# Given a SDN Topology, How Many Controllers are Needed and Where Should They Go?

**Conference Paper** · November 2018



Albert A Lysko
Council for Scientific and Industrial Research, South Africa

# Given a SDN Topology, How Many Controllers are Needed and Where Should They Go?

Lusani Mamushiane[1, 2], Joyce Mwangama[1], Albert A. Lysko[2]
*University of Cape Town[1], Cape Town, South Africa*
*Council for Scientific and Industrial Research (CSIR)[2], Pretoria, South Africa*
lravhuanzwo@csir.co.za, jb.mwangama@uct.ac.za,alysko@csir.co.za

*Abstract*—**Software Defined Networking (SDN) presents a paradigm shift in network management and configuration. The idea of having an externalized control plane opens many unanswered questions regarding scalability, fault tolerance and performance of the controller. An important question that must be answered is, given a network topology, how many controllers are needed and where should they be placed to satisfy user-specific requirements and constraints. Such requirements range from latency constraints, failure tolerance and fair load distribution. These metrics compete with each other, thus no single best placement is available. In this paper, we focus on controller placement to minimize propagation latency (of control traffic) and CapEx associated with installing a new controller. We apply Silhouette Analysis and Gap statistics to compute the optimal number of controllers to use for a given topology. To determine the optimal locations to place the controllers, we use Partition Around Medoids (PAM) clustering algorithm. We evaluate our solution using the Internet2 topology and then expand our scope to over 10 publicly available WAN topologies. As expected, the answers to controller placement are topology-dependent. However, an evaluation of our algorithms on the Internet 2 topology, recommends two controllers as the optimal number of controllers to use. Surprisingly, our results indicate that one controller suffices to meet latency requirements (though certainly not reliability requirements). Finally, the techniques presented in this work can be extended to tackle other similar placement problems, such as baseband unit placement for 5G cloud radio access network (C-RAN) deployment and fog node placement which appears in the context of edge computing.**

***Keywords—SDN, Optimization, Average latency, Worst-case latency, PAM, Controller placement, Clustering***

## I. Introduction

Software Defined Networking (SDN) presents a paradigm shift in communication networks directed towards a logically centralized architecture, which moves control plane functions from the forwarding hardware to dedicated external controller instances running in software. This decoupling enables the network to be directly programmable by various applications via APIs such as REST on the northbound interface and OpenFlow on the southbound interface. This paradigm promises to simplify network management, improve resource utilization, and enable new network innovations, all achieved through abstraction of the complexity of heterogeneous forwarding hardware. SDN is anticipated to play a pivotal role in the envisaged 5G networks. This includes dynamic flow management and orchestration of massive machine type communications.

In SDN deployments, a single controller instance is likely to suffer from scalability, performance and reliability issues as the network size grows. A potential solution to address this is to fragment the network into smaller administrative domains each supervised by a dedicated controller. Fortunately, these controllers achieve a common basic architecture with collaborative efforts enabled by their west/eastbound interfaces. However, deploying multiple-controllers calls for a technique to place them efficiently. This is commonly known as the controller placement problem [1]. The controller placement problem requires the user i.e. decision maker to define a set of objectives that must be considered during placement optimization. These objectives include: reducing latency, enhancing fault tolerance, increasing energy efficiency and so on.

Another relevant aspect that must be considered during controller placement is the number of controllers to use for a given network topology. Factors affecting this decision include but are not limited to load balancing, latency, security and fault tolerance objectives for a given network. It has been shown in [2] that these objectives compete with each other, in that optimizing one typically compromises the other. Therefore, depending on use case, controller placement may need to feature an adequate trade-off between the objectives that are relevant for that particular use case to achieve efficient operation.

The first work to address the SDN controller placement problem is by Heller et al. [1]. These authors modelled the controller placement problem as a facility location problem, an NP-hard problem which typically appears in many contexts such as optimizing the location of factories and warehouses. Instead of resorting to approximations, the authors argue that an exhaustive evaluation of the entire solution space is viable for production networks. The basic idea in their work was to find controller locations that optimize latency, specifically average and worst-case latency. To do this, the authors applied the k-center algorithm. From their simulations, authors conclude that one controller often suffices to meet latency constraints in medium size networks. They also argue that one controller is not sufficient to meet fault tolerance requirements of production networks.

This work explores different approaches to determine the optimal number of controllers for deployment in a given network, with particular focus on the latency objective. This is followed by determining the optimal locations of these controllers to achieve minimum average latency and worst-case latency. Although the controller placement problem has been studied in the past, there is no study (to the best of our knowledge) that explores the solutions to determining the optimal number of controllers to deploy for a given topology. In previous studies, the number of controllers was assumed to be known in advance. Moreover, our optimization solution applies the Partition Around Medoids

(PAM) clustering algorithm for placement which has to the best of knowledge not been used in the context of controller placement. Most previous studies used heuristic algorithms which are commonly known to sacrifice accuracy for speed. On the contrary, our PAM algorithm optimizes accuracy instead of speed, which we believe is of greater significance in the context of controller placement.

The structure of this paper is as follows: Section II presents related work; Section III describes the problem formulation; Section IV presents the simulation results and analysis, and lastly Section V concludes the paper.

## II. Related Work

The publication coverage on research work exploring the controller placement problem can be broadly divided into two categories. In the first category the authors consider node-to-controller latency for their controller placement optimization, as exampled by Heller et al. [1], Hu et al. [3], Hock et al. [4], and Lange et al. [2].

Hu et al. [3] argue that production networks require multiple controllers to maximize network resiliency. Authors introduce and compare different heuristic algorithms such as l-w greedy and brute force to increase the resilience of SDN controllers against node-to-controller link failures. The results from these evaluations show that l-w greedy yields the most optimal results compared to brute force. The drawback of this solution is that the number of controllers was assumed to be known in advance.

Hock et al. [4] and Lange et al. [2] advocate for careful consideration between latency and reliability (defined as resilience against node and link failures and load balancing in the control plane). These authors propose a framework for resilient Pareto-based Optimal Controller placement called POCO that provides decision makers with all pareto-optimal placements of controllers in realistic networks. Their results suggest that in the event of either link or node failure, the best resiliency is achieved when more than 20% of all nodes used are controllers. Their proposed framework can be leveraged to implement a scalable and reliable control plane. However, instead of segmenting the network into multiple domains, they treat the network as a whole and the controllers work collaboratively. This approach requires frequent exchange of state information between the controllers to achieve an accurate global state awareness. This puts the network at risk for inter-controller broadcast storm which significantly affects inter-controller latency. These authors provide a user friendly interactive GUI that enables decision makers to explore the solution space and perform various what-if analysis.

The second category of research work done on controller placement is that by Rath et al. [5], Sallahi et al. [6], Jimenez et al. [7] and Yao et al. [8]. In their research, the authors consider both latency and data plane load to address the controller placement problem. The solution of Rath et al. [5] proposes the use of game theory to ensure maximum utilization of controllers. This solution unfortunately does not define where to place the SDN controllers in the network. Sallahi et al. [6] proposes a mathematical formulation to optimize the number of controllers to deploy. However this solution suffers the same drawback as that presented by Rath et al. in that it does not take into account controller location optimization.Yao et al. [8] proposes the divide and conquer philosophy where the network is partitioned into multiple segments to facilitate load balancing and network stability. The objective of this work was to ensure that no controller is stressed beyond its capacity at a given time. Authors propose a capacitated k-center algorithm. This solution differs from that proposed by Jimenez et al. [7] in the way that the load is considered. The solution of Yao et al. is optimized for heterogeneous data plane traffic. On the contrary, Jimenez et al. assumes homogeneous traffic. Both these solutions define controller placement based on fixed traffic load observed initially, but do not adapt to the dynamic traffic load. This shortcoming is addressed by Bari et al. [9] and Jourjon et al. [10] who propose a solution for dynamic controller placement i.e. controller placements to support load variation over time. The authors focus their metrics on controller utilization and latency (both node-to-controller and inter-controller latency) for optimal management of dynamic flows. These solutions rely on trial and error to estimate the optimal number of controllers to use.

As demonstrated in [4], the controller placement problem involves a number of competing metrics, thus confronting the decision maker with trade-offs between them. Our work strictly focuses on network propagation latency as an important QoS determinant in the network. We aim to address two questions: (i) given a network topology, how many controllers are needed, and (ii) where should the go to optimize propagation latency? To the best of our knowledge, previous studies on controller placement assumed the number of controllers to be known in advance. Our work proposes various approaches to use for determining optimal number of controllers to deploy in SDNs. We also propose the use of an exhaustive algorithm called Partition Around Medoids (PAM) for determining optimal locations to place controllers in SDNs.

## III. Problem Formulation

### A. Assumptions

Our mathematical formulation is based on the following assumptions:

- Switch-to-controller communication is assumed to happen in-band ;
- The bandwidth for all connection links is constant;
- The inter-controller communication has been solved perfectly to address the inter-controller broadcast storm under network segmentation;
- Control path security has been perfectly solved;
- Controller and switches are co-located;
- Switches incur a fixed load.

As stated in Section II, our primary goal is to optimize the number of controllers to deploy in an SDN-enabled WAN. Secondly we determine the optimal locations to place these controllers particularly focusing on two QoS parameters namely, the average propagation latency and the worst-case propagation latency. This is an NP-hard problem as it cannot be solved in polynomial time.

We mathematically formulate our problem as follows: the network topology is modelled as an undirected graph $G(V,E,L)$, with $V$ denoting the network switches, $E$ denoting connections between network nodes and $L$ representing switch locations (longitude and latitudes). For our model, $L_{avg}$ represents the average propagation latency and $d(v,z')$ is the shortest distance from the switch (node $v \in V$) to the controller (node $Z'$), and the number of nodes is $n=|V|$, the average propagation latency for the placement of $Z'$ is determined as per equation (1).

$$L_{avg}(Z') = \frac{1}{(2\times10^8)N} \sum_{v\in V} min\ d(v,z') \quad (1)$$

Another metric to optimize is the worst-case latency, defined as the maximum switch-to-controller latency. Equation (2) shows the formal definition of worst-case latency.

$$L_{wc}(Z') = max_{v\in V}\ min_{z\in Z'} d(v,s) \quad (2)$$

In the corresponding optimization problem, the goal is to find the placement $Z'$ from the set of all placements $Z$ such that $L_{avg}$ and $L_{wc}$ are optimized.

*B. Algorithms*

**Silhouette Analysis:** As mentioned before, deploying a single controller instance in large-scale SDNs affects data plane scalability and presents a single point of attack. To address this, large-scale networks are typically segmented into smaller manageable clusters each supervised by a dedicated controller. An important question that we aim to address is: given a SDN-enabled network, how many controllers are required to meet user-defined objectives, particularly the intra-cluster latency variation? To achieve this we employ Silhouette Analysis [11]. Silhouette is used to study the proximity of nodes in one cluster to those in adjacent clusters. This measure can be used to determine the optimal number of clusters/controllers to use for a given topology by evaluating cluster quality. This is to say, the number of controllers that minimize the total intra-cluster latency variation as defined by equation (3). $C_k$ is the $k_{th}$ cluster and $L(C_k)$ is the intra-cluster latency variation.

$$min\ \sum_{k=1}^{n} L(C_k) \quad (3)$$

Algorithm 1 outlines the structure of the Silhouette approach. The input consists of three parameters namely, the topology graph *G(V,E,L)*, the desired maximum number of controllers *k*, haversine distance function handle, and the clustering algorithm for optimization of the intra-cluster latency variation. The clustering algorithm used is Partition Around Medoids (PAM) [11] described in the paragraphs to follow. The number of controllers that yields the maximum silhouette score is considered the optimal value. This score has a range of [-1, 1].

**Algorithm 1:** Silhouette Analysis

1. **Input** $G(V,E,L)$ ←network graph
2. **Input** $k$ ←number of controllers
3. **Input** $d$, $Clust$ ←distance function handle and clustering algorithm
4. **for** different values of k **do**
   compute intra-cluster variation
5. Calculate the average silhouette coefficient of observations
6. **plot** number of controllers against silhouette coefficients
7. **return** $optimalK$ ←the optimal number of controllers

**Gap Statistic:** This is an alternative method for determining the optimal number of controllers to use for a network topology requiring fragmentation. To do this, Gap statistic compares the total sum of intra-cluster latency variation for a varying number of controllers with the expected null reference distribution of nodes i.e. distribution with no obvious clustering [12]. The objective is to compute the number of controllers that maximizes the gap value ($Gap_n(k)$) as defined by equation (4). The first term on the right hand side of (4) represents the expected intra-cluster latency variation from the reference dataset, $L(C_k)$ is the within-cluster latency variation of the original data, and $Gap_n(k)$ is the gap value for a given number of controllers. Parameter $B$ (in equation (5)) denotes the size of the reference dataset. The optimal number of controllers is one that meets the condition outlined in equation (6), where $s_k$ is the simulation error calculated from the standard deviation of $B$ Monte Carlo replicates [13]. Algorithm 2 outlines the steps followed in our analysis using Gap statistic.

$$Gap_n(k) = E_n^*\{log(\ L^*(C_k)\} - \log(L(C_k)) \quad (4)$$

$$\text{Where} \quad E_n^*\{lo\ g(\ L^*(C_k)\} = \left(\frac{1}{B}\right) \sum_b \log(L^*(C_{kb})) \quad (5)$$

$$Gap(k) \geq Gap(k+1) - s_{k+1} \quad (6)$$

**Algorithm 2:** Gap Statistic

1. **Input** $G(V,E,L)$, $k$ ← network graph, number of controllers
2. **Input** $d$, $Clust$ ← distance function handle and clustering algorithm
3. **for** varying k **do**
4. Run clustering algorithm on original data to find k clusters
5. Calculate $L(C_k)$ ← intra-cluster latency variation
6. Generate a set of reference datasets same ← size as original data
7. **for** $b$=1,2,.......$B$ **do**
8. Calculate $L^*(C_{kb})$ ← intra-cluster latency variation of reference
9. Calculate the mean of reference datasets ← as per equation (5)
10. Compute $Gap_n(k)$ ←gap value
11. Calculate $S_k$ ← standard deviation of B Monte Carlo replicates
12. Take the gap value that satisfy equation (6)
13. **return** $optimalK$ ← optimal number of controllers

**Partition Around Medoids (PAM):** In order to determine the optimal locations to place the controllers we use the PAM [11] clustering solution. The QoS parameter to optimize here is the propagation latency (particularly worst-case and average latency). Unlike clustering algorithms such as k-means, PAM is more robust in the presence of noise and outliers whereas k-means is extremely sensitive to outliers and other extreme values [14]. Since we assume in-band communication, we use Johnson's algorithm to compute the shortest path matrix between all node pairs. The Johnson's algorithm constitutes a combination of Bellman-Ford and Dijkstra's algorithms. Algorithm 3 describes the steps followed in optimizing latency using PAM and Johnson's algorithm. The input parameters are: the network graph $G(V, E, L)$, edge weights, harvesine distance function handle, and the number of controllers defined by the number clusters computed from Silhouette Analysis and Gap Statistic. The output of this algorithm is the cluster indices of each observation, the optimal locations of SDN controllers, and the intra-cluster distances from each switch to the controller. The overall complexity of this algorithm is $O(V^2 \log V + VE)$+$O(k(n-k)^2)$, where $n$ is the network size and $k$ is the number of controllers.

**Algorithm 1:** PAM clustering

1. **Input** $G(V, E, L)$ ←network graph
2. **Input** $d, k$ ← distance function handle, number of controllers
3. **Input** $w$ ← edge weights
4. Compute shortest path matrix using Johnson's algorithm
5. Select k representative switches arbitrarily
6. **for** each pair of non-selected switch $v$ and selected switch $l$ **do**
7. compute the overall swapping cost $TC_{vl}$
8. **for** each pair of $v$ and $l$ **do**
   If $TC_{vl}$< 0,
   Substitute $l$ by $v$

**7.** Assign each non-selected switch to the most similar representative switch

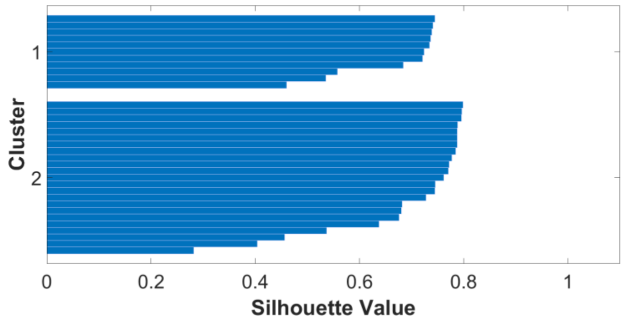
8. Repeat steps 6 -8 until there is no change

9. **Output:** $idx, CL, sumd, d$ ←cluster indices for each observation, optimal controller locations, intra-cluster sums, and intra-cluster distance from each switch to controller

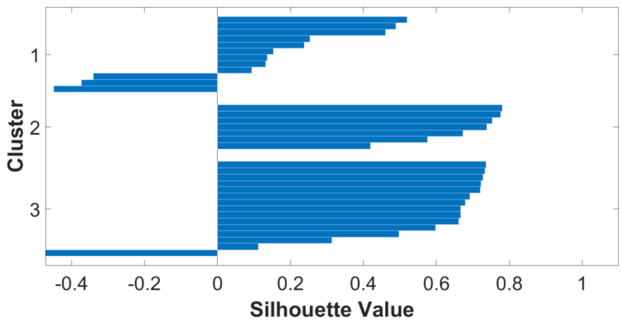
## IV. Simulation Results

Based on the evaluation techniques introduced in Section III, computations with different objectives are performed. The first objective is to identify algorithms that may be used to determine the optimal number of controllers to use for a given topology. Secondly, we define the best location for these controllers to improve overall QoS delivered by the network. The first objective is addressed using Silhouette Analysis and Gap Statistic whereas the second objective is addressed using PAM algorithm. Our assessment of network QoS is based on propagation latency. The key factor in our mathematical model is the distance between node pairs while the bandwidth is constant across all sites. Therefore under constant bandwidth, propagation latency is directly proportional to distance. To maintain realism, our optimization solution was applied to the Internet 2 OS3E topology (an SDN deployment of 34 nodes and 41 fibre links developed for research and testing purposes). The dataset for Internet 2 OS3E was derived from the node locations presented by the topological map in [15].

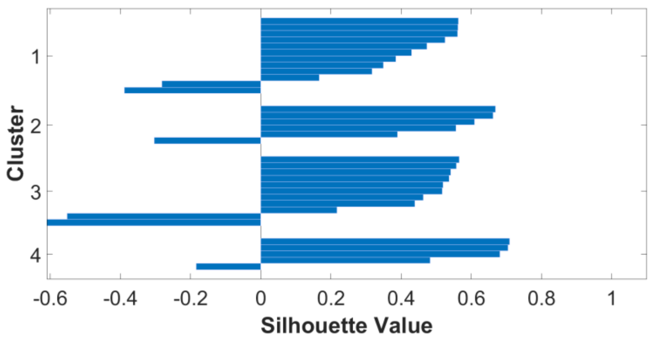
Figure 1 illustrates the clustering quality for different number of controllers measured by the intra-cluster latency variation based on Silhouette Analysis. Silhouette scores near +1 indicate a high dissimilarity between a cluster and its neighbouring clusters. On the other hand, Silhouette scores near -1 indicate high similarity between clusters and is a sign of poor clustering quality. Our results indicate that deploying two controllers is an optimal choice due to a quantitatively large number of nodes with high silhouette scores. However three or four controllers would be a back pick as there are points with negative silhouette scores meaning a high inter-cluster similarity between nodes. This is summarized in Figure 2, which shows a high silhouette score for two controllers, indicating that best number of controllers to use on Internet 2 OS3E is two controllers.




(a)




(b)




(c)

Figure 1: Silhouette analysis to determine optimal number of controllers for (a) $k$ =2, (b) $k = 3$ and (c) $k = 4$ .

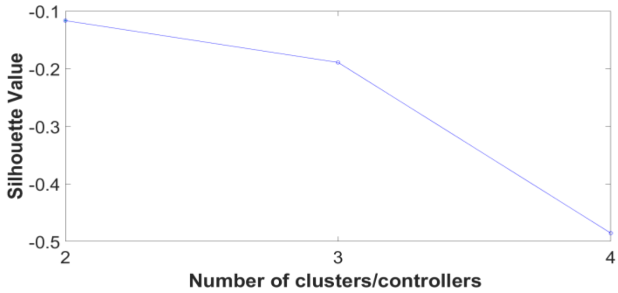



Figure 2: Silhouette evaluation summary

To verify our results from Silhouette Analysis, we use Gap Statistic algorithm. With Gap statistics, the desired number of controllers is one with the highest gap value as it indicates a low intra-cluster latency variation. The results from the gap statistic (as illustrated in Figure 3) recommend 2 controllers as the optimal number that minimizes the intra-cluster variation thus yielding lower propagation latency. This coincides with the results from the silhouette evaluation.

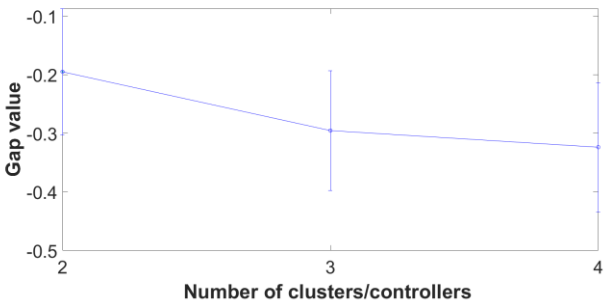



Figure 3: Determining optimal number of controllers using Gap Statistic

Another important metric to consider when choosing the number of controllers to deploy is the cost of installing new controllers in a given network. This metric is important as it considerably contributes to the overall CapEx associated with an SDN deployment. However, there is a significant tradeoff between cost and QoS delivered by the network. Our goal here was to quantify this tradeoff to provide a practical guideline to decision makers, regarding the optimal number of controllers to use in consideration of cost and the latency QoS parameter. We call this tradeoff "cost benefit". We define a figure of merit for the cost benefit by taking the ratio of the controller cost ($C_k$) to average latency as shown by equation (5).

$$cost\ benefit = \frac{k*C_k}{L_{avg}} \quad (5)$$

Unsurprisingly, our results (see Figure 4) indicate that 1 controller is an ideal choice that gives the least tradeoff between cost and propagation latency. However to meet scalability and fault tolerance requirements, we recommend using two controllers. This is because two controllers are the second best choice that provides the least tradeoff between cost and latency, and it also coincides with our solution from Silhouette Analysis and Gap Statistic evaluations.

When we applied our algorithms (Gap Statistic and Silhouette Analysis and Cost benefit) to 10 other topologies, we realised that the optimum number of controllers is not fixed but rather is dependent on the network topology. Moreover, the decision on how many controllers to use also depends on the unique needs and constraints of service providers.

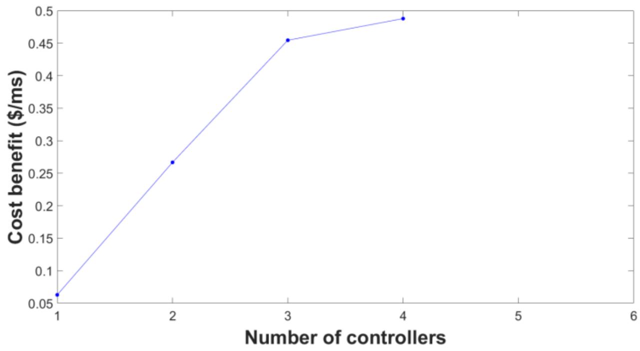



Figure 4: Optimal number of controllers based on cost benefit

The next step in our work was to define where to place the recommended two controllers in the Internet 2 OS3E topology. This involved the use of the PAM algorithm to determine placement locations that minimize the network average latency. As shown in Figure 5, the optimal controller placements when the number of controllers is two (*k*=2) is Salt Lake City and Atlanta which are the best locations that yield minimum average latency ($L_{avg} = 0.0036\ ms$). Deploying SDN controllers in these locations guarantees best network performance with respect to the southbound communication.

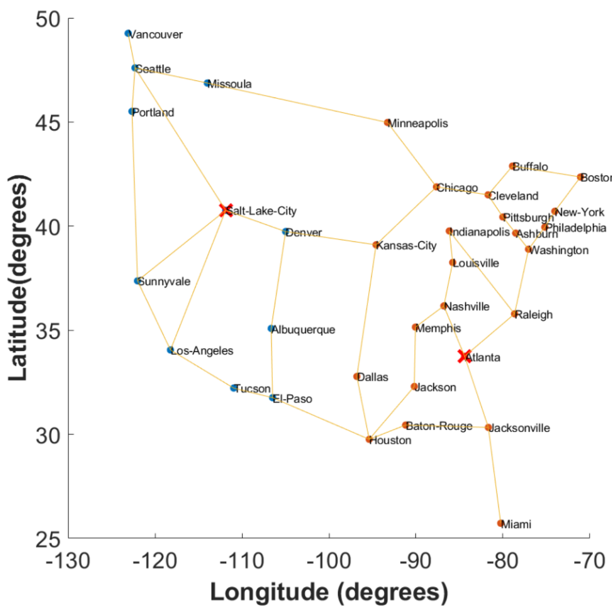



Figure 5: Optimal placements of 2 SDN controllers in Internet 2 OS3E topology

Table 1 summarizes the QoS outcomes of our algorithm when the number of controllers is varied from one to four. From our results, we observe a significant decrease (about 53% reduction) in overall latency when the number of controllers is changed from one to two; in this case network performance was improved. However, further increases have a much less significant effect on overall latency compared to this first increase (as shown also in Figure 6). The same is observed for the worst-case latency.

Table 1: Average and worst-case latency when the number of controllers is varied from one to four.

| | $k = 1$ | $k = 2$ | $k = 3$ | $k = 4$ |
|---|---|---|---|---|
| $L_{avg}$ | 0.0076 | 0.0036 | 0.0029 | 0.0026 |
| $L_{wc}$ | 0.0159 | 0.0075 | 0.0082 | 0.0066 |
| **Names of locations for** $L_{avg}$ | Kansas City | Salt Lake City<br>Atlanta | Washington DC<br>Salt Lake City<br>Dallas | Jackson<br>Salt Lake City<br>Jacksonville<br>Ashburn |
| **Names of locations for** $L_{wc}$ | Vancouver | Vancouver<br>Boston | Nashville<br>Vancouver<br>Chicago | Dallas<br>Vancouver<br>Houston<br>Boston |

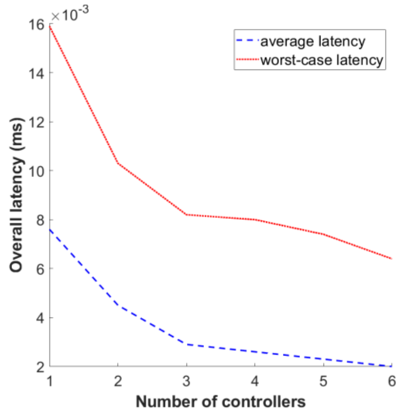



Figure 6: Relation between number of controllers and latency

## V. Conclusion

Decoupling the control plane from the data place poses several challenges with respect to network scalability, fault tolerance, and performance (throughput and latency). Before deploying an SDN-enabled network, a decision must be made on the optimal number of controllers required for a given topology and their respective geographic placements. Our work proposed a guideline to assist decision makers in this regard. We proposed three algorithms namely, Silhouette, Gap Statistics and PAM to address the controller placement problems. These are exhaustive algorithms which don't work well in the presence of time constraints but are more accurate than heuristic algorithms. These algorithms were applied on the Internet 2 OS3E topology and on over 10 topologies obtained from the Topology Zoo database. Our assessment was based on latency and cost of installing SDN controllers. While the solutions are dependent on network topology itself, many networks require a reasonable number of controllers for optimal performance. Although our work assumed in-band communication, it can also be applied to out of band SDNs. Our work can be used by potential service providers who would like to transition to SDN, to mitigate their concerns of having an externalized control plane. This work can also be extended to other placement problems such as baseband unit (BBU) placement for 5G Cloud Radio Access Network (C-RAN) deployment which appears in the context of Network Function Virtualization (NFV) and Fog node placement for edge computing optimization. In future, we intend to integrate dynamic load balancing as well as fault tolerance optimization to our solution. Lastly we plan to evaluate our model on an emulation orchestration platform.

## References


[1] B. Heller, R. Sherwood and N. McKeown, "The controller placement problem," in *in The first workshop on Hot topics in software defined networks*, 2012.

[2] 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.

[3] Y. N. Hu, W. D. Wang, X. Y. Gong, X. R. Que and S. D. Cheng, "On the placement of controllers in software-defined networks," *The Journal of China Universities of Posts and Telecommunications,* vol. 19, pp. 92171-97, 2012.

[4] D. Hock, M. Hartmann, S. Gebert, M. Jarschel, T. Zinner and P. Tran-Gia, "Pareto-optimal resilient controller placement in SDN-based core networks," in *Teletraffic Congress (ITC)*, 2013.

[5] H. K. Rath, V. Revoori, S. M. Nadaf and A. Simha, "Optimal controller placement in Software Defined Networks (SDN) using a non-zero-sum game," in *IEEE 15th Int. Sym. on A World of Wireless, Mobile and Multimedia Networks*, 2014.

[6] A. Sallahi and M. St-Hilaire, "Optimal Model for the Controller Placement Problem in Software Defined Networks",," *IEEE Communications Letters,* vol. 19, no. 1, pp. 30-33, 2015.

[7] Y. Jimenez, C. Cervello-Pastor and A. J. Garcia, "On the controller placement for designing a distributed SDN control layer," in *IFIP Networking*, 2-14.

[8] G. Yao, J. Bi, Y. Li and L. Guo, "On the capacitated controller placement problem in software defined networks," in *IEEE Communications Letters*, 2014.

[9] M. F. Bari, "Dynamic controller provisioning in software defined networks," in *Proc. Int. CNSM*, 2013.

[10] M. T. Huque, G. Jourjon and V. Gramoli, "Revisiting the Controller Placement Problem," in *40th Annual IEEE Conference on Local Computer Networks*, 2015.

[11] Chapman and Hall, Data Clustering: Algorithms and Applications, CRC Press, 2006.

[12] M. Mohajer, K. H. Englmeier and V. J. Schmid, " A comparison of Gap statistic definitions with and without logarithm function," 2011.

[13] R. Tibshirani, G. Walther and T. Hastie, "Estimating the number of clusters in a data set via the gap statistic," *Journal of the Royal Statistical Society: Series B (Statistical Methodology),* vol. 63, no. 2, pp. 411-423, 2001.

[14] I. A. Venkatkumar and S. J. K. Shardaben, "Comparative study of data mining clustering algorithms," in *Data Science and Engineering (ICDSE),* , 2016.

[15] "Internet2," 2017. [Online]. Available: https://www.internet2.edu/media/medialibrary/2017/09/25/I2-Network-Infrastructure-Topology-Alllogos-201705_hr8gwSg.pdf.