DESIGN OF MULTI-AGENT MULTI AGGREGATOR (M-MA) FOR THE DISTRIBUTED WIDE AREA NETWORK

W.Rose Varuna
[1] Assistant Professor, Department of Information Technology,
Bharathiar University, Coimbatore-46, Tamilnadu, India
hvaruna@gmail.com

R.Vadivel
[2] Assistant Professor, Department of Information Technology,
Bharathiar University, Coimbatore-46, Tamilnadu, India
vlr_vadivel@yahoo.co.in

Abstract - A hysteresis multiple agent aggregations (M-MA) based load balancing system is proposed to distribute the Software-Defined Networking (SDN) control plane of Wide Area Network architecture. Agent-based load balancing is based on the Markov chain process. It considers control policy with thresholding and scaling under the eventual distribution of traffic among an overloaded SDN controller (user controller) and lightweight neighboring Controller (server-side). This can be acquired with a more massive discrepancy in resource utilization via dynamically utilizes the Controller's global available capacity. The anticipated M-MA strategy attempts to gain Multi-objective computation appropriate to high availability, scalability, flexibility, agility, resource utilization, energy-saving, and data computation probability without any overhead. This is anticipated via numerical investigations to validate the efficiency of the expected model. This is acquired by steady-state and transient analysis based on suitable performance metrics like transition rate (among server and client), aggregated capacity, throughput computation, and latency computation. This work depicts that the anticipated (M-MA) scheme outperforms prevailing approaches, and resource provision is attained with specific essential requirements.

Keywords**:** SDN, Multi-agent, aggregation, Markov chains, Controller

1. Introduction

The advancements of advanced computation technologies like the Internet of things (IoT), big data, and cloud systems primarily modify the method for storing, acquiring, and transferring data and information [1]. High-speed transmission and resource utilization are elevated based on certain new computational challenges, trends, network management, and up-grading. To deal with these goes up against, Software Defined Networking (SDN) is considered as a promising theory for cutting edge calculation [2]. Despite conventional systems administration, SDN decouples with the data plane from the control plane. Here, the control plane is produced by a set of committed controllers, and each Controller oversees rearranged Packet-Forwarding Switches [3]. As a result, the executives and control capacities are executed at SDN controllers. Along these lines, network and application administrations are pre-involved from giving infra-structure.

In SDN, application-based Controller fills in as network switches and network intellects as straightforward packet sending gadgets modified through an open interface. Open Flow. The improved system architecture is given in Figure 1. Show packet conveyance conspires in Open Flow: 1) if the packet arrives at switch among flow table in the switch that doesn't partner this packet flow, 2) switch will create demand on the packet and transmit it to related Controller. After gathering this solicitation, 3) the Controller will respond to the exchange with a novel sending arrangement, and the switch may refresh the flow table accordingly. Notwithstanding, 4) packet is conveyed dependent on flow table updation.

As delineated in this model, each SDN capacities are performed using continuous message trade among switches and controllers [4]. Hereafter, controller area impacts drastically through message trades and consequently influences SDN execution. The difficulty in examining the reasonable area of controllers is considered as controller position disaster.

This emergency indicates how to put controllers in SDN based system and allocate related changes to those controllers, in this manner to arrive at this target [5]. Controller position is very much suitable for Wide Area Networks, as its asymmetrical topology and immense latency in packet spread.

In solutions to determine controller position problem in SDN, execution measurements have been foreseen in so many publications [6]. With every one of these measurements, network delay/latency among switches and controllers goes about as crucial part, as it impacts the whole SDN execution tremendously. For instance, if the packet does not match with switch-based flow table, the switch has to wait for a longer time period [7]. Therefore, it has to acquire a newer flow table that has to be installed before the packets' execution. In fact, loner time for waiting time will humiliate over SDN performance. For example, time insensible application may more slow down when continuous assignments may consider being infeasible [8]. Normally, links with substantial traffic and obliged bandwidth capacity are inclined to experience congestion, and subsequently, prompts included latency. Owing to unique SDN features, latency outcomes from congestion among switches and controllers are quite restricted in SDN enabled network [9]. Initially, control planes are partitioned from the SDN data plane. Therefore control messages among switches and controllers are transmitted among dedicated channels (out-of-band mode) [10].

Additionally, control messages hold similarly least flow in divergent to data plane burdens. Various strategies have been foreseen in publication to reduce engendering idleness by estimating the essentialness of delay among switches and controllers. Besides, the broadcast delay is counted as one factor to decide on the whole latency. Other elements comprise switch processing latency, packet transmission latency, and queuing controller latency. A quantitative and depth investigation of total latency, specifically for queuing latency among controllers, has acquired constrained analysis in Related work.

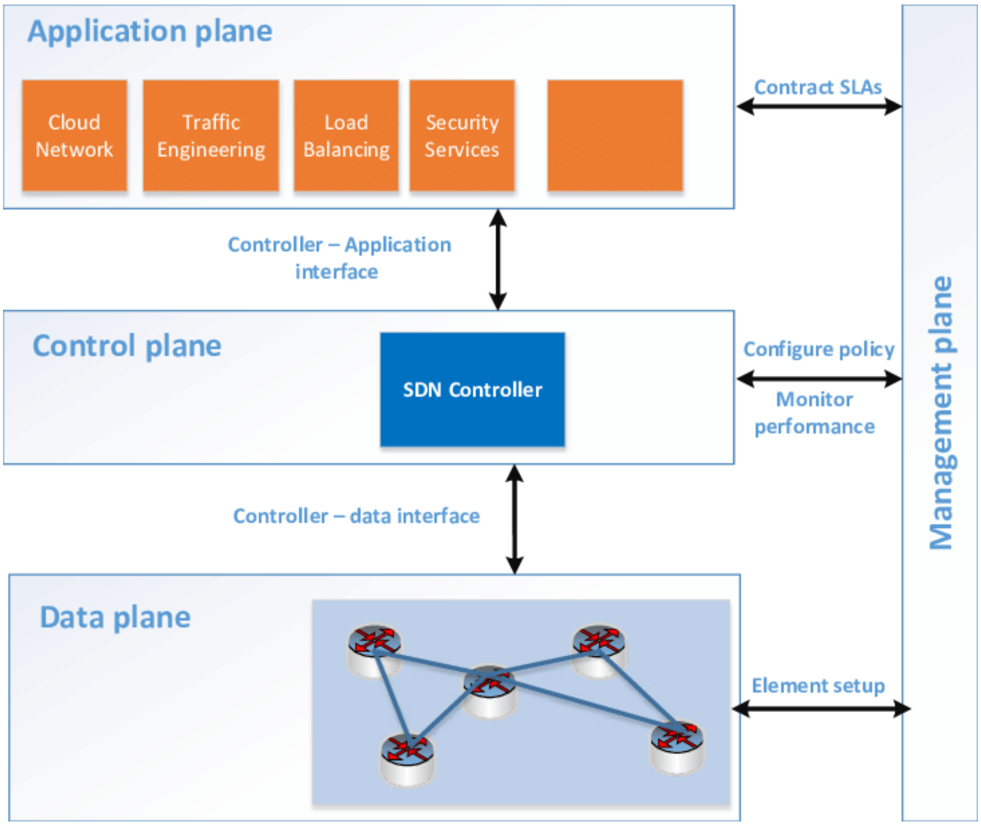

Fig. 1 Generic SD-WAN Architecture

In this work, the overall latency among switches and controllers is analyzed, and solutions are considered to reduce the Wide Area Network [11]. As overall, latency has End-to-End latency (E2E) (switch handling latency, packet broadcast latency, and packet transmission latency), and Controller based lining latency are utilized to address them freely [12]. Prior, a network system distributing is predicted as subjected to diminish End-to-End latency among switches and controllers. A few algorithms, like group-based network segments, are considered to perform organize apportioning [13]. This may satisfy and ensure that each partitioning is skillful to diminish the most End-to-End latency among controllers and related switches. In this manner, various controller positioning among sub-network is foreseen to have enormous switches to reduce queuing latency.

To quantitatively calculate queuing latency, queuing theory is utilized in this work to demonstrate the relationship among the number of controllers and queuing latency [14]. An appropriate measure of controllers is figured for every one of the different controllers and sub-network put among sub-networks to decrease queuing latency [15]. The critical commitments of this work are featured as trails:

•Entire latency among switches and controllers in SDN empowered network is entirely investigated here. In reality of other existing works, there is some credible commitment towards in entire latency. To be explicit, the entire latency involves End-to-End latency and Controller queuing latency. End-to-End latency incorporates packet broadcast inactivity, packet transmission latency, and switch handling latency.

•The thought behind network partition is initiated in this research work to hold controller positioning problem with Multi-agent aggregation. To reduce E2E latency, partitioning network into sub-network and appropriate controllers' position for all sub-networks has been envisioned. Collection based methodology is considered to determine this network segment problem.

•This Markov chain process-based method satisfies that each part is skilled in decreasing maximal E2E latency among switches and controllers.

•Multi agent-based controller placement is anticipated to reduce queuing latency among controllers. Concerning the queuing model, an enormous amount of controllers are computed for all sub-network, multi-controllers are placed in sub-networks inside a dense switching environment and to reduce queuing controller latency. Therefore, it reduces total latency among switches and controllers.

•Simulation is performed in an extensive manner in the NS-2 environment. Simulation outcomes validate that the anticipated model possesses greater latency reduction among switches and controllers compared to prevailing models to resolve controller placement problems.

The rest of the article is sorted out as: Section 2 is related works, Section 3 portrays newly proposed technique, Section 4 is numerical outcomes and discussions, and Section 5 is the conclusion and future extension.

2. Related Works

Reference [16] anticipated a newer community-based detection for controller deployment. The heuristic calculation is improved, and crucial adaptability elements are envisioned to confine the number of nodes in each network and stables the distinction among the nodes among various networks. Reference [17] portrayed a versatile transformative algorithm alongside a greedy approach to deal with better introductory populaces, a smart approach to give escalation and broadening, and new quicker Pareto discoverer for enormous scope multi-target controller improvement algorithm. In addition, this methodology needs some computational time and memory assets. Reference [18] portrays the controller positioning approach based on the bi-partition chart to decrease the weighted match. Earlier, the Kuhn-Munkres approach is pushed off to decide the best match among controllers and switches. Then, with genetic algorithms, a solution for controller placement is determined for considering average propagation latency.

Reference [19] foresee two methodologies for versatile capacitated controller position emergency in SD-WAN. Earlier, deciding controller limit, stream load among switch controller and switches, broadcast latency among controllers, strong controller positioning is envisioned. In this way, in understanding the approach for displaying NP-hard depending on a faction in graph theory, the polynomial-time methodology is expected. Reference [20] envisions k-self adaptive controller positioning methodology for SD-WAN. Versatile spectral grouping approach segments superior network among various tiny areas subject to range grouping to decrease WAN delay and boost controller dependability. The eigenvector structure is pushed off to naturally depict the number of SDN areas and foresee measurements for spectral grouping position.

Reference [21] delineated three noteworthy pointers, including hop count, latency, interface use, and foresee a novel way to decide the controller position approach. With these pointers, diagnostic progressive handling innovation is pushed off to look at Controller's ideal situation [22]. Also, an improved controller positioning put together a genetic algorithm subordinate concerning novel innovation is foreseen to determine position task emergency in CPP(Controller Placement Problem) [23]. With these examinations, instructions at the Multi-controller plane in level controller structure without considering various leveled controller plane [24]. Progressive compositional plans based control structure is intended to dispose of vitality utilization and synchronization latency because of data recurrence and synchronization among controllers in level structure [25]. Various constructed engineering can amazingly improve adaptability in SD-WAN.

Network dividing in even structure decides only correspondence performance among controllers and switches there. In progressive construction, they were dividing impacts not just intra-domain correspondence execution, though between space correspondence also. Unlike the level control structure, in various leveled control designs, the root controller area impacts correspondence execution among the space controller and root controller. Consequently, message correspondence solicitation and business handling among root and switch controller will be affected when between area stream shows up. Thus, it is fundamental to examine the subtleties of system apportioning and organization of root controller, space controller in the progressive control structure.

3. Proposed Methodology

This section considers the essential factors needed to model and distribute SDN in WAN based on control plane placement in centralized logical perception for executing the Markov chain process. It is considered that:

1) Here, structure modeling is completed with SDN control plane with the logical allocation of the controlling structure of the data plane and legitimately centralized view.

2) The data (network) plane comprises a set of delivering switches, and the entire switch is associated with a certain Controller with its nearest location.

3) SDN control plane involves a set of controllers.

4) Every Controller is composed on set of virtual instances.

5) Every Controller has to work on Markov decision processing and acts as an essential role in the server-side and client-side (have competency to handle accepting and responding to the client)

6) There is some communication link among every pair of neighborhood controllers, and every link is subjected to impairments.

When Markov based decision process is triggered, every link related to heavily loaded Controller to moderately loaded Controller may undergo failure or congestion with a certain probability. This thought encourages t to expect to organize hindrances in this methodology.

At the point when the controllers' network is portioned into various non-overloading SDN states, it encourages gradual consumption and adaptability.

• Controller network is given as a shared occasion pool.

• Every Controller contains certain locally reasonable SDN areas, and each space involves a set of switches conveyed over the data plane. The number of switches per state won't surpass over the controller limit.

3.1. Markov Decision Process In Localized Environment Of Every Controller

Typically, SDN controller assets are pooled to act as various switches subject to Multi-tenant models with different virtual and physical assets comfortably and powerfully apportioned and re-allocated to data plane necessities. This work utilizes a novel Multi-agent aggregator based decision model as the primary material for this research. This model considers an underlying structure that has been considerably modeled for this current research work.

To fulfil reproducibility of Multi-agent functionality, decision modeling is available for partitioning control plane. This model makes agents simulate the characteristics of SDN components. Figure 1 depicts an overview of the model. Network-based Multi agents specify network components like a group of clusters, and these may hold varying characteristics. Most of these agents appropriately transmit network packages based on the ordering of software-defined centralized controllers.

Moreover, some of the network agents may possess non-proper or malicious characteristics. These characteristics are intended for compromising privacy and considering non-authorized data. Sometimes it may encounter some malfunctioning features that lead them to lose a higher network rate. From this perspective, these kinds of network agents considered are termed as 'malware agents'. This work specifies it as 'normal network agents' in this research work to differentiate malware agents from others.

After that, the 'controller agent' may represent a centralized SDN controller. It specifies how to route network traffic planes to eradicate saturation. In Multi-agent based aggregation, these agents may work primarily to merge the isolation protocol. These are programmed by SDN designers to model enormous isolation protocols. Controller agents combine the table with trust with every network agent as a 0-100 percentage time interval. This accumulation may recreate a huge number of information from the sender to the receiver. In the conversion process, the controller agent chooses the way indicated by various network operators. The recipient may set up that message was received or lost suitably. Once messages are lost, the controller agent can follow which spot packet path is lost and decreases trust in networking mediators.

Algorithm: Multi Agent-based Controller

Input: Nodes with connected network topology
Output: Average latency computation

Step 1: Allocate with appropriate clustering
Step 2: Compute E2E latency computation between connected nodes
Step 3: Randomize node for network initialization
Step 4: Validate actual network centroid
Step 5: Determine the route between connected nodes
Step 6: Merge controllers agents without message loss and track package route
Step 7: Deploy multi-controllers for core partitioning
Step 8: Compute load of controllers with Eq.(1)
Step 9: Determine objective function with Eq.(2)
Step 10: Determine link utilization rate
Step 11: Compute path reliability among nodes $'i'$ and $'j'$
Step 12: Compute the shortest path between nodes $'i'$ and $'j'$
Step 13: Evaluate Centrality index of normalization degree
Step 14: Compute Multi-objective constrain with Eq. (14), Eq. (15) and Eq. (16)
Step 15: Output latency based on constrained factors

3.2. Markov Decision Process Based Load Balancing

In SD-WAN, the control plane is deployed by multiple controllers; these controllers monitor and handle core partitioning. Controllers manage a load of every partition is considers as controller load. Controller capacity to manage load isn't identified with software structure, anyhow confined by server sources itself. Controller overloading leads to raising in response latency during controller failure and network events. Henceforth, load balancing during partition is considered as an actual performance to resolve controller deployment. Some investigations consider balancing of the number of switches in every partitioning. Load during segmentation involves all switches handle the number of switches in controller supervision and flow size as in Eq. (1) as follows,

$$L = \left(\sum_{i=1}^{k} \alpha \left(\frac{\sum_{j=1}^{N_i} \frac{\lambda_{i,j}}{k}}{\sum_{i=1}^{K} \sum_{j=1}^{N_i} \lambda_{i,j}} - \frac{1}{k} \right) + \beta \left(\frac{N_i}{N} - \frac{1}{k} \right)^2 \right)^{\frac{1}{2}} \tag{1}$$

L is Load calculation, K is the total number of nodes, N is the number of partition, $\lambda_{i,j}$ is request rate of the switch, α & $\beta$and is weighting co-efficient during load balancing. When $\alpha = 1$is load balancing $\beta = 1$the number of switches partitions index. The objective function is specified as in Eq. (2) as follows,

$$\min L \tag{2}$$

Load balancing constraints are as in Eq. (3) as follows,

$$\overrightarrow{f_{vi}} = \overleftarrow{f_{vi}}, V_i \in V \tag{3}$$

$$l_{i,j} \leq 1, \qquad (i,j) \in E \tag{4}$$

$$\sum_{i \in N_i} \lambda_{i,k} \leq \phi_k, \; k \in K \tag{5}$$

Eq. (3), (4) & (5) specifies controller partition-based flow is provided equally. Where f_vi is flow, V_i is Vertex estimation of node I, V is Vertex, and ϕ_k is the load resistance. The connection use of each connection is lesser than 1. The Eq. (5) is a complete load of each domain that can't go over the domain controller's extreme load resilience.

In circulation Multi-controller placement state, the consistency of the control plane is an essential progress measurement. Higher than character, the control plane frequently works with no loss. The loss brings about interference between the domain controller and the switch. Henceforth, reliability is also an essential performance metrics to resolve controller deployment. The link includes communication between domain and switch Controller, and between the root controller and the domain controller.

Path reliability among$'i'$ and $'j'$is depicted as in Eq. (6) as follows,

$$R\ (i,j) = \prod_{e\ \in E_{i,j}} (1 - p_e) \prod_{v\ \in V_{i,j}} (1 - p_v) \tag{6}$$

Where E_(i,j) is connected set of shortest way among node $'i'\ 'j', V_{i,j}$The shortest path set between the node$'i'$ and$'j' p_e$ shortest path edge $p_v$is the fastest path set vertex.

$$R_c(k) = \sum_{j\ \in C_k} R\ (cc\ (k), j) \tag{7}$$

The$C_k k^{th}$ partition where the node$'j'$ is located$R_c(k)$ in the controller's reliability is the$R\ (cc\ (k)$ reliability of the root and domain controller. The path reliability of the root controller is Eq. (8) as follows,

$$R_{rc} = \sum_{k\ \in K} R_c(k). R(cc(k), rc) \tag{8}$$

Different investigations consider path consistency in settling consistency problems, though it does not consider other node factors. Here, not only the reliability of links and nodes are considered; however, request rate and node degrees are considered. Node degree is the number of edges related to the node. The centrality index of the normalization degree is provided as in Eq. (9) & (10) as follows,

$$R_{degree} = \frac{degree\ (i)}{\max_{i\ \epsilon\ N}(\deg(i))} \tag{9}$$

$$R_{\lambda,i} = \frac{\lambda_i}{\max_{i\ \in N}(\lambda_i)} \tag{10}$$

Where $R_{degree}$is node degree and $R_{\lambda,i}$is the centrality index of normalization degree.

If node 'i' is conveyed as root controller, node consistency is portrayed as in Eq. (11) as follows,

$$R_i^{rc} = R_{rc}. R_{degree} R_{\lambda i} \tag{11}$$

Where $R_i^{rc}$is the reliability of root controller i.

When choosing the domain controller and root controller, the objective function as in Eq. (12) & (13) as follows,

$$\max R_i^c \tag{12}$$

$$\max R_i^{rc} \tag{13}$$

The multi-objective constraint is provided as in Eq. (14) as follows,

$$\sum_{i\ \in N_i} \lambda_{i,k} \leq\ \phi_k, \qquad k\ \in K \tag{14}$$

$$\sum_{k \in K} \sum_{i\ \in N_i} \lambda_{i,k} \leq\ \phi \tag{15}$$

$$\sum y_{cc}^{rc} = K \tag{16}$$

Previously mentioned Eq. (14)(15) and (16) depend on each domain's complete load lesser than the maximum load resilience of domain controller, the$\boldsymbol{y_{cc}^{rc}}$ aggregate total of all domains. The whole amount of all domains in the entire network is lesser than the maximal load that root controller.

4. Experimental Results And Discussions

This portion examines in aspect regarding the numerical outcomes and conversations of the proposed model. Here, recreation was completed in the NS-3 simulator, and the network topology model was accepted to arrange the controller plane. The assessment was completed in execution measurements like packet failure, latency, and throughput. The running time of the proposed calculation is 0.6100 seconds. In light of the proposed model, the unwavering quality of the control plane is experienced adequately. Here, the proposed model contrasts with Markov's decision procedure regarding latency, packet failure, and throughput when the controller is positioned.

Table I. Configuration setup

Parameters	Configuration 1 (Markov decision process)	Configuration 2 (M-MA)
Agent reliability	96.0	96.0
Agent with malware	61.0	16.0
Number of network agents	85	85
Number of network agents with malware	16	16
Number of network packages	1000	1000

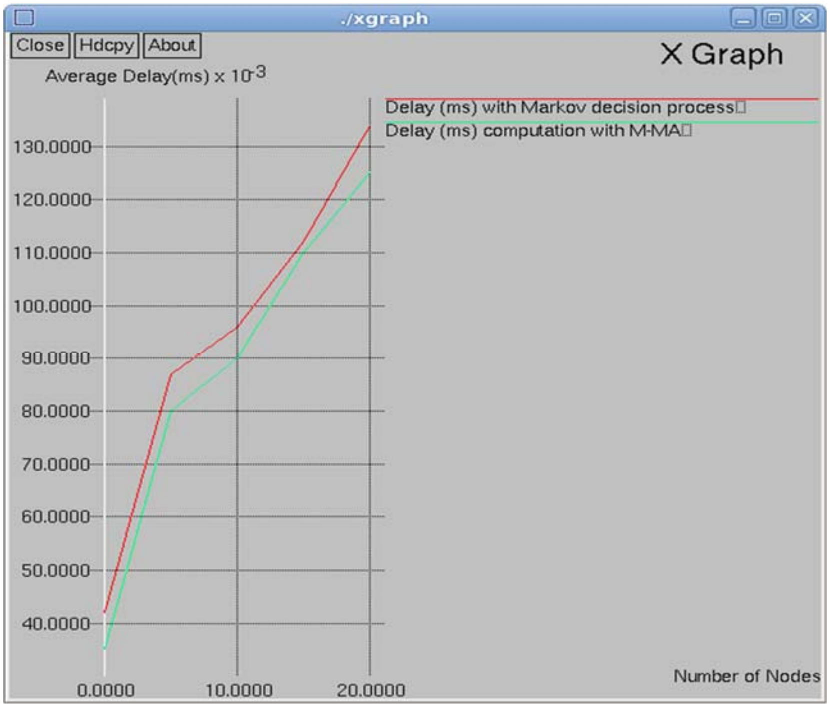

Fig. 2 Latency calculations before node allocation

Figure 2, Figure 3, shows the graphical portrayal of the usual latency calculation in Milliseconds. 'X' axis shows the number of controllers, and 'Y' axis shows the usual latency (*ms*). The proposed model shows a better exchange regarding latency as opposed to winning methodologies. Figure 4, Figure 5, shows the graphical portrayal of the usual packet failure calculation in Milliseconds. 'X' axis shows the number of controllers, and 'Y' axis shows the usual latency (*ms*). The proposed model shows a better exchange of as far as latency as opposed to existing methodologies.

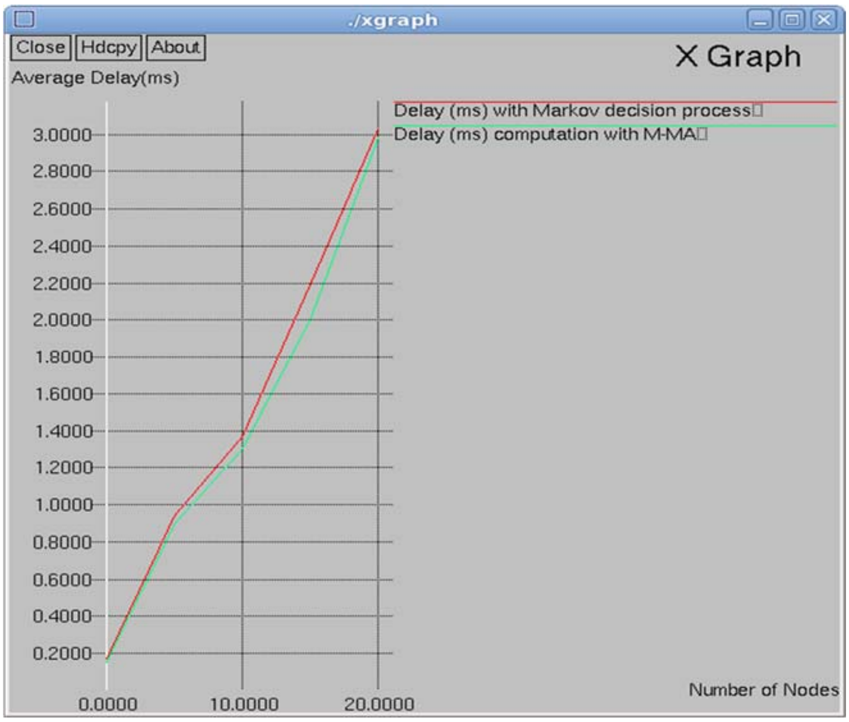

Fig. 3 Latency computation after node placement

Table II. Latency computation before node placement

S.No	No of nodes	Latency (ms) with the Markov decision process	Latency (ms) computation with M-MA
1	0	0.042	0.035
2	5	0.087	0.080
3	10	0.096	0.090
4	15	0.112	0.110
5	20	0.134	0.125

Table III. Latency computation after node placement

S.No	No of nodes	Latency (ms) with the Markov decision process	Latency (ms) computation with M-MA
1	0	0.042	0.035
2	5	0.087	0.080
3	10	0.096	0.090
4	15	0.112	0.110
5	20	0.134	0.125

Table II shows the evaluation of latency experienced before the node positioned in SD-WAN. A complete number of nodes 20 utilized for representation. Latency in node 0, 5, 10, 15, 20 is 0.035, 0.080, 0.090, 0.110 and 0.125 correspondingly.

Table III shows the evaluation of latency experienced after the node positioned in SD-WAN. The complete number of nodes is 20 utilized for representation. Latency calculation with M-MA in node 0, 5, 10, 15, 20 is 0.15, 0.90, 1.3, 2 and 2.98 correspondingly.

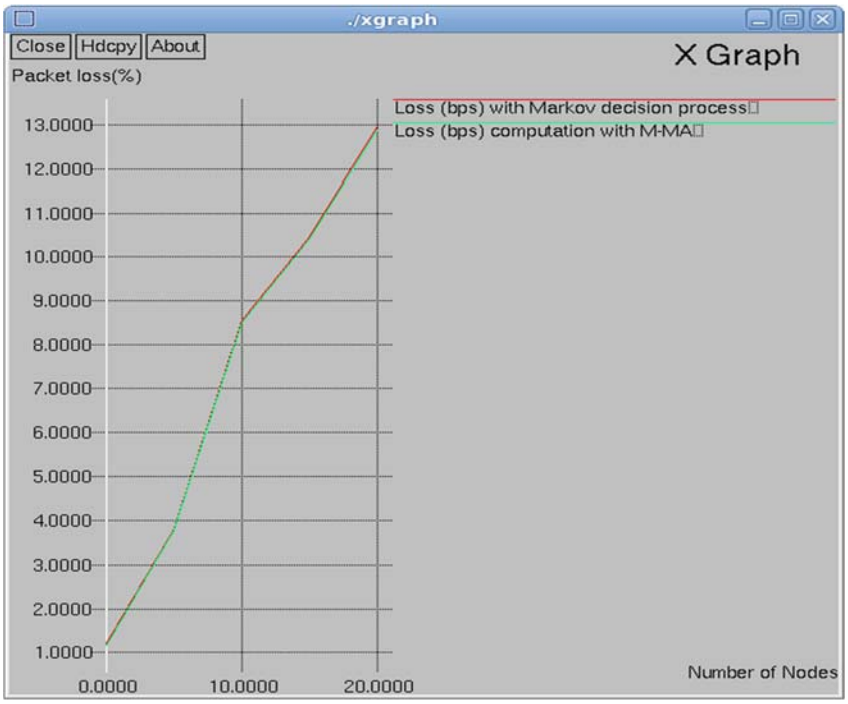

Fig. 4 Packet loss before node placement

Figure 6, Figure 7 delineates the graphical portrayal of the usual throughput calculation in Milliseconds. 'X' axis shows the number of controllers, and 'Y' axis shows the usual latency (ms). The proposed model shows a better exchange of as far as latency rather than existing methodologies.

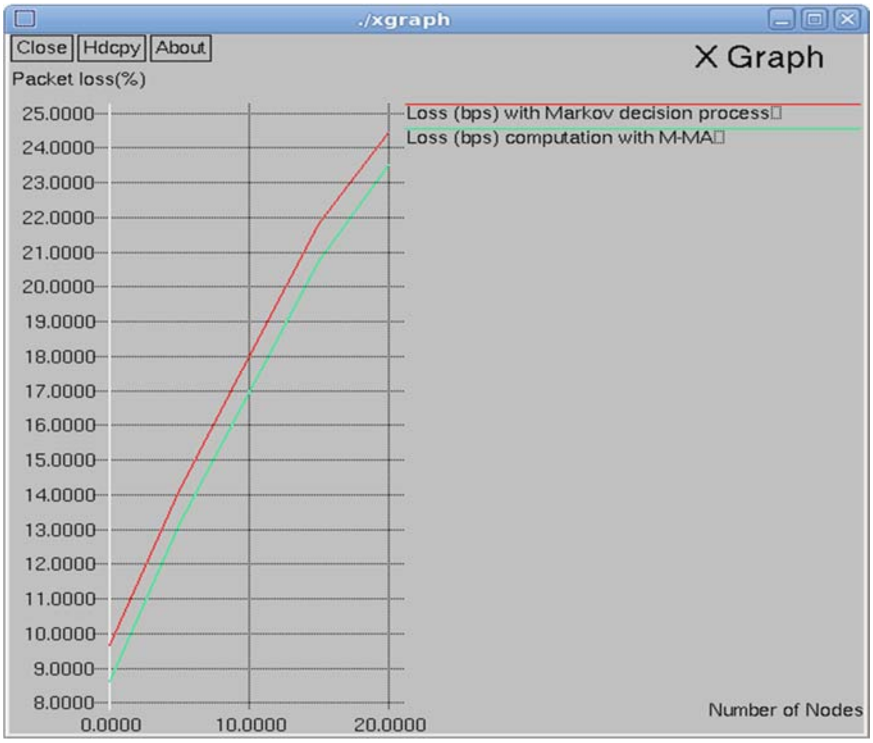

Fig. 5 Packet loss after node placement

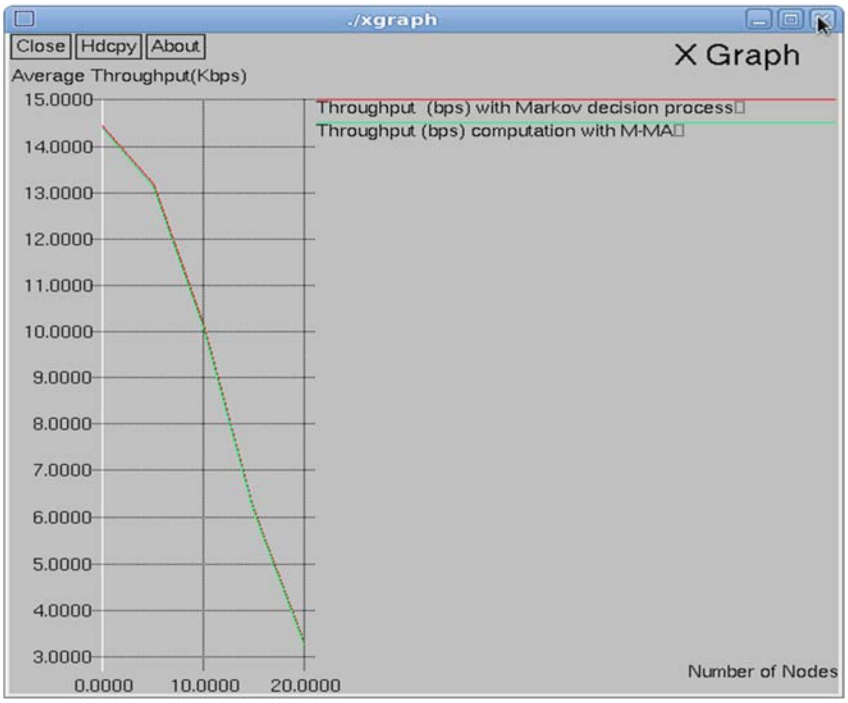

Fig. 6 Throughput before node placement

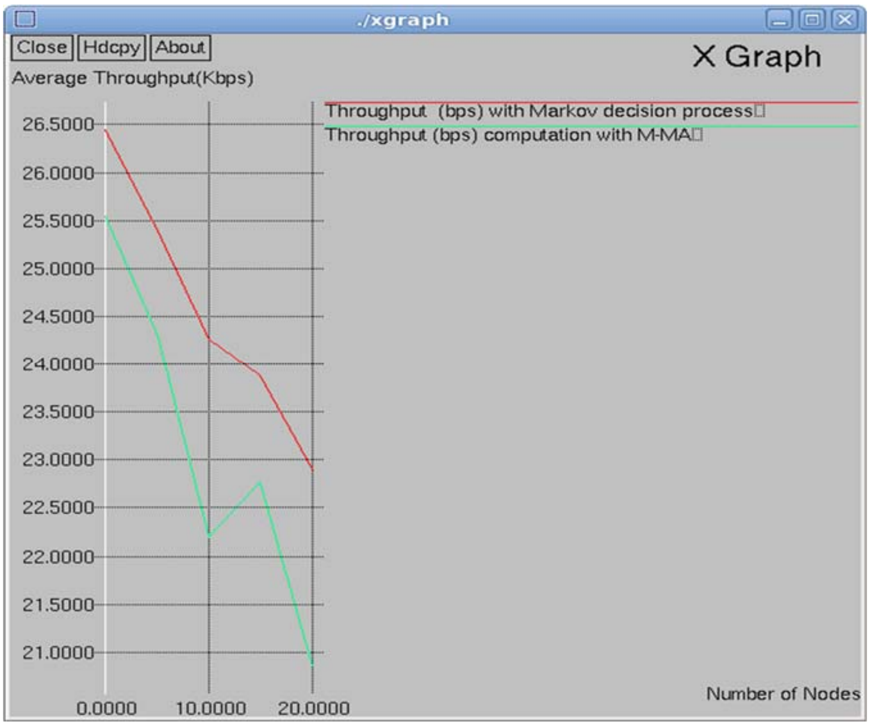

Fig. 7 Throughput after node placement

Table IV. Packet loss before node placement

S.No	No of nodes	Loss (bps) with Markov decision process	Loss (bps) computation with M-MA
1	0	1.20	1.15
2	5	3..82	3..80
3	10	8.56	8.50
4	15	10.47	10.40
5	20	12.98	12.90

Table V. Packet loss after node placement

S.No	No of nodes	Loss (bps) with Markov decision process	Loss (bps) computation with M-MA
1	0	9.67	8.60
2	5	14.14	13.15
3	10	17.98	16.95
4	15	21.82	20.75
5	20	24.46	23.50

Table IV shows the evaluation of packet loss experienced before the node situated in SD-WAN. The complete number of nodes is 20 utilized for representation. Loss in node 0, 5, 10, 15, 20 is 1.15, 3.80, 8.50, 10.40 and 12.90 correspondingly. Table V shows the correlation of packet loss experienced after node arrangement in SD-WAN. The complete number of nodes is 20 utilized for representation. Loss in node 0, 5, 10, 15, 20 is 8.60, 13.15, 16.95, 20.75 and 23.50 correspondingly.

Table VI. Throughput before node placement

S.No	No of nodes	Throughput (bps) with Markov decision process	Throughput (bps) computation with M-MA
1	0	14.45	14.40
2	5	13.18	13.10
3	10	10.17	10.10
4	15	6.21	6.15
5	20	3.31	3.25

Table VII. Throughput after node placement

S.No	No of nodes	Throughput (bps) with Markov decision process	Throughput (bps) computation with M-MA
1	0	26.45	25.54
2	5	25.39	24.30
3	10	24.26	22.20
4	15	23.88	22.77
5	20	22.90	20.85

Table VI shows the examination of throughput experienced before node arrangement in SD-WAN. The complete number of nodes is 20 utilized for representation. Throughput in node 0, 5, 10, 15, 20 is 14.40, 13.10, 10.10, 6.15 and 3.25 correspondingly. Table VII shows the examination of throughput experienced after node positioned in SD-WAN. The absolute number of nodes is 20 utilized for representation. Throughput in node 0, 5, 10, 15, 20 is 25.54, 24.30, 22.20, 22.77 and 20.85 correspondingly.

5. Conclusion

Here, Multi-agent based aggregation in controller deployment is considered in the hierarchical modeling of SD-WAN. Here, the Markov chain process is deemed to be evaluate load balancing, reliability, and request latency for hierarchical control architecture. Initially, Markov-based decision processing is put forward for network partitioning for delay optimization and balancing load between various partitions. Consider controlling plane-based availability and reliability in the control plane to deploy root controllers and domain controllers for SD-WAN. For balancing the load among control planes, the number of switches and flow size must be considered in every partition. The reliability of hierarchical control architecture based degree, path reliability, and flow rate of nodes have to be evaluated. Nodes representation results portray that the foreseen algorithm reduces average latency to some SD-WAN level, enhances load between various segments, and upgrades control plane unwavering quality. The anticipated algorithm acquires the most pleasing effect over efficiency. Various components must be considered during a genuine organization. Future work will be reached out to consider sending cost inside these networks, latency factors of the control plane, etc.

References

[1] Kreutz. D, Ramos. F, Verissimo. P, Rothenberg. C,Azodolmolky. S, Uhlig. S. "Software-defined networking: A comprehensive survey." Proceedings of the IEEE, Vol.103.1., 2015, pp. 891-921.

[2] Kshira Sagar Sahoo, Deepak Puthal, Mohammad S.Obaidat, Anamay Sarkar, Sambit Kumar Mishra, Bibhudatta Sahoo. "On the placement of controllers in software-Defined-WAN using meta-heuristic approach." Journal of Systems and Software, Vol.145., 2018, pp. 180-194.

[3] Yeganeh. S.H, Ganjali. Y. "Kandoo: a framework for efficient and scalable offloading of control applications." Proceedings of the first workshop on hot topics in software defined networks, ACM, 2012, pp. 19-24.

[4] Pankaj Berde, Matteo Gerola, Jonathan Hart, Yuta Higuchi, Masayoshi Kobayashi, Toshio Koide, Bob Lantz, Brian O'Connor, Pavlin Radoslavov, William Snow, Guru Parulkar. "ONOS: Towards an Open, Distributed SDN OS.", HotSDN '14: Proceedings of the third workshop on Hot topics in software defined networking, ACM, 2014, pp. 1-6.

[5] Liu, J. Liu, and R. Xie. "Reliability-based controller placement algorithm in software defined networking." Computer Science and Information Systems, Vol.13.2, 2016, pp. 547-560.

[6] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jonathan Zolla, Urs Hölzle, Stephen Stuart and Amin Vahdat. "B4: Experience with a Globally-Deployed Software Defined WAN." ACM SIGCOMM Computer Communication Review, 2013.

[7] Diego Kreutz, Fernando M. V. Ramos, Paulo Verissimo, Christian Esteve Rothenberg, Siamak Azodolmolky, Steve Uhlig. "Software-Defined Networking: A Comprehensive Survey." Proceedings of the IEEE, 2014, Vol. 103.1.

[8] Mourad Soliman, Biswajit Nandy, Ioannis Lambadaris, Peter Ashwood-Smith. "Exploring source routed forwarding in SDN-based WANs." IEEE International Conference on Communications (ICC), 2014.

[9] Yossi Kanizo, David Hay, Isaac Keslassy. "Palette: Distributing Tables in Software-Defined Networks." Proceedings IEEE INFOCOM, 2013.

[10] Nanxi Kang, Zhenming Liu, Jennifer Rexford, David Walker. "Optimizing the "One big switch" abstraction in software-defined networks." Proceedings of the ninth ACM conference on Emerging networking experiments and technologies, 2013.

[11] Akram Hakiri, Aniruddha Gokhale, Pascal Berthou, Douglas C. Schmidt, Gayraud Thierry. "Software-defined Networking: Challenges and Research Opportunities for Future Internet." Comput. Networks, Vol. 75, 2014, pp. 453- 471.

[12] Sushant Jain, Alok Kumar, Subhasree Mandal, Joon Ong, Leon Poutievski, Arjun Singh, Subbaiah Venkata, Jim Wanderer, Junlan Zhou, Min Zhu, Jonathan Zolla, Urs Hölzle, Stephen Stuart, Amin Vahdat. "B4: Experience with a Globally-Deployed Software Defined WAN." Proceedings of Conference in Special Interest Group Data Communication. (SIGCOMM), 2013.

[13] Angelos Mimidis Kentis, Artur Pilimon, José Soler, Michael S. Berger, Sarah Ruepp. "A Novel Algorithm for Flow-Rule Placement in SDN Switches." Proceedings of 4th IEEE Conference on Network Softwarization and Workshops (NetSoft), 2018.

[14] Bu-Sung Lee, Renuga Kanagavelu, Khin Mi Mi Aung. "An efficient flow cache algorithm with improved fairness in Software-Defined Data Center Networks." Proceedings of IEEE 2nd International Conference on Cloud Networking (CloudNet), 2013.

[15] Bing Xiong, Rengeng Wu, Jinyuan Zhao, Zhuofan Liao, Jin Wang. "Efficient Differentiated Storage Architecture for Large-scale Flow Tables in Software-Defined Wide-Area Networks." Proceedings for 10th USENIX Symposium in Networked System Design Implementation(NSDI), 2013.

[16] David Hock, Steffen Gebert, Matthias Hartmann, Thomas Zinner, Phuoc Tran-Gia. "POCO-framework for Pareto-optimal resilient controller placement in SDN-based core networks." Proceedings for IEEE Network Operations and Management Symposium (NOMS), 2013.

[17] Adlen Ksentini, Miloud Bagaa, Tarik Taleb, Ilangko Balasingham. "On using bargaining game for Optimal Placement of SDN controllers." Proceedings for IEEE International Conference on Communications (ICC), 2016.

[18] Yury Jiménez, Cristina Cervelló-Pastor, Aurelio J. García. "On the controller placement for designing a distributed SDN control layer." Proceedings for IFIP Networking Conference, 2014.

[19] Afrim Sallahi, Marc St-Hilaire. "Optimal Model for the Controller Placement Problem in Software Defined Networks." IEEE Communications Letters, Vol. 19.1., 2015, pp. 30-33.

[20] Anderson Santos da Silva, Paul Smith, Andreas Mauthe, Alberto Schaeffer-Filho. "Resilience support in software-defined networking: A survey." Computer Networks, Vol. 92.1, 2015, pp. 189-207.

[21] Afrim Sallahi, Marc St-Hilaire. "Expansion Model for the Controller Placement Problem in Software Defined Networks." IEEE Communications Letters, Vol. 21.2, 2017, pp. 274–277.

[22] Daphne Tuncer, Marinos Charalambides, Stuart Clayman, George Pavlou. "Adaptive Resource Management and Control in Software Defined Networks." IEEE Transactions On Network And Service Management, Vol. 12.1, 2015, pp. 18–33.

[23] Bang Zhang, Xingwei Wang, Lianbo Ma, Min Huang. "Optimal Controller Placement Problem in Internet-Oriented Software Defined Network." Proceedings for International Conference on Cyber-Enabled Distributed Computing and Knowledge Discovery (CyberC), 2016.

[24] Jianxin Liao, Haifeng Sun, Jingyu Wang, Qi Qi, Kai Li, Tonghong Li. "Density cluster based approach for controller placement problem in large-scale software defined networkings." Computer Networks: The International Journal of Computer and Telecommunications Networking, Vol. 112.C, 2017, pp. 24–35.

[25] Guodong Wang, Yanxiao Zhao, Jun Huang, Robb M. Winter. "On the Data Aggregation Point Placement in Smart Meter Networks." Proceedings for 26th International Conference on Computer Communication and Networks (ICCCN), 2017.