A K-means based algorithm for placing controller in SDN by network partitioning

Kunal Joshi | kyjoshi@go.olemiss.edu

ENGR 661: Computer Networks II | Dr. Feng Wang

***Abstract*—Software Defined Networking (SDN), the novel paradigm of decoupling the control logic from packet forwarding devices, has been drawing considerable attention from both academia and industry. As the delay between a controller and switches is a significant factor for SDN, selecting appropriate locations for controllers to shorten the latency becomes one grand challenge. Here, we will investigate multi-controller placement problem from the perspective of network latency minimization by dividing the network in feasible sub-networks. Distinct from previous works, the network partition technique is introduced to simplify the problem. Specifically, the network partition problem is solved not using the hops but the actual network delay. An Improved K-means algorithm is then proposed to address the problem. This new algorithm partitions the network into k sub-networks. Results demonstrate that the proposed algorithm can divide the network into same sub-networks regardless of the random initial node.**

***Index Terms*—Software Defined Network; Network partition; Controller placement; K-means.**;

I. INTRODUCTION

Software Defined Networking (SDN) is emerging as a promising paradigm for future networking. One distinct change of SDN compared with conventional networking is that the control plane is moved to a set of dedicated controllers and each of them manages one or more simplified packet- forwarding switches. Although the research in SDN has drawn considerable attentions, many challenges remain. Take the control plane as an example, given a specific network with a certain number of nodes, how to place the controllers to appropriate nodes is an open question. To address this issue, metrics such as hops, network latency, resilience, reliability, energy saving and load balancing are often considered. Among these performance metrics, network latency(network delay) plays a critical role in networking performance as it heavily affects quality of service in growing real-time applications. Therefore, the main focus of this work is to design a controller placement approach while minimizing network latency from the controller to its nodes.

In this project, we try to tackle the multi-controller placement issue in SDN and propose a new approach with network partition technique. To the best of my knowledge, it is the first work to analytically investigate the strategy of network partition in SDN while taking into consideration the network delay. The potential gains of partitioning network are two-fold. First, it is an effective way to shorten the response latency between a controller and switches. Second, other performance objectives such as load balancing, resistance/reliability awareness, and energy saving can be implemented to the subnetworks instead of the whole network, which significantly reduce the complexity of the controller placement.

In this approach, the entire network is divided into multiple subnetworks and for each sub-network, one or more controllers are deployed correspondingly. Specifically, the clustering algorithm is leveraged to partition the network into subnetworks and an improved K-means algorithm is proposed to shorten the maximum latency between the centroid and associated switches in the subnetwork. The improved K-means can guarantee that each partition is able to shorten the maximum latency between the centroid and other nodes and hence the performance is remarkably improved compared with the regular K-means.

The main contributions of this paper are briefly highlighted as follows.

- A network partition strategy is proposed in this paper to facilitate the controller placement in SDN. With this strategy, performance objectives can be implemented to the subnetworks instead of the whole network, which significantly reduces the complexity of the controller placement.
- The network partition problem is analytically formulated and the controller placement problem in terms of network latency is presented. Even though the main focus of this project is latency minimization, the proposed network partition strategy will shed lights on the issue of controller placement.
- An improved K-means algorithm is proposed to partition the network. The results demonstrate that every time regardless the starting node, we get the same clusters and its centroids.

II. FORMULATION

In this section, we formulate the network partition problem and controller placement problem in terms of latency. For a network with given nodes and links, the physical topology is denoted by an undirected graph G = (V,E) , where V is the set of switches, E is the set of physical links among switches. The distance between node u and v is denoted by d(u, v) , which is the shortest path between node u and v (u, v ∈ V). The set of possible controller instance is given by C . The number of sub-graphs is given by K , denoting the amount of sub-networks that are divided by the algorithm. The network partition can be defined by SDNi(Vi,Ei) , which subjects to:

$$\bigcup_{i=1}^{k} V_i = V; \quad \bigcup_{i=1}^{k} E_i = E \tag{1}$$

$$SDN_i \cap SDN_j = \phi, \quad \forall\ i \neq j, i, j \in k \tag{2}$$

$$S(SDN_i) = TRUE, \quad \forall\ i \in k \tag{3}$$

$$S(SDN_i \cap SDN_j) = FALSE, \quad \forall\ i \neq j,\ i, j \in k \tag{4}$$

$$SDN_i\ is\ a\ connected\ region \quad \forall\ i \in k \tag{5}$$

Eq. (1) indicate that the total sub-networks need to cover all the network;

Eq. (2) means that one edge or vertex can only be allocated to one sub-network;

Eq. (3) implies that the elements in one sub-network have the same similarity;

Eq. (4) suggests that the elements allocated to different subnetworks have different similarity;

Eq. (5) indicates that all the vertexes in one sub-network are connected by edges.

Note that other similarities can be adopted here such as resilience, reliability, load balancing, energy saving, etc. In this paper, we only consider the similarity in terms of latency, as it is the main focus of this paper.

III. REVISED K-means

The network partition problem resembles the clustering problem. K-means is a straightforward and effective clustering method which is widely used in the partition problems. In this section, we propose an improved K-means algorithm to partition a network. For clarification, the initial nodes which are selected to do K-means are coined as 'center'. The actual node which is found by the K-means algorithm and has the shortest distance to other nodes in the subnetwork is called as 'centroid'.

However, the standard K-means algorithm cannot be directly applied to partition network topologies. Therefore, here is an enhanced version of the standard K-means which is more feasible towards network partition - **"Revised K-means"**.

★ Revised K-means algorithm which can be utilized to partition network topologies:

***step 1**: Initialize centers.*

***step 2**: Distribute the vertex v ($v \in V$) to one of the K clusters using the relation,*

$v \in$ cluster i, if $d(v, ci) < d(v, cj)$, $\forall j \in \{1, 2, \cdots, k\}$ where d(u, v) represents the shortest path between node u and v.

***step 3**: Update centroids $C = \{c1, c2, \cdots ck\}$ such that the sum of the shortest path distance from all points in cluster i to the new centroid ci is minimized.*

$ci = vm$, if $d(vm, v) = minimum$,

$\forall\, m \in$ size(cluster i), $v \in$ cluster i, $i = \{1, 2, \cdots, k\}$

***step 4**: Repeat steps 2 and 3 until there is no change in cluster centroids.*

IV. IMPROVED K-means

The revised Kmeans clustering algorithm works to partition network topologies, the partition, however, varies as the initial centers are selected randomly. Since the objective of the network partition in the SDN is to shorten the maximum latency between the centroid and other nodes in a cluster, a more effective K-means algorithm is required to meet this goal. The centroid initialization method plays a significant role for the K-means algorithm and many solutions have been proposed. But in this one, I have taken network delay between the nodes into consideration instead of hops. This new algorithm is named as **"Improved K-means".**

★ Improved K-means network partitioning algorithm:

step 1*: Compute the shortest paths d(u,v) and delay D(u,v)*

step 2*: Randomly select one node from V as the first initial center (center(1)) of G;*

step 3*: Do the revised K-means algorithm and find out the exact centroid of G (centroid(1)).*

step 4*: Find out the target subgraph (Gtarget) which has the highest delay (latency) between its centroid and nodes. The node which has the highest latency is denoted to center(Gtarget).*

step 5*: Select the previous centroid (or centroids) and the center(Gtarget) as the new initial centers.*

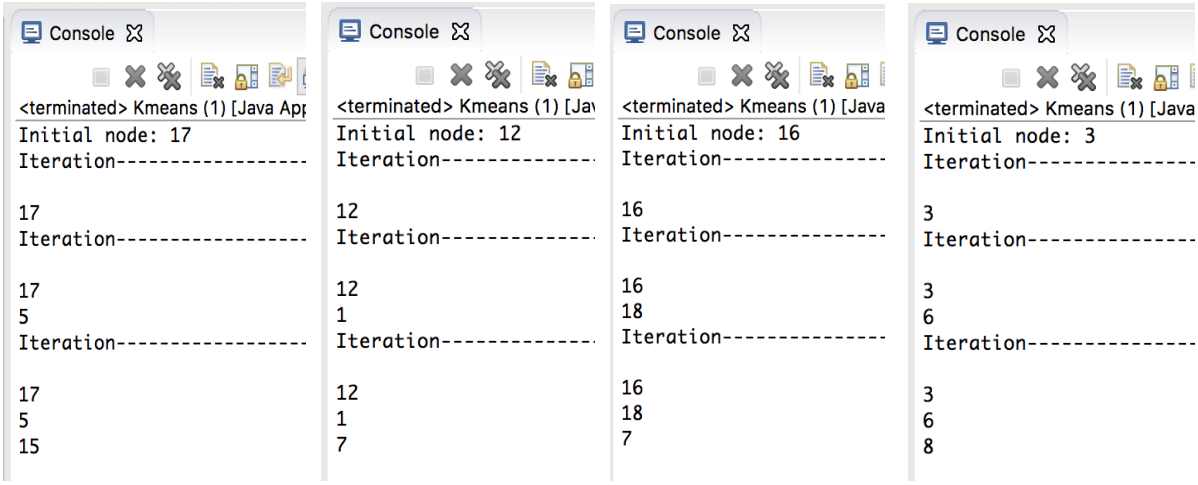
step 6*: Do the revised K-means algorithm again and find out the centroid of each subnetwork.*

step 7*: Repeat step 4, 5 and 6 until the network is partitioned into K subnetworks.*

V. RESULTS

In this section, we evaluate the partition capability of the improved K-means algorithm compared with the standard K-means. Note that the standard K-means utilized in this section is implemented based on the revised K means which is capable to partition networks. The network partition is first conducted by the standard K-means and its centroids are presented. Afterwards, the network partition performed by the improved K-means is illustrated. To evaluate the performance of the improved K-means, we use an example of network with 100 nodes. More specifically we used a file that has delay of all 100 nodes to every other nodes. While for evaluating, we used only a small network of 20 nodes.

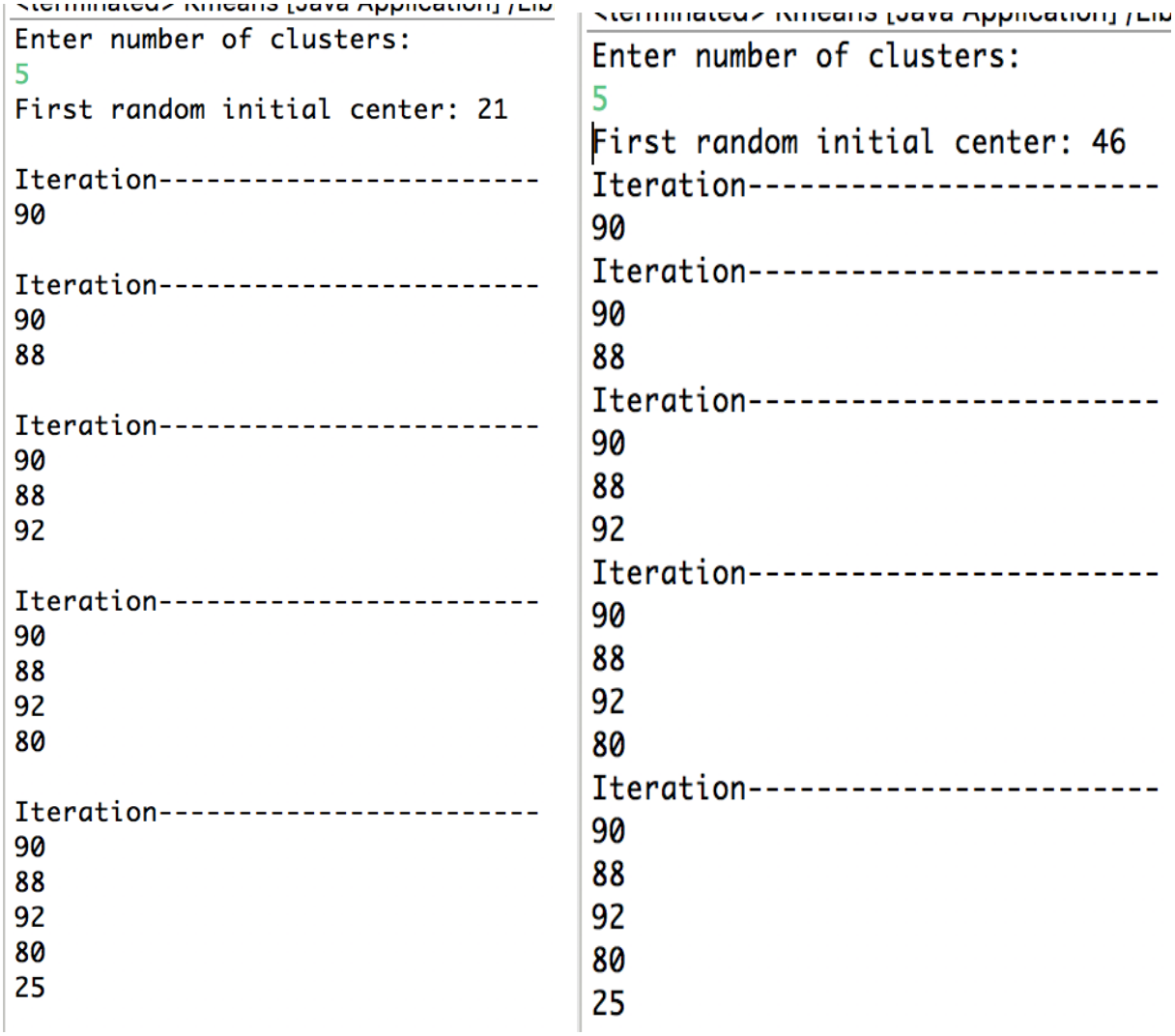
Given below are the snapshots of the results of standard K-means (Revised K-means): Again it's a network of 20 nodes with delay of range 50-100ms between the nodes.



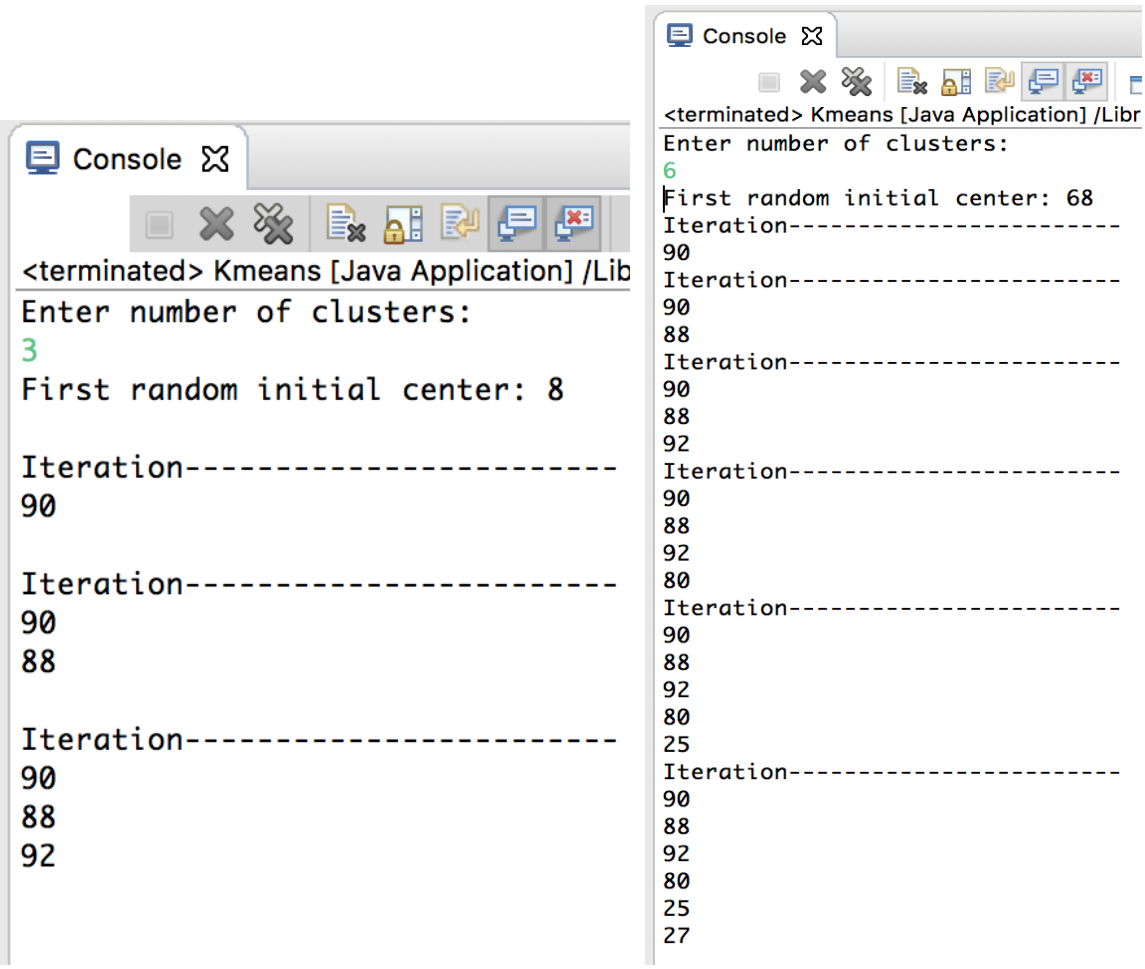
In all the above examples, we have set the number of clusters = 3. As we can see, the centroids we get every time are different due to the random initial node. It is one of the major drawbacks of K-means. We are trying to solve this main issue here with our newly proposed improved K-means algorithm.

In the case of our improved K-means, we tried to make it harder. So, here we work with 100 nodes all having delays of range 50-100ms among each other. Also, here we let the user decide the number of clusters/centroids he wants. The results of it are as below:

Case 1: Here user has entered 5 centroids/clusters. The program takes a random node both the times but still we get the same centroids.



Case 2: Here even with different number of clusters we get the same centroids for the respective number of clusters. Note that, even this time the first initial center is random.



Hence, we can finally say that we are able to minimize the randomness of K-means. Now, with our improved K-means, no matter whatever the initial center we get the same centroids every time. These centroids are potentially capable to be a controller in SDN. Also, we can claim that the delay from the centroid of a cluster to its nodes is minimized.

VI. FUTURE WORK

The claim that we just made in the previous section about the delay being minimized by our algorithm; we can work on the proof of it using our improved K-means. Also along with the network delay, other metrices of network such as reliability, resilience, load-balancing, etc. Can

be implemented using our algorithm. If not the whole network then at least we can start with one of the sub-networks.

VII. CONCLUSION

How to place controllers into SDN is a big challenge. Here, the concept of network partition bas been proposed to tackle the controller placement problem. By modeling the network partition problem, we propose an effective network partition algorithm named Improved K-means to partition network into subnetworks in terms of network latency(delay). Results have verified that our Improved K-means algorithm can effectively partition the network into subnetworks where there is no variation in centroids obtained regardless the initial center, unlike standard K-means. Improved K-means can work on any number of nodes, and any number of desired clusters. More objectives such as load balancing, resistance/reliability awareness, and energy saving can be implemented to the subnetworks instead of the entire network to decrease the complexity of the controller placement.

REFERENCES

1. Guodong Wang, Yanxiao Zhao, Jun Huang, Qiang Duan‡, Jun Li, "A K-means based Network Partitioning Algorithm for Controller Placement in Software Defined Network" IEEE ICC 2016 - Communications Software, Services and Multimedia Applications Symposium
2. My Professor : Dr. Wang, Assistant Professor, Department of Computer Science, University of Mississippi, Oxford, MS 38655