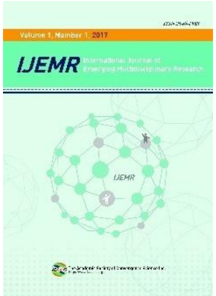

International Journal of
Emerging Multidisciplinary Research

ISSN : 2546-1583
Volume 4, Number 2, June 2020
https://doi.org/10.22662/IJEMR.2020.4.2.001

MLMBN mechanism optimizes network load balance using information with multiple controllers

Y. S. Jeong[1]
[1]Department of Information and Communication Convergence Engineering, Mokwon University, Daejeon, CO 35349, Korea

Abstract
Background/Objectives: In this paper, we propose a Deep Learning Mechanism (DLMBN) mechanism (Deep Learning Mechanism on Blockchain) that optimizes the load balance that can occur in the network by deep learning some important information related to the load balance after connecting the information of multiple distributed controllers into the blockchain. Methods/Statistical analysis: The proposed mechanism binds and manages the load of each controller distributed over the network with a blockchain, thus reducing load time while dynamically balancing the load balance. In particular, deep learning technology was used to ensure that each controller classified as a group would not be biased to one side and would maintain a balanced load balance across the entire network. Findings: As a result of the experiment, the proposed mechanism improved the load balance retention time by 14.6% on average compared to the mechanism previously studied, and the efficiency of SDNs processed in multiple groups by 17.3% on average. In addition, the overhead of SDNs for each group was lowered by 7.9%. Improvements/Applications: Based on the results of this study, future studies plan to apply the results to the actual network and test whether the performance analysis results are applicable to heterogeneous networks consisting of heterogeneous devices.

Index Terms
Distributed Network, Machine Learning, Block-chain, Load-Balancing, Performance Improvement

Corresponding author : Y. S. Jeong
bukmunro@gmail.com

• Manuscript received April 10, 2020.
• Revised May 1, 2020 ; Accepted June 1, 2020.
• Date of publication June 30, 2020.

I. INTRODUCTION

Recently, networks have diversified approaches to network resources and storage due to different types of organizations and devices[1]. In particular, the recent exponential growth of tablets and cell phones has transformed the work of institutions and managers into an environment that individuals can handle. Companies such as Amazon, Google, Microsoft and Rackspac, which also offer cloud services, are offering cloud services individually to institutions or individuals[2-3].

Companies that provide the cloud are researching and developing more efficient and easier-to-manage ways to maintain a load balance on the network of institutions or individuals that require cloud services through a number of devices[4]. Among them, virtualization and multi-tenancy are technologies that allow all network resources to be shared with each other, and are used to efficiently utilize network available resources. And Software Defined Networking (SDN) is a network concept that helps increase network automation, network program possibilities and innovation, and is used for rapid service deployment and virtualization to help network dynamic configuration, policy updates and resource management.

There are still many unresolved aspects of virtual machine (VM) microphone support, although many studies and technologies have been developed regarding the network to date[5-8]. Heterogeneous networks that process big data dynamically switch to other networks for access, adjusting the load balance for network throughput and performance.

Prior to SDN technology, load balancing techniques were performed only in the local area due to the limitations of the network architecture, but today multiple vendors that supply SDNs jointly provide a data plane platform with an API abstraction layer, reducing the complexity of network configuration and operations[9]. The SDN infrastructure significantly improves the user experience using specific applications so that new protocols and network services can be easily deployed to users due to high operational abstractions.

In this paper, we propose a Deep Learning Mechanism (DLMBN) mechanism to optimize the network by deep learning some important information related to load balance after connecting information from multiple distributed controllers to blockchain in order to optimize the load balance of different network devices operating in heterogeneous environments. The proposed mechanism has features that minimize the delay caused by the network process because it binds and manages the load of each controller distributed over the network with a blockchain. In addition, the SDNs were classified into multiple groups to dynamically organize networks to balance load balance, and then deep-learning SDN information so that each controller could maintain a balanced load balance across the entire network without being biased to one side. The proposed mechanism periodically checks each controller to bind the information (handles and storage load information, etc.) of each controller with a block chain so that overloaded controllers in the region are not collected. In particular, the proposed mechanism is grouping SDNs into blockchain, reducing load frequency to reduce possible processing and communication overhead in the network load balance.

The composition of this paper is as follows. In Chapter 2, we learn about Software Defined networking and existing research. Chapter 3 proposes a Deep Learning Mechanism (DLMBN) mechanism that optimizes network load balance, while Chapter 4 compares and evaluates the evaluation of proposed techniques with existing techniques. Finally, I conclude in chapter 5.

II. RELATED WORKS

A. Software defined network(SDN)

SDN refers to a network virtualization approach to optimize network resources and quickly adapt the network to changing business requirements, applications and traffic. SDN is implemented by separating the network control flane from the data flane and creating a software programmable infrastructure that is distinct from the physical equipment [10-13].

With SDN, the functions of network orchestration, management, analysis and automation become the work of SDN controllers. These controllers are not networking equipment, so they can benefit from scalability, performance, modern cloud computing, and the availability of storage resources. More and more SDN controllers are being built on open platforms using open standards and open APIs, enabling orchestration, management and control of network equipment purchased from different vendors [14].

SDN brings a variety of business benefits. Separation of control and transmission layers is increasing flexibility and speeding up the launch of new applications. The ability to respond more quickly to problems and outages is increasing network availability. And programming capabilities enable IT organizations to reduce operating costs by simply automating network functions.

SDN also works well with another technology, Network Functions Virtualization (NFV). NFV offers the ability to virtualize appliance-based network features such as firewalls, load balancers and WAN accelerators. Centralized control provided by SDN enables efficient management and orchestration of virtual network functions enabled by NFV [15].

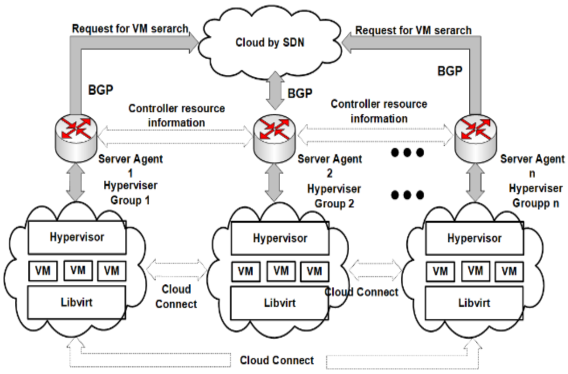

Fig. 1. Software defined network

B. Previous Research

Various studies are underway regarding network load balancing. In particular, to address traffic engineering problems and network scalability problems, it is implemented as a centralized and distributed approach to the network based on SDN [16-18].

Randles et al. [19] dynamically analyzes load balancing of different network types, enabling relative analysis of load balancing in the network. Inspired by the behavior of the biological bees, Randles et al. has organized the services of the data center into multiple virtual servers to calculate the request value of the server that needs the service, if the server is interested, to process the request and otherwise return to the scout action.

Hu et al. [20] proposed an algorithm to virtualize the network and distribute loads between virtual machines. The proposed algorithm optimizes operating costs by selecting the virtual machine with the least request processing. However, the disadvantage is that increasing the number of requests will take longer and longer to process virtual machines.

Achar et al. [21] proposed an algorithm to maintain load balancing using network traffic. This algorithm used Xen technology, enabling dynamic allocation of network resources at the request of users. However, because the algorithm periodically examines the resources of RAM and CPU, it has the problem of wasting resources even when there is no service request.

Zhao et. al [22] proposed an algorithm that works according to sampling to reduce the time it takes to migrate between VM(Virtual Machine)s. This algorithm is characterized by reduced downtime relocation and VM-to-VM migration with shared storage.

Ashwin et. al [23] has proposed a PSO-based algorithm that can effectively allocate VMs using cloud computing technology. This algorithm minimizes the load on the server by requiring the allocation of resources based on the state of the VM.

III. MLMBN MECHANISM OPTIMIZED NETWORK LOAD BALANCING

A. Network

Networks that use SDNs are Fig. It has a structure in which SDN allocates and distributes network resources at service requests from different devices in the same form as 2. In the proposed mechanism, Fig. In order to reliably optimize network-wide load balancing in network environments such as 2, the network is optimized by linking controller information from SDN to blockchain and then deep learning some important information related to load balancing. Since the proposed mechanism binds and manages each controller distributed over the network with a blockchain, it finds a controller that causes a delay and requests the service minutes from the controller in Idel state. In addition, the proposed mechanism deep-learning information related to load balancing among SDN information to ensure that SDNs classified into multiple groups are dynamically load balanced so that each controller is not biased to one side.

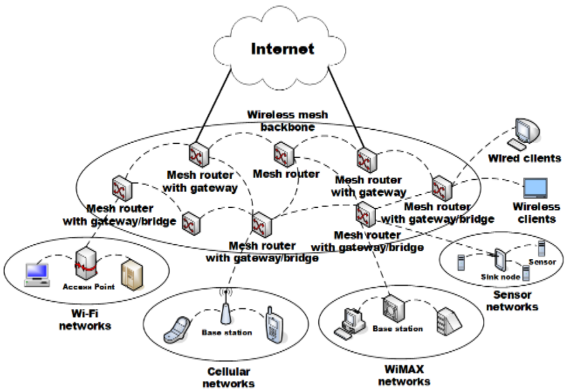

Fig. 2. SDN based Network

As Fig. 2, the proposed mechanism periodically checks each controller to reassemble each controller into a block chain based on status information, such as handle and storage load, to reduce the processing and communication overhead of locally overloaded controllers to maintain load balance across the entire network.

B. Blockchain-based SDN Controller Configuration

The controls on the SDNs that make up the network are configured to connect to each other by blockchain, as shown in Figure 3. Each configured controller delivers important information through the SDN server. At this time, the SDN server checks the similarity of each controller's critical connection information and selects a seed. The proposed mechanism selects the link information for each controller that constitutes the network as a seed and binds similar information for controllers on different SDNs with hashchains.

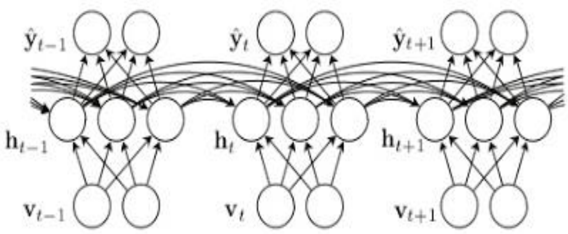

Fig. 3. Configure blockchain-based controller connections

The proposed mechanism selects controllers selectively according to different SDN networks to group linked information hierarchically according to the similarity of controllers. At this time, the network server stores controller information from the SDN connected by the blockchain in the database and compares it with the existing stored information to adjust the load balance.

C. Create controller information

SDN generates information from each controller as shown in Expression (1) to balance the load of the entire network using controller information processed in a heterogeneous network environment. Information processed by each controller, such as Expression (1), is sampled with information related to network resources using N-1 polynomials. The reason for using this polynomial of N-1 is to sample information that the overall load balance of the SDN network does not maintain.

$$CI_x = \begin{cases} i_1 + i_2x^1 + \cdots + i_nx^{n-1} & , if\ i, n > 1 \\ 1 & , otherwise \end{cases} \quad (1)$$

Where, n means the total number of information processed by each controller. i means SDN network resource information.
The IBD of the controller placed on the network through Expression (1) generates the information of the controller distributed to the SDN as shown in Expression (2).

$$\text{SDNCI}_n = \sum_1^n CI_n \mod Z \quad (2)$$

Expression (2) distributes the information of each controller in the SDN network and overlaps the information Z(=$CI_1 + CI_2 + \cdots + CI_n$) to identify the information of each controller. The reason for this overlap is to ensure that information on each controller is treated as lossless.

IV. Performance Evaluation

A. Experimental Environment

The proposed mechanism used MatLab to conduct the experiment, and the network configuration was Pig. The same environment was formed as in 4.

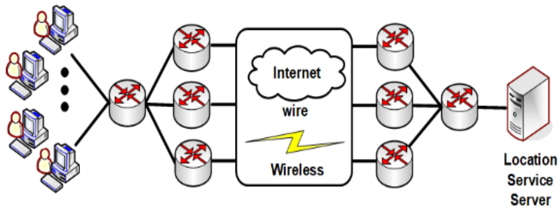

Fig. 4. Network for Experimental Environments

Table 1 shows the setting values of the parameters used in the experimental environment.

Table 1. Parameters for Simulation

Parameter	Value
Network topology	Mesh topology
Number of server	1
Number of Controller	1-50
Number of relay SND	5
Link capacity	1 Gbps
Link round trip delay	10ms
Internet Timeout Timer	500ms
Number of SND	1, 2, 4, 8, 10
Number of Controller Information	1, 5, 10, 25, 50, 100
Number of Property	1, 2, 3, 4, 5
Buffer	40 packet
Traffic	5 pkts/s

B. Load Balance Retention Time

Load Balance retention time has achieved the same results as Fig 5 by linking important information about SDN's controllers to the entire network. As a result of the performance evaluation in Fig. 5, the average time to maintain balance was 14.6% better than the model without the control of SDN as a result of linking the SDN's controller to each other. This is the result of the proposed mechanism being applied to the entire network after comparing the similarity of control information in the SDN.

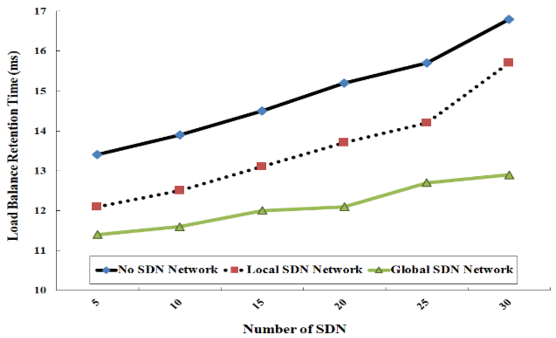

Fig. 5. Load Balance Retention Time

C. Efficiency

In order to evaluate the efficiency of the controller in the SDN network, the important information processed by the controller was extracted and the efficiency of the load balance of the network was assessed, resulting in the same result as Fig. 6. As a result of Figure 6, the proposed mechanism improved the efficiency of SDNs processed in multiple groups by an average of 17.3% when blockchain-based deep learning technology was applied when processing information extracted from controllers on servers. These results are due to the use of the controller's linked information as the seed value of the hierarchical subnet as a probability value.

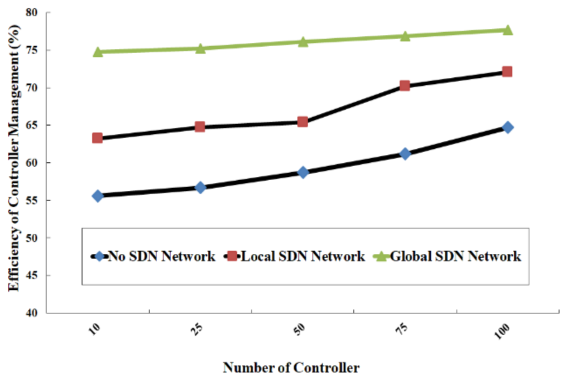

Fig. 6. Efficiency of Controller Management

D. Overhead

The evaluation of the overhead change of SDN-configured networks in heterogeneous network environments resulted in the same result as Fig. 6. The overhead of each group's SDN was 7.9% lower on average compared to the existing algorithm because the probability-linked information of the controller was applied to deep learning. The result was that probability-linked information about critical information in the controller was seeded by the server.

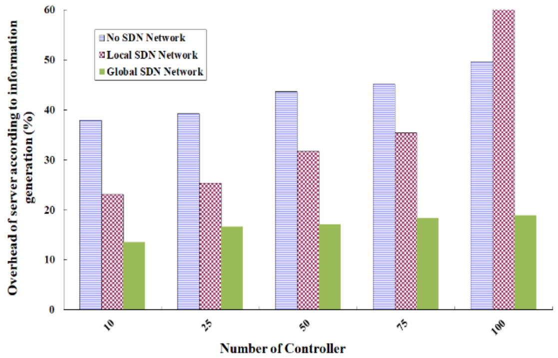

Fig. 7. Overhead of server according to information generation

V. Conclusion

Recently, various heterogeneous devices are receiving data services in the network environment, so network efficiency is being valued. In particular, network processing technologies for easy-to-carry devices such as mobile phones and tablets are required.

Most network management technologies are used to optimize the performance of the network, such as energy consumption and communication delays, but there are still many problems to be In this paper, we proposed a Deep Learning Mechanism on Blockchain Network (DLMBN) mechanism that can optimize load balancing of the entire network by managing the information of controllers located in multiple distributed networks.

The proposed mechanism has improved the performance of the network by linking information from distributed controllers to blockchain and then deep learning some important information related to load balance. In addition, the proposed mechanism dynamically balances the load on a blockchain basis to reduce the load on each controller distributed over the network.

In particular, the deep learning technology was applied to each of the grouped controllers so that they could maintain a balanced load balance across the entire network without being biased against a specific network area.

As a result of the experiment, the proposed mechanism improved the load balance retention time by 14.6% on average compared to the mechanisms previously studied, and the efficiency of SDNs processed in multiple groups by 17.3% on average. In addition, the overhead of SDNs for each group was lowered by 7.9%. Based on the results of this study, future studies plan to apply the results to the actual network and test whether the performance analysis results are applicable to heterogeneous networks consisting of heterogeneous devices.

REFERENCES

[1] Wang R., Bidkar S., Meng F., Nejabati R., Simeonidou D. (2019). Load-aware nonlinearity estimation for elastic optical network resource optimization and management, *IEEE/OSA Journal of Optical Communications and Networking*, 11(5), 164-178

[2] Li J., Wang Y., Wu Z, Feng S. Qiu X. (2017). A prediction-based dynamic resource management approach for network virtualization, *Proceedings of the 2017 13th International Conference on Network and Service Management (CNSM)*, 1-5.

[3] Mosahebfard M., Vardakas J. Remantas K., Verikoukis C. (2019). SDN/NFV-Based Network Resource Management for Converged Optical-Wireless Network Architectures, *Proceedings of the SDN/NFV-Based Network Resource Management for Converged Optical-Wireless Network Architectures*, 1-4.

[4] Feng Z., Sun W., Zhu J., Shao J. Hu W. (2017). Resource allocation in electrical/optical hybrid switching data center networks, *IEEE/OSA Journal of Optical Communications and Networking* ,9(8), 648-657.

[5] Thar K., Tran Nguyen H., Son J. Hong C. S. (2016). Resources management in virtualized Information Centric Wireless Network, *Proceedings of the 2016 18th Asia-Pacific Network Operations and Management Symposium (APNOMS)*, 1-4.

[6] Zhang P., Yao H, Liu Y. (2018). Virtual Network Embedding Based on Computing, Network, and Storage Resource Constraints, *IEEE Internet of Things Journal*, 5(5), 3298-3304.

[7] Zhang P., Yao H., Liu Y. (2016). Virtual network embedding based on the degree and clustering coefficient information, *IEEE Access*, 4, 8572–8580.

[8] Ogino N., Kitahara T., Arakawa S., Murata M. (2017). Virtual network embedding with multiple priority classes sharing substrate resources, *Computing Networks*, 112, 52–66.

[9] Zhang P., Yao H., Fang C., Liu Y. (2016). Multi-objective enhanced particle swarm optimization in virtual network embedding, *EURASIP Journal on Wireless Communications and Networking*, 2016(167), 1–9.

[10] Gharbaoui M., Contoli C., Davoli G., Cuffaro G., Martini B., Paganelli F., Cerroni W., Cappanera P. Castoldi P. (2018). Demonstration of Latency-Aware and Self-Adaptive Service Chaining in 5G/SDN/NFV infrastructures, *Proceedings of the 2018 IEEE Conference on Network Function Virtualization and Software*, pp. 1-2.

[11] Tatang D., Quinkert F., Frank J., Ropke C., Holz T. (2017). SDN-GUARD: Protecting SDN Controllers Against SDN Rootkits, *Proceedings of the 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 297-302.

[12] Baddeley M., Nejabati R., Oikonomou G., Gormus S., Sooriyabandara M., Simeonidou D. (2017). Isolating SDN control traffic with layer-2 slicing in 6TiSCH industrial IoT networks, *Proceedings of the 2017 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 247-251.

[13] Baik S., Lim Y., Kim J., Lee Y. (2015). Adaptive flow monitoring in SDN architecture, *Proceedings of the 2015 17th Asia-Pacific Network Operations and Management Symposium (APNOMS), 468-470*

[14] Raghunath K., Krishnan P. (2018). Towards A Secure SDN Architecture, *Proceedings of the 2018 9th International Conference on Computing, Communication and Networking Technologies (ICCCNT)*, 1-7.

[15] Flathagen J., Mjelde Terje M., Bentstuen Ole I. (2018). A combined Network Access Control and QoS scheme for Software Defined Networks, *Proceedings of the 2018 IEEE Conference on Network Function Virtualization and Software Defined Networks (NFV-SDN)*, 1-6.

[16] Aslanzadeh S., Chaczko Z. (2015). Load Balancing Optimization In Cloud Computing: Applying Endocrine-Particle Swarm Optimization, *Proceedings of the 2015 IEEE International Conference On Electro/Information Technology (Eit)*, 165-169.

[17] Wen Wt., Wang Cd., Wu Ds., Xie Yy. (2015). An Aco-Based Scheduling Strategy On Load Balancing In Cloud Computing Environment, *Proceedings of the 2015 Ninth International Conference On Frontier Of Computer Science And Technology*, 364-369.

[18] Babu Kr, Joy Aa, Samuel P. (2015). Load Balancing Of Tasks In Cloud Computing Environment Based On Bee Colony Algorithm, *Proceedings of the 2015 Fifth International Conference On Advances In Computing And Communications (Icacc)*, 89-93.

[19] Randles, M., Lamb,andTaleb-Bendiab, A. (2010). A comparative study into distributed load balancing algorithms for cloud computing. *Proceedings of the IEEE 24th International Conference on Advanced Information Networking and Applications Workshops (WAINA)*, 551-556.

[20] Hu, J., Gu, J., Sun, G. and Zhao, T. (2010). A scheduling strategy on load balancing of virtual machine resources in cloud computing environment. *Proceedings of the 2010 Third International Symposium on Parallel Architectures, Algorithms and Programming (PAAP)*, 89-96.

[21] Achar R., Thilagam Ps, Soans N. Vikyath P. V., Rao S. Vijeth A. M. (2013). Load Balancing In Cloud Based On Live Migration Of Virtual Machines, *Proceedings of the 2013 Annual IEEE India Conference (Indicon)*, 1-5.

[22] Zhao Y., Huang W. (2009). Adaptive Distributed Load Balancing Algorithm Based On Live Migration Of Virtual Machines In Cloud, *Proceedings of the Fifth International Joint Conference on INC, IMS and IDC(INC 2009)*, 170-175.

[23] Ashwin Ts., Domanal Sg., Guddeti Rm. (2014). A Novel Bio-Inspired Load Balancing Of Virtual Machines In Cloud Environment", *Proceedings of the 2014 IEEE International Conference On Cloud Computing In Emerging Markets (Ccem)*, 1-4.