Spectral clustering based approach for controller placement problem in software defined networking

Jie Lu, Zhen Zhang, Tao Hu
National Digital Switching System Engineering & Technological Research Center, Zhengzhou, China
luwangli@hotmail.com

Abstract. In a large-scale SDN, the network was effectively managed by deploying multiple controllers. In order to improve reliability and reduce delay in the existing multi-controller deployment scheme, we proposed a Multi-Controllers Elastic Placement (MCEP) algorithm based on reliability and delay. First, the multi-controller deployment problem was transformed into the row vector classification problem of the network topology matrix based on spectral clustering. Then, the k-medoids algorithm based on simulated annealing was used to classify the row vectors to achieve the flexible deployment of multiple controllers. The simulation results show that, compared with the existing algorithms, while guaranteeing low delay, the reliability of control paths increase by 17% on average.

1. Introduction
The logically centralized architecture in traditional networks has begun to shift toward the separation of control and data planes in software-defined networking (SDN) [1]. This shift facilitates the management and maintenance of the network and has attracted extensive attention from academics and industry. Initially, SDN was designed with only one single centralized controller, which actually results in some drawbacks such as long response delay and poor scalability. In order to improve the scalability of the network and avoid single point of failure, the researchers proposed to divide the network into multiple domains on the basis of centralized control and deploy multiple controllers to implement logically centralized and physically distributed SDN networks [2].

In recent years, the research of multi-controller placement problem (MCPP) has two main categories. The first schemes [3-6] consider the delay, load and other indicators, and solve the controller deployment position through multi-objective optimization. Such programs aim to improve the communication quality, the utilization, but ignore the influence of the apparatus failure. The second schemes [7-9] consider the reliability of the network. When the controller is deployed, the reliability of nodes and links is used as an indicator. The inadequacies are the lack of consideration for the delay of the establishment of new flows, and it is easy to cause unbalanced loads.

Based on the idea of spectral clustering, this paper proposes a multi-controller Elastic Placement (MCEP) algorithm to solve the problem of reliability and load optimization. MCEP first uses the similarity between nodes to represent the network topology as a similarity matrix, transforms the multi-controller deployment problem into a similar matrix row-vector classification problem, and then uses an improved k-medoids algorithm based on simulated annealing to classify the row vectors. Investigation of current literature, the realization of multi-control deployment through spectrum clustering is a new attempt, which provides a new mathematical model and ideas for the design and implementation of multi-controller deployment.
The rest of this paper is organized as follows. Section II describes the controller placement problem. Section III introduces the proposed approach. Simulation results are presented in Section IV and this paper is concluded in Section V.

2. Problem formulation

2.1. Network Model

We model the topology of the network as an undirected graph $G(V,E,B)$, where $V$ represents the set of the nodes (switches), $E$ represents the set of edges, and $B$ represents the set of bandwidth. $|V|$ denote the cardinalities of $V$, $e_{ij}$ and $b_{ij}$ represents the length and bandwidth of the direct links between the nodes.

$V_c$ represents the position of the controller, $V_c \subseteq V$, and $V'_j$ indicating the location of the controller $j$. Each controller manages a domain, and the controller $j$ manages sets of switches $SC_j = (s_1, s_2, \ldots, s_n)$, $|SC_j|$ representing the number of switches, correspondingly $SC'_j$ indicates that switch $i$ controlled by controller $j$. For any given node $s$ and $t$, $P_{st}$ represent the shortest path between node $s$ to node $t$, $P_{st} = (v_1, v_2, \ldots, v_{st})$ refers to the node on the path. $d_{st} = e_{s_1s} + \ldots + e_{i(s-1)s}$ representing the shortest path distance between the node and the node. If $s \in V'_c, t \in SC_j$ then $P_{st}$ represents a control path.

2.2. Controller Placement Problem

We find that more controllers in a SDN network can make the control network more reliable. However, we cannot improve the reliability only by adding controllers. Because the amount of controllers or control domains increases, more communication among controllers is needed to exchange information. As a result, the whole network becomes more complex and more difficult to be managed. Since one important motivation of SDN is to simplify the network management, we try to place as less controllers as possible, which can ensure the manageability of the whole network. Moreover, in SDN environment, the delay between switches and their controllers has to meet a response time requirement, or the network cannot work effectively.

3. The proposed controller placement approach

3.1. Two Metrics

In order to measure the reliability and delay, we proposed two metrics. The main objective of controller placement is to minimize the average latency between the switches and controllers. Generally speaking, delay can be divided into four aspects: propagation delay, processing delay, queuing delay and sending delay. For WAN, the propagation delay is greater than the other three delay. This paper mainly considers the SDN research of wan, so the propagation delay approximation is used to represent the network delay.

**Definition 1** control path average delay (CPAD): as mentioned in chapter 2, node $i$ represents controller and node $j$ represents switch. If switch $j$ is assigned to controller $i$, then $p_{ij}$ is a control path. So control path delay (CPD) can be defined as follow:

$$CPD_{ij} = \sum_{e_{st} \in P_{ij}} e_{st}$$

Then we defined CPAD as follow:

$$CPAD = \frac{1}{|V|} \sum_{j \in V_c} \sum_{i \in V'_c} CPD_{ij} = \frac{1}{|V|} \sum_{j \in V_c} \sum_{i \in V'_c} e_{st}$$

(2)
to analyze reliability of the SDN control network, we define a reliability metric similar to what has been proposed in [7]. Our reliability metric is defined as the expected percentage of control path loss. For metric in [7], ignore broken node. The optimization target is then to minimize the expected percentage of control path loss.

**Definition 2** control path average failure probability (CPAFP): we assume that the probability of link failure is related to the length of the path, and the probability of failure of the unit length link is \( p_a \). Therefore, for a control network, we define control path failure probability (CPFP) as follow:

\[
CPFP = 1 - \prod_{e_i \in P_a} (1 - p_a)^{e_i}
\]

Then we defined CPAFD as follow:

\[
CPAFD = \frac{1}{|V|} \sum_{e_i \in P_a} \left(1 - \prod_{e_i \in P_a} (1 - p_a)^{e_i}\right)
\]

Based on the definitions of the two metrics, the problem of low delay and high reliable multi-controller placement problem is to find a reasonable method, which makes the two metrics CPAFD and CPAFP achieve the minimum. Defined as follow:

\[
\min \{CPAD, CPAFP\}
\]

\[
\text{s.t } CPFP < P, i \in SC, j \in V_c
\]

\[
CPD < D, i \in SC, j \in V_c
\]

3.2. **Spectral Clustering**

As mentioned before, MCPP is a constrained multi-objective optimization problem, and it is difficult to obtain the optimal solution in polynomial time. We proposed spectral clustering in this paper to get a feasible solution. In order to make a reasonable partition of the network, when clustering the nodes on the graph, the principle will cluster the nodes closer to each other. According to the requirements of this paper, the distance between nodes can be defined by the delay and reliability of communication between two nodes.

**Definition 3** Node similarity. Node similarity can indicate the degree of similarity between two nodes, and the higher the similarity degree, the greater the probability that two nodes will be classified as one class. Because CPFP is a probability value of \((0, 1)\), and CPD is a specific value, we normalized CPD for convenience:

\[
CPD^* = \frac{CPD - \min CPD}{\max CPD - \min CPD}
\]

Where \( \max CPD \) indicates the longest delay of the link in the network. Similarly, \( \min CPD \) indicates the shortest delay. Node similarity \( w_{ij} \) defined as follow:

\[
w_{ij} = \begin{cases} 
\frac{1}{\alpha CPD^* + (1-\alpha)CPFP} & \text{if } CPD < D \& \& CPFP < P \\
0 & \text{else}
\end{cases}
\]

Where \( \alpha \) represents the weight between delay and reliability. When \( 0 \leq \alpha \leq 0.5 \), the cluster is more biased towards reliability. When \( 0.5 \leq \alpha \leq 1 \), the clustering principle is biased towards delay. From the similarity of each node, a similarity matrix \( W \) representing the entire network topology can be obtained.

For a network topology \( G \), it is cut into two subgraphs \( A, B \), and the cut between \( A \) and \( B \) is defined as:
According to the definition of cutting, if the network topology is cut into \( k \) subgraphs, the minimum cut can be expressed as:

\[
\text{Cut}(A_1, A_2, \cdots, A_k) = \frac{1}{2} \sum_{i=1}^{k} W(A_i, \overline{A_i})
\]

(11)

Where \( \overline{A_i} \) is the complement of \( A_i \) and 1/2 means avoiding that each edge is calculated twice. The principle of this expression is to classify nodes with large similarities into one class, and similarity of nodes belonging to different classes is as small as possible. However, this partitioning principle does not take into account the load conditions of the number of nodes between sub-domains.

Based on the consideration of load balancing, this article uses RatioCut as the expression of graph partition. RatioCut not only considers minimizing cut, but also consider maximizing the number of each subgraph node, expressed as

\[
\text{RatioCut}(A_1, A_2, \cdots, A_k) = \frac{1}{2} \sum_{i=1}^{k} \frac{W(A_i, \overline{A_i})}{|A_i|}
\]

(12)

Where \( |A_i| \) represents the number of nodes in the area \( A_i \). When the sub-divisions are not uniform, for example, if a certain area contains only a few nodes, the sub-graph \( |A_i| \) will be particularly small, resulting in an increase in the overall RatioCut value. The number of nodes in the entire graph is a fixed value, namely \( |A_1| + |A_2| + \cdots + |A_k| = |V| \), and it can be known from the mean inequality that, when the number of nodes in each area tends to be even, RatioCut becomes smaller. So the solution of equation (5) can be transformed into the problem of finding the minimum value of RatioCut:

\[
\arg \min \text{RatioCut}(A_1, A_2, \cdots, A_k)
\]

(13)

Since \( W \) is a similarity matrix, according to the spectral theory, the degree matrix \( D \) is expressed as:

\[
D = \text{diag}(d_1, d_2, \cdots, d_n)
\]

(14)

Where \( d_i = \sum_{j=1}^{n} w_{ij} \) represents the addition of the columns, then the Laplace matrix \( L \) is expressed as:

\[
L = D - W
\]

(15)

For any n-dimensional vector \( f = [f_1, f_2, \cdots, f_n] \):

\[
f^T L f = f^T D f - f^T W f = \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} w_{ij} f_i f_j = \frac{1}{2} \left( \sum_{i=1}^{n} d_i f_i^2 - \sum_{i,j=1}^{n} w_{ij} f_i f_j + \sum_{j=1}^{n} d_j f_j^2 \right) = \frac{1}{2} \sum_{i,j=1}^{n} w_{ij} \left( f_i - f_j \right)^2
\]

(16)

In order to cluster network nodes, each node needs to be assigned a vector to indicate the node's category. Introducing the indicator vector \( \mathbf{h}_{\text{init}} = [h_1, h_2, \cdots, h_n] \) for any n-dimensional vector \( \mathbf{h} \), the element \( h_{ij} \) is defined as:

\[
h_{ij} = \begin{cases} 0 & v_i \notin A_j \\ 1 & v_i \in A_j \\ \sqrt{|A_j|} & v_i \in A_i \end{cases}
\]

(17)

Then for \( \mathbf{h}_i^T L \mathbf{h}_i \), there is:
\[ h_i' L h_i = \frac{1}{2} \sum_{a=1}^{n} \sum_{m=1}^{k} w_{ma} (h_{ma} - h_{na})^2 \]
\[ = \frac{1}{2} \left( \sum_{m=1}^{n} w_{ma} \left( \frac{1}{\sqrt{|A|}} - 0 \right)^2 + \sum_{m=1}^{n} \left( 0 - \frac{1}{\sqrt{|A|}} \right)^2 \right) \]
\[ = \frac{1}{2} \left\{ \text{cut}(A, \overline{A}) \frac{1}{|A|} + \text{cut}(\overline{A}, A) \right\} \]
\[ = \frac{\text{cut}(A, \overline{A})}{|A|} = \text{RatioCut}(A, \overline{A}) \quad (18) \]

Where the first equation variant uses the result of formula (16), the second equation variant uses the definition of the indicator vector. Formula (18) represents the result of splitting into two subgraphs. Therefore for \( k \) subgraphs, RatioCut can be represented as:

\[ \text{RatioCut}(A_1, A_2, \cdots, A_k) = \sum_{i=1}^{k} h_i' L h_i \]
\[ = \sum_{i=1}^{k} (H' LH)_{ii} \quad (19) \]
\[ = \text{Tr}(H' LH) \]

Therefore, solving the minimum value of RatioCut translates into the problem of solving matrix \( H' LH \) traces, that is:

\[ \arg \min \text{Tr}(H' LH) \quad s.t. \quad H' H = I_k \quad (20) \]

According to the Rayleigh-Ritz Theorem[10], if the real symmetric matrix has an eigenvalue \( \lambda_1 \leq \lambda_2 \leq \cdots \leq \lambda_n \) and its corresponding eigenvector \( v_1, v_2, \cdots, v_n \) is orthogonal, then:

\[ \min \text{Tr}(H' LH) = \lambda_1 + \lambda_2 + \cdots + \lambda_k \quad (21) \]

The minimum corresponding to the matrix \( H = [v_1, v_2, \cdots, v_k] \).

Since the value of the vector in \( H \) is affected by the constraint condition, the value is discrete and the solution to the problem is NP-Hard. To solve this minimum value, relax it and relax the value of \( h_i \) to the real number range[11], then the smallest real number solution appears in the matrix \( L \) that consists of the eigenvectors corresponding to the first \( k \) least eigenvalues of the matrix \( L \).

At this time, the \( H \) matrix consists of \( n \) dimensional line vectors, which respectively represent the \( k \)-dimensional coordinates of \( n \) nodes. The \( k \)-dimensional coordinates of each node represent the \( k \)-dimensional attributes of the node respectively. The more similar the nodes, the \( k \)-dimensional attributes are more similar.

After obtaining the matrix \( H \), spectral clustering usually adopt the k-means algorithm to classify \( n \) points in the \( k \)-dimensional coordinate system. From the above inference, the essence of spectral clustering is a process of dimension reduction. The classification of \( n \times n \) dimensional matrices is transformed into the classification problem of \( n \times k \) dimensional matrices. Usually \( k \ll n \), it can effectively reduce the time complexity. Considering that the original k-means algorithm in RatioCut is sensitive to extremum and may be trapped into local optimal solution, this paper adopts improved simulated annealing algorithm instead of k-means algorithm.

3.3. Improved Simulated annealing

This section proposes an MCEP algorithm to solve the problem mentioned before. The MCEP algorithm is based on the improved RatioCut algorithm and uses simulated annealing to optimize its clustering process. The innovations are mainly reflected in: (1) Using k-medoids instead of the original k-means clustering in RatioCut, limiting the selection of center points to the collection of data.
points contained in the current cluster, and preventing outlier clustering Impact; (2) After initial clustering, add an adjustment step for simulated annealing to avoid local optimum. The specific algorithm is shown below:

**Table 1 MCEP algorithm**

| Input: number of controller, \( k \); weight value, \( \alpha \); flow matrix, \( F \); exit temperature, \( T_{\text{exit}} \); initial temperature, \( T_{\text{org}} \); cooling rate, \( r \); |
| Output: controller location, \( c_{\text{best}} \); switch and controller mapping \( \text{bestSC} \) |

1. for each \( i,j \in V \) do
2. \( CPD \leftarrow \text{formulation (1)}, CPFP \leftarrow \text{formulation (3)} \)
3. end for
4. \( W \leftarrow \text{formulation(9)}, D \leftarrow \text{formulation (14)}, L = D - W, [H] = eigs(L, k) \)
5. \( [H] = eigs(L, k), tempV_{c} \leftarrow \text{initKCenter}(H, k), T = T_{\text{org}} \)
6. \( tempV_{c} \leftarrow \text{initKCenter}(H, k), T = T_{\text{org}} \)
7. call for k-medoids, get \( tempV_{c}, tempSC \)
8. \( \text{bestSC} \leftarrow SC \leftarrow tempSC \)
9. while \( (T > T_{\text{min}}) \)
10. Randomly select a node, call for k-medoids, get \( tempV_{c}, tempSC \)
11. \( \Delta = \text{obj}(tempSC) - \text{obj}(SC) \)
12. if \( \Delta \leq 0 \) \( SC \leftarrow tempSC \)
13. if \( \text{obj}(SC) < \text{obj}(\text{bestSC}) \)
14. \( \text{bestSC} \leftarrow SC, \text{bestV}_{c} \leftarrow V_{c} \)
15. endif
16. else
17. \( SC \leftarrow \text{assign tempSC with exp}(-\Delta/T) \)
18. endif \( T = T \ast r \)
19. end while
20. return \( bestSC, bestV_{c} \)

In the MCEP algorithm, from the first row to the fifth row, the eigenvalues and eigenvectors of the matrix \( L \) are obtained, and the eigenvectors corresponding to the \( k \) smallest eigenvalues are selected to constitute the matrix \( H \). Lines 6 through 8 use the traditional k-medoids algorithm to perform the first classification of \( k \)-dimensional row vectors representing nodes. Lines 9 to 19 indicate that the classification results are adjusted based on simulated annealing. Line 20 outputs the optimal classification result.

The algorithm firstly calculates the shortest path between any two points by the Johnson algorithm[12], then obtains the corresponding CPD and CPFP through formula (1) and formula (3), the time complexity is \( O(V^{3} \log V + VE) \); the time complexity of the calculation matrix \( W, D, L \) is \( O(V) \); Compute a matrix \( H \) composed of eigenvectors whose time complexity is \( O(V^{3}) \); the time complexity of k-medoids is \( O(V^{2} k t) \), where \( t \) is the number of iterations of k-medoids; since k-medoids is called in every step of the simulated annealing are classified, so their adjustment time is \( O(V^{2} k t \log _{r}(T_{\text{min}}/T_{\text{org}})) \). Therefore, the total time complexity of this algorithm is \( O(V^{3} + V^{2} \log _{r} T)) \).

4. Performance Evaluation
In this section, we first adjust the proportion of delay and reliability, and get deployments under different requirements, and then conduct simulation approach to compare our approach with two existing approaches in the delay and reliability. In order to evaluate the effectiveness of the model and
the feasibility of the algorithm, this paper takes the actual network topology OS3E[13] as the simulation topology. The topology consists of 34 nodes and 42 edges. The controller deployment method was implemented using Matlab simulation. The testing machine was configured with Intel Core i5 2.4GHz, 8GB RAM.

4.1. MCEP algorithm with different parameters
The first experiment verifies the controller deployment and partitioning of the MCEP algorithm under different parameter settings. Different parameters of the algorithm were simulated. The network partition and controller deployment results are shown in Fig 1. The same symbol indicates that it is in the same subdomain, and the black solid symbol indicates the control deployment position. Fig. 1(a) and 1(b) show the effect of different values on the network partition when the network is divided into two areas. In this case, it means that only the zoning of the link reliability index is considered. In this case, the longer link is more likely to be cut off. Although both time delay and reliability are related to the length of the link, the probability of link failure is exponential with the link length, and the delay is linear with the link length. Therefore, reliability is more sensitive to the link length. The parameters in Fig. 1(c) and Fig. 1(d) are: and the networks are divided into 3 and 4 areas respectively. At this time, the size of the divided subnets is uniform.

4.2. Comparision With Existing Approach
To verify our approach, denoted as MCEP with two typical existing approaches: the approach proposed by Qin et al. (KCBP)[11] and the approach proposed by Wang et al. (Optimized k-means) [6]. In this experiment, the weight $\alpha$ of MCEP defaults is 0.5.
Fig 2 shows the average delay of different partitioning algorithms. It can be seen that as the number of controllers increases, the average delay of the control paths for different partitioning algorithms decreases. Among them, the MCEP and KCBP algorithms have numerical values and declining trend. Both are close; the optimized k-means algorithm is relatively poor in terms of latency. This is because the optimized k-means algorithm only considers load balancing and there are no other constraints; MCEP and KCBP algorithms use delay as a reference factor for controller deployment. This shows that the MCEP algorithm can effectively guarantee the average delay of the control path, making the average delay at an acceptable lower value.

Fig 3 shows the reliability of the control paths of different algorithms which the weight of the MCEP set as 0.5. It can be seen that the reliability of the three algorithms has been improved with the increase of the number of controllers. This is because the increase in the number of controllers can make each sub-domain more compactness improves reliability. At the same time, the MCEP algorithm always has the highest reliability among the three algorithms when the number of controllers is the same. Compared with the other two algorithms, the reliability increases by an average of 17%.

5. Conclusion
This paper addresses the problems of poor control path reliability and communication delay in existing multi-controller deployment scenarios. In order to improve the reliability and reduce communication delay, we proposed a multi-controller resilient deployment (MCEP) algorithm based on spectral clustering. Comparing the KCBP algorithm and the optimized k-means algorithm, the matlab simulation results show that the MCEP algorithm proposed in this paper increases the average reliability of the control path by 17% on the basis of guaranteeing low delay. Future work will consider how to use spectral clustering algorithm to improve the robustness of multi-controller placement problem.

References
[1] Kreutz D, Ramos F M, Verissimo P E, et al. Software-defined networking: A comprehensive survey[J]. Proceedings of the IEEE, 2015,103(1):14-76.
[2] Yeganeh S H, Tootoonchian A, Ganjali Y. On scalability of software-defined networking[J]. IEEE Communications Magazine, 2013,51(2):136-141.
[3] Heller B, Sherwood R, McKeown N. The controller placement problem: Proceedings of the first workshop on Hot topics in software defined networks, 2012[C]. ACM.
[4] Yao G, Bi J, Li Y, et al. On the Capacitated Controller Placement Problem in Software Defined Networks[J]. IEEE Communications Letters, 2014,18(8):1339-1342.
[5] Hock D, Hartmann M, Gebert S, et al. Pareto-optimal resilient controller placement in SDN-based core networks: Teletraffic Congress (ITC), 2013 25th International, 2013[C]. IEEE.

[6] Wang G, Zhao Y, Huang J, et al. A K-means-based network partition algorithm for controller placement in software defined network: Communications (ICC), 2016 IEEE International Conference on, 2016[C]. IEEE.

[7] Hu Y, Wendong W, Gong X, et al. Reliability-aware controller placement for software-defined networks: Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on, 2013[C].

[8] Müller L F, Oliveira R R, Luizelli M C, et al. Survivor: an enhanced controller placement strategy for improving SDN survivability: Global Communications Conference (GLOBECOM), 2014 IEEE,

[9] Nencioni G, Helvik B E, Gonzalez A J, et al. Impact of SDN Controllers Deployment on Network Availability[J], arXiv preprint arXiv:1703.05595, 2017.

[10] Gallier J. Geometric methods and applications: for computer science and engineering[M]. Springer Science & Business Media, 2011.

[11] Kuang-Yu, Q., Chuan-He, H. Balanced multiple controllers placement with latency and capacity bound in software-defined network, Journal on Communications, (11), 90-103(2016).

[12] Johnson, D.B. Efficient algorithms for shortest paths in sparse networks[J], Journal of the ACM (JACM), 24(1), 1-13(1977).

[13] Knight S, Nguyen H X, Falkner N, et al. The internet topology zoo[J]. IEEE Journal on Selected Areas in Communications, 2011,29(9):1765-1775.