I. J. Computer Network and Information Security, **2019, 2, 36-42**
Published Online February 2019 in MECS (http://www.mecs-press.org/)
DOI: 10.5815/ijcnis.2019.02.05

PSO and TLBO based Reliable Placement of Controllers in SDN

Ashutosh Kumar Singh
CSED, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India 211004
E-mail: ashuit89@gmail.com

Naveen Kumar
CSED, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India 211004
E-mail**:** nk10121989@gmail.com

Shashank Srivastava
CSED, Motilal Nehru National Institute of Technology Allahabad, Prayagraj, India 211004
E-mail: shashank12@mnnit.ac.in

Received: 20 November 2018; Accepted: 18 January 2019; Published: 08 Febuary 2019

Abstract**—**SDN (software defined networks) is a programmable network architecture that divides the forwarding plane and control plane. It can centrally manage the network through a software program, i.e., controller. Multiple controllers are required to manage the current software defined WAN. Placing multiple controllers in a network is known as controller placement problem (CPP). Only one controller is not capable to handle the scalability and reliability issues. To tackle these issues, multiple controllers are required. Efficient deployment of controllers in SDN is used to improve the performance and reliability of the network. To the best of our knowledge, this is the first attempt to minimize the total average latency of reliable SDN along with the implementation of TLBO and PSO algorithms to solve CPP. Our experimental results show that TLBO outperforms PSO for publicly available topologies.

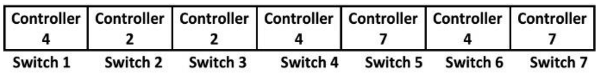
Index Terms**—**Controller placement, SDN, CPP, latency, reliability.

I. Introduction

Software defined networking (SDN) is a new edge network paradigm which aims to provide separation between the control plane and data plane [1]. The data plane is responsible for forwarding network traffic which is controlled by the control plane. Both the data plane and control plane can be managed through software controls i.e. *controller*. Whenever a switch gets a new flow, it sends a "Packet In" information to the respective controller for setting up flow rules along with the best flow path. The controller is responsible for managing the routing of flows by interacting with the switches through a secure channel (SSL and TLS). It guides to the switches that how packets should be forwarded by installing new flow rules and policies. The efficiency of the controller is a significant factor during scalability and reliability of SDN are considered.

In a large-scale network, only one controller is not sufficient to manage a many number of switches that are geographically distributed as it cannot ensure acceptable latencies between switches and controllers. Controller installed on a particular server (or switch) has a limited resource capacity for handling a large number of *"Packet In"* generated by switches. As a result, SDN based WAN uses multiple controllers to improve the network performance. Placing multiple controllers is a good option if the decision of controller deployment is based on placement metrics like the average switch-controller latency, maximum switch-controller latency, the controller-controller latency, etc. In 2012, Heller et al. in [2] introduced controller placement problem (CPP).

CPP is an NP-hard problem. The papers [3,4] discuss CPP and provide heuristic based optimization solutions for it. Gao et al. [4] introduced a PSO-based algorithm to solve this problem. They did not address the reliable placement of controllers. This is the first attempt to map the reliable CPP with the PSO (particle swarm optimization) and TLBO (teaching learning based optimization). In this letter, we discuss the PSO and TLBO based reliable placement of controllers. Eberhart and Kennedy [5] introduced PSO algorithm in 1995. It is a nature inspired population based optimization algorithm. Earlier, PSO algorithm is used to minimize the total average latency of the SDN network. Rao et al. [6] introduced TLBO algorithm in 2011. It works on the influence of a teacher on learners.



Controller 4	Controller 2	Controller 2	Controller 4	Controller 7	Controller 4	Controller 7
Switch 1	Switch 2	Switch 3	Switch 4	Switch 5	Switch 6	Switch 7

Fig.1. Mapping of Particle

In our mapping of the particle for both PSO and TLBO algorithms, a particle refers to the particular placement of controllers and switches attached to respective controllers.

Here, the particle is taken as a *d*-dimensional vector, where *d* is the total number of switches in the network i.e. *n*. And value in each dimension indicates controller number. For example, let a network consists of seven switches and three controllers, these controllers are placed at switch number 2, 4 and 7. Where, controller at switch 2 controls switches 2, 3, and controller at switch 4 controls switches 1, 4 and 6. And controller at switch 7 controls switches 5 and 7. For this case, the mapping is shown in Fig.1.

The main contributions of this letter are summarized as:

- We define the reliable controller placement problem (RCPP) and use PSO and TLBO algorithm to solve this problem.
- To the best of our knowledge, we consider total average latency of the reliable network for the first time.
- We compare the solution of PSO with the solution of TLBO where the performance of TLBO algorithm is found better.

The rest of the paper is structured as follows. In this paper, we discuss the related work in section II. In section III, we present the proposed model. Section IV shows the experimental results and discussions. Finally, the paper is concluded in section V.

II. Related Works

Heller et al. [2] discussed CPP as facility location problem which is known to be an NP-Hard class of problem. To solve the CPP, authors provide their solutions through optimization algorithms, clustering algorithms, Integer programming, and so on. The other school of thought is to take CPP as a clustering problem in which a large-scale network is divided into various small-scale network domains. Each domain managed by the single controller. Bari et al. in [7] introduced dynamic CPP named as controller provisioning problem (DCPP) in 2013. Multiple controllers are deployed that work simultaneously to manage and control the SDN infrastructure. Authors of [7] use integer linear programming (ILP) to formulate optimal DCPP and minimize the flow setup time and communication overhead. In [8], the authors proposed an algorithm named LiDy to find the minimum controllers and their positions as well as minimize the switch-controller latency and maximize the controller utilization. Authors extended their work as *LiDy+* in [9] that provides a better result than *LiDy*. In general, controllers are placed only to locations of switches. However, authors of [8,9] provide their solution in which the deployment locations of controller are not limited to switch locations.

Table 1 shows key points that are considered by different authors in the literature. Most of the works in literature have not considered the reliable controller placement. Gaining the understanding of CPP, we came to a conclusion that reliability is main issue in the CPP. So, reliability should be taken into attention while placing the controllers in the SDN. In this paper, we consider placement of controllers on the basis of ensuring reliability for a given controller. We solve the reliable controller placement problem (RCPP) by PSO and TLBO methods.

Table 1. Important Points Taken from Literature for Reliable CPP (S-C = switch-controller and C-C = controller-controller)

		Latencies			Solutions based on		
Ref.	WAN Network	S-C	C-C	Total	Clustering	Optimization	Reliability
Bari *et al.*[7]	✓	✗	✗	✗	✓	✗	✗
Hu *et al.* [10]	✓	✗	✗	✗	✗	✓	✗
Perrot and Reynaud [11]	✓	✓	✗	✗	✗	✓	✓
Hock *et al.*[12]	✗	✓	✓	✗	✗	✓	✗
Cheng *et al.*[13]	✓	✗	✗	✗	✓	✗	✗
Cheng *et al.*[14]	✓	✓	✗	✗	✓	✗	✗
Yao *et al.*[15]	✓	✗	✗	✗	✓	✗	✗
Liu *et al.*[16]	✓	✗	✗	✗	✓	✗	✓
Lange *et al.*[17]	✓	✓	✓	✗	✗	✓	✓
Gao *et al.*[4]	✓	✓	✓	✓	✗	✓	✗
Sallahi and Hilaire [18]	✗	✗	✗	✗	✗	✓	✗

III. Model for RCPP

The network topology can be represented as an undirected graph G(S, E), where S and E represent set of switches and set of links respectively. Let P = 2^S - 1 denotes set of all possible placement of controllers (-1 operation probabily refers to excluding the empty set). Let *d(s, c)* be the shortest path between switch sϵS and controller cϵP, and d(c_i, c_j) be the shortest path between controllers c_i,c_j ϵP. X and D represent the particle and distance matrix (adjacency matrix of switches) . Average switch-controller latency ($\pi^{\text{avgS-Clatency}}$(P)), maximum switch-controller latency ($\pi^{\text{maxS-Clatency}}$(P)), average controller-controller latency ($\pi^{\text{avgC-Clatency}}$(P)) and maximum controller-controller ($\pi^{\text{maxC-Clatency}}$(P)) are calculated by Equation (1), (2), (3), and (4) respectively.

$$\pi^{avgS-Clatency}(P) = \frac{1}{|S|}\sum_{s\epsilon S} \min_{c\epsilon P} d(s,c) \qquad (1)$$

$$\pi^{maxS-Clatency}(P) = \max_{s\epsilon S} \min_{c\epsilon P} d(s,c) \qquad (2)$$

$$\pi^{avgC-Clatency}(P) = \frac{1}{p_c}\sum_{c_i,c_j \, \epsilon P} d(c_i,c_j) \tag{3}$$

$$\pi^{maxC-Clatency}(P) = \max_{c_i,c_j \, \epsilon P} d(c_i,c_j) \tag{4}$$

With the above definitions, we can calculate the total average latency $T^{avg\text{-}latency}(P)$ and total maximum latency $T^{max\text{-}latency}(P)$ of the SDN by Equation (5) and (6) respectively.

$$T^{avglatency}(P) = \frac{1}{|S|}\sum_{s\epsilon S}\min_{c\epsilon P} d(s,c) + \frac{1}{p_c}\sum_{c_i,c_j \, \epsilon P} d(c_i,c_j) \tag{5}$$

$$T^{maxlatency}(P) = \max_{s\epsilon S}\min_{c\epsilon P} d(s,c) + \max_{c_i,c_j \, \epsilon P} d(c_i,c_j) \tag{6}$$

In this paper, our goal is to minimize the total average latency of the reliable SDN ($T^{avg\text{-}latency}(P)$). In today's real world, traffic is increasing day by day, so only one controller cannot control huge traffic. In this type of scenario, the load of the switches and the capacity of controllers as a parameters are required.

Algorithm 1: Calculating value of objective function

```
Input: X, k(number of controllers), D, l(s), L(c)
Output: objVal
controllerSet = get k controllers from X that have
            maximum probabilities
Calculate power set of controllerSet excluding null set
for i = 1 to $2^k$ -1 do
    controllerSub = get $i^{th}$ set from power set
    placement, penalty = compute placement of
      controllers and penalty as per Algorithm 2 using
      controllerSub, D, l(s), L(c)
    latency = compute latency as per Algorithm 3
      using placement and D
    $objValSub_i$ = latency*(1 + penalty)
end
objVal = mean(objValSub)
return objVal
```

Algorithm 2: Algorithm for placement computation

```
Input: controllerSub, D, l(s), L(c)
Output: placement, penalty
k = number of controllers in controllerSub
n = number of switches in D
T = Get list of controllers with increasing order of
      distance from every switch using D
CapAvail = L(c) for each controller in controllerSub
for i = 1 to k do
    for j = 1 to k do
        $T_{i,j} = j^{th}$ nearest controller from $i^{th}$ switch
        temp = CapAvai for controller $T_{i,j} - l(s_i)$
        if temp ≥ 0 then
            $placement_i = T_{i,j}$
            CapAvai for controller $T_{i,i}$ = temp
            break
        end
    end
end
penalty = square of sum of loads of un-allocated
  switches
return placement, penalty
```

Algorithm 3: Particle's latency computation

```
Input: placement, D
Output: $T^{avg-latency}$
c = get unique controllers from placement
k = number of controllers in c
n = number of switches in D
S-Clatency = 0
for i = 1 to n do
    s-c = get controller assigned to switch i from
          placement
    dist = D(i, s-c)
    S-Clatency = S-Clatency + dist
end
$\pi^{avgS-C}$ = S-Clatency / n
C-Clatency = 0
for i = 1 to k do
    for j = i+1 to k do
        dist = D($c_i$, $c_j$)
        C-Clatency = C-Clatency + dist
    end
end
$p_c = \binom{k}{2}$
$\pi^{avgC-C}$ = C-Clatency / n
$T^{avg-latency} = \pi^{avgS-C} + \pi^{avgC-C}$
return $T^{avg-latency}$
```

When we are considering these parameters then TLBO algorithm gives better results as compared PSO algorithms. Minimum total average latency based CPP is defined by Equation (7). Equation (8) guarantees that the total loads on switches do not exceed the capacity of respective controller.

$$T^{avg-latency}(P) \tag{7}$$

Suject to:

$$\sum_{s\epsilon S(c)} l(s) \leq L(c), \forall c\epsilon P \tag{8}$$

Let C = 2^P - 1 denote all alternative placements (-1 operation probably refers to excluding the empty set) that result from the failure of up to k-1 controllers. $T_C^{avg-latency}(P)$ denotes the total average latency (average switch-controller and average controller-controller latency) for failure free cases and failure cases up to k-1 controllers. L(c) and l(s) represent capacity of controllers and load of switches respectively and S(c) denotes set of switched under the control of controller c. Equation (9) shows the formal definition of average latency.

$$T_C^{avg-latency}(P) = \frac{1}{|C|}\sum_{P\epsilon C}\left(T^{avg-latency}(P)\right) \tag{9}$$

In this paper, our goal is to minimize the total average latency of the network $\left(T_C^{avg-latency}(P)\right)$for all possible failure scenarios in all possible placement scenario. Equation (10) shows the minimization of total latency of the network and Equation (11) guarantees that the total loads on switches do not exceed the capacity of respective controller.

$$T_C^{avg-latency}(P) \tag{10}$$

Suject to:

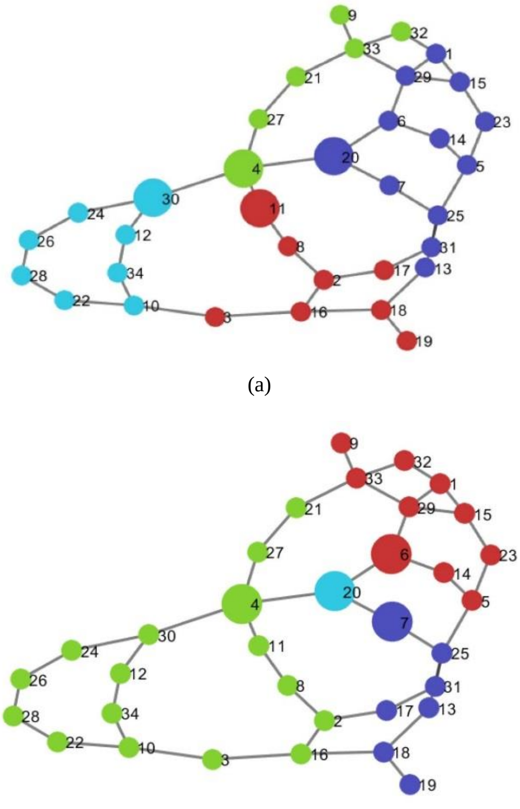
$$\sum_{s \epsilon S(c)} l(s) \leq L(c), \forall c \epsilon P \tag{11}$$

When controllers (up to k-1) fail due to some reasons then additional calculations is required for the finding global latency of the reliable controller placement problem. Equation (10) calculates the latency between switches and controllers as well as latency between controllers and controllers.

Algorithms 1 and 2 calculate the value of the objective function (see Equation (10) and (10)) and placement of controllers respectively. Algorithm 3 calculates the latency of the particles. On the using these values as input for heuristic approaches (PSO and TLBO) to compute the total average latency of the reliable controller placement problem.

IV. Results and Discussions

We have used MATLAB 2018b for the simulation of the optimization based solution of RCPP. The system consists of Windows 8.1 (64-bit) with Intel Core i7-4770 CPU @ 3.40 GHz and 16 GB RAM. For all experiments of optimization methods, we have set population size equal to 100.

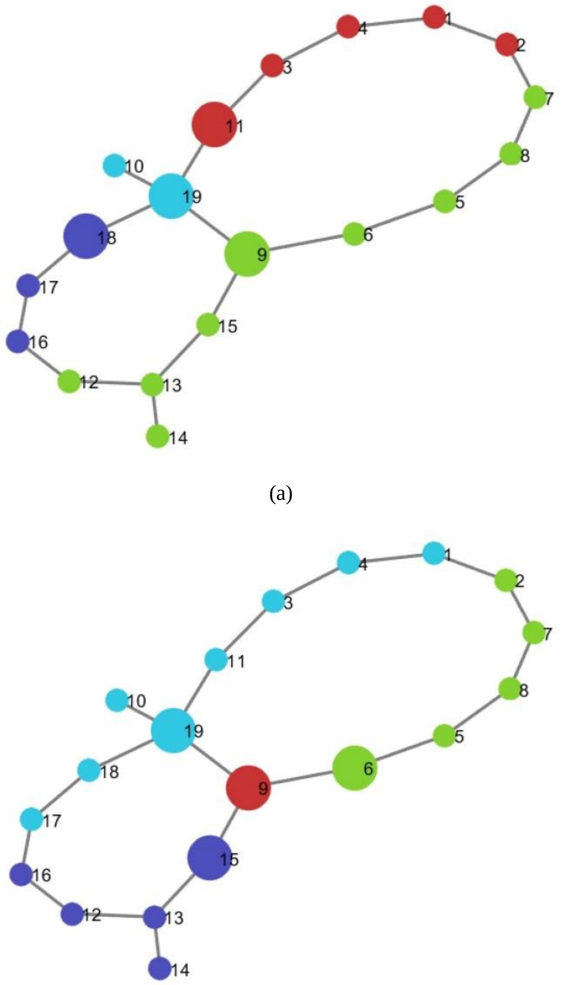


(a)

(b)

Fig.2. Internet2 OS3E Topology view for Reliable Controller Placement Problem (controllers = 4) by (a) PSO and (b) TLBO Algorithms

Our experimental results tell that TLBO based solutions are better than PSO based solutions. We use publicly available network topologies in our experiments and use PSO and TLBO algorithms for finding optimal placement in a given topology. Experimental results show that TLBO outperforms PSO. Here, we give the results for two most popular topologies (Internet2 OS3E with 34 nodes and Savvis with 19 nodes). Here, numbers in a circle represents the switch number and controller number. Big circle denotes controllers and switches is represented by both big and small circles. Same colors nodes represent the controller and their controlled switches. Switches assigned to the controllers and this assignment is based on the shortest distance to the closest controller.

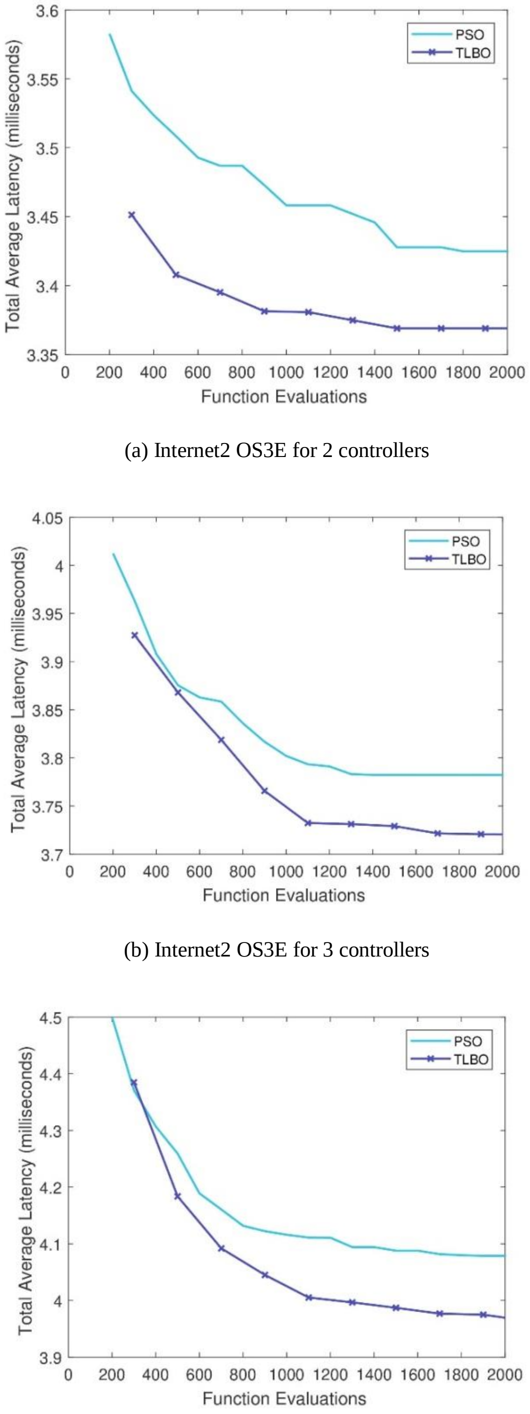


(a)

(b)

Fig.3. Savvis Topology view for Reliable Controller Placement Problem (controllers = 4) by (a) PSO and (b) TLBO Algorithms

Fig.2 shows the Internet2 OS3E topology view after the reliable placement of controllers in SDN. From Fig. 2(a), it is observed that Internet2 OS3E topology uses 4 controllers i.e., 4, 11, 20, and 30 for PSO based solution.



(a) Internet2 OS3E for 2 controllers

(b) Internet2 OS3E for 3 controllers

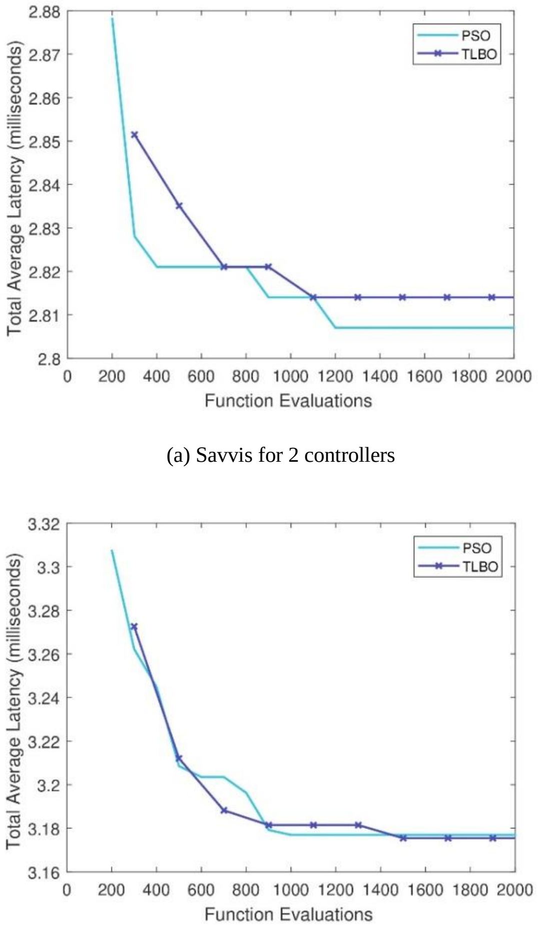
(c) Internet2 OS3E for 4 controllers

Fig.4. Convergence Plots for Reliable Controller Placement Problem (controllers = 2, 3, 4) by PSO and TLBO Algorithms

Controller number 4 controls 6 switches i.e., 4, 9, 21, 27, 32, and 33 (for example, all nodes are represented by same color for the simplicity). Controller number 11 controls 8 switches i.e., 2, 3, 8, 11, 16, 17, 18, and 19. Controller number 20 controls 12 switches i.e., 1, 4, 6, 7, 13, 14, 15, 20, 23, 25, 29, and 31. Controller number 30 controls 8 switches i.e., 10, 12, 22, 24, 26, 28, 30, and 34. Similarly, Fig. 2(b) i.e. Internet2 OS3E uses 4 controllers i.e., 4, 6, 7, and 20 for TLBO based solution. Controller number 4 controls 16 switches, controller number 6 controls 10 switches, controller number 7 controls 7 switches and controller number 20 controls only one switch i.e., switch number 20.

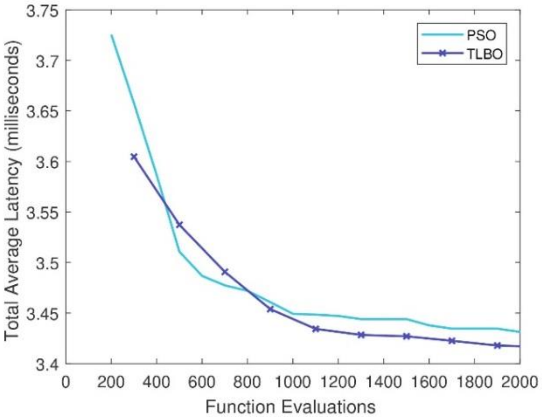
Fig. 3 shows the Savvis topology view after the reliable placement of controllers in SDN. From Fig. 3(a), it is observed that Savvis topology uses 4 controllers i.e., 9, 11, 18, and 19 for PSO based solution. Controller number 9 controls 10 switches i.e., 5, 6, 7, 8, 9, 12, 13, 14, and 15. Controller number 11 controls 5 switches i.e., 1, 2, 3, 4, and 11. Controller number 18 controls 3 switches i.e., 16, 17, and 18. Controller number 19 controls 2 switches i.e., 10 and 19. Similarly, Fig. 3(b) i.e. Savvis uses 4 controllers i.e., 6, 9, 15, and 19 for TLBO based solution. Controller number 6 controls 5 switches, controller number 9 controls only one switch i.e., switch number 9, controller number 15 controls 5 switches and controller number 19 controls 8 switches.

We test PSO algorithm as well as TLBO algorithms for reliable controller placement problem with 30 independent runs. We find out the results and convergence rate for these two optimization methods. As shown in Fig. 4 and Fig. 5, it is clear that TLBO gives better convergence rate as compared to PSO algorithms. In Fig. 5(b), if number of controllers is 2 then start-up of PSO gives better convergence rate as compared to TLBO, but after some function evaluations and it goes uniform.



(a) Savvis for 2 controllers

(b) Savvis for 3 controllers



(c) Savvis for 4 controllers

Fig.5. Convergence plots for Reliable Controller Placement Problem (controllers = 2, 3, 4) by PSO and TLBO Algorithms

V. Conclusions

We Placement of controllers is an essential aspect in the large-scale network. Efficient deployment of controllers tries to minimize the total average latency of reliable SDN for maximizing the performance of SDN. To the best of our knowledge, we consider the total average latency of the reliable network for the first time. In this letter, we discuss and analyze the reliable controller placement problem. Our experimental results show that TLBO based solution gives better results as compared to PSO based solution for CPP.

In the future, we will try to solve this problem with other optimization algorithms and will propose a new optimization algorithm for it.

References

[1] AK Singh, S. Srivastava. "A survey and classification of controller placement problem in SDN". *Int J Network Mgmt*. Available: https://doi.org/10.1002/nem.2018, e2018, 2018.

[2] B. Heller, R. Sherwood, N. McKeown. "The controller placement problem". *In: ACM*. ; pp. 7–12, 2012.

[3] A. Jalili, V. Ahmadi, M. Keshtgari, M. Kazemi. "Controller placement in software-defined WAN using multi objective genetic algorithm". *In: IEEE*. Pp. 656–662, 2015.

[4] C. Gao, H. Wang, F. Zhu, L. Zhai, S. Yi. "A Particle Swarm Optimization Algorithm for Controller Placement Problem in Software Defined Network". *In: Springer, pp.* 44–54, 2015.

[5] R. Eberhart, J. Kennedy."A new optimizer using particle swarm theory". *In: IEEE*, pp. 39–43, 1995.

[6] RV Rao, VJ Savsani, D. Vakharia. "Teaching–learning-based optimization: a novel method for constrained mechanical design optimization problems". *Computer-Aided Design*, 43(3), pp. 303–315, 2011.

[7] MF Bari, AR Roy, SR Chowdhury, et al. "Dynamic controller provisioning in software defined networks". *In: IEEE*, pp. 18–25, 2013.

[8] Ul Huque MTI, G. Jourjon, V. Gramoli. "Revisiting the controller placement problem". *In: IEEE*, pp. 450–453, 2015.

[9] Ul Huque MTI, W. Si, G. Jourjon, V. Gramoli. "Large-Scale Dynamic Controller Placement". *IEEE Transactions on Network and Service Management*, 14(1): pp. 63-76. doi: 10.1109/TNSM.2017.2651107, 2017

[10] Y. Hu, T. Luo, NC Beaulieu, C. Deng "The Energy-Aware Controller Placement Problem in Software Defined Networks". *IEEE Communications Letters*, 21(4): pp. 741–744, 2017

[11] N. Perrot, T. Reynaud. "Optimal placement of controllers in a resilient SDN architecture". *In: IEEE*, pp. 145–151, 2016.

[12] D. Hock, M. Hartmann, S. Gebert, T. Zinner, P. Tran-Gia. "POCO-PLC: Enabling dynamic pareto-optimal resilient controller placement in SDN networks". *In: IEEE*, pp. 115–116, 2014.

[13] TY Cheng, M. Wang, X. Jia. "QoS-guaranteed controller placement in SDN". *In: IEEE*, pp. 1–6, 2015.

[14] G. Cheng, H. Chen, H. Hu, J. Lan. "Dynamic switch migration towards a scalable SDN control plane". *International Journal of Communication Systems*, 29(9): pp. 1482–1499, 2016.

[15] L. Yao, P. Hong, W. Zhang, J. Li, D. Ni. "Controller placement and flow based dynamic management problem towards SDN". *In: IEEE*, pp. 363–368, 2015.

[16] J. Liu, R. Xie. "Reliability-based controller placement algorithm in software defined networking". *Computer Science and Information Systems*, pp. 547-560, 2016.

[17] S. Lange, S. Gebert, T. Zinner, et al. "Heuristic approaches to the controller placement problem in large scale SDN networks". *IEEE Transactions on Network and Service Management*, 12(1), pp. 4–17, 2015.

[18] A. Sallahi, M. St-Hilaire. "Expansion Model for the Controller Placement Problem in Software Defined Networks". *IEEE Communications Letters*, 21(2), pp. 274-277. doi: 10.1109/LCOMM.2016.2621746, 2017

Authors' Profiles

Ashutosh Kumar Singh obtained his B.Tech degree in Information Technology from Uttar Pradesh Technical University Lucknow, India in 2011 and M.Tech degree in Computer Science and Engineering from Indian Institute of Information Technology and Management Gwalior, India in 2014. Now he is a Ph.D. student in the Department of Computer Science and Engineering, Motilal Nehru National Institute of Technology Allahabad, Prayagraj 211004, India. He is having a membership of IEEE and ACM. His research interest includes network optimization and Software Defined Networking.

Naveen Kumar obtained his B.Tech degree in CSE from Uttar Pradesh Technical University Lucknow, India in 2012 and M.Tech degree in Computer Science and Engineering from Motilal Nehru National Institute of Technology (MNNIT) Allahabad, Prayagraj, India in 2014. He is currently a Ph.D. student in the Department of Computer Science and Engineering, MNNIT Allahabad, Prayagraj 211004, India. He is having a membership of IEEE and ACM. His research interest includes peer to peer systems, future internet technologies and Named Data Networking.

Shashank Srivastava obtained his Ph.D. degree at Indian Institute of Information Technology Allahabad, India in 2014. He is currently working as Assistant Professor in the Department of CSE, Motilal Nehru National Institute of Technology Allahabad, 211004, India. He possesses an experience of more than six years in the field of teaching and research. He has published various research papers in the area of Network and security. At present he is guiding five Ph.D. students in the field of Software Defined Networking (SDN), Named Data Networking (NDN), Network flow optimization and security. He is having the Membership of IEEE, ACM, CSI and CRSI (Cryptographic Research Society of India). His areas of expertise are SDN, NDN, information security, and future internet technologies.

How to cite this paper: Ashutosh Kumar Singh, Naveen Kumar, Shashank Srivastava,"PSO and TLBO based Reliable Placement of Controllers in SDN", International Journal of Computer Network and Information Security(IJCNIS), Vol.11, No.2, pp.36-42, 2019.DOI: 10.5815/ijcnis.2019.02.05