Research on SDN Multi Controller Deployment based on K-means++

Li Yue¹, Chen Junyan*¹,², Liang Chuxin¹, Lei Xiaochun¹,³

¹(School of computer and information security, Guilin University of Electronic Technology, Guilin 541004, China; ²Guangxi cloud computing and big data collaborative innovation center, Guilin 541001, China; ³Guangxi key laboratory of image and graphic intelligent processing, Guilin 541004, China)

*Corresponding author’s e-mail: 56546632@qq.com

Abstract: In large-scale SDN network, a single centralized controller can not meet the demand, and multiple controllers are needed to deal with the problem, which leads to the problem of multi control balanced deployment. In this paper, the topology of SDN switches and links is known. The main research contents are as follows: the mathematical model of SDN multi controller deployment is established, and the appropriate network topology is selected and the multi controller deployment problem is solved. In the research, the number of controllers needed in the network and the switches managed by each controller is determined by the algorithm results, and the mapping relationship between controllers and switches is established. By analyzing the deployment results of the same network topology under different algorithms, the influence of different clustering algorithms on the experimental results is obtained. At the same time, the better deployment experiment results are simulated.

1. Introduction
In a large network, it is difficult for a single controller to handle the requests of many switches. The controller is faced with the problem of too high load, which will cause the network performance degradation or congestion. It needs multiple controllers to share the pressure of the whole network. If multiple controllers are deployed in different locations, the delay and load balance of the whole network will be different, and the performance of the whole network will also be different. At the same time, if the controller does not balance the network traffic properly, it will also affect the network balance. Therefore, the way to realize the balanced deployment of multiple controllers is an important topic of SDN network research.

2. Research on SDN Multi Controller Deployment
In the network, different network nodes need to communicate with each other, sometimes need to transmit information frequently, so the delay between switches and the delay between switches and controllers will have an impact on the speed of information transmission. When a new data flow passes through the switch, the switch needs to query the controller, the switch sends the Packet-In message to its controller, and the controller returns the control information after processing, and the processing scheme of the data flow got as well, then the query delay produced\(^1\).
In this paper, clustering algorithm is used to divide the network, so as to complete the deployment of multiple controllers. Based on the complexity of the algorithm model and the difficulty of the simulation model, when clustering, it only considers the relationship between the links of nodes to cluster, that is, whether there is connectivity between nodes. The results of the two algorithms are compared and a further conclusion is drawn. In the simulation experiment, we need to further consider the actual situation to complete the balanced deployment of multiple controllers[4].

This paper mainly studies the SDN multi controller balanced deployment problem. Given a network topology and the link relationship between network nodes and network nodes, the network is divided by clustering algorithm. Then the placement position of multiple controllers and the switch composition managed by the controller are obtained. This research is divided into two parts: theory and experiment. The theoretical part mainly includes the establishment of SDN multi controller deployment model, the selection of algorithm, the description of controller load balancing problem and the selection of network topology[7]. The part of simulation experiment mainly uses mininet to simulate. The simulated topology is the network topology with better effect of multi controller placement. After the corresponding network topology is created, the switch is connected to the corresponding controller according to the algorithm results. The controller runs the network traffic balancing program to check the network connectivity and network traffic.

3. SDN Multi Controller Deployment Model

Limited by the processing capacity and bandwidth of the controller, the number of switches a controller can manage is limited, and the load of the controller should not be exceeded when allocating the switches it manages to the controller[2]. At the same time, considering the propagation delay between switches and controllers, the propagation delay is proportional to the distance between links and inversely proportional to the bandwidth. In this paper, we only consider the link situation, and implement the deployment strategy of multi controller by clustering algorithm. In the SDN multi controller deployment model in this paper, the location of controller deployment is selected from the switch nodes, so the delay from the ideal controller to the deployed switch nodes is 0, and there is no separate physical link between the controller and the switch[6][10].

The SDN network topology is abstracted as an undirected graph \( G = (V, e) \). The switch node in the network is equivalent to the point in the undirected graph, and the physical link between the switch and the switch is equivalent to the edge of the undirected graph. And the \( V \) represents the set of all nodes in the network, the \( E \) represents the set of all edges in the network[2][3]. Assuming that the whole network needs to deploy \( K \) controllers, the network will be divided into \( k \) classes, and the switches in each class will be managed and controlled by a unique controller. The number of switches in the network is \( N \). And the number of controllers is \( K \).

Set of switches. The formula is shown as follows:

\[
V = \{v_1, v_2, ..., v_N \mid v_i \in R, i = 1,2, ..., N\}
\]  (1)

Set of controllers. The formula is shown as follows:

\[
C = \{c_1, c_2, ..., c_K \mid c_j \in R, j = 1,2, ..., K\}
\]  (2)

Set of nodes deployed by the controller. The formula is shown as follows:

\[
M = \{c_j \mid c_j \in V, j \in C\}
\]  (3)

Switch managed by controller j. The formula is shown as follows:

\[
C_j = \{v \mid v \in V, v \text{ is managed by controller } j\}
\]  (4)

Physical link between switches. The formula is shown as follows:

\[
d(m, n), m \in V, n \in V
\]  (5)

The \((m, n)\) represents the shortest path from switch \( m \) to switch \( n \), only considering whether there is a connection relationship between links, the weights of the connected edges between switch nodes are set to 1.

Based on the demand of load balancing among multiple controllers, the concept of controller load difference degree is proposed. The specific calculation method selected in this paper is shown as
follows:

\[ \frac{1}{K} \sum_{j=1}^{K} \left| C_j - \frac{N}{K} \right| \]  

At the same time, in order to deploy SDN multi controller better, this model puts forward other constraints, hoping to get more appropriate deployment results, and the research content is more perfect. In the actual SDN network, the cost of the controller is relatively high, so we hope to deploy as few controllers as possible while meeting the requirements, and the number of switches managed by a single controller does not exceed the given upper limit [2].

4. SDN Multi Controller Deployment Algorithm

4.1. k-means Algorithm

The central idea of K-means algorithm is to randomly generate k points as the clustering centers of K classes. By calculating the distance from all points to these clustering centers, these points can be classified into the nearest cluster. Then reselect the clustering center in the K categories that have been divided, recalculate the center points according to the previous steps, and classify the rest points until the results no longer change. K-means algorithm has several obvious defects. First of all, the value of K can only be given in advance, which is difficult to determine. Secondly, K-means is random at the beginning of cluster center selection. If there is one point, which is far away from other points, it is likely to become an isolated point. Finally, if there are many data features, the cluster center needs to be calculated many times, which will increase the cost of the algorithm [3][5].

4.2. k-means++ Algorithm

K-means++ algorithm in order to solve the problem that the selection of clustering centers in k-means algorithm is too random, and the results are too sensitive to the initial clustering centers, the way of selecting the initial clustering centers is improved. In this algorithm, the nodes which are far away from each other are selected as clustering centers. For the selection of the initial clustering center, select a random point from the existing points as the first clustering center, and for the other points x in the data, calculate the distance D (x) between point X and the point already selected as the clustering center, and then select a point as the new clustering center. The basic principle of selecting a new clustering center is: the point with larger D (x) has higher probability to be selected. Repeat the previous steps of selecting cluster centers until K cluster centers are selected. After selecting the initial K clustering centers, the next steps are the same as the previous k-means algorithm.

K-means++ generally solves the problem of data model is also a point in space. When this algorithm is used to cluster network topology, the steps of clustering method are the same as k-means algorithm.

4.3. Network Traffic Balancing Algorithm

After the algorithm is implemented, the corresponding placement of multi controllers can be obtained. However, if the controller does not choose the appropriate strategy for the network it manages, there may be too many packets transmitted in a single link, and the problem of unbalanced transmission of network traffic. With the development of SDN and OpenFlow protocol, it is possible to find multiple routes in the network by programming the network, and give certain weight to these routes, so that a large number of packets in the network can reach the destination host from different paths [8].

With DFS algorithm, all possible paths can be found. After obtaining all paths, we must judge the quality of these paths. The standard of judgment is the cost of this path. In different routing algorithms, the cost calculation of this path is different. Please refer to the calculation method in OSPF. In the interface cost calculated by OSPF protocol, the interface cost is inversely proportional to the interface bandwidth. Define a reference bandwidth constant. According to the ratio of this constant and link bandwidth, get the cost, and calculate the cost PW formula of a single link. The formula is shown as follows:

\[ \frac{1}{K} \sum_{j=1}^{K} \left| C_j - \frac{N}{K} \right| \]
pw = 10000000/bw bps

With the cost of a single link, the cost of the whole path can be calculated. Assuming that the whole path has n links, the calculation formula of the cost PW of the whole path is as shown in formula (8):

\[ PW = \sum_{i=1}^{n} pw_i \]  

(8)

Next, calculate the bucket weight of each path, assuming that there are n paths. For path P, the calculation formula is as shown in formula (9):

\[ bw(p) = \left(1 - \frac{PW(p)}{\sum PW(0)}\right) \times 10 , 0 \leq bw(p) < 10 \]  

(9)

When choosing the path, the ideal path is to choose the path with lower cost, but in order to balance the network traffic, we also need to choose other paths. By calculating the bucket weight, we can get the following conclusion: the path cost lower, the bucket weight will be higher. The weight of buckets in OpenFlow protocol higher, the priority of selection will be higher.

The controller can convert these rules into OpenFlow rules through the above calculation. If the switch has multiple ports with paths, the creation type is select (OFPCT_SELECT), otherwise install the normal flow table. After these things are installed in the corresponding OpenFlow switch managed by the controller, the switch no longer needs to ask the controller how to forward the data flow every time[9].

5. Experiment and Analysis

In SDN network, if the problem of multi controller deployment needs to be considered, it shows that the network scale is large, because when a single controller can meet the requirements of network performance, it is not necessary to deploy multi controller. Therefore, when deploying multiple controllers, we should choose a network topology with more nodes, more complex links, and no rules to find. In the previous section, we have established the relevant deployment model for SDN multi controller deployment problem and introduced the selected relevant algorithm. On this basis, we need to select the network topology used. The selected network topology should meet the conditions that the number of switch nodes is large, irregular and without isolated points, so if the network topology is abstracted into a graph model, it should be a connected graph. Due to the experimental comparison and analysis, two network topologies are selected for comparison, and their structures are shown in Figure 1 and Figure 2 respectively.

Because k-means algorithm and K-means ++ algorithm use random function in the process of execution, the result of each execution may be different. However, from the perspective of generality, the result with the most times appears in the case of multiple runs, is also the most likely solution that the algorithm can find under the current conditions. This solution has great reference value for clustering effect analysis of this algorithm. The results of these two algorithms in Noel network and ALS2 network are listed below, which are the results when k value of the algorithm is taken as 4 and 5 respectively.

In order to facilitate the analysis and visual comparison of the results, the nodes divided into the same class are represented by the same color, and the central node in the same class is larger than other nodes. The results are shown in figures 3 to 6.
Figure 3 K-means algorithm divides Noel network into five categories, with the central node of 2, 4, 7, 12, 16

Figure 4 K-means ++ algorithm divides Noel network into four categories, with the central node of 2, 7, 12, 15

Figure 5 The k-means algorithm divides the ALS2 network into four categories, with the central nodes of 6, 12, 28, 37

Figure 6 The K-means ++ algorithm divides the ALS2 network into four categories, with the central node of 6, 12, 28, 37

According to the classification and the load difference formula of the controller, the data shown in figure 7 and figure 8 are obtained.

Figure 7 Load difference under Noel network

Figure 8 Load difference under ALS2 network

The experimental results show that the two algorithms are effective when the network is divided into four classes. In the network with fewer nodes, K-means algorithm performs better, and in the network with more nodes, K-means ++ algorithm performs better. At the same time, when the two algorithms partition the same class, the central node is similar. In the implementation of these two algorithms, each time the center node is reselected, the distance sum of all nodes in the class will be calculated, and the minimum point of the distance sum will be reselected as the center node. Therefore, the center nodes in the algorithm's execution results are relatively close, which means that it is more appropriate to select these nodes as the center nodes. Because the initial nodes of the algorithm are randomly assigned, and there is no effective constraint on the number of nodes in each class when classifying the nodes, the results may be uneven, or even the number may vary too much.

In the above algorithm results, when ALS2 network is divided into four classes, the implementation effect is the best. Finally, related experiments will be carried out with this network topology.
6. Summary and Prospect
This paper focuses on SDN multi controller deployment. In order to solve this problem, a mathematical model is established, which turns the problem of SDN network multi controller deployment into the problem of finding the undirected connected graph and dividing it into several subgraphs. Through clustering algorithm, the undirected connected graph is divided under the condition that the link relationship of network nodes is known, and the deployment result is obtained, which is the mapping relationship between controller and switch nodes. At the same time, these results are compared, and the network topology with better partition effect is selected for simulation experiment. When there are multiple paths to the destination, the switch can adopt a certain strategy, according to a certain standard, the packets are forwarded from different path ports.

Acknowledgment
This work is supported by Project to Improve the scientific research basic ability of Middle-aged and Young Teachers (No.2019ky0222, 2020KY05033), Student’s Platform for Innovation and Entrepreneurship Training Program under Grant (No.201910595092)

References
[1] Yang YT, Wang Q, Gao LR, et al. (2018) SDN multi controller deployment based on bat algorithm. J. Journal of Chongqing University. 41(09):57-65.
[2] Qin KY, Huang CH, Wang CH, et al. Balanced deployment of multiple controllers in SDN network limited by delay and capacity. J. Journal of Communications. 37(11):90-103.
[3] Yang YT. Research on controller deployment based on SDN. Tianjin University. 2018.
[4] Zhu W. Research on placement of multiple controllers based on load balancing and delay limitation in SDN. Hefei Polytechnic University. 2018.
[5] Xu H. Research on multi controller load balancing optimization algorithm based on delay constraint in software defined networks. Hefei Polytechnic University. 2019.
[6] Hao YY. Research on optimal deployment strategy of multi controller in SDN environment. Henan University. 2018.
[7] Chen FY. Research on Key Technologies of load optimization of multi controller in software defined network. PLA University of Information Engineering. 2015.
[8] Lan WJ. Research on multi controller load balancing technology based on SDN. Wuhan University of Technology. 2018.
[9] Sheng JQ. Research on multi controller cooperation mechanism and secure communication for SDN network. Anhui University. 2018.
[10] Yao J. Research on deployment strategy of controller in SDN. Chongqing University of Posts and Telecommunications. 2019.