# Affinity Propagation based Network Partitioning for Controller Placement in Software Defined Networks

Sminesh C. N.
Dept. of Computer Science and Engineering
Govt. Engineering College Thrissur
Thrissur, India

Dr. E. Grace Mary Kanaga
Dept. of Computer Sciences Technology
Karunya University
Coimbatore, India

Amalchithra Stanly.
Dept. of Computer Science and Engineering
Govt. Engineering College Thrissur
Thrissur, India

**Abstract— Software defined networking (SDN) is a new network architecture where SDN control plane is separated from data plane. The control plane logic is transferred to external entity referred as controllers and data plane consists of packet forwarding devices. When the network elements and traffic flows grow up a single controller cannot efficiently handle the growing load. In large SDN networks to improve performance and scalability, multiple controllers need to be deployed. In a multi controller scenario the load balancing aspect of control plane is an open research challenge. In a large scale SDN the control plane load balancing is called controller placement problem. The traffic engineering solutions in SDN (SDN TE) need to provide optimal number, locations and load distribution of controllers. To develop computationally simple solutions for the controller placement problem in WAN, network partitioning algorithms can be employed. The proposed method partitions the network using clustering algorithm. The affinity propagation based clustering scheme which automatically computes the number of clusters is used for finding optimal number and placement of SDN controllers. The simulation of the proposed system is done using standard network topologies as input. The proposed scheme automatically identifies the required number of clusters and places the SDN controller such a way that controller to switches communication latency is significantly minimized. The calculated imbalance metric can be used to ensure controller load balancing.**

***Keywords— SDN; controller placement;traffic engineering affinity propagation***

## I. INTRODUCTION

Software defined networking is a promising new network architecture which provides solution to many problems faced by traditional packet switched network. In SDN, control plane is decoupled from the data plane[2]. The control plane makes the packet forwarding decision that the data plane consists of packet forwarding devices. SDN controller communicates with the data plane by the standard programming interface provided by OpenFlow protocol. With a centralized architecture, SDN controller has the global view of the underlying network enabling enormous research works in the area of traffic engineering[15].

For large scale network a single controller cannot work efficiently and scale up with the increased number of network elements and the growing number of traffic flows. To meet the growing demands multiple controllers need to be deployed which paves way to another open research problem in optimal placement of controllers and control plane load balancing[15]. Controller placement problem is one of the key research problems in SDN. SDN TE solutions for controller placement should find a trade-off between communication latency and load balancing [2]. Multiple controllers are adopted to solve the scalability and reliability issues in SDN TE Onix[10], Hyper flow[9], are two approaches towards implementation of distributed controllers.

The controller placement problem was first proposed by Heller et.al in [4] in 2012. In this K- Center based approach only communication latency is considered. In capacitated K-Center approach [5] the authors propose controller load balancing solutions. Conceiving the controller placement problem as multi objective optimization authors propose solutions for controller placement using Pareto optimal[6] and heuristics[7] based schemes. However the computational complexity of the implementation of the above approaches are significantly high.

To simplify controller placement, network partitioning algorithms can be used. In K- Means clustering [1] based approach it is required to initialize the number of partitions. For a WAN network partitioning using spectral clustering[3] identifies optimal number of clusters but that method does not identify an exemplar for each cluster.

In the proposed method the network is partitioned into sub networks using affinity propagation algorithm. Affinity propagation (AP) clustering[11 ] automatically determines the number of sub networks using an exemplar based message passing clustering method. An exemplar is a data point that is representative of the cluster and is used for the placement of SDN controller. The performance metric chosen is controller to switches communication latency. The proposed scheme places exemplars such a way that it minimizes the maximum distance between exemplars and other nodes in the cluster. Hence the communication latency from controller to all other nodes in the cluster will be minimized. The proposed system also computes Imbalance, a measure of the irregularities in the number of nodes associated with a controller.

The simulator for the proposed system is developed using Python language. The network partitioning scheme using affinity propagation is implemented. The standard input dataset for the proposed system is taken from the Internet Topology Zoo[12]. The proposed system automatically identifies the required number of controllers such that the communication latency between controller and associated switches are minimized. This is the first paper which automatically determines the required number of controllers.

The rest of the paper is arranged as follows. Section II discusses the existing literature in controller placement problem. Section III states the problem definition in detail, section IV discusses the proposed strategy for optimal placement of controllers, section V discusses the simulation and performance analysis of the proposed system. The paper is concluded in section VI.

## II. RELATED WORKS

Controller placement problem tries to find optimal number and placement of SDN controllers. The placement of controllers is done in such a way that multiple contradicting requirements are satisfied. For the optimal placement of controllers, we have to minimize the propagation latency between switches and controllers. The location of controllers and the number of controllers affect the value of propagation latency among controllers and between controller and switches. The controller placement problem can be regarded as a multi objective combinatorial problem. It is analogues to a facility location problem which is NP Hard.

The controller placement problem was first proposed by Heller et.al [4] in the authors examines the impact of the controller placement on average and worst case propagation latencies. By solving K-median problem, the average propagation latency between controller and switches is minimized. K-center based algorithm finds the worst case latency between switches and controller. In [4] only propagation latency was considered. In [5] the authors consider the load of controllers.

In [5] the placement of controllers are done in such a way that the propagation latency is minimized and checks whether load on each controller has not exceeded the capacity. Multi objective controller placement was done in [6] and [7].

Wang et al.[1] proposed a new approach with network partition technique to solve the controller placement problem. In the paper the authors proposed an effective network partition algorithm named optimized K-means to partition network into k sub networks in terms of latency. The network partition technique is introduced to simplify the controller placement problem. Xiao et al[3] proposed another clustering approach called spectral clustering for controller placement. Compared to the above schemes the clustering based approaches significantly reduce the computational complexity.

However the above approaches do not provide proper solutions to controller placement in the perspective of load balancing, scalability and optimal number of controllers.

## III. PROBLEM STATEMENT

In the existing literature addressing control plane load balancing specifies the need for multiple controllers. It is observed that in a multi controller scenario optimal placement of controllers is indispensible for control plane load balancing. The K-center, Pareto optimal and heuristic based approaches for control plane load balancing results in high computational complexity. The clustering based approaches considerably reduces the algorithmic complexity however better algorithms need to be developed in the perspective of optimal number, placement and load balancing perspective of SDN controllers.

The research objective is to determine the optimum number of controllers and their location in software defined networking using affinity propagation based clustering in the perspective of load balancing and minimizing the communication latency between SDN controller and switches.

## IV. PROPOSED SYSTEM

Finding the optimal number of controllers and their location in the network is a major concern in the case of multiple controller placements. One of the performances metric for SDN controller placement is the propagation latency proportional to the distance between the controller and the switches. Controller placement problem can be simplified by considering it as network partition problem. For a given network we have to find the optimal number of controllers and their location satisfying the controller placement metric.

The network is partitioned into sub networks using affinity propagation based clustering scheme. A controller is placed in each sub network. Affinity propagation based clustering is an exemplar based message passing method[11]. An exemplar is a node which represents a cluster it is analogous to cluster head. The proposed method identifies exemplars and their locations. The SDN controllers are placed in the nodes which are identified as exemplars. The input to this algorithm is the pair wise similarities between nodes denoted by s(i,j). We consider propagation latency between switch and controller as placement metric. Propagation latency is proportional to distance between nodes hence the proposed work choose distance between nodes as the similarity function s(i,j). After calculating the pair wise similarity, pass messages between nodes to find exemplars. The proposed system for network partitioning and placement of controllers using affinity propagation is shown in the figure 1.

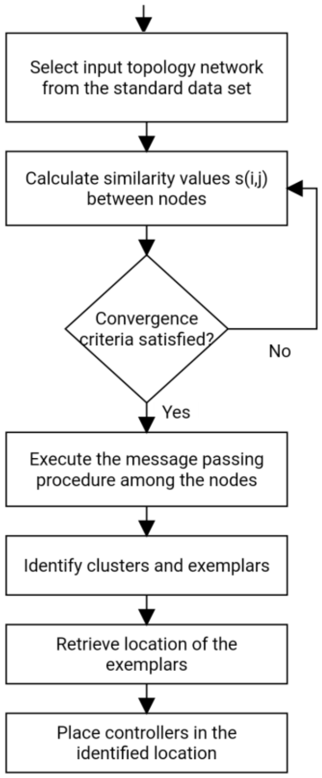


Figure 1 : System architecture for controller placement

## V. SIMULATION AND RESULTS

The Python language is used to develop the simulator[13][14] for the proposed system. The similarity measure chosen is the distance between the nodes. The simulator identifies the location of the exemplar and places the controller in the identified location.

The standard dataset for network topology is taken from the Internet Topology Zoo[12]. This platform provides dataset containing geographical location, including longitude and latitude of the interconnecting devices as well as details of edges between nodes. From the several available topologies the details of dataset used in the proposed work is shown in table 1.

Table 1 : Standard dataset of Network Topologies

| Sl Number | Name of Network Topology | Number of Nodes | Number of Edges |
|---|---|---|---|
| 1 | CWIX | 35 | 41 |
| 2 | Columbus Network | 69 | 85 |
| 3 | Bestel | 84 | 101 |

The average latency and Imbalance are the performance measures chosen to ensure the correctness of controller placement, the details are shown below.

The Average Latency : The performance metric chosen for controller placement the average propagation latency, proportional to the distance between the placement of exemplar and all associated nodes within the cluster. For the identified cluster C1 with 'n' nodes, let d(v,s) be the minimum distance between node v∈C1 and exemplar s∈C1. The average latency Lavg [4] can be calculated using the formula.

$$Lavg = \frac{\sum_{v \in C1} \text{MIN } d(v,s)}{n} \qquad (1)$$

Imbalance: The Imbalance metric[7] of a network topology is the difference between maximum and minimum number of switches associated to a controller in the network. Let C be the set of controllers and $n_c$ be the number of switches controlled by controller c∈C. Imbalance metric can be measured using the formula.

$$Imbalance = \max{}^{c \in C} n_c - \min{}^{c \in C} n_c \qquad (2)$$

The network topologies shown in table 1 are given as input to the proposed system. The simulation results shows optimal number of controllers were identified for the network topologies selected. The number of controllers and computed propagation latency is shown in Table 2.

Table 2: Number of Controllers and latency

| **Network topology** | **Number of nodes** | **Number of controllers** | **Propagation Latency(in ms)** |
|---|---|---|---|
| Bestel | 83 | 4 | 50 |
| Columbus Network | 69 | 4 | 5 |
| CWIX | 35 | 3 | 5 |

It is observed that the performance metric like propagation latency, which is proportional to the distance between controller to switches, also influence the decision of number of controllers the depiction is shown in the figure 2.

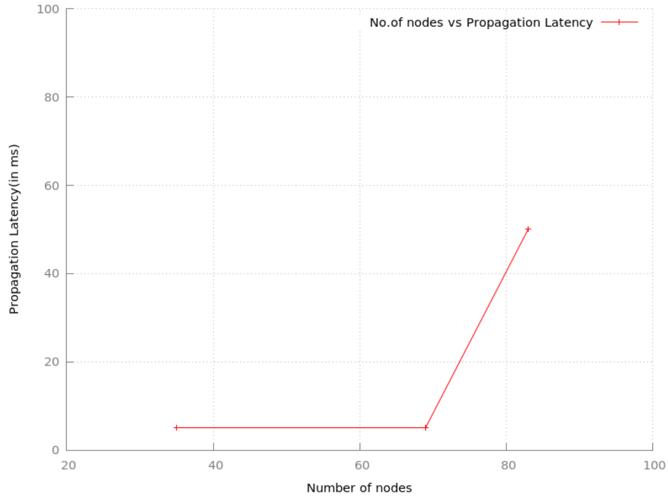


Figure 2. Variation of Propagation latency with number of nodes

The proposed system computes Imbalance as a measure difference between maximum and minimum number of switches associated to a controller in the network shown in Table 3.

Table 3: The computed Imbalance measure

| Network topology | Number of nodes | Number of controllers | Maximum number of nodes under a controller | Minimum number of nodes under a controller | Imbalance |
|---|---|---|---|---|---|
| Bestel | 83 | 4 | 15 | 28 | 13 |
| Columbus Network | 69 | 4 | 15 | 18 | 3 |
| CWIX | 35 | 3 | 7 | 17 | 10 |

It is observed the imbalance is measure of deviation from the uniform distribution of nodes among the controllers. A good controller placement scheme should minimize the measure of imbalance. The observed imbalance is plotted in Figure 3.

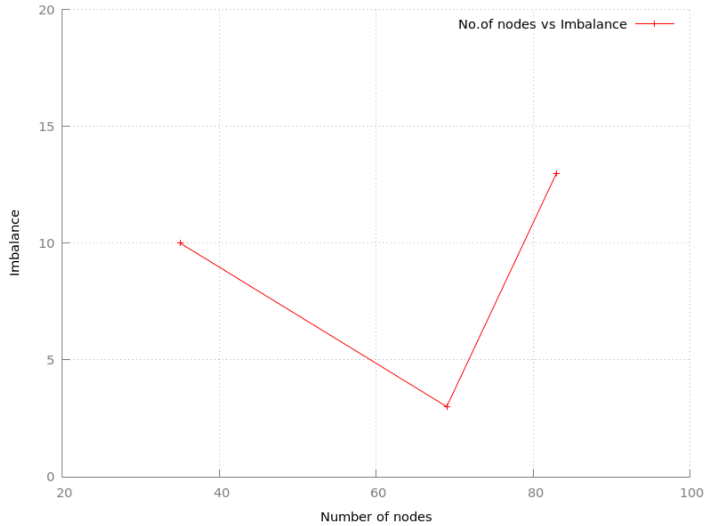


Figure 3: Imbalance of standard network topologies

## VI. CONCLUSION AND FUTURE WORKS

In this paper the proposed affinity propagation based clustering method automatically identifies the required number of network partitions. The placement the SDN controllers are done to minimize the average latency between controller and other nodes in the cluster. The imbalance metric calculated can be used to ensure controller load balancing.

As a future work we are planning to provide a solution for controller load balancing by minimizing the imbalance metric to ensure uniform distribution of switches among the controllers. Performance comparison can be made between proposed method and other clustering schemes like K-means and spectral clustering. We also plan to extend the performance analysis to find the average and worst case communication latency among the controllers and between switches and associated controller.

## REFERENCES


[1] Guodong Wang, Yanxiao Zhao, Jun Huang, Qiang Duan, Jun Li: "A K-means-based Network Partition Algorithm for Controller Placement in Software Defined Network",IEEE,2016

[2] Ian F. Akyildiz, Ahyoung Lee, Pu Wang, Min Luo, and Wu Chou, May/June 2016 “Research Challenges for Traffic Engineering in Software Defined Networks”, published in IEEE Network ,.

[3] Peng Xiao, Wenyu Qu, Heng Qi , Zhiyang Li, Yujie Xu:The SDN Controller Placement Problem for WAN,IEEE,2015.

[4] Brandon Heller, Rob Sherwood, Nick McKeown"The Controller Placement Problem;” ACM conference paper,2012

[5] Guang Yao, Jun Bi, Member, IEEE, Yuliang Li, and Luyi Guo," On the Capacitated Controller Placement Problem in Software Defined Networks", IEEE COMMUNICATIONS LETTERS, AUGUST 2014

[6] David Hock, Matthias Hartmann, Steffen Gebert, Michael Jarschel, Thomas Zinner, Phuoc Tran-Gia:"Pareto-Optimal Resilient Controller Placement in SDN-based Core Networks",IEEE, Proceedings of the 201325th International Teletraffic Congress.

[7] A. Jalili, V. Ahmadi, M. Keshtgari and M. Kazemi:"Controller placement in software-defined WAN using multi objective genetic algorithm";IEEE;2015 2nd International Conference on Knowledge-Based Engineering and Innovation (KBEI), Tehran, 2015, pp. 656-662

[8] S. Hassas Yeganeh, Y. Ganjali, Kandoo: a framework for efficient and scalable offoading of control applications, in: Proceedings of the First Workshop on Hot Topics in Software Defined Networks, HotSDN12, August 2012, pp. 1924.

[9] A. Tootoonchian, Y. Ganjali, Hyperfow: a distributed control plane for openflow, in: Proceedings of the 2010 Internet Network Management Conference on Research on Enterprise Networking, INM/WREN10, April 2010, p. 3.

[10] T. Koponen, M. Casado, N. Gude, J. Stribling, L. Poutievski, M. Zhu, R. Ramanathan, Y. Iwata, H. Inoue, T. Hama, et al., Onix: a distributed control platform for large-scale production networks, in: Proceedings of the 9th USENIX Conference on Operating Systems Design and Implementation, OSDI10, vol. 10, October 2010, pp. 16.

[11] B.J. Frey and D. Dueck, Clustering by Passing Messages between Data Points, Science, vol. 315, no. 5814, pp. 972-976, Feb. 2007

[12] S. Knight, H. Nguyen, N. Falkner, R. Bowden, and M. Roughan, “The Internet Topology Zoo,” IEEE Journal on Selected Areas in Communications, vol. 29, no. 9, pp. 1765–1775, 2011.

[13] Mark Lutz ,Programming Python, 4th Edition,Powerful Object-Oriented Programming,O'Reilly Media, December 2010

[14] Allen Downey, Think Python, How to Think Like a Computer Scientist Version 2.0.17, Copyright © 2012.Green Tea Press.

[15] Akyildiz, I., Lee, A., Wang, P., Luo, M. and Chou, W. (2014). A roadmap for traffic engineering in SDN-OpenFlow networks. Computer Networks, 71, pp.1-30.


-