Optimum Placement of SDN Controllers in African Backbones: SANREN and ZAMREN as a Case Study

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) has emerged as a potential solution to the ICT inequality challenge in emerging markets. This technology promises to revolutionize the telecommunications industry by introducing decoupled architectures to facilitate network management and configuration. A consensus was reached that a huge portion of OpEx comes from the cost associated with the management and configuration of tightly coupled legacy networks. This has contributed to operators' reluctance to extend broadband coverage to the poor rural areas and sparsely populated areas due to the potential low profit margins. SDN opens unprecedented opportunities such as non-discriminatory infrastructure sharing, hardware commoditization (through the use of cheap commodity hardware), and business agility. This is likely to encourage operators to cover rural areas with low or no network footprint. At the heart of SDN is a controller with a global view of the current network status. It is critical that this controller is placed in a manner that optimizes network performance. This design choice is commonly known as the controller placement problem (CPP). This paper proposes an algorithm for placement of the controller that optimizes network performance, particularly propagation latency. The algorithms are tested on two African backbones, namely SANREN and ZANREN.**

Keywords—**SDN, Optimization, Average latency, worst-case latency, Partition Around Medoids (PAM), Controller placement, SANREN, ZAMREN, Johnson's algorithm.**

I. INTRODUCTION

To date, broadband penetration is critically low in most emerging economies. This is predominantly because of the cost associated with rolling out a new broadband infrastructure relative to the potential profit margins. Operators are more comfortable to rollout network infrastructure in urban areas than in rural areas, due to the attractive return on investment promised by urban areas. The repercussion of this is a wide digital divide between urban and rural areas. In order to bridge this divide, a more robust and cost effective telecom infrastructure is indispensable. Additionally, a stronger telecom infrastructure is pivotal to economic growth [1]. Without this in place, the vision of the South African National Development Plan (NDP) to eliminate poverty and reduce inequality by 2030 remains farfetched.

The vision of the NDP are well embraced in the National Broadband Policy ("SA Connect") which aims to achieve 100% broadband connectivity by the end of 2030 [2]. In order to develop the telecom infrastructure necessary to realise this vision, there needs to be a paradigm shift in networking. This means adopting an architecture that minimizes both network infrastructure deployment costs (CapEx) and running costs (OpEx), while maintaining a high service quality level and ensuring business agility. It has been shown in [3] that a large portion of OpEx comes from the costs associated with the configuration and management of the telecom infrastructure. This is because legacy networks make use of manual and vendor-specific configurations which is both time consuming and error prone, thus requiring specially trained networking professionals. On the other hand, CapEx largely constitute deployment costs related to networking equipment and trenching costs which are significantly high for rural coverage [4].

Software Defined Networking (SDN) and Network Function Virtualization (NFV) have emerged as the next-generation networking paradigms that can be leveraged to deploy a much stronger telecom infrastructure to help combat the poverty and inequality plaques in emerging economies like South Africa. NFV allows virtualizing the network functions through hardware commoditization, which considerably facilitates the deployment and the orchestration of the network resources and reduce CapEx. SDN promises to dramatically improve network management and configuration, reduce operating costs, and stimulate innovation and evolution within the telecoms arena. To achieve this, SDN decouples the control logic from the forwarding hardware (e.g. routers and switches) and moves it to a centralized controller. This results in a three-tiered architecture constituting high level abstractions namely, northbound and southbound interfaces. The northbound API hides the complexity of the underlying forwarding hardware from applications such as firewall, load balancing, and orchestration. This enables applications to apply new services as policies instead of mechanisms. Leveraging the exposed northbound API, applications directly push new traffic engineering policies to the controller. Using its global view of the network status and its southbound API, the controller dynamically manages and configures the behaviour of the forwarding hardware based on the traffic engineering policies programmed by applications. Some of the southbound APIs supported by SDN controllers include but not limited to, OpenFlow, PCEP, LISP, BGP-LS, etc. [5]. A prevalent choice for a northbound API is the RESTful networking protocol [5].

Decoupling the control logic from the forwarding hardware means the delivery time of control traffic to the forwarding devices largely impacts the performance of SDNs, especially when in-band communication is used. The delivery time constitutes, propagation, queuing, and processing latency. Propagation latency is a measure of the time taken by packets to reach their intended destination. Queuing latency is the time packets wait to be transmitted. Finally, processing latency is the time taken to examine the packets' headers and determine where to direct packets. There is a general consensus that propagation latency typically dominates in WANs. Propagation latency is primarily a function of the location of the source and destination nodes. In the context of SDN, it means that the location of the controller from the perspective of the SDN switches and/or routers is directly proportional to the propagation latency. Therefore, controller placement is an important design problem that must be addressed prior to SDN deployment. This is indispensable in wide area production networks where guaranteed QoS is desired. These networks are typically segmented into several smaller administrative domains each supervised by a dedicated controller. This is necessary to facilitate network scalability, incremental deployment, and to address potential security threats [6]. Thus for a given network topology, it is important to determine the optimal number of controllers to use such that the overall latency is minimized while maintaining a fair distribution of load between controllers. This problem is henceforth referred to as the Controller Placement Problem (CPP).The publication coverage of this topic is quite limited in the context of emerging markets, mainly because SDN adoption in emerging markets is still in its infancy.

In this paper, we study the CPP in the context of emerging markets by optimizing propagation latency through proper placement of the SDN controller in the SANREN and ZAMREN backbones. We propose the use of Johnson's algorithm and the Partition Around Medoids (PAM) clustering algorithm for optimal placement of SDN controllers.

This paper is organized as follows: Section II defines the problem statement; Section III presents related work; Section IV describes the system model and problem formulation; Section V presents the simulation results and analysis, and lastly Section VI concludes the paper.

II. Problem Statement

Since SDN controllers are primarily responsible to provide services such as programming flow entries on the switches, load balancing, threat detection and link layer fault recovery, the switch-to-controller latency (propagation latency) is an important QoS parameter for network services. Other metrics matter, such as reliability, load balancing and throughput. However, our focus is the WAN where latency dominates. This latency is largely dependent on the placement of the controller relative to the switches. Controller placement within local area networks (LANs) like data center networks (DCNs) has been presented to be relatively simpler. However in WANs, controller placement is an NP hard combinatorial optimization task that cannot be solved in polynomial time. In summation to this, it is paramount to determine the optimal number of controllers to use for a given WAN prior placement. This is because a single controller is likely to yield suboptimal performance in terms of latency, load balancing, scalability and security. Therefore the overall problem that must be addressed is: given an SDN-enabled WAN, how many controllers are needed and where should they go to optimize propagation latency?

III. Related Work

To date the most relevant studies exploring diverse algorithms in an attempt to address the controller placement problem can be found in [7]- [13] . Heller et al. [7] studied the controller placement problem by examining the effect that placement has on propagation latencies namely, average-latency and worst-case latency. The algorithm used for this study is called the k-center algorithm. Authors carried out their study on the Internet2 OS3E topology [8] as well as on over 100 public WAN topologies. They find that most networks show diminishing returns for each added controller along with tradeoffs between worst-case latency and average latency. Authors conclude that one controller often suffices to meet existing response time requirements in medium-size networks. However they also state that one controller is not enough to meet fault tolerance requirements.

Another important design metric to consider during controller placement is reliability also known as fault tolerance. Reliability-aware controller placement has been studied and explored in [9]- [10]. Hu et al. [9] proposes the use of multiple controllers on large scale SDN networks. The design objective in this study was to maximize the expected percentage of valid control paths. Authors define control paths as routes between switches and controllers as well as routes between controllers. To optimize reliability, authors compare the performance of optimization algorithms namely, random placement, l-w greedy and brute force. After running simulations on the Internet2 OS3E topology and an ISP topology called Rocketfuel [11], random placement produced dismally poor performance results, while brute force obtained optimal results after an extremely long execution time. Consequently, l-w greedy was chosen as the most optimal placement algorithm. However latency optimization was not considered in this study. Moreover, the number of controllers was assumed to be known in advance.

Wendong et al. [10] compared (through simulation) the benefits of several algorithms namely, random placement, l-w greedy and simulated annealing. Their simulation results showed that simulated annealing algorithm performs the best. Moreover, the results showed that although the placement of controllers may be optimal, the number of controllers used must be chosen carefully. It was shown that using too few controllers reduces reliability. Similarly using too many controllers has an adverse effect on reliability especially in networks with hierarchical control architectures. Lastly, authors analysed tradeoffs between latency and reliability and found that there exist significant tradeoffs between these metrics.

Yao et al. [12] proposed the divide and conquer philosophy were the WAN is segmented into several smaller administrative domains to facilitate load balancing and network stability. The idea was to optimize load balancing by ensuring that the load of each controller does not exceed its capacity at a given time. The authors propose an algorithm called the capacitated k-center algorithm, which proved to significantly reduce the number of required controllers compared to the k-center algorithm proposed by Heller et al. [7].

As demonstrated by Hock et al. [13], there exists a significant tradeoff between latency, load balancing and reliability with solving SDN controller placement problems. This means it is virtually impossible to optimize one metric without compromising the other. This work attempts to solve the controller placement with strict focus on latency. This metric has emerged as an important QoS determinant in SDN and must be optimized during network planning. Our primary focus is on optimizing African backbones namely, SANREN and ZAMREN to facilitate deployment of SDN in emerging markets

IV. System Model and Problem Formulation

The main goal of this work is to optimize an SDN-enabled WAN through controller placement, particularly focusing on two QoS parameters namely, the average propagation latency and the worst case propagation latency. This is an NP-hard problem which requires an input for k (the number of controllers to place). In our mathematical model, the network topology (or WAN) is modelled as an undirected graph $G(V, E, L)$, V representing the network nodes, E representing the edges (fiber links) and L representing the GPS location of the switches in degrees. For our model, L_{avg} represent 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 latency for the placement of Z' is:

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

An alternative metric to optimize is the worst-case latency, defined as the maximum switch-to-controller latency:

$$L_{wc}(Z') = (max_{v\in V} min\ d(v, s)_{z\in Z'}) \frac{}{2\times10^8} \qquad (2)$$

In the corresponding optimization problem, the goal is to find the placement Z' from the set of all placements Z such that either L_{avg} or L_{wc} is minimized.

A. Assumptions

The development of the mathematical model was based on the following assumptions:

- All switches possess the capability to run a software-based SDN controller;
- Node-controller communication is assumed to happen in-band i.e. control and regular traffic share the same physical links;
- The bandwidth for all fiber links is constant;
- The inter-controller communication has been solved perfectly to address the inter-controller broadcast storm in large scale networks;
- Control path security has been perfectly solved;
- Controller and switches are co-located;
- Documentation exists for all WAN switch locations.
- The controller can handle the load under its supervision

B. Algorithms

Partition Around Medoids (PAM): As mentioned before, deploying a single control entity in SDN-enabled WANs presents a single point of failure/attack and affects the scalability of the network. As a result, WANs are normally partitioned into smaller administrative domains each managed by a dedicated controller. However, it is critical to ensure proper placement of these controllers to guarantee QoS which in our case is the propagation latency. In our mathematical model, we use the Partition Around Medoids (PAM) clustering solution for network partitioning while guaranteeing minimum propagation latency (worst-case and average latency). This is because unlike the classical partitioning methods such as k-means clustering, PAM is more robust in the presence of noise and outliers whereas k-means is extremely sensitive to outliers and other extreme values [14]. Our clustering solution as shown in Algorithm 1 accepts $G(V, E, L)$, a network graph with switch geographical locations (longitude and latitude), d custom distance function and k the number of clusters which is analogous to the number of controllers. The initial step is to randomly select k controller locations. The next step is to associate each switch to the closest controller location using the custom distance function. For each controller location l and switch v associated to l, v and l are swapped. Next the average dissimilarity TC_{vl} of switch v to all the switches associated to l is computed. Finally, the point with the lowest dissimilarity is selected as the best location. The output of this algorithm is the cluster indices of each observation, the geography location of controllers, and the distance from each switch to the controller in its domain. The computation complexity of this algorithm is $O(k(n-k)^2)$, where n is the number of switches and k is the number of controllers.

Algorithm 1: PAM clustering

1. Input: $G(V, E, L)$ network graph with switch locations

2. Input: d, k distance function and number of controllers

3. Select k representative switches arbitrarily

4. for each pair of non-selected switch v and selected switch l , calculate the total swapping cost TC_{vl}

5. for each pair of v and i, If TC_{vl}< 0, i is replaced by v

6. then assign each non-selected object to the most similar representative object

7. Repeat steps 4 -6 until there is no change

8. Output: $idx, CL, sumd, d$ cluster indices of each observation, controller locations, within cluster sums, and distance from each switch to controller

Johnson's algorithm: This algorithm is used to compute the shortest distance matrix used by the PAM for optimal controller placement. Johnson's algorithm is a well-known optimization algorithm for computing the shortest path between all node pairs in a network [15]. Johnson's algorithm constitutes three major steps as outlined in Algorithm 2. First, an artificial source node q with zero-edge weights is added to the network graph $G(V, E)$, to get a modified graph G'. Next the Bellman-Ford algorithm is run on graph G' with source node q to find all shortest paths $h(v)$

from q to each node v. If this step detects a negative weight cycle, the Bellman-Ford algorithm is terminated. Next the edges of the original graph $G(V, E)$ are recalculated using output from Bellman-Ford algorithm. Finally, q is removed and Dijkstra's algorithm is run to compute shortest paths from each node v to every other vertex in the reweighted graph. The overall time complexity is $O(V^2 \log V + VE)$.

Algorithm 2: Johnson's algorithm

---

1. Input: $G(V, E), w$ network graph and associated edge-weights

2. Compute G' where $V|G'| = G|V| \, U \, \{s\}$

$E|G'| = E|G|U \, (s, v): v \in V[G]$

3. for all $v \in V[G]$ **do**

$w(s, v) = 0$

end

4. if Bellman-Ford $(G', w, s) = FALSE$ **then**

Print "negative weight cycles are forbidden"

end

5. for each switch $v \in V[G]$ **do**

Set $h(v)$ to the value of $\delta(s, v)$ computed by the Bellman-Ford algorithm

end

6. for each link $(u, v) \in E[G']$ **do**

$\widehat{w}(u, v) \leftarrow w(u, v) + h(v) - h(u)$

7. for each switch $u \in V[G]$ **do**

Run Dijkstra's $(G, \widehat{w}, u)$ to compute $\hat{\delta}(s, v)$ for all $v \in V[G]$

8. for each switch $v \in V[G]$ **do**

$d(u, v) \leftarrow \hat{\delta}(s, v) + h(v) - h(u)$

end

end

end

9. Output: $dist$, shortest path matrix

Haversine Distance: Given that the locations of switches are given in GPS coordinates (longitude and latitude), we carry out our calculations on the basis of a spherical earth (ignoring ellipsoidal effects) – which is accurate enough for most purposes. We use the Haversine formula to determine the great-circle distance between switches. The great-circle distance is the shortest distance between two locations on the surface of a sphere, measured along the surface of the sphere (as opposed to the ordinary Euclidean distance). An alternative method to compute geographic distances is the Law of Cosines. This method is optimal for shorter distances and does not work well with longer distances. To compute the great-circle distance, equation 3 which defines the Harvesine approach is used, where φ_1 and φ_2 denote the latitudes of point 1 and 2 respectively, λ_1 and λ_2 denote the longitudes of point 1 and 2 respectively and r denotes the radius of the earth, a constant with is equal to 6371 km.

$$2r\arcsin\left(\sqrt{\sin^2\left(\frac{\varphi_2-\varphi_1}{2}\right)+\cos(\varphi_1)\cos(\varphi_2)\sin^2\left(\frac{\lambda_2-\lambda_1}{2}\right)}\right) \quad (3)$$

C. Topologies

To maintain realism, our optimization solution was applied to real-world national backbones namely SANREN (a South African national backbone) and ZAMREN (a Zambian backbone). The dataset for these backbones were downloaded from Topology Zoo [16], which is a database of network topologies published by network operators. SANREN is a WAN of 7 nodes distributed across 7 cities in South Africa. ZAMREN on the other hand encompasses 14 nodes. The key factor in our mathematical model is the distance while the bandwidth is constant across all sites. Therefore under constant bandwidth, propagation latency is directly proportional to distance.

D. Model Description: Flow chart

As depicted in Figure 1, we use the Geography Markup Language (GML) to generate the network topology. To determine the best controller placements, we first calculated the distance matrix by applying the Haversine formula. The next step involved generating edge weights by implementing the adjacency matrix between all node pairs. Then, Johnson's algorithm was used to generate the shortest path matrix for the network graph. Lastly, the PAM algorithm was implemented to determine the best placements that minimize average latency and worst-case latencies.

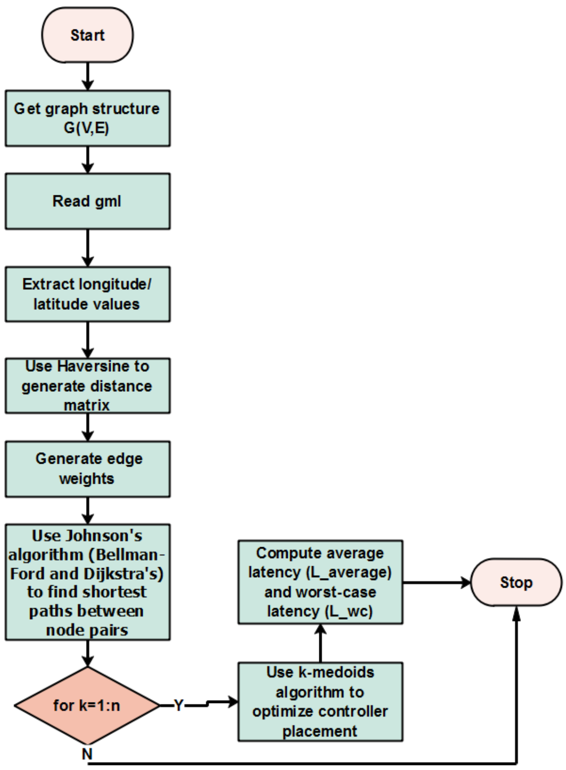


Figure 4: Flowchart: Model description

V. SIMULATION RESULTS

We use MATLAB to implement our model formulations. These models were developed to provide a means for assisting Internet service providers (ISPs) who wish to move to SDN, so that they can optimize network performance by computing the optimum controller locations during SDN planning. In particular the emerging market use case is considered by featuring the SANREN and ZAMREN topologies. Our assessment of network performance was based on propagation latency. Figure 2 and 3 show the optimal controller placement results for the SANREN and ZAMREN topologies respectively (when the number of controllers is set to two). For the SANREN case study, our model output recommends Johannesburg and East-London as the best locations for controller placement, that is the locations that yield lowest average latency ($L_{avg} = 1.6ms$). When we apply our model to the ZAMREN topology, the best locations are found to be Chainama and Kabwe ($L_{avg} = 1.5ms$). The worst controller locations for the SANREN and ZAMREN topologies are Cape Town ($L_{wc} = 4ms$) and Evelyn Hone College ($L_{wc} = 2.8ms$).

Figure 4, shows that increasing the number of SDN controllers has a significant impact on the average latency. When the number of controllers is varied from one to two, a reduction of up to 43% (from 2.8 ms to 1.6 ms) in the average latency is achieved (for the SANREN network topology). Similarly, for the ZANREN topology a reduction of up to 29% (from 2.1ms to 1.5 ms) is achieved.

Network operators and ISPs are more concerned about the CapEx associated with deploying networks. Therefore it makes sense to consider the cost of installing new controllers when determining the optimal number of controllers to use for a given topology. Additionally to minimize the trade-off between performance and cost, average latency is also considered. Thus we define a figure of merit for the cost benefit by taking the ratio of the controller cost (Ck) to average latency $(Lavg)$ as shown in equation (4).

$$\textit{The cost benefit} = k * Ck / Lavg \qquad (4)$$

At the writing of this report, the cost of an SDN controller at NEC Corporation was sitting at R1 500 000. Using the latency results from our simulation, the optimal number of controllers that minimizes the cost benefits was found to be two controllers for both SANREN and ZAMREN topologies.

Figure 2: Optimal Controller placement on SANREN

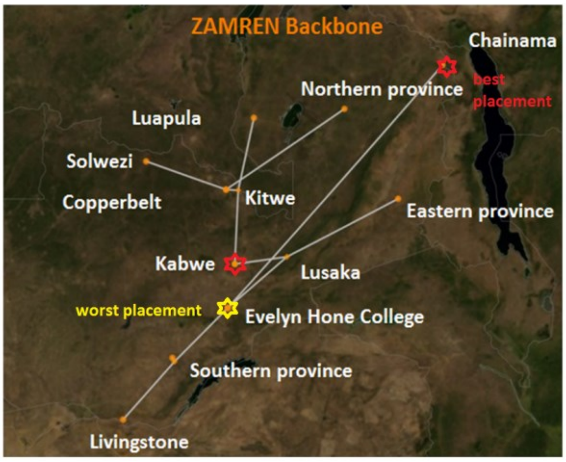
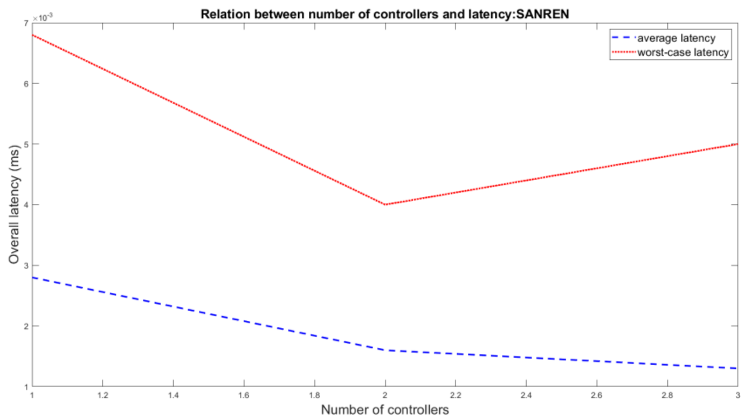
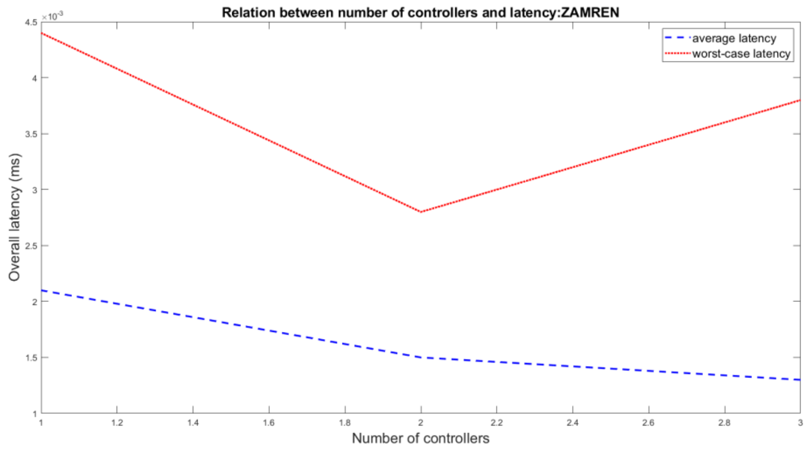


Figure 3: Optimal Controller placement on ZAMREN



(a)



(b)

Figure 7: Average and worst-case latency under varying number of controllers on (a) SANREN, (b) ZAMREN

VI. Conclusion

In this paper, we identified the controller placement problem in SDN in the context of emerging markets. We presented a mathematical formulation for solving the controller placement problem with particular focus on propagation latency (average latency and worst-case latency). The algorithms used to implement our mathematical model are Johnson's algorithm and Partition Around Medoids (PAM). Our simulation results show that running a single controller causes high average propagation delay as some switches are located further away from the controller. This is witnessed by the significant decrease in latency when the number of controllers is varied from one to three. However, the answer to the question of how many controllers to deploy is mostly dependent on the unique needs and constraints of each service provider. For this study, we show that using two controllers is the most efficient way to achieve the best QoS outcomes at reasonable CapEx. This work was done with the goal of assisting internet service providers (ISPs) in emerging markets to optimally transition to SDN.

In future we intend to test our optimization algorithm on larger scale topologies. We also intend to explore more on the different methods for determining the optimal number of controllers to use for topologies with varying requirements and constraints.

References

[1] S. A. Government, 2017. [Online]. Available: https://www.gov.za/issues/national-development-plan-2030..

[2] "Mybroadband," 2015. [Online]. Available: https://mybroadband.co.za/news/government/93243- south-africa-connect-the-new-broadband-policy.html..

[3] E. Hernandez-Valencia, S. Izzo and B. Polonsky, "How will NFV/SDN transform service provider opex?," *IEEE Network,* vol. 29, no. 3, pp. 60-67, 2015.

[4] L. Mamushiane and S. Dlamini, "Leveraging SDN/NFV as key stepping stones to the 5G era in Emerging Markets," in *Global Wireless Summit (GWS)*, 2017.

[5] D. Kreutz, F. M. Ramos, P. E. Verissimo, R. C. E, A. S and S. Uhlig, "Software-defined networking: A comprehensive survey," *Proceedings of the IEEE,* vol. 103, no. 1, pp. 14-76, 2015.

[6] H. Xie, T. Tsou, D. Lopez, H. Yin and V. Gurbani, "Use cases for alto with software defined networks," IETF Internet-Draft, 2012.

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

[8] GlobalNoc, " Internet 2," [Online]. Available: https://docs.globalnoc.iu.edu/sdn/oess/using-oess.html.

[9] 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.

[10] Y. Hu, W. Wendong, X. Gong, X. Que and C. Shiduan, "Reliability-aware controller placement for software-defined networks," in *Integrated Network Management*, 2014.

[11] U. O. Washington, 2017. [Online]. Available: research.cs.washington.edu/networking/rocketfuel.

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

[13] 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.

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

[15] N. Damak, "Distance problems in networks—theory and practice. Classical shortest-path algorithms," 2010.

[16] U. O. Adelaide, " The Topology Zoo," [Online]. Available: http://www.topology-zoo.org/dataset.html. [Accessed May 2018].

[17] H. Aoki and N. Shinomiya, "Controller placement problem to enhance performance in multi-domain SDN networks," in *ICN*, 2016.

Lusani Mamushiane received the B.Eng. degree in Electrical and Electronic Engineering from the University of Johannesburg, South Africa in 2014. In 2015, she joined Microsoft where she worked as an Electronic and Telecoms Engineer for Intelligent Traffic Systems. In 2016 she then joined Advanced and Network Architecture Systems research group in Meraka,CSIR as a Researcher. She is currently pursuing her MSc at the University of Cape Town under the supervision of Dr Joyce Mwangama. Her research interests include but not limited to optical networks, programmable networks using software defined network (SDN) and Network Function Virtualization (NFV), intelligent traffic systems.