International Journal on Emerging Technologies 11(3): 82-90(2020)

ISSN No. (Print): 0975-8364
ISSN No. (Online): 2249-3255

Latency Optimization of Structured Multi-Controllers in Software defined Networks

Mythrayee Ramasamy[1] and Sanjay Pawar[2]
[1]Research Scholar, Department of Electronics and Communication, UMIT, SNDT, Mumbai (Maharashtra), India.
[2]Professor, Department of Electronics and Communication, UMIT, SNDT, Mumbai (Maharashtra), India.

(Corresponding author: Mythrayee Ramasamy)
(Received 03 February 2020, Revised 27 March 2020, Accepted 30 March 2020)
(Published by Research Trend, Website: www.researchtrend.net)

ABSTRACT: In Software-Defined networking (SDN), the large-scale network is enhanced with the help of multi-controller structure. Moreover, latency optimization is considered as one of the significant challenge in SDN. This paper concentrates the latency optimization in SDN network using a latest optimization algorithm. Initially, multi-controller architecture of SDN is designed into two different structures, named as flat structure and hierarchy structure. The multi-controller architecture calculates the latencies like Packet Transmission Latency, Queuing Latency, Propagation Latency, and Controller Processing Latency for both the structures. From these values we attain the Average Latency. Finally, the Sea Lion Optimization (SLO) is proposed to optimize the calculated latencies. The metrics such as Packet Loss Rate, Average Latency, and Packet Delay Rate are used to show the efficiency of the suggested optimization algorithm. Optimization method is compared with conventional methodologies in order to prove effective performance of SLO based on latency optimization in SDN. The optimized values of Average Latency, Packet Delay Rate and Packet Loss Rate obtained from the latest algorithm are compared with the existing algorithms. The simulation results show that the metrics are been optimized using SLO algorithm and the graphs show the values of the SLO algorithm with that of the existing techniques.

Keywords: Software Defined Network (SDN), Hierarchical structure, multi-controller, Latency, and Sea Lion Optimization (SLO) algorithm.

I. INTRODUCTION

In various network developments, SDN model based on OpenFlow is one of the most promising models in networks. When the system is presented with huge number of incoming data packets, there exists a bottleneck with the use of single controller in SDN. So, the SDN faces some challenges like scalability, failure of data transmission etc. An integrated interface is created between the devices, switches and routers by the controllers of SDN. This controller enhances the network management, enlarges the programmability and acts as the network operating system of SDN architecture. The controllers of SDNs are dynamic in nature and they efficiently adjust the data traffic forwarding paths to distribute the data packets in an even manner. The controllers require more investigations, if they not balance the traffic loads between several network paths. Hence, the plane design is simply controlled by the centralized controller based architecture in SDN.

In the architecture of SDN, there exists distributed controller architecture and Centralized architecture. Distributed architecture contains multi-controllers and centralized architecture consists of single controller. Main reason for developing multi-controller architecture is to replace the single controller when it fails to transmit data packets. Adaptation of multiple controllers in control plane is better solution for this failure of single controller. Single controller usage in centralized network does not achieve three critical requirements which are main tendency for proposed SDN architectures. Efficiency is the initial requirement which is not established with centralized controller. The idea of multi-controller is developed from the scalability which is the second most requirements in centralized network.

Finally, third one is high availability which has two important parameters such as security and redundancy [1]. Plenty of SDN architecture considers redundancy as an important parameter. But there is no required mechanisms are formed to enforce access control for certain applications. The process flow rules are efficiently generated to control the applications which are infected by some malicious codes [2-3]. To effectively manage networks and provide fast paths, operators must continually monitor the latency on all paths that the traffic of an application could traverse and quickly route packets away from high-delay segments. Latency is most significant factor which is calculated by the layout of multiple controller architecture of SDN [4]. Latency is defined by the time taken to send the data packet from source to receive the data packet in destination. For SDN architecture, this metric is considered as significant factor to manage the packet traffic problems which may be slow down the process of network [5].

The main contribution of our work is given as follows:

– To design a multi-controller structure.

– To measure the latencies like Propagation Latency, Packet Transmission Latency, Switch Queuing Latency and Controller Processing Latency in a SDN with the help of designed Multi-controller.

– To use a recently developed Sea Lion Optimization algorithm for optimizing the calculated latencies.

– To show the performance, the comparative analysis of the proposed method with the existing techniques gives the strength of the method.

Paper organization is given as follows: The recent works related with this topic is elaborated in Section II. The latency measurement with proposed optimization algorithm is described in Section III. Simulation results and performance analysis is explained in Section IV. Finally, Section V presents the conclusion of the paper.

II. LITERATURE REVIEW

Some recent related works are given below:
Ateya *et al.,* (2019) developed a chaotic salp swarm algorithm for SDN multi-controller networks. The flexibility of network was managed by the promising network known as software-defined networking. Also, a single controller had limitations on scalability and performance due to the increase in network capacity. Fault-tolerant and scalability was satisfied with the help of distributed multi-controller deployment strategy. The Salp Swarm optimization algorithm was proposed to optimize the performance of SDN controller. This method validates optimal connections among controllers and switches as well as optimum amount of controllers in multi-controller structure. The optimization algorithm optimizes the performance of proposed algorithm and the local optima were prevented with the help of chaotic maps [6].
For Software Defined Wireless Networks (SDWN), Li *et al.*, (2019) developed a multi-controller resource management. The mobile users get low energy consumption and latency services from this SDWN which was decoupled with infrastructure and control layer. Latency-aware resource management and an energy-efficient multi-controller placement method were proposed for SDWN. Also, the multi-controller placement problem was optimized with Particle Swarm optimization (PSO) algorithm as well as a Deep Reinforcement Learning (DRL) algorithm was established to help the resource allocation strategy. This DRL algorithm solves the only the resource allocation problem [7].
A multi-controller placement strategy was proposed by Fan *et al.,* (2019) to optimize delay and reliability in SDN. The networks ability was affected by the layout of the controllers. So, this problem was prevented by placement of controllers in an optimal manner. This paper considers link failure factors, worst-case delay among switch and controller and number of control paths rerouting in each link failure state to perform multi-objective optimization problem. So, this paper proposed PSO algorithm to achieve optimal controller placement in SDN. In most link failure states, the delay and reliability of the control layer was guaranteed by this proposed method [8].
Secure SDN-enabled Inter-data centre overlay networks was developed by Francois and Gelenbe (2016) over cognitive routing. Plenty of businesses deployed their applications with different cloud providers to show better QoS to the customers. An optimized secure software-defined overlay network interconnects the geographically-dispersed applications. The overhead monitoring was minimized by the logically centralized Cognitive Routing Engine (CRE) performed based on the Random Neural Networks with Reinforcement learning. The CRE was validated against five different public clouds in the overlay network [9].
Keupondjo *et al.*, (2019) establishes hybrid routing with latency optimization in SDN networks. The latency and packet loss strongly affects the overall network management policy of OpenFlow switches over the OpenFlow protocol. Also, latency and packet loss were two important parameters in multi-controller applications. Reactive and proactive approaches were generally proposed in the SDN to optimize the transmission time in the data networks. The reactive approach validates the quality of optimum paths and proactive approach minimizes the time and does not consider the parameters like failure of a node which was the part of transfer path. The routing algorithm optimizes the routing functions by simply placing traffic to avoid the overload of the network [10].
Previous existing methods [8] utilize optimization algorithms to optimally place the controllers in a SDN network. But, they didn't optimize the latency values. So, in our work, the SDN network is structured into two different structures using multi-controllers. The multi-controllers of SDN architecture. Initially, the multi-controller structure of SDN is designed to perform different processes. In this work, we have considered flat and hierarchy structure for latency calculation. The network latencies like Switch Queuing Latency, Packet Transmission Latency, Propagation Latency, and Controller Processing Latency are calculated with the help of multi-controller in SDN. Here, we apply the optimization algorithm to optimize the latency parameters. Finally, the performance of proposed method is compared with conventional approaches.

A. Structure of SDN

In SDN's structure, the data forwarding plane is separated from the network control plane. This is done to enhance network management, network resource utilization, minimize operating cost and finally promote progression and invention. Application, Data-plane, and Control-plane layer are three different kind of layers presented in SDN architecture. Data-plane layer contains routers, OpenFlow switches and other infrastructure elements. The SDN controllers like Pox, Nox, Floodlight, Open Daylight and Beacon controllers are present in control plane layer. Applications like network virtualization, monitoring, routing, traffic engineering, and QoS applications are done using the application layer. Southbound and northbound APIs are utilized by these layers to communicate with each other. In a huge data plane and the control plane with single controller some issues occurs due to more number of data. Hence, multi-controller architecture of SDN is adopted to reduce the traffic loads and to scale up larger networks. Likewise, distance among any closest available controller and switches is minimized by distributing workloads of distributed control planes [11].

B. Multi-controller structure of SDN controller

The network operating system of SDN is considered as the controllers either it is single controller or multiple controller. The nature of SDN model is defined by the controllers. A unified view of network is provided by these controllers and also it is responsible to communicate between switches and controllers. In this work, we have considered multi-controller architecture of SDN. Multi-controller architecture contains multiple numbers of controllers which are responsible to manage the entire SDN-enabled network. There are three structures considered in the multi-controller architecture which is named as flat structure, hierarchical structure and hybrid structure. The network topologies of these structures are divided into number of domains and each and every domain contains switches and routers which are controlled by the controllers. Here we consider only two models flat and hierarchical model.

C. Flat structure of Multi-controller

In this structure, every switch is controlled by controller's own local network view. Let's consider we have M number of controllers.

Each Controller is assigned with several numbers of switches. Let's consider we have N number of switches connected to each controller. Depending upon the number of switches the design of the network changes. All the parameters of the network changes with respect to the complexity of the network.

$C= \{C_1, C_2, C_3.....C_M\}$ (1)

$S= \{S_1, S_2, S_3.....S_N\}$ (2)

Each and every controller communicates with each other over their interfaces to maintain the global view of SDN-enabled network topology. Flat structure of multi-controller architecture is displayed in Fig. 1. Capability of control plane is extended by the flat design. It does not require for the root controller to control the architecture.

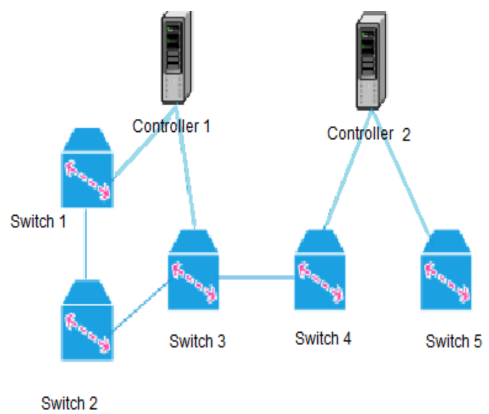

Fig. 1. Flat structure of multi-controller architecture.

D. Hierarchical structure of multi-controller

Fig. 2 displays the hierarchical structure of SDN controller architecture. There are three layers in this architecture and are named as physical layer, area controller layer and root controller layer. Here we have M number of controllers as in Eqn. (1) known as Area controller and for each area controller N number of switches as in Eqn. (2) is connected. In addition to this structure, we have one more controller, which controls these N controllers. This controller is known as Root Controller. Every Root controller will have several area controllers connected under it and the work for the area controllers are assigned by the Root controller. The huge amount of OpenFlow switches and routers are presented in the physical layer. Intra-area topology management, intra-area routing request processing, connecting physical devices and link information are performed by the area controller layer. Also, it sends area network details to root layer. Root controller is presented in the top of the architecture which treats area controllers as devices [13].

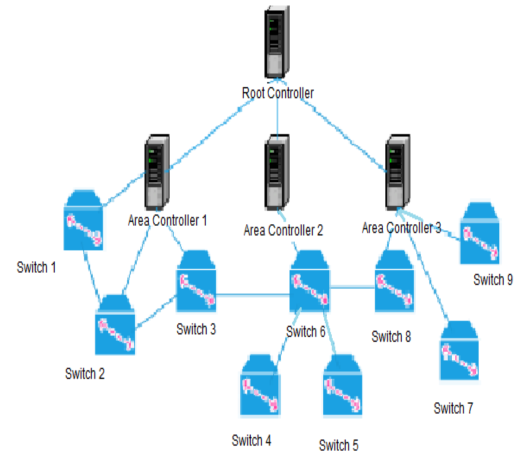

Fig. 2. Hierarchical structure of multi-controller architecture.

E. Description about Controller Assignment

The controller structure contains number of controllers, switches and links (connections among switches). Initially, multi-controller architecture is established in different locations. Only one controller is used to charge the entire network and the network is divided into multiple numbers of control domains. The communication between controller and switch is performed based on an in-band model. The given below instructions explain the way of establishing control relationship between controller and switches.

– Initially, controller sends messages to their switches to create control relationship.

– Switches find their controllers and send them a control requests.

– The routing depends upon the flow table of switches.

The multi-controller architecture of SDN is defined by G(S, C, L, P) in which S is mentioned as set of switches, C mentions the controllers, L denotes the location of controllers and P expose the physical links of networks. Memory size, processor and access bandwidth determines the limited capacity of every single controller. To avoid the overload, limited amount of switches are allocated to one specific controller [14].

F. Flow Initialization model of Multi-controller

The processing flow of a controller is defined in this section. Here, the requests are denoted as R. Initially, the controller judges whether the request is local or not. The local module processes the local tasks. A binary decision variable is defined for non-local tasks and it is denoted as A_{Cr}^{R} . Here, Cr denotes the controller. The local module process the task when $A_{Cr}^{R}=1$. Otherwise, the task is propagated to its parent controller. In latency measurement, A_{Cr}^{R} is considered as decision variable. λ_{mc}^{TL} and λ_{mc}^{L} mentions the arrival rate of non-local and local tasks in c_{mc} . The higher service rate, lower computation complexity, and capacity of larger controller are denoted as C_{mc}. Service rate of local and non-local tasks are explained in the given below expression.

$$\eta_{mc}^{L} = \frac{C_{mc}}{(h_{mc})^2} \quad (3)$$

$$\eta_{mc}^{NL} = \frac{C_{mc}}{(\sum_{i=1}^{L} h_{mc})^2} \quad (4)$$

Here, h_{mc} denotes the domain size of controllers. The given below expression defines the processing time of local and non-local tasks in c_{mc} .

$$T_L = \frac{1}{\eta_{mc}^{L} - \lambda_{mc}^{L}} \quad (5)$$

$$T_{NL} = \frac{1}{\eta_{mc}^{L} - \lambda_{mc}^{TL}} \quad (6)$$

Processing time of local and non-local tasks are mentioned as T_L and T_{NL}.The flow model is the fundamental task of SDN controllers in which the flow table is distributed based on the routing method. Here, $fl_{ij}(1 \le i, j \le L_1)$ mentions the flow from i[th] to j[th] sub-domain and the new flow generating rate is denoted by λ_{ij} of fl_{ij} . There are two stages in initial process flow of fl_{ij} and are mentioned as upward requesting and downward executing. During upward requesting, each and every request will get propagate upwards and this propagation continues till reaching the controller. The decision mode variable for this upward flow is set as 1. However for downward executing, the requests are separated into various local requests, and then transmit such requests toward the respective controller for accomplishing flow table distribution [15].

G. Latency Measurement

In SDN architecture, latency of control message is considered as one of the important parameter. During the routing process, controllers provide only the strategies and traffic to the network. The switches are not intelligent to process the packets without any instruction from the controller. There exists a constant exchange of messages between controller and switches during the routing operation. The latency among the switch and controller is measured by the exchange of messages in the network. It can be achieved with the help of creating timestamps in each payload of certain packets. Initially, the message is send from point A to point B with the timestamp. Then, point B extracts the timestamp and estimates the time taken to travel from A to B in the network.

In this work, we consider latency parameters like packet transmission latency, propagation latency, controller processing and switch queuing latency. The most commonly measured latency is propagation latency. Total time taken to forward the control messages over an entire network is defined as propagation latency. It is otherwise known as the time required for the message bits to reach destination from the source. Distance and propagation speed are two different factors which is based on the propagation delay.

The propagation latency of control message is calculated by the given below formula.

$$T_\alpha = \sum_{e \in \alpha} \eta_e d_e, \forall \alpha \in A \quad (7)$$

Here, η_e mentions the propagation latency of one accessible path e per kilometre, the distance of link is mentioned by d_e and the set of latency between switches and controllers is mentioned by A.

Then, the processing rate of switch is used to measure the message queuing latency. Time taken by the control messages to be line up in the routing queues is defined by Queuing latency. Main reason for this latency is intermediate switches, originating switches and call receiver servicing switches. The queuing latency can be calculated by the given below formula.

$$Q_v = \frac{\rho_v}{\mu_v(1-\rho_v)} - \frac{m_v \rho_v^{m_v}}{\mu_v(1-\rho_v^{m_v})}, \forall v \in A \quad (8)$$

Here, the estimated queue length is mentioned by $\rho_v = \frac{\lambda_v}{\mu_v}$ and m_V is taken as a fixed value for calculating Q_v.

When the control messages are successfully transferred, then switch to controller latency is calculated based on the given below expression.

$$R_\alpha = T_\alpha + \sum_{v \in A} Q_v \quad , \forall_\alpha \in A \quad (9)$$

Because of the short length of the queue, some messages may be dropped. So, the given below expression denotes the message loss rate of switch.

$$\delta_v = \frac{1-\rho_v}{1-\rho_v^{m+1}} \rho_v^m, \forall v \in A \quad (10)$$

Here, m and ρ have the same definition in Eqn. (8). Also, the packet transmission latency is well-defined by the time required to put a data packet to the link. It depends on length of packet and size of the switch. It can be represented in the following formula.

$$T_v = \frac{S_m}{B} \quad (11)$$

Here, length of data packet is mentioned by S_m and size of the switch is denoted by B. control messages should be re-sent from source when they dropped by switches. After certain time interval, control messages will be re-sent from source point. One-chance re-sending method is followed in this model due to the resource wastage while continuous resending. Here, network condition is enhanced through the addition of multiple controller or switches. By considering dropped message retransmission, switches to controller delay is explained as,

$$L_\alpha = R_\alpha + F(1 - \prod_{v \in A}(1-\delta_v)), \forall \alpha \in A \quad (12)$$

Here, the maximum toleration time is denoted as F. Multi-paths are presented to forward the control messages in a hierarchical structure. The given below expression calculates the minimum latency between controller and switches.

$$w(u,c) = \min_{\alpha \in A_{uc}} L_\alpha, \forall u \in S, c \in C \quad (13)$$

Here, group of latency constrained available paths between switch u and controller c is mentioned by A_{uc} [14]

Calculated latencies are optimized with the help of SLO algorithm. The procedure of SLO algorithm is explained in the next section.

H. Sea Lion Optimization (SLO) Algorithm

Sea lions are lived in huge societies as they have more number of members also they are considered as one of the most intellectual animals [16]. They are lived in a huge colonies so that, they have many subcategories also have own hierarchy within them. In their living time, they traverse around these subcategories. Quickly responds to a fish movements and it is considered as a significant characteristics of sea lions. Furthermore, it has the capability to recognize the position of fish also it immediately shows reaction based on this collected information. His identification process is normally performed to gather the fishes towards the shallow water which is near to the shore and to the ocean surface. In dark water also they have the ability to sense the availability of fishes.

Sea lions grouped together to hunt the large number of fishes so that, it upsurge the prospects of obtaining more prey. The sea lions went together to hunt when there are huge number of prey and when the number of prey is less they hunt individually. Given below facts mentions sea lion's hunting behaviour in three stages.

– Whiskers to chase and hunt the prey.

– Encircle the prey with the help of other members in subgroup.

– Attack towards the prey.

Optimization of latency parameter is the main objective of this work.

I. SLO algorithm's mathematical model

Tracking, encircling and attacking prey are three different types of process which are mathematically modelled in this subsection.

J. Detecting and tracking phase

The size, shape and location of the prey is identified with the help of whiskers of sea lions. The whiskers direction is opposite to the water wave's direction so that, they can identify the prey and their location. After finding the location of prey, it calls other sea lions to its subgroup to pursuit the prey. For this hunting behaviour, this sea lion is deliberated as leader and other sea lions update their location to target prey. In this algorithm, optimal solution or best solution is assumed as the target prey. Also, the sea lion is considered as switches and the target prey is mentioned as controllers. In the initial stage, population initialization is done by place the controllers and switches in a SDN network. So, distance between controllers and switch is calculated based on the given below expression.

$$\vec{D} = \left| 2\vec{B}.\vec{C}(t) - \vec{S}(t) \right| \quad (14)$$

Here, current iteration is mentioned as t, the distance among controllers and of search agent, random vector $\vec{B}$ is multiplied with two to upsurge the search space which is in the range of [0,2]. In next iteration, the data flow is start from the switch to controller. The given switch is represented as $\vec{D}$, the position vectors of switch and controller is illustrated as $\vec{S(t)}$ and $\vec{C(t)}$. To find optimum or near optimum solution below expression explains the mathematical model of this behaviour.

$$\vec{S(t+1)} = \vec{C(t)} - \vec{D}.\vec{X} \quad (15)$$

Here, next iteration is represented as (t+1). Likewise $\vec{X}$ is reduced from two to zero through the course of iterations. Since this reduction helps the main switch to transmits data packets from switch to controller.

K. Fitness calculation

Main objective of our optimization algorithm is to minimize four different kinds of latencies like switch queuing latency, packet transmission latency, propagation latency, and controller processing latency. Given below expression mention the objective function of our work.

$$fitness\ function\ f(x) = \min(T_\alpha \times Q_v \times T_v \times R_\alpha) \quad (16)$$

Here, the propagation latency, Queuing latency, packet transmission latency and controller switching latency is mentioned by T_α, Q_v, T_v and R_α.

Pseudo code for SLO algorithm

Input : Controllers, switches, maximum iteration number, optimization parameters such as T_α, Q_v, T_v and R_α

Output : Optimized latencies

Begin

Initialize the optimization parameters such as propagation latency, queuing latency, transmission latency and controller switching latency (four different latencies).

Define objective function based on Eqn. (16)

Initialize current best search agent

while (I<maximum number of iteration)

for each search agent (switch)s

calculate the speed of leader

calculate latencies using Eqns. 7, 8, 9 and 10

if ($\vec{SP_{leader}} < 0.25$)

if(C<1)

update the distance

else if (C>1)
Choose random controller to calculate latency
end if
end if
if ($\overrightarrow{SP_{leader}} > 0.25$)
Update the location of current search agent
end if
end for
update the location
Compute fitness function for every search agent (switch) using (16)
Update S if there are any better solution
return minimum latency

III. SIMULATION RESULTS AND EXPLANATION

Simulation of controller design in MATLAB platform is described in this section. Proposed method is compared with the existing methods to show the efficiency of our optimization algorithm. For simulation setup, some nodes are considered to calculate the latency of controller. Both flat and hierarchy structure is considered for evaluation. In flat structure, we assume two or three controllers to calculate the latency of control message flow. In hierarchy structure, one node is considered as root controller and three or four nodes are considered as area controller and remaining nodes are considered as switches. The data transmission speed is mentioned by *de* = 5 µs/km. The service rate of all switches are same and the service rate is considered as μ = 5 Megabits/ms. Arrival rate λ ranges from 4 to 5 Megabits/ms. The queue length of switch is mentioned by m and it is denoted as *m* = 40 Megabits. Each and every controller is connected to different number of switches to avoid the data overload in the links.

A. Performance analysis

The given below figure shows the packet delay rate for different number of switches to show the efficiency of the proposed method. Both flat and hierarchical structure is designed to calculate the packet delay rate with different number of switches. Here, M denotes the number of controllers and N denotes the number of switches. Generally, increasing number of switches increase the packet delay rate in both flat and hierarchical structures are shown in the Figs. 3 & 4 respectively.

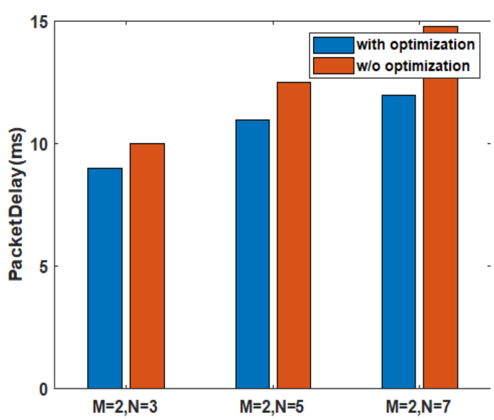

Fig. 3. Packet delay rate for Flat structure.

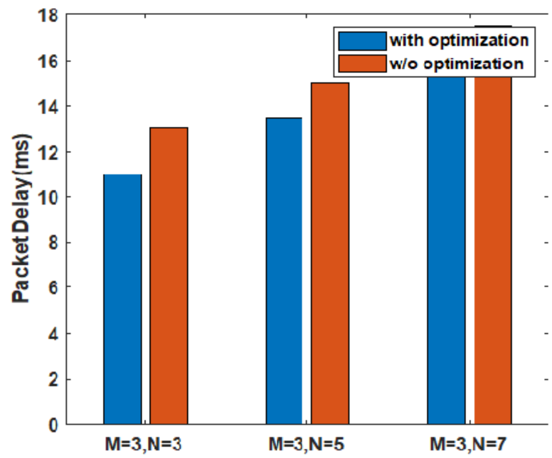

Fig. 4. Packet delay rate for Hierarchical Structure.

From the figures, it is clearly shown that the proposed optimization algorithm achieves less packet delay in both structures.

B. Average latency

Average latency of both flat and hierarchical structure is displayed in the given below figure. Here, M number of controllers and N number of switches. From the Figs. 5 & 6, it is clearly shown that the latency is increased with increasing number of switches in both structures. The proposed optimization algorithm calculates less latency while the other method achieves higher latency for different number of controllers and switches.

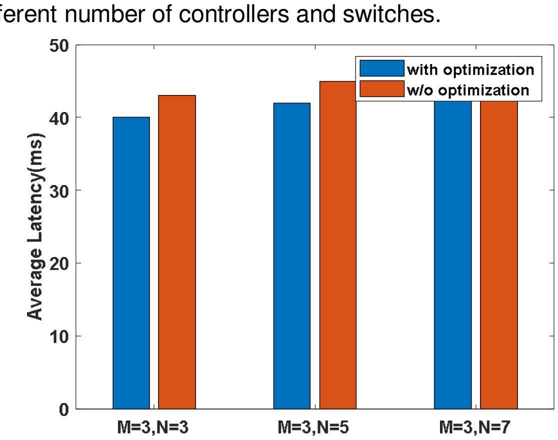

Fig. 5. Average Latency in Hierarchical structure.

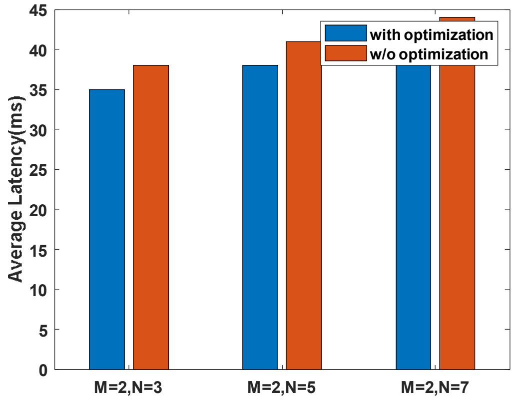

Fig. 6. Average Latency in Flat structure.

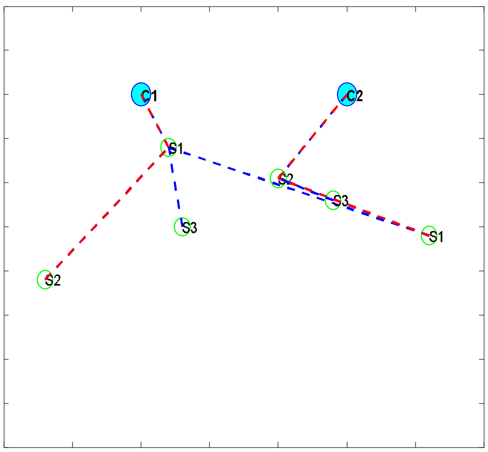

Fig. 7. Flat structure of multi-controller SDN when M=2(controllers) and N=3 (switches). Here we have 2 controllers as C_1, C_2 and for each controller 3 switches S_1, S_2, S_3 are connected to it. The communication between the controller and the switches are shown in the network diagram.

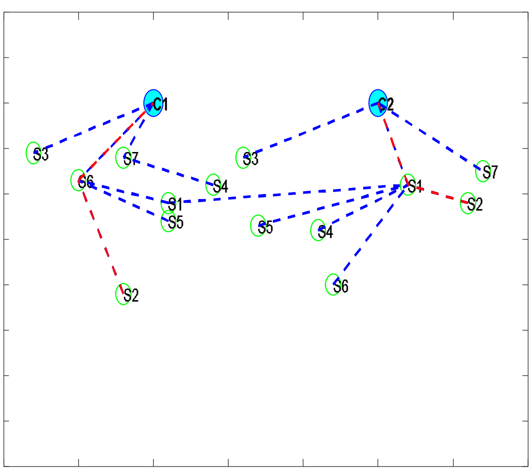

Fig. 8. Flat structure of multi-controller SDN when M=2 (controllers) and N=5 (switches). Here we have 2 controllers as C_1, C_2 and for each controller 5 switches S_1, S_2, S_3, S_4, S_5 are connected to it. The communication between the controller and the switches are shown in the network diagram.

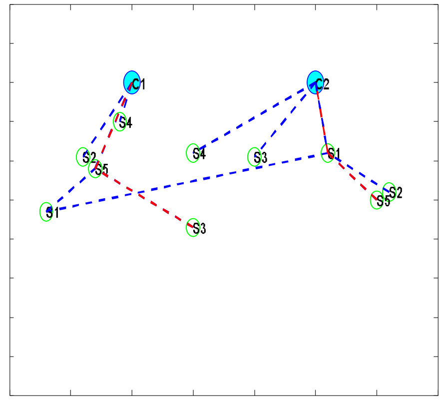

Fig. 9. Flat structure of multi-controller SDN when M=2 (controllers) and N=7 (switches). Here we have 2 controllers as C_1, C_2 and for each controller 7 switches S_1, S_2, S_3, S_4, S_5, S_6, S_7 are connected to it. The communication between the controller and the switches are shown in the network diagram.

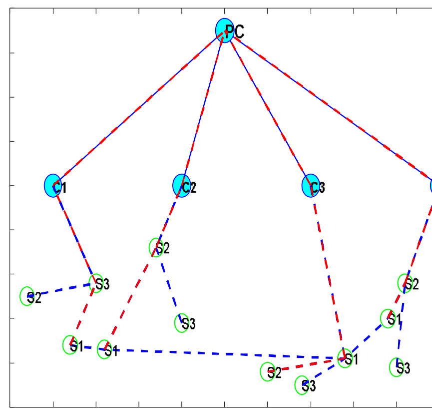

Fig. 10. Hierarchy structure of multi-controller SDN when M=4 (controllers) and N=3 (switches). Here we have 4 controllers as C_1, C_2, C_3, C_4 and for each controller 3 switches S_1, S_2, S_3 are connected to it. The communication between the controller and the switches are shown in the network diagram.

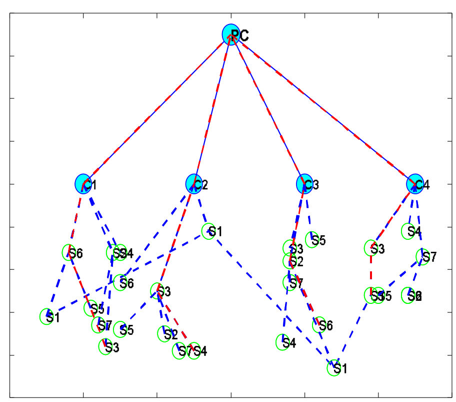

Fig. 11. Hierarchy structure of multi-controller SDN when M=4 (controllers) and N=5 (switches). Here we have 4 controllers as C_1, C_2, C_3, C_4 and for each controller 5 switches S_1, S_2, S_3, S_4, S_5 are connected to it. The communication between the controller and the switches are shown in the network diagram.

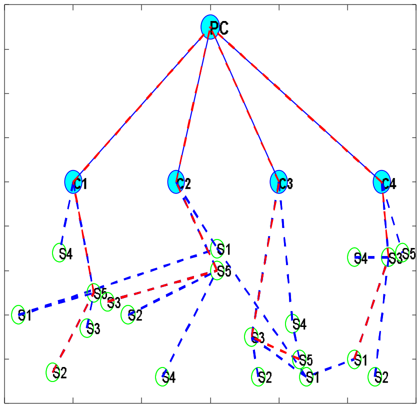

Fig. 12. Hierarchy structure of multi-controller SDN when M=4 (controllers) and N=7 (switches). Here we have 4 controllers as C_1, C_2, C_3, C_4 and for each controller 5 switches S_1, S_2, S_3, S_4, S_5, S_6, S_7 are connected to it. The communication between the controller and the switches are shown in the network diagram.

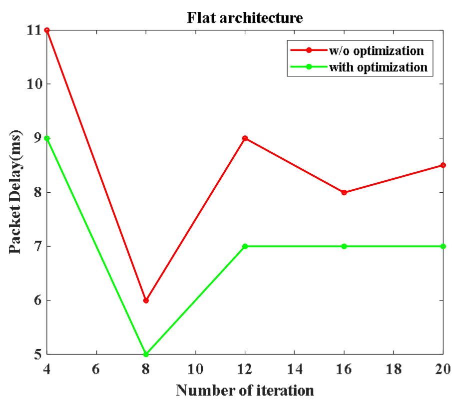

Fig. 13. Packet Delay for Flat Structure.

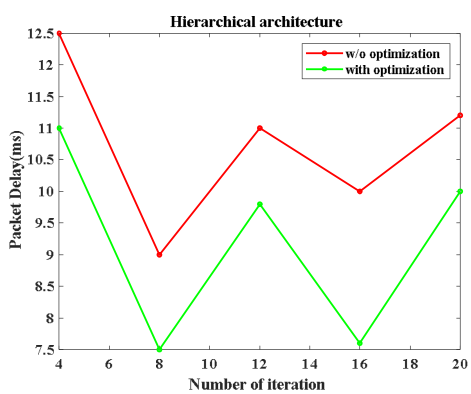

Fig. 14. Packet Delay for Hierarchical Structure.

The Figs.13 and 14 gives the view of Packet Delay which is been measured with respect to the different number iterations for both the structures. The Figs. 15 and 16 displays the result of Drop of packets in the network which is measured in Packet Loss Rate. The performance of proposed method is compared against the existing methods, Legacy Ignored Assignment (LIA) and Legacy Based Agreement (LBA) technique [14].

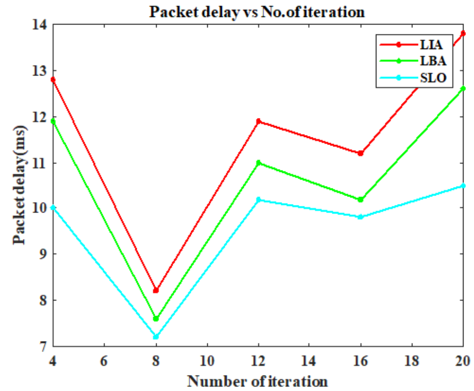

Fig. 15. Comparison graphs of the optimization algorithms.

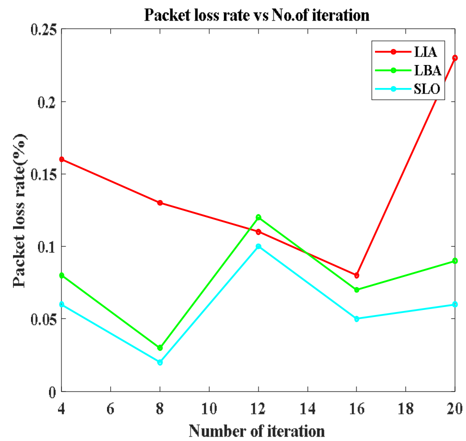

Fig. 16. Analysis of Packet loss rates of the optimization algorithms.

IV. CONCLUSION

Latency optimization of SDN controller is considered in this paper. We have designed a flat and hierarchical model multi-controller structure and proposed the SLO algorithm to optimize the calculated latencies. Initially, the multi-controller architecture is structured into flat and hierarchical model. Then the latencies like controller processing latency packet transmission latency, propagation latency, and switch queuing latency are calculated with the help of designed multi-controller architecture. Finally, the SLO algorithm is proposed to optimize the calculated latencies. In this optimization, latencies are minimized and the network scalability is enhanced. The performance of proposed method is calculated in terms of packet loss rate, average latency and packet delay with varied number of controllers. Effectiveness of the proposed method is displayed in the simulation results.

V. FUTURE SCOPE

The current study has proven the performance in optimizing the latency with other conventional methodologies. This can be extended and performed in Real-time scenario.

Conflict of Interest. There is no conflict of interest by any of the authors in this paper.

REFERENCES

[1]. Maity, I., Mondal, A., Misra, S., & Mandal, C. (2018). CURE: consistent update with redundancy reduction in SDN. *IEEE Transactions on Communications*, *66*, 3974-3981.

[2]. Tantayakul, K., Dhaou, R., & Paillassa, B. (2016). Impact of SDN on mobility management, *In 2016 IEEE 30th International Conference on Advanced Information Networking and Applications (AINA)*, IEEE, 260-265.

[3]. Schaller, S., & Hood, D. (2017). Software defined networking architecture standardization. *Computer Standards & Interfaces*, *54,* 197-202.

[4]. Aly, W. H., & Al-anazi, A. M. (2018). Enhanced Controller Fault Tolerant (ECFT) Model for Software Defined Networking. *In 2018 Fifth International Conference on Software Defined Systems (SDS), IEEE,* 217-222.

[5]. Smida, K., Tounsi, H., Frikha, M., & Song, Y. Q. (2019). Delay Study in Multi-controller Software Defined Vehicular Network Using OpenDaylight for Emergency Applications. *In2019 15th International Wireless Communications & Mobile Computing Conference (IWCMC)*, *IEEE*, 615-620.

[6]. Ateya, A. A., Muthanna, A., Vybornova, A., Algarni, A. D., Abuarqoub, A., Koucheryavy, Y., & Koucheryavy, A. (2019). Chaotic salp swarm algorithm for SDN multi-controller networks. *Engineering Science and Technology, an International Journal*, *22*(4), 1001-1012.

[7]. Li, F., Xu, X., Yao, H., Wang, J., Jiang, C., & Guo, S. (2019). Multi-controller resource management for software-defined wireless networks. *IEEE Communications Letters*, *23*(3), 506-509.

[8]. Fan, Z., Yao, J., Yang, X., Wang, Z., & Wan, X. (2019). A Multi-Controller Placement Strategy Based on Delay and Reliability Optimization in SDN. In *2019 28th Wireless and Optical Communications Conference (WOCC),* 1-5. IEEE.

[9]. Francois, F., & Gelenbe, E. (2016). Optimizing secure SDN-enabled inter-data centre overlay networks through cognitive routing. In *2016 IEEE 24th International Symposium on Modeling, Analysis and Simulation of Computer and Telecommunication Systems (MASCOTS),* 283-288. IEEE.

[10]. Keupondjo, G. A., Anoh, N. G., Adepo, J. C., & Oumtanaga, S. (2019). Hybrid Routing With Latency Optimization In Sdn Networks. *Journal of Engineering Science and Technology*, *14*(5), 3062-3072.

[11]. Sharma, P. K., Singh, S., Jeong, Y. S., & Park, J. H. (2017). Distblocknet: A distributed blockchains-based secure sdn architecture for iot networks. *IEEE Communications Magazine*, *55*(9), 78-85.

[12]. Prete, L. R., Shinoda, A. A. Schweitzer, C. M. de R., & Oliveira, L. (2014). Simulation in an SDN network scenario using the POX Controller. *In 2014 IEEE Colombian Conference on Communications and Computing (COLCOM), IEEE*, 1-6.

[13]. Selvi, H., Gür, G., & Alagöz, F. (2016). Cooperative load balancing for hierarchical SDN controllers. *In 2016 IEEE 17th International Conference on High Performance Switching and Routing (HPSR), IEEE,* 100-105

[14]. Yuan, J ., Li, E., Kang, C., Chang, F., Yuan, T., & Li, X. (2019). Latency-Based Dynamic Controller Assignment in Hybrid SDNs: Considering the Impact of Legacy Routers. *Future Internet*, *11*(8), 1-14.

[15]. Li, J., Liu, J., Gao, Q., & Huang, T. (2019). The Decision Latency Optimization Problem in SDN With Multi-Controller. *IEEE Communications Letters*.

[16]. R. Masadeh, B.A. Mahafzah, A. Sharieh (2019), Sea Lion Optimization Algorithm. *Sea*, *10*(5), 388-395.

How to cite this article: Ramasamy, M. and Pawar, S. (2020). Latency Optimization of Structured Multi-Controllers in Software defined Networks. *International Journal on Emerging Technologies*, *11*(3): 82–90.