Solving Multi-Controller Placement Problem in Software Defined Network

Kshira Sagar Sahoo [a], Bibhudatta Sahoo[a], Ratnakar Dash[a], Mayank Tiwary[b]
kshirasagar12@gmail.com, bibhudatta.sahoo@gmail.com, ratnakar@nitrkl.ac.in, mayank.tiwary@sap.com

[a]National Institute of Technology,Rourkela, Odisha,India, 769008

[b]SAP Labs, Bangalore, India

***Abstract*—The Software Defined Network (SDN), is the next generation Internet technology, which not only solve the ossification of the Internet, but also creates new innovations and simplify the network management. The key idea behind SDN is separation of control plane from data plane, as a result devices in the data plane becomes the forwarding device and transfer all the decision making activities in a centralized system called a controller. In SDN architecture, it is a great challenge to find a solution to the controller placement problem (CPP). The CPP decides where to place the K number of controllers with a limited amount of resources within the SDN network. Among many proposed solutions to this problem, we apply clustering algorithms such as K-median, K-center on the topology which are obtained from a topology-zoo. A spectral clustering technique has been proposed for partitioning the network and place the controller within the sub-domain. Latency is one of the performance matrices that we have considered in the simulation. The simulation results show that the graph partition method reduces the inter-controller latency within the network.**

I. Introduction

The Software Defined Network has become an emerging technology in industries in recent years and has become more popular network architecture in different kinds of work environments [1][2]. The performance could be based on several parameters like latency in the network, load among devices, resiliency of the network or distance between the nodes,etc. SDN divide the network into three planes namely the data plane, the control plane and the management plane. The data plane is tasked with the job of processing the packets it has received. When a packet arrives, the data plane functionality uses its information about the local forwarding state and the information that is contained in each packets header to make a decision of whether to drop a packet or forward it. If the data plane decides to forward it, then it decides which computer to send it to and which port on that computer packet should be received. To keep pace with all the packets that are arriving to the device, the processing capability of data plane must be extremely fast.

The forwarding state of the packets can be calculated using distributed algorithms or centralized algorithms by the control plane. The data plane and the control plane are very different. On the other hand, the management plane is accountable for managing the interaction among the control plane and the data plane. The control plane comprises of an intelligent physical or software entity called controller, which attaches and manages the switches of the computer network. The controller is responsible for reducing the congestion as well as balance the load in the network. The OpenFlow protocol is a standard way to control the flow tables in commercial switches and routers. This protocol facilitates the communication between the control and data planes of supported network devices (usually switches and routers)[3].

A single controller in an SDN environment is advantageous since it has the global view of the whole network and all the decision are made by the same controller. But single point failure is one of the main disadvantages of this single controlled network [4]. Deploying multiple controllers in the SDN network is the alternate to the said problem. Again in the case of multiple controller scenarios the placing of controllers having an impact on the performance of the network. In [5], the author is trying to show the performance of the network by varying the placement of the controller in the network. Hence it is important to have a better solution to find the locations and number of controllers in an SDN architecture. The switches communicate with their controller through the standard and secure channel via TCP connection. In a multi-controller Software-Defined Wide Area Network (SDWAN), the latency among controllers plays a major role. To achieve the global state of the network, the inter-controller communication is much important because a huge network traffic flows for thousands of application within the network. So, both inter-controller latency and switch to controller latency must take into consideration while placing the controller. In this article, we have considered two clustering algorithms, i.e. K-median and K-center of the controller placement problem. These algorithms find the K number of controllers considering distance as a metric between controllers and controller to switch [6] Later a graph partitioning technique has introduced which partition the network into smaller domains for placing the controller. In this work we have used the spectral clustering technique to partition the network into several sub domains and in each sub domain can be controlled by a single controller [7]. In a multi controller SDN environment the switch migration from one controller to another controller is a very complex and rigid process which severely affect the response time of the WAN controllers. Secondly, various network optimization techniques

like load balancing, energy saving etc. are applicable for the entire network [8].
If the network is divided into smaller domains all the performance parameters can applicable to the sub SDN domains instead of the entire network which reduces the complexity of the controller. To the best of our knowledge the proposed method is a new strategy for placing the controller in the network. The selected controllers can perform their work efficiently in their domain and on its own abstract view of the network.

The contribution of our work is summarized below:

- Consider the real topologies from the www.topology-zoo.org and apply the spectral clustering technique for partition the network.
- Apply both the K-median and K-center on the real topologies and compare the result with graph partitioning technique to find the best controller position.
- In all the cases the inter-controller latency has considered.

The remainder of this paper is prepared as follows: next section presents a comparative discussion of current network architecture and SDN architecture. The relevant related work on placement problems of the SDN is discussed in section 2. Then the problem statements and system models are formulated in section 3 . After the problem formulation, we have discussed K-median and K-centre algorithm along with graph partitioning technique in section 4. For analyzing the best location of the controller we have considered some well-known topologies and after a comparative performance evaluation we conclude with the future work.

II. Related Work

The goal of CPP in SDN domain is to place K number of controllers so as to achieve an optimization function. Heller [5] in his paper has taken a minimum average distance to place the controller in a network, where he considers the matrices as propagation latency and try to find the solution for this problem. Also, they try to answer, how many controllers will be sufficient to properly function a SDN network. Stanislav Lange [9] use heuristic approach, POCO a Matlab based framework, that use Pareto optimal placement with respect to different matrices. Md Faizul Bari [10]use dynamic controller provisioning for placement of the controller, this technique is capable of dynamically adopt the number of controller and their location with changing network condition, where it minimize the flow setup time communication overhead. Rath et al. [11] has used non-zero game technique to place the controller in a network. This technique improves the QoS - minimum packet drop and delay, also can save the cost of deployment and operation. Guang Yao [12] has considered the load of the controller as a basic parameter for the placement of a controller in an SDN and introduce an algorithm CCPP (Capacitated Controller Placement Problem) to solve the problem. Different authors have studied the problem in their own way and find solution for the given problem. From the literature survey, we come to a conclusion that as there are many solutions are provided, but there are many to come as it is an optimization problem of type NP hard.

III. System Model and Problem Formulation

In case of multiple controllers in a SDN environment, the inter controller latency has a major importance. Thus, both inter controller latency and switch to controller latency must be considered during the placing of controllers.

A. System Model for K-median and K-center

In this work, the network under study is represented by a graph $G(V, E)$ with node set represented as V and edge set represented as E.The network elements, i.e., switches and controllers that are connected through communication links. Further the node set V classified into controller set C and switch set S.

$$V = S \cup C \tag{1}$$

Furthermore, the shortest path latency between each pair of nodes are stored in a distance matrix D. The k-center problem and the k-median problem are the most well-known cluster analysis algorithm [13]. These two techniques are used to find the controller locations in an SDN topology where the topology is logically divided into K clusters and each cluster headed by one controller. These two problems are as computationally hard as set cover problem, hence they are NP-hard problem [14].
More formally let us, we have a set of nodes at different locations, i.e. $V = 1, 2, 3, , n$ and $\{D_{ij} | i, j \in N \; and \; D_{ii} = 0, D_{ij} = D_{ji}\}$ where D_{ij} is the distance between nodes. The goal of the work is to find out the location of the controller so as to minimize the switch to controller distance and controller to controller distance. In the worst case, if there is no limitation of controllers required to set up, the solution is to place a controller at each switch location, but for a best case the number of controller locations should be restricted to $1 < k < V$. For a given set of controller locations, a switch (s) is being served by a controller (C_k) which is nearer to controller, hence a cost $d(s, c_k)$ is associated with it. The $d(c_i, c_j)$ represent the shortest distance between the controller c_i and c_j. To keep the above information the global average latency can be written as :

$$min(G(X)) = \frac{1}{n} \sum_{v \in V} d(s, c_k) + \frac{1}{n_c} \sum_{c_i, c_j \in C} d(c_i, c_j) \tag{2}$$

The objective is to find the placement from the set of all possible controller placements , such that the overall latency i.e. $G(X)$ would be minimum.

B. System model for partitioning algorithm

Let the network represented as $G(V, E)$; where V is the set of nodes as represented in equation 1 partition into K subgraphs. Each subgraph represented as $SDN_i(V_i, E_i)$ where

$$\bigcup_{i=1}^{K} V_i = V \tag{3}$$

$$\bigcup_{i=1}^{K} E_i = E \tag{4}$$

$$SDN_i \cap SDN_j = \phi \ \forall i \neq j \tag{5}$$

The above equations indicate that the union of all nodes in each subgraph covers the whole network and no node or edge can allocate to more than one sub-graph. All the nodes within a sub graph must be connected and each subgraph can have dissimilar in nature.Further more after the division of the graph, the controller must be place in the center so as to achieve maximum performance.

$S = \{s_i, s_2...., s_n\}$ is the set of switches indicated in equation 1; and s_i represented as the switches controlled by the controller c_i.

$$\forall s_i, c_i \in SDN_i \tag{6}$$

IV. Algorithms for Controller Placement Problem

In this section, the performance of the three algorithms are discussed. We are considering the spectral graph theory in this paper. The use of similarity matrix and the Laplacian matrix are the core concept behind the spectral theorem.

A. K-median algorithm

The K-median algorithm starts its execution by assigning the switches to the nearest controller that minimizes the average propagation latency (L_{avg}).

$$L_{avg} = \frac{1}{V} \sum_{s \in V} d(s, c_k) \tag{7}$$

The distance between two nodes is identified by the shortest link between them. The nodes that satisfy $d(s, c_k) \leq \alpha$ are joined in the cluster.This process continues for searching better controller position which minimizes the average delay and the controller list updated accordingly till no controller position has changed.

Algorithm 1 K-median algorithm for controller selection problem

INPUT: Require: $(N \times N)$ Delay Matrix (α is the required delay)
OUTPUT: Location of the controllers
while node $\notin$ cluster **do**
  $cluster_k \leftarrow$ find nodes that satisfy $d(s, c_k) \leq \alpha$
  for each node $s \in cluster_k$ **do**
    Evaluate L_{avg}
  end for
  select the node s as controller from $controller_k$that minimizes L_{avg}
  $\{cluster\} = \{cluster\} \cup \{cluster_k\}$
  Choose the node k which is nearest to the current cluster and $k \notin cluster_k$
end while

B. K-center algorithm

The K-center algorithm randomly selects K controller from the network and creates a cluster that satisfies the equation 7. The objective of this algorithm is to searches K number of nodes as the controller that minimizes the maximum delay to other nodes within the network. Repeat the above process for the furthest node to the cluster.

Algorithm 2 K-centre algorithm for controller selection problem

INPUT: Require: $(N \times N)$ Delay Matrix (α is the required delay)
OUTPUT: Location of the controllers
k $\leftarrow$ random node selection
while node $\in$ cluster **do**
  $cluster_k \leftarrow$ find nodes that satisfy $d(s, c_k) \leq \alpha$
  for each node $s \in cluster_k$ **do**
    Evaluate L_{k_c}
  end for
  select the node c as controller that minimizes $d(s, c_k)$
end while

Both for K-median and K-center algorithm, the time complexity for finding a controller is $\theta(V)$ and for k number of controllers, the time complexity would be $\theta(Vk)$.

C. Spectral Clustering Algorithm

The spectral clustering technique is one of the modern day clustering techniques [15]. In a graph, the spectrum is defined as a set of eigen values of its adjacency matrix (A). The well known spectral properties is fit for undirected and unweighted simple graph. The eigen value of the Laplacian matrix is always preferable i.e.

$$L = 1 - A' \tag{8}$$

Algorithm 3 Spectral clustering algorithm for controller selection problem

INPUT: Require: Similarity Matrix $S(k$ is the required partition)
OUTPUT: Final partition with location of the controllers
Compute the Laplacian Matrix L
Select the first K eigen vectors from L.
Sort them in the clolumns of matrix U
The i^th row of U, represent node from graph $(G).(V_i)$
Call K-means to matrix U
Find $\pi^k = c_1, c_2, ..., c_k$
Form the partition by assigning $\forall_{1 \leq i \leq n}\ s_i$ to c_w
return Controller position of each SDN_i sub-graph

The spectral clustering technique is based on the computation of eigen vector corresponding second smallest eigen value of the normalized Laplacian.

V. SIMULATION AND RESULTS EVALUATION

Our algorithms are written with MATLAB 2014, and runs on the machine equipped with Intel Core i5 4-Core processors and 8GB RAM. The detailed process of simulation is described below.

- The adjacency matrix is calculated from the graphML file for the respective topology from the www.topology-zoo.org. The graphML contain the information of the nodes in the form of pairs of latitudes and longitudes. Hence, to obtain the adjacency matrix we compute the distance between the two nodes from their (latitude, longitude) using Haversine formula. The Haversine formula is explained below.
- To do so, their latitude and longitude are taken and the distance between them is calculated using Haversine formula.
- The Haversine formula is explained below:

$$hav(\frac{d}{r}) = hav(2\phi - \phi_1) + cos(\phi_1)cos(\phi_2)hav(\alpha_1 - \alpha_2) \tag{9}$$

and

$$d = 2 * r * \sin^{-1} \sqrt{\sin 2(\frac{\phi_1 - \phi_2}{2}) + cos(\phi_1)cos(\phi_2)\sin 2(\frac{\alpha_1 - \alpha_2}{2})} \tag{10}$$

where
d= distance between two pints
r=radious of the sphere
ϕ_1 and ϕ_2 are the latitude of the points.
α_1 and α_2 are the longitude of the points.

TABLE I: General specification of the network

Topology Type	Node range	Name of the topology
Dense	100-150	TataNld , coltnet
Medium	340-60	Forthnet, usSignal
Sparse	20-40	DeutscheTelekom, Lambdanet

We have considered three different type of network in our simulation. Based on the type of network we have chosen two different topology from the topology-zoo dataset. The general specification of the network is given in Table 1.

The result of various simulations is present here. We have used the K-center and K-median in this simulation for the comparative study and use three different real time topology. In [6] authors have listed the number of controllers with required delay value(α) from controller to switches. For instance, when five controllers to be deployed the required delay for K-median algorithm managed between 14-40 μsec which varies from sparse to dense type of topology.

The Fig.1 shows the Cumulative Distribution Function of all possible placement of the controller within the topology. The CDF has simulated on TataNld topology with the number of controller is five. The pink , red curve line and the blue

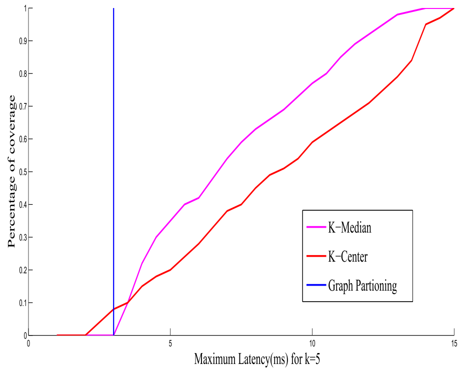

Fig. 1: CDF for all possible position of controller

straight line represent the maximum latency Cumulative Distribution Function (CDF) of K-median K-center and spectral clustering algorithm respectively. After 50 times run the algorithms on the same topology, it has seen that there is a very minimal chance to reach the least maximum latency, whereas in the case of spectral clustering method the least maximum latency is fixed for each time i.e. 3.4 ms because the algorithm has fixed the center for each partition.

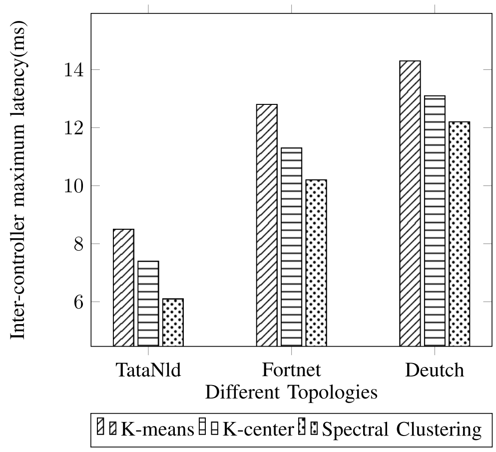

Fig. 2: Controller to controller latency (k=5)

When we consider the total number of controllers to be placed is 5 and 6, the controller to controller latency has been reduced relatively by using graph partitioning technique. For instance, in case of K-median the maximum inter-controller latency in TataNld topology is 8.5 ms whereas the spectral clustering technique reduces it to 6.2 ms which has shown in Fig. 2. In other two topologies the spectral technique performs better than others.

Fig.3 displays the maximum inter-controller latency in the control layer selected by all the considered methods. It can be observed that using the proposed method the maximum delay of the Forthnet topology has dropped from 8.2 ms to 5.8 ms

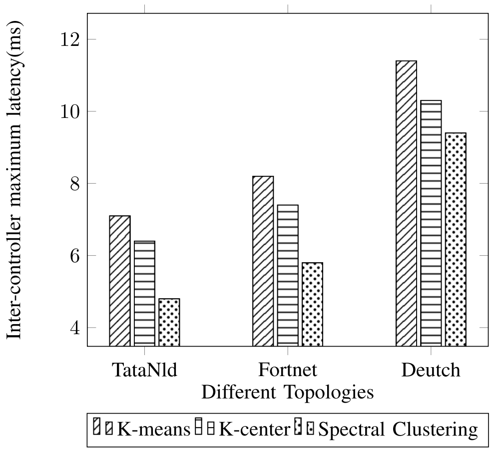

Fig. 3: Controller to controller latency (k=6)

and in the DeutscheTelkom topology it has dropped from 11.4 ms to 9.4 ms.

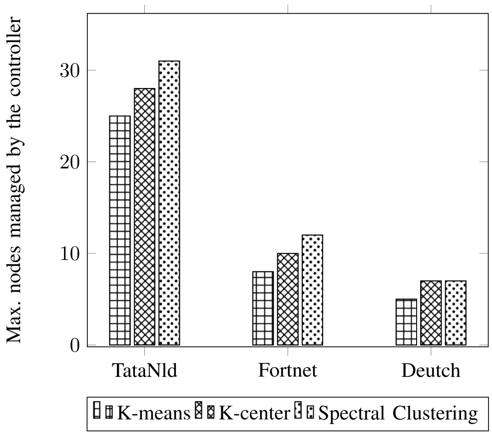

Fig. 4: Nodes managed by the controller(K=5)

Though the number of nodes varies in each category of topology, we observed the average number of nodes managed by a controller is also varies. The fig.4 shows the maximum number of switches covered by the controller by different approaches. The graph shows that in all type of networks the spectral clustering technique has better node distribution than other two.

VI. Conclusion

We have gone through the detail of SDN architecture and various research articles where we found that still SDN have many challenges such as security, traffic control, load balancing, QoS, etc. Controller placement is one of them and many works have already done on this problem. Among many static solutions we have considered two clustering techniques to analyze the effect of inter controller latency with the proposed spectral clustering technique.Our simulation works are carried out on the real topology and our extensive work discovered that in most of the cases the graph partitioning technique performs better than K-center and K-median algorithms. In our future work we will consider dynamic provisioning for controller placement and load as the metric on the same real topologies.

References

[1] Y. Jarraya, T. Madi, and M. Debbabi, "A survey and a layered taxonomy of software-defined networking," *IEEE Communications Surveys & Tutorials*, vol. 16, no. 4, pp. 1955–1980, 2014.

[2] M. Jammal, T. Singh, A. Shami, R. Asal, and Y. Li, "Software defined networking: State of the art and research challenges," *Computer Networks*, vol. 72, pp. 74–98, 2014.

[3] K. S. Sahoo, S. Mohanty, M. Tiwary, B. K. Mishra, and B. Sahoo, "A comprehensive tutorial on software defined network: The driving force for the future internet technology," in *2016 International Conference on Advances in Information Communication Technology and Computing (AICTC)*. ACM, 2016.

[4] K. S. Sahoo, B. Sahoo, and A. Panda, "A secured sdn framework for iot," in *2015 International Conference on Man and Machine Interfacing (MAMI)*. IEEE, 2015, pp. 1–4.

[5] B. Heller, R. Sherwood, and N. McKeown, "The controller placement problem," in *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012, pp. 7–12.

[6] Y. Jiménez, C. Cervelló-Pastor, and A. J. Garcia, "Defining a network management architecture," in *2013 21st IEEE International Conference on Network Protocols (ICNP)*. IEEE, 2013, pp. 1–3.

[7] M. C. Nascimento and A. C. De Carvalho, "Spectral methods for graph clustering–a survey," *European Journal of Operational Research*, vol. 211, no. 2, pp. 221–231, 2011.

[8] K. S. Sahoo and B. Sahoo, "Sdn architecture on fog devices for realtime traffic management: A case study," *International Conference on Signal, Networks, Computing, and Systems [ICSNCS-2016]*, vol. 1, pp. 323–29, 2016.

[9] S. Lange, S. Gebert, T. Zinner, P. Tran-Gia, D. Hock, M. Jarschel, and M. Hoffmann, "Heuristic approaches to the controller placement problem in large scale sdn networks," *IEEE Transactions on Network and Service Management*, vol. 12, no. 1, pp. 4–17, 2015.

[10] M. F. Bari, A. R. Roy, S. R. Chowdhury, Q. Zhang, M. F. Zhani, R. Ahmed, and R. Boutaba, "Dynamic controller provisioning in software defined networks," in *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*. IEEE, 2013, pp. 18–25.

[11] H. K. Rath, V. Revoori, S. Nadaf, and A. Simha, "Optimal controller placement in software defined networks (sdn) using a non-zero-sum game," in *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*. IEEE, 2014, pp. 1–6.

[12] G. Yao, J. Bi, Y. Li, and L. Guo, "On the capacitated controller placement problem in software defined networks," *IEEE Communications Letters*, vol. 18, no. 8, pp. 1339–1342, 2014.

[13] L. Guo-Hui and G. Xue, "K-center and k-median problems in graded distances," *Theoretical computer science*, vol. 207, no. 1, pp. 181–192, 1998.

[14] S. K. Sahni, "Algorithms for scheduling independent tasks," *Journal of the ACM (JACM)*, vol. 23, no. 1, pp. 116–127, 1976.

[15] S. Agarwal, J. Lim, L. Zelnik-Manor, P. Perona, D. Kriegman, and S. Belongie, "Beyond pairwise clustering," in *2005 IEEE Computer Society Conference on Computer Vision and Pattern Recognition (CVPR'05)*, vol. 2. IEEE, 2005, pp. 838–845.