*International Journal of Computer Applications (0975 – 8887)*
*National Conference on Next Generation Computing and its Applications in Computer Science & Technology, 2016*

# Analyzing Controller Placement in Software Defined Networks

Kshira Sagar Sahoo
NIT Rourkela, India

Sampa Sahoo
NIT Rourkela, India

Sambit Kumar Mishra
NIT Rourkela, India

Sagarika Mohanty
IGIT Sarang, India

Bibhudatta Sahoo
NIT Rourkela, India

## ABSTRACT
Software Defined Networks is an emerging network paradigm which introduces programmability to networks and has the capability to dynamically configure the network. In a traditional IP based network the control part and the data forwarding elements are imposed in a single box that has very limited ability to configure the network, some vendor specific codes run on the forwarding elements to perform this task. SDN takes another approach by decoupling the controller part from the data plane part. In a large network such as WAN, centralize of controller approach have many limitations related to the performance and scalability. The placement of the controller is one of them, which affects the scalability and performance. For a large network it is very difficult to decide how many controllers is sufficient to manage the network and where they should be placed. In this work we are trying to solve the problem of controller placement in an SDN network by using two clustering techniques. Latency is one of the measuring matrices that we have chosen.

## General Terms
Your general terms must be any term which can be used for general classification of the submitted material such as Pattern Recognition, Security, Algorithms et. al.

## Keywords
Controller, Software Defined Network, Topology

## 1. INTRODUCTION
In a traditional network, controller software and forwarding element, i.e., switches, controller software is integrated in a single box. Software instruction simply dictates on the forwarding element by imposing rules. These networks are complex and hard to manage. When we want to implement some new network policies. The network operator has to configure each and every individual device separately by using low–level and other vender specific command. It becomes more complicated when the current networks are also vertically integrated. The control plane (which control the network traffic) and the data plane (i.e. Forwarding devices) are integrated into single network devices, reducing flexibility and no chance for new innovation, so there is no elevation of network infrastructure.

Software defined network (SDN) [1] [2] is a new paradigm of networking. It separates the control plane from the data plane. In other word, it decouples the control plane from the data plane. The control part is taken away and is placed in a centralize location by means of the server. We realize this separation through a programming interface between switches and controller. The controller directly can control over the data plane elements through these programming interfaces (API). The most well-known API is OpenFlow [3] [12] switches it has one or more packet handling rule, where each rule has some specific task to do. These rules match a subset of traffic and perform certain actions on the traffic. According to the rules installed by the controller applications. Switches will be instructed by the controller behave like, router, switches, firewall etc. [8]. With this emerging technology this brings some new challenges such as: scalability, availability, security, deployment, management. From the literature review, we have found the placement of the controller in control layer is one of the key problem areas and we are working on to solve this problem. The contributions of the paper are summarized below:

- We have considered two clustering algorithms, i.e. K-medoids and K-center for the controller placement problem in the control plane of SDN.
- These algorithms find the K number of controllers considering distance as a metric between controllers and controller to switch.
- Evaluate these algorithms with varying number of controllers on real topologies which have been taken from topology.zoo.org.

In section II we have discussed the literature review on controller layer and controller placement techniques. The next section describes a deep insight into controller layer. Then problem formulation with the system model discussed in section IV. Two different placement algorithms are discussed in section V. Result analysis, conclusion and future work of the paper discussed in section VI and VII respectively.

## 2. RELATED WORK
In the control layer of SDN, among many, controller placement is a well-known and challenging problem. Heller [4] 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 network. Stanislav Lange [5] use heuristic approach, POCO a MATLAB based framework that use Pareto optimal placement with respect to different matrices. Md Faizul Bari [6] 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. Hemant Kumar [7] 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 [9] has consider 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 so many solutions are possible for the given problem.

# 3. CONTROL LAYER OF SDN

During a long run two factors such as, performance requirements, and cost optimizations of introducing new functions make no difference between the network and the cloud, as more and more functions will be executed either on the network or in the cloud [15] [17] [18] [20]. Introduction of more and bigger data applications creates bottlenecks in the network [21]. In this perspective, by looking at the advancement of telecom industry, the so-called role of a “software-defined operator” is achieving more grip.

Traditional operating system provides a high level abstractions for the accessing lower level devices. It manages the concurrent access to the under-lying resources, e.g. network adapter, CPU, memory etc. This functionality and the resource are the key elements for increasing the productivity, making the life of the system and application development easier. On the other hand network is so far is manage by using some low level devices specific configuration mostly some close proprietary NOSs. Some of the examples of NOSs are Cisco IOS, juniper junOS. It is promised to facilitate network management with less burden to solve the networking problems. It can achieve by means of the logically centralize control offer by the NOS. In traditional operating system, the task of NOS is to provide an abstraction, essential services, and common API to develop generic functionalities such as device discovery, network topology information, and sharing of network configuration. But in SDN architecture NOS or the new term as a controller is one of the important elements as it generates the control logic for the network configuration based on the same policies which are defined by the network operator. The working principle is similar to the traditional operating system by abstracting the internal details of the connected forwarding devices.

In an SDN network centralized controller is a single entity that manages all the connected forwarding elements in a network which has depicted in Fig.1. So it presents a single point of failure, so it has scaling limitation. A single controller is not sufficient for managing a large network. Centralized system such as NOX [10], Floodlight [11], Maestro and Beacon [12] are designed for the highly concurrent system. For example beacon controller can deal with 12 million flows per second because it supports multithreading with parallel execution.

Contrary to the centralized design of the controller, in distributed design the number of controllers can be scaled up to achieve the requirement. As more and more applications use multimedia content the big data for SDN may create bottlenecks for the controller.

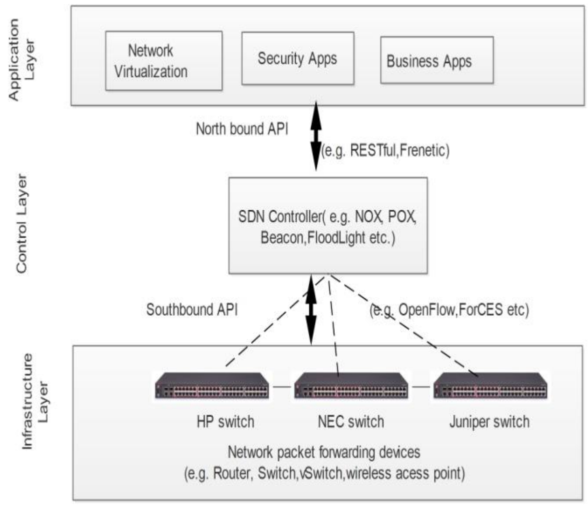


**Fig.1. SDN architecture**

In today’s cloud and big data scenario we need a dedicated network, which can handle a huge volume and high speed data. To manage all these sufficient numbers of controllers and the controllers should be in a suitable place in a network so that it will easier to manage the network. So, a distributed control layer architecture is needed for the time [13] [19]. Onix [14], HypeFlow [16], HP VAN, beacon are the some controller have been designed for the distributed SDN architecture.

# 4. SYSTEM MODEL

As the SDN technology is realized by decoupling the control plane from the data plane. The placement of the controller is related to the network efficiency, scalability, reliability, security which dramatically increases the performance of the entire network when the controller is placed in an appropriate position in a network. To solve the controller placement for a small network a single controller is sufficient, but for a large scale network such as WAN multiple controllers is needed.

To realize the physical network, let the topology is considered to be graphed *G(V, E)* Where *‘V’* denotes the nodes (either switch or controller) in the network and *‘E’* represent the link connection in between the switches. $D(u, v)$ represent the distance between the switches ‘$u$’ and ‘$v$’. Let $K$ denotes the set of controllers $\{k_1, k_2, k_3, \ldots\ldots k_n\}$. In the controller plane where, $K \subset V. E = \{e_1, e_2, e_3 \ldots . e_n\}$ is the set of possible links between the switches and the controllers. L is the set of location for the controller to be placed.

The objective is to minimize the average latency between the controller (v) and switch (u).

$$L = \frac{1}{n}\sum_{v,u\in V} d(u, v) \quad (1)$$

The above equation (1) is used to find the average minimum distance between the switch to the controller of a given topology.

# 5. PLACEMENT ALGORITHMS

For placement of controller we have used graph partition technique. We have chosen a standard graph which obtained from the network dataset. We have used K means clustering algorithm to obtain the partition of network graph. Explanation of the K-medoids and K-center clustering technique is given below.

**K-medoids algorithm:**

This is a clustering algorithm which chose the center first and take an approach of minimizing the sum of dissimilarity between the points and marked to be in a cluster and a data point chosen to be the center of that cluster.

*Steps:*

*initial gauss for the center* $c_1 \ldots\ldots\ldots c_k$

*Repeat:*

*1.Minimize over C: for each i=1…n find the cluster center* $c_k$ *closest to* $P_i$

*2. Minimize over* $c_i \ldots\ldots\ldots c_k$*: for each k=1…..K*

*3. Stop until inter-cluster variation doesn't change.*

**K-center algorithm:**

This is another clustering algorithm. The goal of this algorithm is to select $K$ points from the given data points which minimizes the maximum distance from the controller to the switches.

*1. Require: (N×N) Shoterst Path Matrix. and Required delay (r)*

*2. k← Select randomly a node*

*3. While there are nodes not belonging to the cluster do*

*4.* $cluster_k \leftarrow$ *Find the nodes v that satisfy d(k; v)* $\leq$ *r, where v* $\notin$ *Cluster*

*6. for each node v* $\in cluster_k$ *do*

*7. Evaluate* $\max(\min(d(v, cluster_k)))$

*8. end for*

*9. choose the node as controller s which minimizes the d(v, s)*

*Find the furthest node k from Cluster*

*end while*

## 6. RESULTS AND DISCUSSION

How many controllers are needed the answer of this question is depends on many factors, one is the network size. Other factors are traffic, load on the network. We have classified topology into three classes; dense, medium and sparse having different range, which has given in the Table 1. For our experiment we have used various topologies from www. http://topology-zoo.org.

**Table 1. Different type of topology of varying nodes**

| Topology type | Nodes Range | Name of topology |
|---|---|---|
| Dense | 100-150 | TataNld |
| Medium | 40-60 | Forthnet |
| Sparse | 20-40 | DeutscheTelekom |

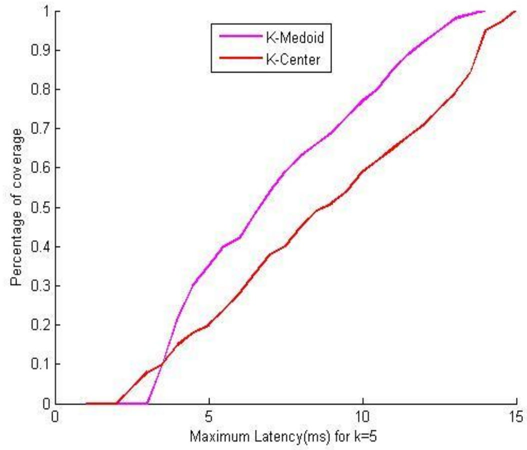


**Fig.2. Maximum latency CDF for all possible locations (K=5)**

In [4], it has observed that, by simply choose a placement at random for a small value of k, the average latency is between 1.4x and 1.7x larger than that of the optimal placement. Hence we have taken the maximum latency Cumulative Distribution Function (CDF) of all possible locations in DeuthTelecom network topology. The above considered algorithms are run for 50 times each and found that K=5 is the optimal number of controller to be placed because it covers the entire network which has shown in Fig.2.

The propagation latency between the nodes can be calculated using the following formula:

$$X = \sin^2\left(\frac{\alpha}{2}\right) + \cos(lat_i)\cos(lat_j)\sin^2\left(\frac{\beta}{2}\right)$$

Where $s_i(lat_i, long_i)$ represent the latitude and longitude of the point $s_i$ and $\alpha = |lat_i - lat_j|$ and $\beta = |long_i - long_j|$

The result of the other simulations is presented below. We have used two algorithms, k-center and k-median for the comparative study and from above, we use three different real time topology from different categories. As the number of controllers deployed in the networks the latency from controller to switch become reduced. We have observed the reduction of latency by varying the number of controllers from 2 to 5 in three different topologies.

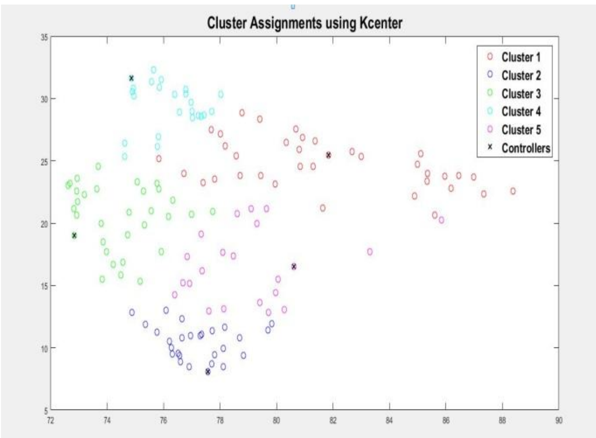


**Fig.2. Cluster of TataNld topology using the k - center algorithm.**

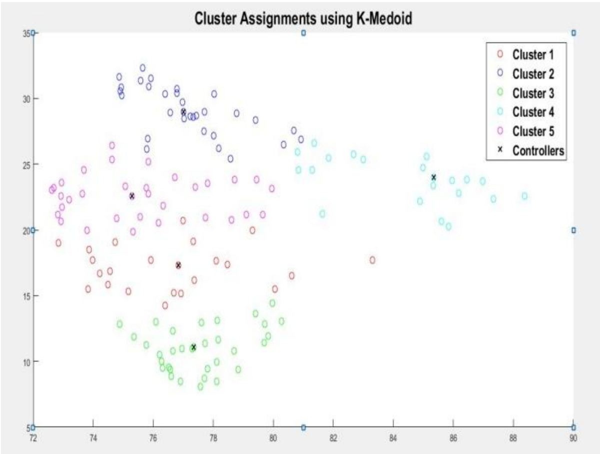


**Fig.3. Clustering of TataNld using k-medoid algorithm**

We have used the two different algorithms on the TataNld topology. The location of the controller is different in different simulation which is depicted in the Fig.2 and Fig.3 respectively.

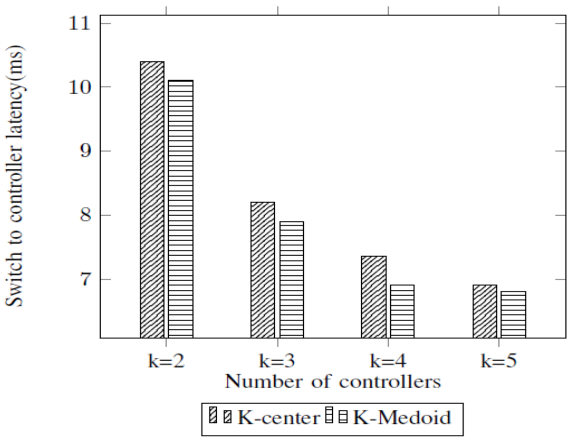


**Fig.4. Switch to controller latency on TataNld Topology**

From the observation, we have found that for dense and medium topology K-medoid is performing better. For the sparse topology like DeuthTelecom network the K-center gives better result than K-medoids. In each case the controller to switch latency decreases on the deploying number of controllers in the network.

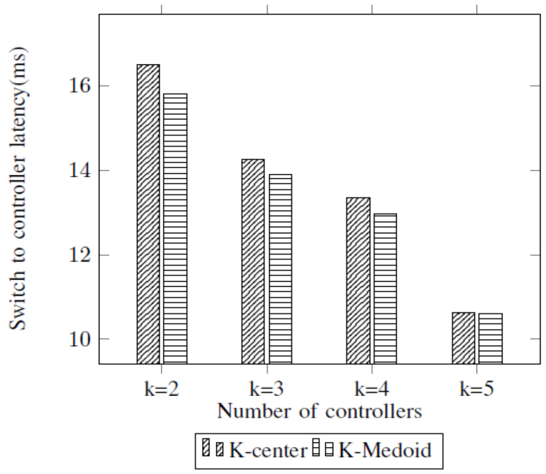


**Fig.5. Switch to controller latency on Forthnet Topology**

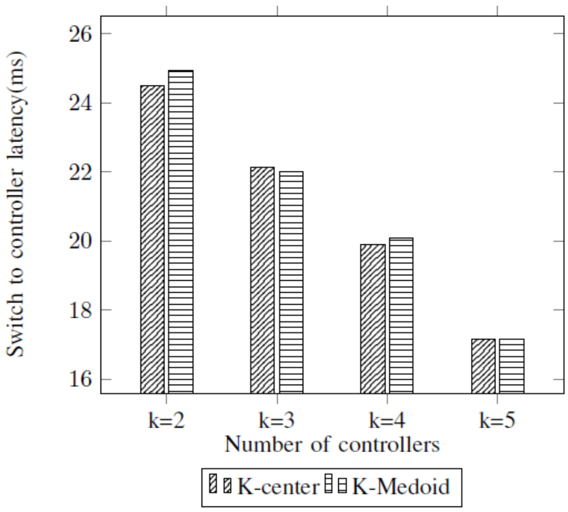


**Fig.6. Switch to controller latency on DeuthTelecom topology**

In case of TataNld topology when the number of controller (K) is 2, the switch to controller latency is 10.35ms, whereas on K=5 it has reduced to 6.9 ms. But in a sparse topology like DeuthTelecom, when k is 2 the latency is more compared to other two topologies. The locations have chosen to use the K - center algorithm showing better result than K-Medoids technique. When the number of controllers is 2 in the network, the latency is reduced to 22.15 ms to 24.5 ms But, in case of K=2 both algorithms returns the same results.

## 7. CONCLUSION

Controller placement is one of challenging research issue in SDN. In this article we have gone through the detail of SDN control layer and discussed related work on the controller placement problem. The goal of this paper is to minimize the latency between the controller and the switches in a given topology. To manage the WAN topology, a sufficient number of controllers should be placed in a suitable location in the network so that it will easier to manage the network. In this work we observed that the deploying number of controllers reduces the controller to switch latency drastically.

The article works towards enumerating the trade-off that results from using two different clustering techniques as well as providing strategies with respect to algorithm choice for different use cases. Placing the same number of controllers varies the latency in two different algorithms in the considered topology. Although latency minimization between the switches and the controller is the main objective of this paper, in the future work; the load balancing, energy saving like objective will consider on the real topologies. In addition to this, we will apply some heuristic approach to solve the controller placement problem.

## 8. REFERENCES

[1] Kreutz, Diego, et al. "Software-defined networking: A comprehensive survey." *Proceedings of the IEEE* 103.1 (2015): 14-76.

[2] Sahoo, K.S., et al. " A Comprehensive Tutorial on Software Defined network:The Driving Force for the Future Internet Technology" *International Conference on Advances in Information Communication Technology & Computing (AICTC),* ACM, 2016.

[3] Nunes, Bruno Astuto A., et al. "A survey of software-defined networking: Past, present, and future of programmable networks." *IEEE Communications Surveys & Tutorials* 16.3 (2014): 1617-1634.

[4] Heller, Brandon, Rob Sherwood, and Nick McKeown. "The controller placement problem." *Proceedings of the first workshop on Hot topics in software defined networks*. ACM, 2012.

[5] Lange, Stanislav, et al. "Heuristic approaches to the controller placement problem in large scale SDN networks." *IEEE Transactions on Network and Service Management* 12.1 (2015): 4-17.

[6] Bari, Md Faizul, et al. "Dynamic controller provisioning in software defined networks." *Proceedings of the 9th International Conference on Network and Service Management (CNSM 2013)*. IEEE, 2013.

[7] Rath, Hemant Kumar, et al. "Optimal controller placement in Software Defined Networks (SDN) using a non-zero-sum game." *World of Wireless, Mobile and Multimedia Networks (WoWMoM), 2014 IEEE 15th International Symposium on a*. IEEE, 2014.

[8] Sahoo, Kshira Sagar, and Bibhudatta Sahoo. "SDN Architecture on Fog Devices for Realtime Traffic Management: A Case Study." (2016).

[9] Yao, Guang, et al. "On the capacitated controller placement problem in software defined networks." *IEEE Communications Letters* 18.8 (2014): 1339-1342.

[10] Gude, Natasha, et al. "NOX: towards an operating system for networks." *ACM SIGCOMM Computer Communication Review* 38.3 (2008): 105-110.

[11] Wallner, Ryan, and Robert Cannistra. "An SDN approach: quality of service using big switch's floodlight open-source controller." *Proceedings of the Asia-Pacific Advanced Network* 35 (2013): 14-19.

[12] Sahoo, Kshira Sagar, Bibhudatta Sahoo, and Abinas Panda. "A secured SDN framework for IoT." *2015 International Conference on Man and Machine Interfacing (MAMI)*. IEEE, 2015.

[13] Erickson, David. "The beacon openflow controller." Proceedings of the second ACM SIGCOMM workshop on Hot topics in software defined networking. ACM, 2013.

[14] Koponen, Teemu, et al. "Onix: A Distributed Control Platform for Large-scale Production Networks." *OSDI*. Vol. 10. 2010.

[15] Sahoo, Abhaya Kumar, Kshira Sagar Sahoo, and Mayank Tiwary. "Signature based malware detection for unstructured data in Hadoop." *Advances in Electronics, Computers and Communications (ICAECC), 2014 International Conference on*. IEEE, 2014.

[16] Tootoonchian, Amin, and Yashar Ganjali. "HyperFlow: A distributed control plane for OpenFlow." *Proceedings of the 2010 internet network management conference on Research on enterprise networking*. 2010.

[17] Sahoo, Sampa, et al. "Execution of real time task on cloud environment." *2015 Annual IEEE India Conference (INDICON)*. IEEE, 2015.

[18] Tiwary, Mayank, et al. "CPS: a dynamic and distributed pricing policy in cyber foraging systems for fixed state cloudlets." *Computing* (2016): 1-17.

[19] Sahoo, Bibhudatta, Sudipta Mohapatra, and Sanjay Kumar Jena. "A genetic algorithm based dynamic load balancing scheme for heterogeneous distributed systems." (2008).

[20] Mishra, Sambit Kumar, et al. "Improving energy consumption in cloud." *India Conference (INDICON), 2015 Annual IEEE*. IEEE, 2015.

[21] Panigrahi, Chhabi Rani, et al. "Big Data and Cyber Foraging: Future Scope and Challenges." *Techniques and Environments for Big Data Analysis*. Springer International Publishing, 2016. 75-100.