

# Reliability-Based Controller Placement Algorithm in Software Defined Networking

Jiang Liu<sup>1,2</sup>, Juan Liu<sup>1</sup>, and Renchao Xie<sup>1,2</sup>

<sup>1</sup> State Key Laboratory of Networking and Switching Technology  
Beijing University of Post and Telecommunications, Beijing 100876, China  
{liujiang, renchao.xie}@bupt.edu.cn, kululu815@126.com

<sup>2</sup> Beijing Advanced Innovation Center for Future Internet Technology  
Beijing 100124, China  
{liujiang, renchao.xie}@bupt.edu.cn

**Abstract.** Software Defined Networking (SDN) change the routing architecture from distribution to centralization. Then the centralized controller placement becomes an important issue since network delay and synchronization will be affected. In this paper, the problem of controller placement to optimize the network average reliability is studied. We first consider the optimization problem for controller placement under the assumption with the shortest path between controller and switches, and a clustering based optimal controller placement algorithm and sub-optimal algorithm base on greedy algorithm are proposed, respectively. Then, we extend the controller placement problem to more general case with multi-paths between controller and switches. In this case, to reduce the computation complexity, a Reliability Factor is defined to replace the network average reliability. After that, modified clustering based and greedy based algorithms are proposed to make the optimal and sub-optimal solution, respectively. Finally, simulation results demonstrate the performance of the proposed algorithms better than regular placement stratagem with the Internet2 OS3E and the Internet Topology Zoo.

**Keywords:** controller placement, reliability optimization, clustering algorithm, greedy algorithm, reliability factor

## 1. Introduction

Software Defined Networking (SDN) decouples the control plane and data plane, and moves the control logic into a remote centralized controller [14, 18, 19, 23]. This new network paradigm can simplify the data forward, flexible manage the network and deploy services base on a global network view, which could be used in many advanced networking environment, such as cloud network, content delivery network, data center network, local area network, distribute file system and so on [5, 15, 20, 21, 28].

In SDN, many challenges are needed to be solved, such as the problem of scalability [9, 26], and robustness [24] for control plane, virtualization [8, 16, 29], communication consistency [17], programing language [27], controller placement and so on. This paper will focus on the issue of controller placement. A logically centralized and physically distributed architecture to improve SDNs scalability are proposed in [4, 13, 25]. The authors of [4] design the hierarchy architecture for control plane, local controllers on the bottom layer is deployed to deal with OpenFlow events based on local topology information, the

root controller on the top layer is used to subscribe to topology events from local controllers to form the network-wide state. In [25], a distributed event-based control plane for OpenFlow named HyperFlow has been proposed. It can minimize the control plane response time for data plane requests by passively synchronizing network-wide views of OpenFlow controllers and localizing decision making to individual controllers. A platform on top of a network control plane to realize a distributed system is proposed in [13], which can provide a general API for control plane implementations, while allowing them to make their own trade-offs among consistency, durability, and scalability.

The problem of controller placement has been studied in [3,6,7,10]. The authors in [6] discuss the impact of quantity and location of controllers for latency of control-traffic in WAN deployment, then the trade-off between them is evaluated. In [7], the authors study the problem of controller placement to achieve a tradeoff between performance and resilience metrics, and then the resilient pareto-based optimal controller placement algorithm (POCO) is proposed. In [10], the problem of controller placement is studied to maximize the reliability of the network. Authors of [3] propose a self-adaption version to dynamic adjust controller quantity and placement according to scale and patterns of network traffic.

Although some works have been done for the controller placement, most of them focus on the performance of latency or the trade-off between quantity and location of controller. To the best of our knowledge, the reliability problem is not well considered for the controller placement. However, the reliability between controller and switches is an important performance metric. This is because that the link faulty between controller and switches will result in the failure of flow forward, which makes the switches cannot receive instructions from the controller. Therefore, to fill this gap, we study the problem of controller placement to optimize network reliability in this paper. The contributions of this paper can be summarized as follows.

- Different from previous works, we study the problem of controller placement to optimize the performance of the network reliability. We first study the optimization problem based on the assumption of shortest path between controller and switches. Then, we extend the optimization problem to the case of Multi-paths between controller and switches.
- For the case of shortest path between controller and switches, a clustering based global optimization algorithm and suboptimal controller placement algorithm based on greedy algorithm are proposed, respectively, which can realize the controller placement and get a good performance in term of the network reliability.
- For the case of Multi-paths between controller and switches, the reliability optimization problem is NP-hard problem. Therefore, in order to simplify the computation complexity, we first define a Reliability Factor (RF), which is used to denote the network reliability. Then the modified clustering based global optimization algorithm and sub-optimal algorithm are proposed, respectively.

Finally, simulation results are illustrated to demonstrate the effectiveness of the proposed algorithms. The rest of the paper is organized as follows. In Section 2, the system model is described, and the controller placement to optimize network reliability under the assumption of shortest path between controller and switches is formulated. In Section 3, the optimal and sub-optimal controller placement algorithms are proposed, respectively.

We extend the optimization problem to the case of Multi-paths between controller and switches in Section 4. Then the simulation results are evaluated in Section 5. Finally, we conclude the paper in Section 6.

## 2. System Model Description

The reliability problem between the controller and the switches for SDN can be viewed as the reliability problem of SAT (source-to-all-terminal) in active network. In each control domain, the controller is equivalent to a source node, all switches in the domain is regarded as the terminal nodes. Links and nodes have two states: reliable and failure. Under this setup, we assume that there are  $M$  controllers and  $N$  switches in SDN, without loss of generality, let  $M \leq N$ . Hence, we can use the graph theory to formulate the problem of controller placement. Let graph  $G(V, E)$ , where  $V$  denotes the node with  $N + M = |V|$ ,  $E$  is the edge denoting the link of network. We assume that there is multi-paths between node  $u$  and node  $v$ , and let  $p_{u,v}$  be the shortest path between node  $u$  and node  $v$ . We use the index function  $X_{l,p_{u,v}} = \{0, 1\}$ ,  $X_{n,p_{u,v}} = \{0, 1\}$  to denote that whether the link  $l$  and node  $n$  is on the shortest path  $p_{u,v}$ , respectively.  $X_{l,p_{u,v}}=1$ , means that the link  $l$  and node  $n$  are on the shortest path  $p_{u,v}$ . otherwise,  $X_{l,p_{u,v}}=0$ ,  $X_{n,p_{u,v}}=0$ . We use  $r_l$  and  $r_n$  to denote the reliability of link  $l$  and node  $n$ , respectively. Here, we have  $0 < r_l < 1$  and  $0 < r_n < 1$ . We assume that all links are non-correlation, and the failure rate is independent. Let the switch set controlled by the controller  $m$ ,  $m \in \{1, 2, \dots, M\}$  be  $S_m = \{s_{m1}, s_{m2}, \dots\}$ . Therefore, the reliability between node and node under the assumption of the shortest path can be expressed as

$$R_{u,v} = \prod_{l \in E} r_l X_{l,p_{u,v}} * \prod_{n \in V} r_n X_{n,p_{u,v}} \quad (1)$$

Hence, the problem of controller placement with the shortest path to optimize network average reliability can be expressed as

$$\min \left\{ 1 - \frac{\sum_{m=1}^M \sum_{s_{mj} \in S_m} R_{m,s_{ij}}}{M} \right\} \quad (2)$$

The optimization problem in (2) is to realize the controller placement to optimize the average reliability of network between controllers and switches.

## 3. Optimal and Sub-Optimal Controller Placement Algorithms Under the Assumption of Shortest Path

To solve the optimal controller placement problem formulated in the above section, we can view the optimization problem in (2) as clustering analysis problem, and use the typical clustering algorithm named K-means algorithm [11, 22] to find the global optimal solution.

For the K-means algorithm, the process can be described as follows. It first divides the data set into K different classes, and the objective criteria function is computed and compared. Then executing the iteration process until the optimal solution reached. Hence, for the controller placement problem, the function of the clustering criterion is the overall network average reliability, each cluster represents a control domain, which includes the controller (assuming that each control domain has only one controller) and the number of switches controlled by controller.

In order to eliminate the impact for the clustering results from the initialization and accelerate the convergence rate for clustering algorithm, we first select a candidate  $M$  nodes for controller placement and let  $r = \text{int}(N/M) - 1$ , where  $r$  denotes the number of switches controlled by controller in each control domain. Then we check all the control domain based on the principles of minimal reliability loss, and remove some of the switches from the control domain with more than switches to the control domain with switches less than  $r$ , until there is no control domain with switches less than  $r$ .

The clustering based optimal controller placement algorithm can be described as follows.

---

**Algorithm 1** Clustering Based Optimal Controller Placement Algorithm.

---

- 1: Initialization: computing  $R_{all,u} = \sum_{V, V \neq u}^{|V|} R_{u,V}$  for all nodes  $u$ , and sorting  $R_{all,u}$  with descending, selecting the first  $M$  nodes as an initial iteration controller and let the corresponding set  $C_{old} = \{c_{o1}, c_{o2}, \dots, c_{oM}\}$ , and  $R_{old} = 0$
  - 2: **for**  $v \in V$  and  $v \notin C_{old}$  **do**
  - 3:     compute the  $R_{v,c_{oi}}$  between node  $v$  and node  $c_{oi}$  in  $C_{old}$ , and allocate the node  $v$  to the controller with larger  $R_{v,C_{oi}}$
  - 4:     generating the clustering  $clusters = clu_1, clu_2, \dots, clu_m$
  - 5:     **for**  $clu_i \in clusters$ , and  $u \in clu_i$  **do**
  - 6:         Compute the total reliability  $R_{clu_i,u}$  for node  $u$  and other node in  $clu_i$ .
  - 7:         Choose the node  $u$  with largest  $R_{clu_i,u}$  as a new controller  $c_{ni}$  in clustering  $clu_i$
  - 8:         Generate a new controller set  $C_{new} = \{c_1, c_2, \dots, c_M\}$
  - 9:         Compute the total network reliability  $R_{new} = \sum_{i=1}^M R_{clu_i,c_{ni}}$
  - 10:          $C_{old} \leftarrow C_{new}, R_{old} \leftarrow R_{new}$
  - 11:     **end for**
  - 12:     Repeating the step 6 – 10, **until**  $C_{old} = C_{new}$  and  $R_{old} = R_{new}$
  - 13: **end for**
  - 14: Counting the number of switches  $ns_i$  for each clustering
  - 15: **if** there is  $ns_i < r$  **then**
  - 16:     **for**  $clu_i$  with  $ns_i < r$  **do**
  - 17:         **for**  $clu_j$  with  $ns_j > r$  **do**
  - 18:             **for**  $u$  in  $clu_j$  **do**
  - 19:                  $R_{lost,u} = R_{c_i,u} - R_{c_j,u}$ , and choose the node with minimum  $R_{lost,u}$  from clustering  $clu_j$  and add to the clustering  $clu_i$ , until all  $ns_i > r$
  - 20:             **end for**
  - 21:         **end for**
  - 22:     **end for**
  - 23: **end if**
-

Although the above clustering based global optimization algorithm can search the global optimal solution, the computation complexity of the algorithms are enormous. Therefore, to simplify the computation complexity, we propose a sub-optimal algorithm based on the greedy algorithm for placement controller. In the iteration, the algorithm can guarantee that local optimal solution is reached. The sub-optimal algorithm based on greedy algorithm is shown as follows.

---

**Algorithm 2** Greedy based Controller Placement Algorithm.

---

```

1: Let  $i = 0, V_1 = V$ 
2: for  $u \in V_1$  do
3:   compute the  $R_{u,v}$  for node  $u$  and node  $v, v \in V_1, u \neq v$ 
4: end for
5: sorting the  $R_{u,v}$  for node  $u$  as descending, and select the first  $r$  node, and return  $R_u = \sum_r R_{u,v}$ 
6: find the node  $u$  with Maximum  $R_u$ , and add it to the controller set  $C$ 
7: select the first  $r$  node for node  $u$  as a switch set  $S_u$ 
8: remove the node  $u$  and node in set  $S_u$  from set  $V_1$ 
9: Let  $i = i + 1$ 
10: If  $i < m$  repeat step 2 – 10
11: else if  $i = m$ 
12: if  $V_1 \neq \emptyset$  then
13:   for  $v \in V_1$  do
14:     Compute the reliability  $R_{v,c_i}$  between  $v$  and controller in set  $C$ 
15:     Allocate  $v$  to the control domain with maximum  $R_{v,c_i}$ 
16:   end for
17: else
18:   stop
19: end if

```

---

#### 4. Reliability Optimization with the Case of Multi-paths between Controller and Switches

In the above section, we study the controller placement to realize reliability optimization for the shortest path case. However, in the practical network, there are also Multi-paths between controller and switches. Hence in this section, we extend the reliability optimization to the case of Multi-paths.

For the case of Multi-paths, we assume that there are total  $n_{u,v}$  paths between node  $u$  and node  $v$  the corresponding set is denoted as  $\Omega_{u,v} = \{p_{u,v,1}, p_{u,v,2}, \dots, p_{u,v,n_{u,v}}\}$  the length of path  $i^{th}$  is  $d_{u,v,i}$ ,  $\psi_{u,v} = \{d_{u,v,1}, d_{u,v,2}, \dots, d_{u,v,n_{u,v}}\}$  is the set of path length. To simplify the problem analysis, we assume the node is reliable and link is unreliable, let  $r_l, 0 < r_l < 1$  denote the reliability of link  $l$ . Assuming that all links are non-correlation and the failure rate is independent. Let the switch set controlled by controller  $m, m \in \{1, 2, \dots, M\}$  be  $S_m = \{s_{m1}, s_{m2}, \dots\}$ . We use the index function  $X_{l,p_m,s_{mj},i} = \{0, 1\}$  to denote that whether the link  $l$  is on the path  $P_{m,s_{mj},i}$ .  $X_{m,s_{mj},i} = 1$  denotes that the link  $l$  is on the path  $P_{m,s_{mj},i}$ , otherwise,  $X_{m,s_{mj},i} = 0$ . Therefore, the problem of reliability optimization with the multi-paths can be expressed as

$$\min 1 - \sum_{m=1}^M \sum_{s_{m,j} \in S_m} \frac{R_{m,s_{m,j}}}{n_{m,s_{m,j}}} \tag{3}$$

where  $R_{m,s_{m,j}}$  denotes the reliability between node  $m$  and  $S_{m,j}$ . When the path between  $m$  and  $S_{m,j}$  is non-correlation,  $R_{m,s_{m,j}}$  can be expressed as

$$R_{m,s_{m,j}} = 1 - \prod_{i=1}^{n_{m,s_{m,j}}} (1 - R_{p_{m,s_{m,j},i}}) \tag{4}$$

When the path between  $m$  and  $S_{m,j}$  is correlation,  $R_{m,s_{m,j}}$  can be expressed as

$$R_{m,s_{m,j}} = \sum_{i=1}^{\tilde{n}_{m,s_{m,j}}} \tilde{R}_{p_{m,s_{m,j},i}} \tag{5}$$

where  $\tilde{n}_{m,s_{m,j}}$  denotes the correlation between node  $m$  and node  $S_{m,j}$ .  $R_{m,s_{m,j}} = \prod_{l \in E} r_l X_{l,p_{m,s_{m,j},i}}$  denotes the reliability of  $l^{th}$  link between node  $m$  and  $S_{m,j}$ ,  $\tilde{R}_{p_{m,s_{m,j},i}} = r_l^{n(1)}(1 - r_l)^{n(0)}$  denotes the reliability of the  $i^{th}$  path between node  $m$  and node  $S_{m,j}$  after disjoint cubes.

#### 4.1. Definition of Reliability Factor

For the optimization problem with Multi-paths between controller and switches, the computation complexity is enormous, especially for the large-scale network with lots of nodes. Hence, to reduce the computation complexity, we first define the reliability factor, and use the reliability factor to replace the network reliability.

In [30], the authors claim that the reliability can be represented by a combination of possible paths between two nodes, an average length of path, and the maximum occupancy of the link. However, using the maximum occupancy of the link for all paths between two nodes to denote the reliability is not sufficient. For example, as shown in Fig. 1, the total paths between the node , and the node , are 2, the average length is 4, the maximum link utilization is 1. However, we can obviously observe that the reliability between node and is superior to the reliability between node and . And it is easy to see that between node and seven links are used, the average link occupancy ratio (i.e, the correlation between the average path) is  $8/7*2$ . And six links are used between node and node , the average link occupancy ratio is  $8/6*2$ . The lower the average link occupancy ratio, the higher the reliability of the two nodes, which also applies to the path between nodes in a more general scenario.

In addition, for some nodes, even though the number of paths, the average length, and average correlation are the same, the reliability may be different. For example, as shown in Fig. 1, the paths between node  $a, m$  and node  $a, h$  are 2, the average length is 4, the average occupancy ratio of the link is  $8/7*2$ , it can be clearly seen the reliability between  $a, m$ , is superior to  $a, m$ . This is because the path between the two points  $a, h$  , is greater than node  $a, m$  . The most extreme case can use  $x^2+y^2 \geq 2xy$  to intuitively explain. Extended to the more general case, this theory is also true that the greater the difference degree of multi-path length between two points, the higher reliability. Therefore, in order

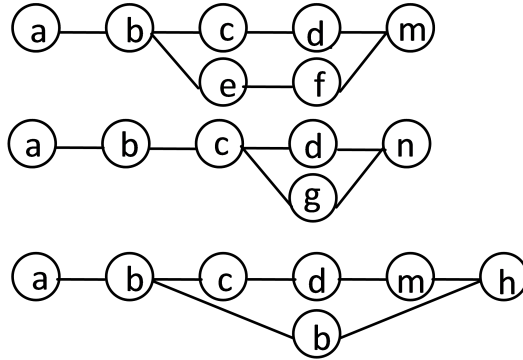


Fig. 1. Example of reliability factor

to compensate the drawback of [30], we give a definition of new reliability factor between node  $u$  and  $v$ . Before giving the definition of reliability factor, we first give the following formula

$$\bar{d}_{u,v} = \frac{\sum_{i=1}^{n_{u,v}} d_{u,v,i}}{n_{u,v}} \tag{6}$$

$$rev_{u,v} = \frac{\sum_{i=1}^{n_{u,v}} \left\{ \frac{(d_{u,v,i} - \bar{d}_{u,v})}{\bar{d}_{u,v}} \right\}^2}{n_{u,v}} \tag{7}$$

$$rev_{u,v} = \frac{\sum_{i=1}^{n_{u,v}} \left\{ \frac{(d_{u,v,i} - \bar{d}_{u,v})}{\bar{d}_{u,v}} \right\}^2}{n_{u,v}} \tag{8}$$

Where (6) represents the average length of all reachable paths between node  $u$  and  $v$ , and (7) denotes the difference degree of all reachable path length between  $u$  and  $v$ , and (8) represents the average correlation for all reachable paths between node  $u$  and  $v$ , which contained a link to the path of the average occupancy rate of repetition, where  $n_{u,v,l}$  is the number of links are used for all paths between  $u$  and  $v$ . Based on the above described formula (6) (7), (8), we define the reliability factor as

$$rf_{u,v} = \frac{n_{u,v}}{d_{u,v} * dep_{u,v} * (1 - rev_{u,v})} \tag{9}$$

The meaning of the expression (9) is that the more the number of paths between two nodes, the shorter the average path length is, and the smaller the average correlation, the higher difference degree between the path length, so the higher reliability of connection has between these two nodes.

#### 4.2. Reliability Optimization Algorithm for the Case of Multi-paths

Based on the definition of reliability factor, we can use the modified algorithm 1 and algorithm 2 to implement the controller placement with multi-paths scenario.

The reliability based optimal controller placement algorithm can be described as follows: for the clustering based global optimal controller placement algorithm proposed in

Algorithm 1, we use the  $rf_{all,u} = \sum_{v,u \neq v}^n rf_{u,v}$  to replace  $R_{all,u}$ , and use  $rf_{v,c_{0i}}$ ,  $rf_{clu_i,u}$ ,

$rf_{new} = \sum_{i=1}^m rf_{clu_i,c_{ni}}$  and  $rf_{lost,u} = rf_{c_i,u} - rf_{c_j,u}$  to replace  $R_{v,c_{0i}}$ ,  $R_{clu_i,u}$ ,  $R_{new}$  and  $R_{lost,u}$ , respectively.

Then the reliability based greedy algorithm is showed as follows: for the sub-optimal controller placement algorithm based on greedy algorithm proposed in Algorithm 2, we only need to use the  $rf_{u,v}$ ,  $rf_u$ , and  $rf_{v,c_i}$  to replace  $R_{u,v}$ ,  $R_u$  and  $R_{v,c_i}$ , respectively.

### 5. Simulation Results

In this subsection, we will evaluate the performance of the proposed algorithms for the reliability optimization under the assumption of shortest path and multi-paths. In [12], the authors point out that the number of topology nodes for nearly half of the network has less than 20, only about 10% of network has the topology nodes more than 51. Therefore, we use the Internet2 OS3E [1], and the Internet Topology Zoo [2] in the simulation. And we compare the proposed algorithms with the random placement algorithm. The random placement algorithm is as follows. We first randomly selected a node  $u$  as a controller from the set of  $V$ , and then randomly select nodes as switches in the control domain; Repeat the above steps until the controllers and corresponding switches are selected. For the random algorithm, we run 100 times during the simulation to avoid the randomness, and the average value is used as the output.

#### 5.1. Simulation Results for the Case of shortest path

For the case of shortest path, we use OS3E topology including 34 nodes. We assume that there are 4 controllers to be mapped. Assuming that the failure rate for nodes and link follows the uniform distribution between 0 to 0.1, respectively. In the simulation, we use global clustering and local greedy to represents the clustering based optimal controller placement algorithm and sub-optimal algorithm based on greedy algorithm, respectively. And random denotes the random algorithm, which mapping the controller randomly.

For the Fig. 2, we evaluate the performance of average failure rate for the proposed algorithms. From the Fig. 2, we can observe that the clustering based global optimization algorithm can get the optimal performance. This is because that clustering based controller placement algorithm is based on the global information, then repeat the iteration process until find the nodes with best performance, while the local search algorithm can only find the local optimal point for each iteration. However, the computation complexity for the local algorithm is less than the clustering based global algorithm, where the computation complexity of global clustering algorithm is , while local greedy algorithms simply iterate times. Therefore, in a large network topology or with low reliability requirements in the



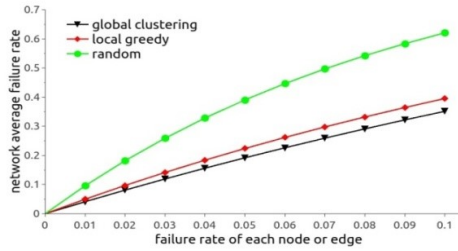


Fig. 2. Network average failure rate versus failure rate of each node or edge with shortest path

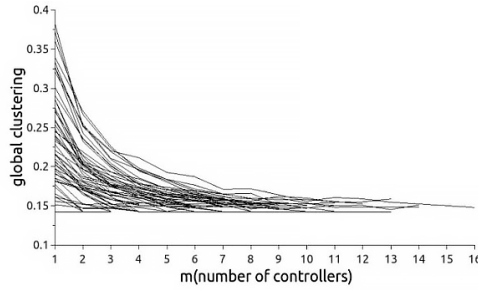


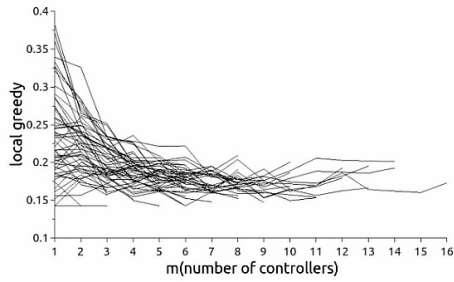
Fig. 3. The average failure rate of global clustering algorithm on the Topology Zoo under the assumption of the shortest path

network environment, we can use simple and fast local greedy algorithm SDN controller placement to realize reliability optimization.

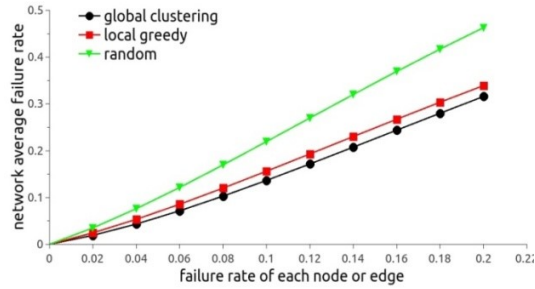
To further verify the effectiveness of the proposed algorithms, we simulate on the more general network topology. We use the Internet Topology Zoo including 55 network nodes, and set the link and node failure rate as 0.05, respectively.

In Fig. 3, we evaluate the average failure rate of the clustering based optimal algorithm on the Topology Zoo. As can be seen from Fig. 3, all of the network average failure rate decreases with the number of controller increasing for all of the network topology for the clustering based global optimization algorithm. This is because that only the shortest path between the controller and switches is used, the failure rate of each control path and the path length has a close relationship, and as the number of controllers increases, the average transmission delay of network between the controller and the switch will decrease, the average failure rate will be reduced.

In Fig. 4, we evaluate the average failure ratio of the suboptimal algorithm based on greedy algorithm on the Topology Zoo. As can be seen from Fig. 4, for all network topologies, using suboptimal algorithm based on greedy algorithm to obtain the network average failure rate is roughly inversely proportional to the number of controllers. However, with the number of controllers increasing, the average failure ratio is fluctuations, which means



**Fig. 4.** The average failure rate of the local algorithm on the Topology Zoo under the assumption of the shortest path



**Fig. 5.** Network average failure rate versus failure rate of each node or edge with Multi-paths

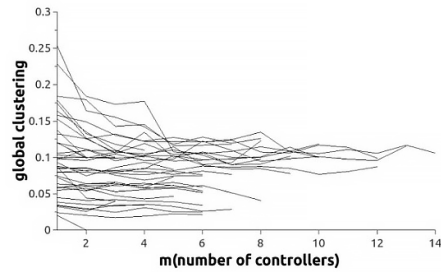
the performance of convergence is not good as the clustering based optimum algorithm. This is because that local algorithm is easy to fall into local optimum.

**5.2. Simulation Results for the Case of Multi-paths**

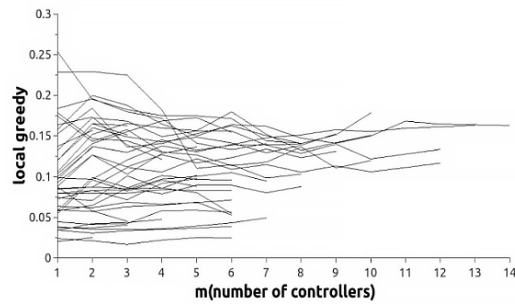
For the case of multiple paths between controllers and switches, we assume that each node is completely reliable in order to simplify computation complexity. The failure rate of link is from 0 to 0.2. Based on the reliability factor (RF), we use Bren (composed of 37 nodes) topology based on Internet Topology Zoo, we assume that there are 6 controllers.

In Fig. 5, we evaluate the performance of average failure rate for the proposed algorithms. From the Figure, we can see that the proposed clustering based global optimization algorithm and greedy algorithm based local optimization outperform the random algorithm. This means that reliability factor defined in the above section can be used to represent the actual reliability.

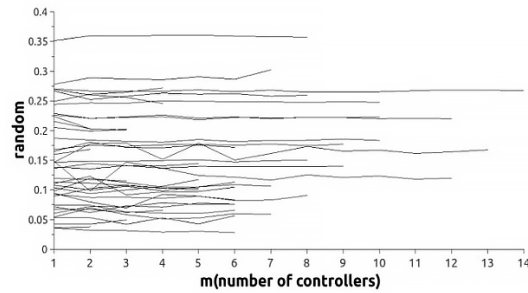
To further verify the effectiveness of the proposed algorithms, we simulate on the more general network topology. We use the Internet Topology Zoo including 43 network nodes, and set the link failure rate as 0.1.



**Fig. 6.** The average failure rate of global clustering based algorithm on the Topology Zoo with multi-paths.



**Fig. 7.** Network average failure rate versus failure rate of each node or edge with Multi-paths



**Fig. 8.** The average failure rate of random algorithm on the Topology Zoo with multi-paths.

From Fig. 6 to Fig. 8, we can observe that global clustering based optimal algorithm and local greedy algorithm based on reliability factors can optimize the performance of network reliability for the case of Multi-paths. And the performance of global clustering algorithm is better than the local greedy algorithm. Both algorithms are not affected by

the number of controllers. This is because the reliability of multi-paths between two nodes is affected by many factors, such as the number of paths, the path correlation and so on, and the length of path is only one factor.

## 6. Conclusions and Future Work

In this paper, we have studied the problem of controller placement, and the cases with the shortest path and multi-paths have been considered, respectively. For the multi-paths case, we have defined the reliability factor to replace the network reliability. Then the clustering based global optimization algorithm and the local optimization algorithm based on greedy algorithm have been proposed, respectively. Finally, we have verified the performance of the proposed algorithms based on the Internet2 OS3E and Internet Topology Zoo.

In our future work, multiple objectives optimization will be considered for the controller placement.

**Acknowledgement.** The work described in this paper was fully supported under the National High Technology Research and Development Program (863) of China (Project Number: 2015AA016101, 2015AA015702), the National Natural Science Funds (Project Number: 61302089, 61300184).

## References

1. Internet2 open science, scholarship and services exchange. <http://www.internet2.edu/network/ose/>
2. The Internet Topology Zoo. <http://www.topology-zoo.org/dataset.html>
3. Bari, M.F., Roy, A.R., Chowdhury, S.R., Zhang, Q., Zhani, M.F., Ahmed, R., Boutaba, R.: Dynamic controller provisioning in software defined networks. In: Network and Service Management (CNSM), 2013 9th International Conference on. pp. 18–25. IEEE (2013)
4. Hassas Yeganeh, S., Ganjali, Y.: Kandoo: a framework for efficient and scalable offloading of control applications. In: Proceedings of the first workshop on Hot topics in software defined networks. pp. 19–24. ACM (2012)
5. He, H., Feng, Y., Li, Z., Zhu, Z., Zhang, W., Cheng, A.: Dynamic load balancing technology for cloud-oriented cdn. Computer Science and Information Systems 12(2), 765–786 (2015)
6. Heller, B., Sherwood, R., McKeown, N.: The controller placement problem. In: Proceedings of the first workshop on Hot topics in software defined networks. pp. 7–12. ACM (2012)
7. Hock, D., Hartmann, M., Gebert, S., Jarschel, M., Zinner, T., Tran-Gia, P.: Pareto-optimal resilient controller placement in sdn-based core networks. In: Teletraffic Congress (ITC), 2013 25th International. pp. 1–9. IEEE (2013)
8. Hu, J.W., Chu-Sing, Y., Te-Lung, L.: Design and implementation of an efficient and programmable future internet testbed in taiwan. Computer Science and Information Systems 10(2), 825–842 (2013)
9. Hu, J., Lin, C., Li, X., Huang, J.: Scalability of control planes for software defined networks: Modeling and evaluation. In: Quality of Service (IWQoS), 2014 IEEE 22nd International Symposium of. pp. 147–152. IEEE (2014)
10. Hu, Y., Wendong, W., Gong, X., Que, X., Shiduan, C.: Reliability-aware controller placement for software-defined networks. In: Integrated Network Management (IM 2013), 2013 IFIP/IEEE International Symposium on. pp. 672–675. IEEE (2013)
11. Kanungo, T., Mount, D.M., Netanyahu, N.S., Piatko, C.D., Silverman, R., Wu, A.Y.: An efficient k-means clustering algorithm: Analysis and implementation. Pattern Analysis and Machine Intelligence, IEEE Transactions on 24(7), 881–892 (2002)

12. Knight, S., Nguyen, H.X., Falkner, N., Bowden, R., Roughan, M.: The internet topology zoo. *Selected Areas in Communications, IEEE Journal on* 29(9), 1765–1775 (2011)
13. Koponen, T., Casado, M., Gude, N., Stribling, J., Poutievski, L., Zhu, M., Ramanathan, R., Iwata, Y., Inoue, H., Hama, T., et al.: Onix: A distributed control platform for large-scale production networks. In: *OSDI*. vol. 10, pp. 1–6 (2010)
14. Lara, A., Kolasani, A., Ramamurthy, B.: Network innovation using openflow: A survey. *Communications Surveys & Tutorials, IEEE* 16(1), 493–512 (2014)
15. Li, X., Xiao, L., Ke, X., Dong, B., Li, R., Liu, D.: Towards hybrid client-side cache management in network-based file systems. *Computer Science and Information Systems* 11(1), 271–289 (2014)
16. Liang, C., Yu, F.R.: Wireless virtualization for next generation mobile cellular networks. *Wireless Communications, IEEE* 22(1), 61–69 (2015)
17. Liu, W., Bobba, R.B., Mohan, S., Campbell, R.H.: Inter-flow consistency: A novel sdn update abstraction for supporting inter-flow constraints. In: *Communications and Network Security (CNS), 2015 IEEE Conference on*. pp. 469–478. IEEE (2015)
18. McKeown, N., Anderson, T., Balakrishnan, H., Parulkar, G., Peterson, L., Rexford, J., Shenker, S., Turner, J.: Openflow: enabling innovation in campus networks. *ACM SIGCOMM Computer Communication Review* 38(2), 69–74 (2008)
19. Nunes, B.A., Mendonca, M., Nguyen, X.N., Obraczka, K., Turletti, T.: A survey of software-defined networking: Past, present, and future of programmable networks. *Communications Surveys & Tutorials, IEEE* 16(3), 1617–1634 (2014)
20. Park, H.W., Yeo, I.Y., Lee, J.R., Jang, H.: Study on network architecture of big data center for the efficient control of huge data traffic. *Computer Science and Information Systems* 11(3), 1113–1126 (2014)
21. Secci, S., Murugesan, S.: Cloud networks: Enhancing performance and resiliency. *Computer* 47(10), 82–85 (2014)
22. Selim, S.Z., Ismail, M.A.: K-means-type algorithms: a generalized convergence theorem and characterization of local optimality. *Pattern Analysis and Machine Intelligence, IEEE Transactions on* (1), 81–87 (1984)
23. Sezer, S., Scott-Hayward, S., Chouhan, P.K., Fraser, B., Lake, D., Finnegan, J., Viljoen, N., Miller, M., Rao, N.: Are we ready for sdn? implementation challenges for software-defined networks. *Communications Magazine, IEEE* 51(7), 36–43 (2013)
24. Tilmans, O., Vissicchio, S.: Igp-as-a-backup for robust sdn networks. In: *Network and Service Management (CNSM), 2014 10th International Conference on*. pp. 127–135. IEEE (2014)
25. Tootoonchian, A., Ganjali, Y.: Hyperflow: A distributed control plane for openflow. In: *Proceedings of the 2010 internet network management conference on Research on enterprise networking*. pp. 3–3 (2010)
26. Veisllari, R., Stol, N., Bjornstad, S., Raffaelli, C.: Scalability analysis of sdn-controlled optical ring man with hybrid traffic. In: *Communications (ICC), 2014 IEEE International Conference on*. pp. 3283–3288. IEEE (2014)
27. Voellmy, A., Wang, J., Yang, Y.R., Ford, B., Hudak, P.: Maple: simplifying sdn programming using algorithmic policies. In: *ACM SIGCOMM Computer Communication Review*. vol. 43, pp. 87–98. ACM (2013)
28. Weng, C.E., Wen, J.H., Chen, H.C., Yang, L.: The performance analysis of direct/cooperative transmission to support qos in wlans. *Computer Science and Information Systems* 11(3), 1143–1156 (2014)
29. Xia, M., Shirazipour, M., Zhang, Y., Green, H., Takacs, A.: Sdn and optical flow steering for network function virtualization. In: *Presented as part of the Open Networking Summit 2014 (ONS 2014)* (2014)
30. Zhang, Y., Beheshti, N., Tatipamula, M.: On resilience of split-architecture networks. In: *Global Telecommunications Conference (GLOBECOM 2011), 2011 IEEE*. pp. 1–6. IEEE (2011)

**Jiang Liu** received his B.S degree in electronics engineering from Beijing Institute of Technology, Beijing, China, in 2005, the M.S. degree in communication and information system from Zhengzhou University, Zhengzhou, China, in 2009, the Ph.D. degree from Beijing University of Posts and Telecommunications, Beijing, China, in 2012. He is currently an assistant professor in Beijing University of Posts and Telecommunications. His current research interests include network architecture, network virtualization, software defined networking (SDN), information centric networking (ICN), and tools and platforms for networking research and teaching.

**Juan Liu** received her B.S degree in Tianjin University, Tianjin, China, in 2011, the M.S. degree in communication and information system from Beijing University of Posts and Telecommunications, Beijing, China, in 2014. Her current research interests include software defined networking (SDN).

**Renchao Xie** received the Ph. D. degree in the School of Information and Communication Engineering from Beijing University of Posts and Telecommunications in 2012. From Jul. 2012 to Sep. 2014, he worked as a Postdoctoral in China Unicom. From Nov. 2010 to Nov. 2011, he visited Carleton University, Ottawa, ON, Canada, as a Visiting Scholar. He is currently associate professor and master supervisor in Beijing University of Posts and Telecommunications. His current research interests include future network architecture design, information centric networking, mobile content delivery network. He has published more than 20 journal and conference papers. He has served for several journals and conferences as a reviewer, including IEEE Trans. on Communications, IEEE Trans. On Vehicular Technology, IET Communications, EURASIP Journal on Wireless Communications and Networking, the 2011 IEEE Global Communications Conference and so on.

*Received: February 25, 2016; Accepted: May 18, 2016.*