does not travel at the same speed in other media. In ordinary fibers, the effective speed of light is reduced by 31%, so we determine the communication latency among nodes in the light of dividing the distance between points by two-thirds the speed of



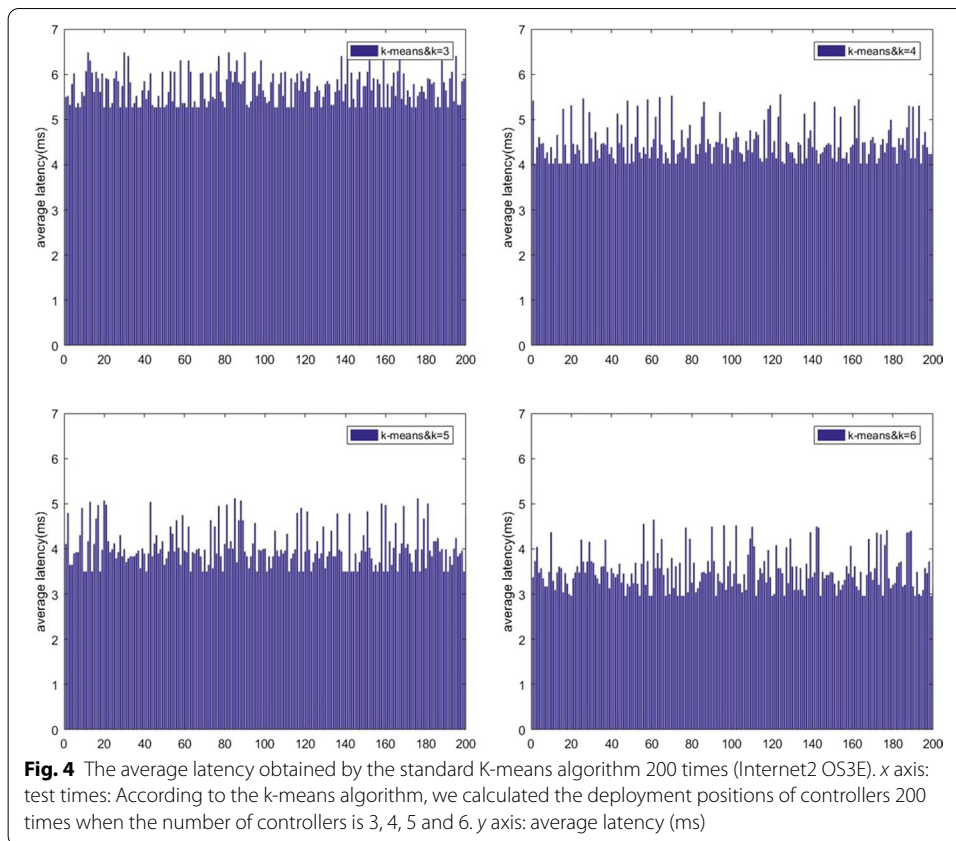
light. Next, we calculate the shortest path which is based on the Dijkstra algorithm to replace the Euclidean distance between points. First, the decision graph of the topology graph is calculated, and the result is shown in Fig. 2.

Next, based on the triangle stability idea of the BeeDPC algorithm and the judgment of the SC metrics, the sub-network area belonging to the switch node is judged. The position of the controller is determined by optimized DPC algorithm. In general, the average latency will decrease with the quantity of controllers increasing in the network. The distribution of the quantity of controllers from 1 to 6 is shown in Fig. 3.

It can be easily seen that the placement positions of the controllers are at the center of the cluster-like cluster, which can ensure that the network topology diagram has good reliability and stability. In light of the clustering principle the average latency from the switch to its controller in its network partition is also the smallest. Figure 3 demonstrates the optimal position for the number of controllers from 1 to 6. We found that we should place the controller in Chicago when the number of controllers is 1 so that we can minimize the average latency. And the controller should be placed in Chicago and Salt Lake City when the number of controllers is 2. When the number of controllers is 3, the controllers should be located in Washington, Salt Lake City, and Nashville so that we can minimize the average latency. And when the number of controllers is 4, the controllers should be located in Washington, Nashville, Houston, and Seattle. When the number of controllers is 5, the controllers should be located in Washington, Nashville, Houston, El Paso, and Seattle. Finally, the controllers should be located in Washington, Nashville, Houston, El Paso, Salt Lake City and Seattle when the number of controllers is 6. These locations can balance high density and low density, so as to better control the average latency in the SDN network and improve the utilization efficiency of the network.

## 5 Performance evaluation

In this section, we obtained the average communication latency in different quantity of controllers when it was calculated by k-means algorithm, optimized k-means algorithm and optimal placement algorithm, and compared the simulation results of the three algorithms with the results of the optimized DPC algorithm.

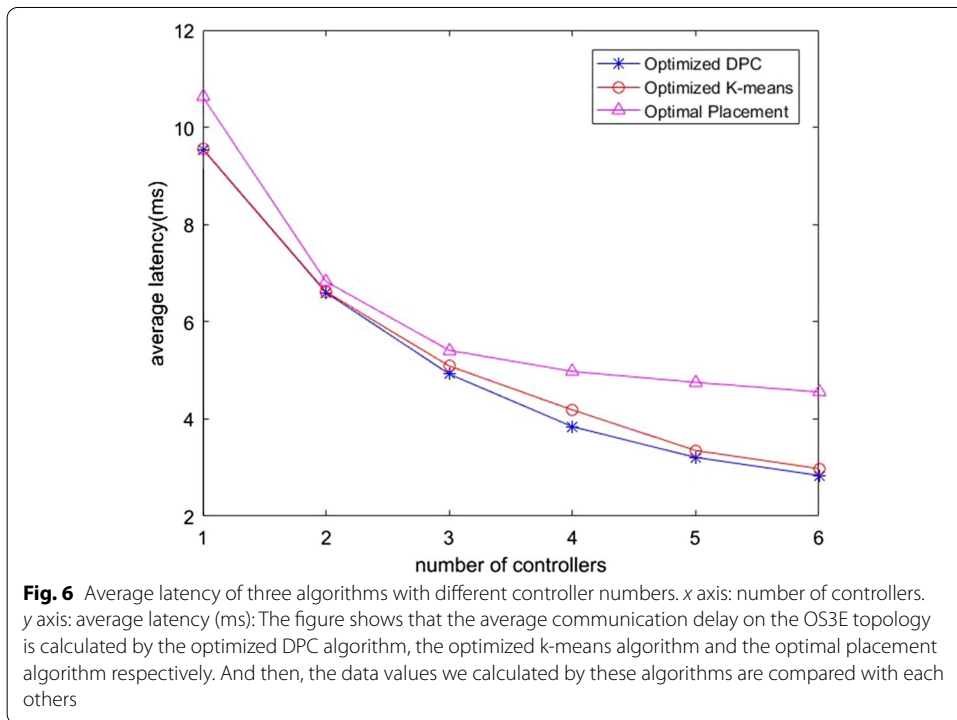
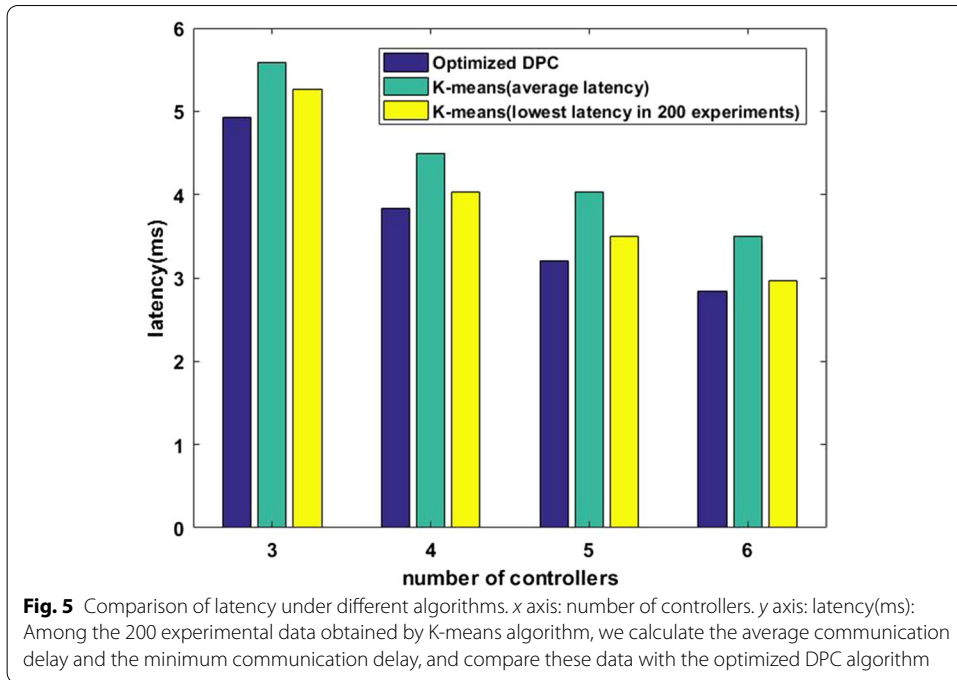


First we compare K-means algorithm with the optimized DPC algorithm. In order to verify the communication latency when the number of controllers is 3, 4, 5 and 6, we ran the standard K-means algorithm on the Internet2 OS3E network topology diagram 200 times and obtained 200 sets of communication Latency data. As shown in Fig. 4.

Next, calculate the average of the results obtained from the 200 experiments, record the minimum latency when the controller is 3, 4, 5, 6 and compare it with the results of the optimized DPC algorithm, as shown in Fig. 5.

On the basis of the data in the above figure, we know that when the number of controllers is 3, 4, 5 and 6, we can know that the average latency obtained by optimized DPC algorithm is lower than K-means algorithm from this picture. We know that k-means is a traditional clustering algorithm. The clustering results of this algorithm are often not globally optimal. Meanwhile, the processing of noise points by k-means algorithm is only a simple process. But some new definitions of the Optimized DPC algorithm address them. Therefore, it can be seen from the figure that with the increasing number of controllers, both the average delay and the minimum delay of k-means algorithm are lower than optimized DPC algorithm. Therefore, optimized DPC algorithm has obvious progress and improvement in solving optimized clustering and noise point processing.

The average communication latency of the optimized DPC algorithm, optimized K-means algorithm and optimal placement algorithm under different quantity of controllers is shown in the following Fig. 6.



As we can see from this picture that when the quantity of controllers is from 1 to 6, the average latency we calculated by the optimized DPC algorithm is the lowest among these algorithms, and we know that the lower the time delay, the more efficient the algorithm.

Therefore, we found an algorithm that can effectively reduce the network communication delay of OS3E topology.

When  $k=1$ , the average latency of the optimized K-means algorithm is 1.001 times that of optimized DPC algorithm, and the average latency of the optimal placement algorithm is 1.113 times that of optimized DPC algorithm.

When  $k=2$ , the average latency of the optimized K-means algorithm is 1.002 times that of optimized DPC algorithm, and the average latency of the optimal placement algorithm is 1.034 times that of optimized DPC algorithm.

When  $k=3$ , the average latency of the optimized K-means algorithm is 1.033 times that of Optimized DPC algorithm, and the average latency of the optimal placement algorithm is 1.099 times that of optimized DPC algorithm.

When  $k=4$ , the average latency of the optimized K-means algorithm is 1.089 times that of optimized DPC algorithm, and the average latency of the optimal placement algorithm is 1.296 times that of optimized DPC algorithm.

When  $k=5$ , the average latency of the optimized K-means algorithm is 1.044 times that of optimized DPC algorithm, and the average latency of the optimal placement algorithm is 1.481 times that of optimized DPC algorithm.

When the quantity of controllers is 6, the average latency of the optimized K-means algorithm is 1.0498 times that of optimized DPC algorithm, and the average latency of the optimal placement algorithm is 1.605 times that of optimized DPC algorithm.

Therefore, the optimized DPC algorithm has a significant effect on reducing the average communication latency when compared with the three algorithms which include K-means algorithm, the optimal placement algorithm and the optimized K-means algorithm.

## 6 Cost–benefit evaluation

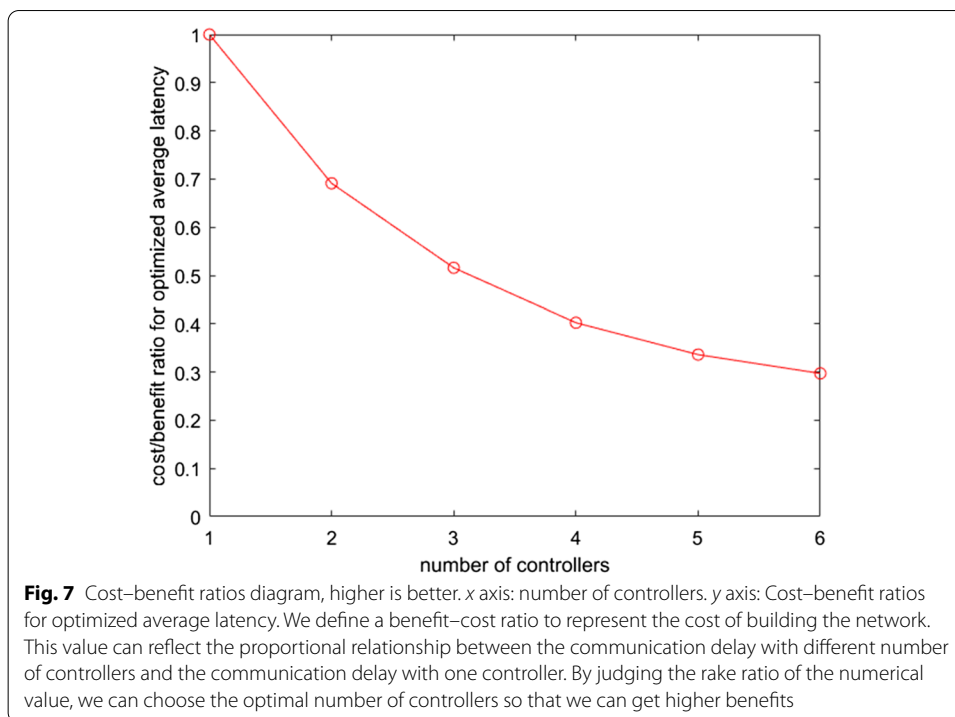
The question of how many controllers should be used which depends on the different network topologies. Different network topologies have different numbers and locations of deployed controllers. In V part of the simulation experiment we learn that we can determine the average delay of the internet2 OS3E network topology with different numbers of controllers in the light of the optimized DPC algorithm. At the same time, the average communication latency will gradually decrease with the quantity of controller increasing. But choosing the right number of controllers directly affects the cost of building the network.

Here, we define a benefit–cost ratio, the formula is:

$$\text{ratio} = \frac{al_k}{al_1}$$

where  $al_k$  represents the average latency when the quantity of controllers is  $k$  and  $al_l$  represents the average latency when the quantity of controllers is 1. Accordingly, on the basis of this formula we get the benefit–cost diagram (Fig. 7).

We can see that the benefit–cost ratio is gradually decreasing. We need three controllers to cut down half of the average communication latency that when the quantity of controllers is 1, and its slope begins to decrease with the quantity of controllers



increasing. Consequently, we can reduce the cost to the greatest extent and obtain higher benefits when  $k = 3$ .

### 7 Discussion and conclusion

Because of the restriction and obstruction of realistic factors, the traditional network is more and more difficult to meet the needs of the development of modern network, which results in the emergence of SDN network. At the moment, the deployment of the controller in SDN network also affects the communication quality in the Internet. In the existing SDN controller deployment research scheme, the main deployment algorithm is to divide the network area. When compared with several existing controller deployment algorithms, the controller deployment scheme obtained by optimized DPC algorithm can effectively reduce the communication delay in SDN, which can also improve the reliability and stability of the network.

But at the same time, there are some limitations in this study. For example, in the process of designing the controller deployment algorithm, we did not consider the traffic load between the controller and the switch, but just divided the SDN network area. Although we found a more optimized controller deployment scheme, the management overhead of network traffic may be large in the actual network traffic scheduling process.

In the course of our research, it is known that the design of the controller deployment scheme is to divide the existing network area into several sub-network areas, but it is a huge problem to choose which method to divide reasonably. After compared with several existing controller deployment algorithms, we find that they all have some problems. For example, the K-means algorithm adopts iterative algorithm and only considers the local optimization of controller deployment. Therefore, in order to reduce the impact



of these problems, we found the DPC algorithm which is a clustering algorithm based on fast search and peak density detection. This algorithm can overcome the shortcomings of general clustering algorithm in data requirements, and achieve efficient clustering of arbitrary shape data, which is more in line with the deployment of real switches.

However, this algorithm only calculates the distance between the points of some sub-regions and the center points of each sub-region, then compares the distance and divides the points into the nearest region. But in some cases, this simple calculation method is obviously not feasible. Therefore, we need to introduce some other measurement schemes to calculate this point. So we find the SC metric, which is a reasonable way to divide these points by comparing the intra cluster compactness and inter cluster separation. In this way, the impact of the original DPC algorithm defects can be reduced. Perhaps the impact of this defect application in the actual situation is very small, the impact may be just a dozen of switches. However, in the case of global consideration, our measurement algorithm can optimize the controller deployment scheme as a whole.

The optimal deployment of controllers in SDN networks is a huge challenge and one of the difficulties faced by the future development of modern networks. So that to solve the controller layout question, the thought of partitioning the network region is put forward. In this article, we use the DPC clustering algorithm as the basis, and combine the idea of triangle stability in the BeeDPC algorithm with the SC metric so that we can solve the problem of controller placement in the WAN more reasonable, which means that we can mathematically model the placement of the controller from the perspective of reducing the average communication latency. At the same time, we also introduce the closeness centrality to calculate the position of the controller. The controller deployment problem is still a clustering problem in nature, so we can simplify the controller deployment problem to the problem of finding the optimal cluster center point. We perform simulation experiments on the Internet2 OS3E network topology diagram. At the same time, it had already compared with several kinds of algorithms under different quantity of controllers. The simulation experimental results which are in the light of the optimized DPC algorithm demonstrate that the novel scheme of the controller deployment has the lowest average latency, which can effectively improve network performance and improve communication quality between switches. Finally, we know that when the number of controllers is three, we can minimize the network construction cost on the OS3E network topology map and achieve greater benefits.

#### **Abbreviations**

SDN: Software-defined network; IoT: Internet of thing; NFV: Network functions virtualization; SC: Separation and compactness; WAN: Wide area network.

#### **Authors' contributions**

XC and XG proposed the idea and revised this paper. XG and XC wrote the manuscript. XG participated in the simulation. YM and WW gave some suggestions. YM and WW participated in the paper revision. All authors read and approved the final manuscript.

#### **Funding**

This work was supported by the CERNET Innovation Project of China (NGII20190110).

#### **Competing interests**

The authors declare that they have no competing interests.

#### **Author details**

<sup>1</sup> School of Computer Science and Technology, Shandong University of Technology, Zibo, China. <sup>2</sup> Institute of Network Technology, Beijing University of Posts and Telecommunications, Beijing, China. <sup>3</sup> School of Bioinformatics, Chongqing

University of Posts and Telecommunications, Chongqing, China. <sup>4</sup> School of Information and Mechanical Engineering, Shanghai Normal University, Shanghai, China.

Received: 26 March 2020 Accepted: 21 October 2020

Published online: 03 December 2020

## References

1. S. Sun, M. Kadoch, L. Gong, B. Rong, Integrating network function virtualization with SDR and SDN for 4G/5G networks. *IEEE Netw.* **29**(3), 54–59 (2015)
2. N. Zhang, N. Cheng, A.T. Gamage, K. Zhang, J.W. Mark, X. Shen, Cloud assisted HetNets toward 5G wireless networks. *IEEE Commun. Mag.* **53**(6), 59–65 (2015)
3. Y. Wu, B. Rong, K. Salehian, G. Gagnon, Cloud transmission: a new spectrum-reuse friendly digital terrestrial broadcasting transmission system. *IEEE Trans. Broadcast.* **58**(3), 329–337 (2012)
4. B. Rong, Y. Qian, K. Lu, H. Chen, M. Guizani, Call admission control optimization in WiMAX networks. *IEEE Trans. Veh. Technol.* **57**(4), 2509–2522 (2008)
5. N. Chen, B. Rong, X. Zhang, M. Kadoch, Scalable and flexible massive MIMO precoding for 5G H-CRAN. *IEEE Wirel. Commun.* **24**(1), 46–52 (2017)
6. L. Xie, Y. Ding, H. Yang, X. Wang, Blockchain-based secured and trustworthy internet of things in SDN-enabled 5G-VANETS. *IEEE* **7**, 56656–56666 (2019)
7. D. Sinh, L.-V. Le, L.-P. Tung, B.-S. Paul Lin, The challenges of applying SDN/NFV for 5G&IoT, in *2018 IEEE 5G World Forum (5GWF)* (At Silicon Valley, CA, 2018), pp. 138–142
8. A. Muthanna, A.A. Ateya, A. Khakimov, I. Gudkova, A. Abuarqoub, K. Samouylov, A. Koucheryavy, Secure and reliable IoT networks using fog computing with software-defined networking and blockchain. *J. Sens. Actuator Netw.* **8**(1), 128–136 (2019)
9. B.P.R. Killi, S.V. Rao, Towards improving resilience of controller placement with minimum backup capacity in software defined networks. *Comput. Netw.* **149**, 102–114 (2019)
10. T. Hu, Z. Guo, P. Yi, T. Baker, J. Lan, Multi-controller based software-defined networking: a survey. *IEEE Access* **6**, 15980–15996 (2018)
11. P.C.D.R. Fonseca, E.S. Mota, A survey on fault management in software-defined networks. *IEEE Commun. Surv. Tuts* **19**(4), 2284–2321 (2017)
12. Y. Zhang, L. Cui, W. Wang, Y. Zhang, A survey on software defined networking with multiple controllers. *J. Netw. Comput. Appl.* **103**, 101–118 (2018)
13. A.K. Singh, S. Srivastava, A survey and classification of controller placement problem in SDN. *Int. J. Netw. Manag.* **28**(3), e2018 (2018)
14. G. Wang, Y. Zhao, J. Huang, W. Wang, The controller placement problem in software defined networking: a survey. *IEEE Netw.* **31**(5), 21–27 (2017)
15. B. Heller, R. Sherwood, N. Mckeown, The controller placement problem, in *Proceeding of the 1st Workshop on Hot Topics in Software Defined Networks* (ACM Press, New York, 2012), pp. 7–12
16. G. Yao, J. Bi, Y. Li, L. Guo, On the capacitated controller placement problem in software defined networks. *Commun. Lett.* **18**(8), 1339–1342 (2014)
17. Q. Zhong, Y. Wang, W. Li, X. Qiu, A min-cover based controller placement approach to build reliable control network in SDN, in *Proceedings of IEEE/network operations and management symposium (NOMS)* (2016), pp. 481–487
18. T. Yuan, X. Huang, M. Ma, J. Yuan, Balance-based SDN controller placement and assignment with minimum weight matching, in *Proceedings of IEEE International Conference on Communications (ICC)* (2018), 1–6
19. T. Wang, F. Liu, J. Guo, H. Xu, Dynamic SDN controller assignment in data center networks: stable matching with transfers, in *Proceedings of IEEE INFOCOM 35th Annual IEEE International Conference on Computer Communications* (2016), pp. 1–9
20. Y. Hu, W. Wang, X. Gong, X. Que, S. Cheng, On reliability-optimized controller placement for software-defined networks. *China Commun.* **11**(2), 38–54 (2014)
21. B.P.R. Killi, E.A. Reddy, S.V. Rao, Cooperative game theory based network partitioning for controller placement in SDN, in *Proceedings 10th International Conference on Communication Systems and Networks (COMSNETS)* (2018), pp. 105–112
22. A. Jalili, M. Keshtgari, R. Akbari, R. Javidan, Multi criteria analysis of controller placement problem in software defined networks. *Comput. Commun.* **133**, 115–128 (2019)
23. A. Jalili, M. Keshtgari, R. Akbari, Optimal controller placement in large scale software defined networks based on modified NSGA-II. *Appl. Intell.* **48**(9), 2809–2823 (2018)
24. A. Chehour, R. Younes, J. Perron, A. Ilinca, A constraint handling technique for genetic algorithms using a violation factor. *J. Comput. Sci.* **12**(7), 350–362 (2016)
25. F. Bannour, S. Souihi, A. Mellouk, Scalability and reliability aware SDN controller placement strategies, in *Proceedings of 13th International Conference on Network and Service Management (CNSM)* (2017), pp. 1–4
26. H. Kim, N. Feamster, Improving network management with software defined networking. *IEEE Commun. Mag.* **51**, 114–119 (2013)
27. M.S.A. Vanjari, D.R.B. Ingle, P.G. Student et al., Network traffic control using distributed control plane of software defined networks. *Int. J. Comput. Sci. Trends Technol.* **3**(5), 5–12 (2015)
28. A.R. Curtis, J.C. Mogul, J. Tourrilhes, P. Yalagandula, P. Sharma, S. Banerjee, DevoFlow: scaling flow management for high performance networks. *ACM SIGCOMM Comput. Commun. Rev.* **41**(4), 254–265 (2011)
29. S. Hassas Yeganeh, Y. Ganjali, Kandoo: a framework for efficient and scalable offloading of control applications, in *Proceedings of the First Workshop on Hot Topics in Software Defined Networks* (ACM, 2012), pp. 19–24

30. A. Tootoonchian, Y. Ganjali, HyperFlow: a distributed control plane for OpenFlow, in *Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking* (USENIX Association, 2010), p. 3
31. A.-W. Tam, K. Xi, H.J. Chao, Use of devolved controllers in data center networks, in *2011 IEEE Conference on Computer Communications Workshops (INFOCOM WKSHPS)* (IEEE, 2011), pp. 596–601
32. A. Rodríguez, A. Laio, Clustering by fast search and find of density peaks. *Science* **344**, 1492–1496 (2014)
33. L. Ge, G. Xu, Collaborative filtering algorithm based on the effectiveness of fuzzy c-means clustering. *Comput. Technol. Dev.* **1**, 22–26 (2018)
34. H.K. Al-Mohair, J.M. Saleh, S.A. Suandi, Hybrid human skin detection using neural network and K-means clustering technique. *Appl. Soft. Comput.* **33**, 337–347 (2015)
35. C. Jose, L.A.R. Calla et al., Parallelization of the algorithm K-means applied in image segmentation. *Int. J. Comput. Appl.* **88**(17), 237–256 (2014)
36. M.E. Celebi, H.A. Kingravi, P.A. Vela, A comparative study of efficient initialization methods for the k-means clustering algorithm. *Expert Syst. Appl.* **40**(1), 200–210 (2013)
37. Y. Zhang, N. Beheshti, M. Tatipamula, On resilience of split-architecture networks, in *2011 IEEE Global Telecommunications Conference-GLOBECOM* (IEEE Press, New York, 2011), pp. 1–6
38. S. Guo, S. Yang, Q. Li, et al., Towards controller placement for robust software-defined networks, in *2015 IEEE 34th International Performance Computing and Communications Conference (IPCCC)* (IEEE Press, New York, 2015), pp. 1–8

### Publisher's Note

Springer Nature remains neutral with regard to jurisdictional claims in published maps and institutional affiliations.

Submit your manuscript to a SpringerOpen<sup>®</sup> journal and benefit from:

- ▶ Convenient online submission
- ▶ Rigorous peer review
- ▶ Open access: articles freely available online
- ▶ High visibility within the field
- ▶ Retaining the copyright to your article

---

Submit your next manuscript at ▶ [springeropen.com](https://www.springeropen.com)

---