

Using fuzzy control routing for dynamic load balancing over Software-Defined Networks

Wen-Shyang Hwang

*Department of Electrical Engineering, National Kaohsiung University of Science and Technology,
Kaohsiung 807, Taiwan; Email: wshwang@nkust.edu.tw*

Ming-Syuan Wu*

*Department of Electrical Engineering, National Kaohsiung University of Science and Technology,
Kaohsiung 807, Taiwan; Email: i111154101@nkust.edu.tw*

Cheng-Han Lin

Department of Health-Business Administration, Fooyin University, Kaohsiung 831, Taiwan

Yan-Jing Wu

Department of Information Technology and Communication, Shih Chien University, Kaohsiung 807, Taiwan

Ming-Hua Cheng

Department of Digital Media Design, Tzu-Hui Institute of Technology, Pingtung, 926, Taiwan

Sian-Fong Huang

*Department of Electrical Engineering, National Kaohsiung University of Science and Technology,
Kaohsiung 807, Taiwan*

Abstract

The load-balancing and forwarding mechanisms of traditional software-defined networks (SDNs) often rely on static path selection. Conventional mechanisms lead to uneven resource utilization and bottlenecks under high-traffic conditions. Therefore, the existing methods require more flexibility in decision-making. This study introduces a load-balancing algorithm based on fuzzy logic to address the difficulty of setting multivariable input thresholds. The algorithm uses fuzzy logic to combine the network parameters (queue length, link utilization, link delay, and packet loss rate) as fuzzy inputs. Converting the fuzzy input into a score is the key to achieving optimal traffic allocation in the mechanism. The proposed mechanism improves resource utilization and reduces bottlenecks and reliability in dynamic network-load situations. The results indicated that the proposed method achieved a higher throughput under high-load conditions. Moreover, the results maintained low packet loss and delay rates. Load balancing based on fuzzy logic provides an effective solution for SDN environments.

Keywords: Software Defined Network, fuzzy logic, load balancing

1. Introduction

Software-defined networks (SDNs) [1] have become an essential part of modern network architecture. SDNs makes the network management and configuration more flexible and efficient by separating the control plane from the data plane. However, load balancing remains a critical and challenging issue in an SDN implementation. Most existing load-balancing mechanisms are based on static configurations, which have many problems and limitations in practical applications. Static load-balancing mechanisms usually set fixed paths for data forwarding, which are simple and limited in a dynamically changing network environment. As network traffic changes, fixed-path configurations cannot adapt to real-time needs, causing some paths to be overloaded, while other paths remain idle. Rough network resource utilization significantly affects the network performance. Particularly, in high-traffic situations, single-path selection can create bottlenecks, causing network congestion and increasing latency. Static load balancing lacks flexibility and cannot dynamically adjust path selection based on the real-time network status. Because traffic patterns change and links fail, the system cannot promptly adapt, thereby reducing network reliability and stability. In addition, the existing SDN load-balancing mechanisms require greater flexibility in the decision-making process. This study proposes a load-balancing mechanism to improve the flexibility and adaptability of

the load-balancing mechanism through fuzzy logic. The remainder of this paper is organized as follows: Section 2 introduces the background and related work. Section 3 describes our research methodology. Section 4 describes the evaluation and experiments. Finally, Section 5 presents conclusions and future work.

2. Backgrounds and related works

2.1. Fuzzy logic theory and Equal-cost multi-path (ECMP)

Fuzzy logic is the mathematical logic theory [2]. This allows the processing of uncertainty and fuzziness by introducing the concept of a fuzzy set. Unlike traditional binary logic, fuzzy logic allows partial truth values. The variables range from 0 to 1.

The fuzzy-control process consists of three stages: input, processing, and output. These stages form a fuzzy inference system that manages uncertain and fuzzy data for effective control and decision making. The Mamdani fuzzy inference system, one of the earliest types, includes four substages: fuzzification, inference engine, fuzzy rules, and defuzzification. ECMP [3] is a load-balancing algorithm that evenly distributes traffic across multiple equal-cost paths. However, ECMP cannot improve network resource utilization and performance.

2.2. An effective routing mechanism based on fuzzy logic for software-defined data center networks

Fuzzy logic intelligently evaluates data packet paths and is primarily used in data center networks. The mechanism considers factors such as hop count and bandwidth utilization. In [4], the model converted these factors into fuzzy variables by assigning a corresponding label to each variable. With an essential fuzzy variable set, a set of fuzzy rules is designed to control the path selection based on network requirements. These rules make precise decisions for the SDN controller, whose path must be selected under specific network conditions. The decision-making process selects routes using multiple indicators and identifies the best path for effective load balancing.

3. Fuzzy logic load balancing algorithm

3.1. System architecture

In the experiment, a Ryu controller and Mininet simulator were selected to build the SDN simulation environment. Fig. 1 shows the operation process of the fuzzy logic load-balancing algorithm (FLLB) in an SDN network.

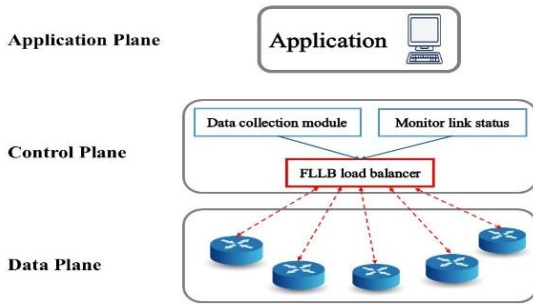


Fig. 1 System architecture

3.2. FLLB mechanism operation

The operation process of the FLLB algorithm is suitable for SDNs. The system was designed to select links dynamically for efficient load balancing and network performance optimization. The link-screening process evaluates the load score (L_i) and quality score (Q_i) of each link using a fuzzy logic system, which confirms that the selected link can reach the destination in the FLLB mechanism. The connection control module generates new flow-table rules based on the final decision results, which are dynamically delivered to the switch to ensure that the data packets are transmitted along the best link. After calculating the comprehensive score (S_i) of each link, link selection is performed based on the score. First, it is judged whether S_i is greater than or equal to 0.5. If the value was less than 0.5, the link was excluded as a candidate link. If S_i of the candidate links is less than or equal to 1, then the original link is followed for traffic forwarding. If S_i of the candidate links exceeds 1, the link with the highest comprehensive score is selected. After

selecting the best link, we must determine whether the link can reach the destination. Otherwise, the candidate link with the second highest comprehensive score is chosen.

3.2.1 Fuzzy logic load balancer

In the experiment, the collected information was passed to a fuzzy-logic load balancer in the FLLB mechanism. The load balancer applies fuzzy logic to assess the link feasibility and effectiveness. It selects optimal links based on these evaluations.

The queue length, link utilization, delay, and packet loss rate are the fuzzification input variables. These factors determine the link-load capacity and quality. Queue length is the wait time for the switch port to process a packet. The fuzzifying queue length addresses the uncertainty in network conditions. Link utilization indicates the traffic load as a percentage of the maximum capacity, which directly reflects the resource usage status. The queue length and link utilization were used as fuzzification input variables. According to [5], the fuzzification of these two variables can be improved to obtain a more accurate L_i , which can evaluate the load status of the link more comprehensively.

Link latency is the time required for a packet to transition from a source node to a destination node on a specific link, which directly reflects the transmission performance of the link. The link delay is used as a fuzzy input variable because it can reflect link performance and predict link congestion. The packet loss rate directly reflects link reliability during transmission. The packet loss rate and link delay are the fuzzification inputs for obtaining Q_i . This score reflects link reliability and performance as well as simplified decision-making. According to [6] and [7], the equation for calculating the fuzzification of each input parameter is: Fuzzification generates a fuzzy input for each input using a membership function. The inference engine produces nine fuzzy outputs based on the L_i rule table. Q_i rule table produced six fuzzy outputs, as illustrated in Fig. 2.

Load Score (L_i)			Quality Score (Q_i)		
Queue Length	Link Utilization	Output	Delay	Packet Loss Rate	Output
Low	Low	VL	Low	Low	H
Low	Medium	L	Low	Medium	MH
Low	High	ML	Low	High	M
Medium	Low	L	High	Low	ML
Medium	Medium	M	High	Medium	L
Medium	High	MH	High	High	VL
High	Low	ML			
High	Medium	MH			
High	High	H			

Fig. 2 Fuzzy inference engine rule table

The fuzzy output generated by the inference engine must be converted into a crisp output through the defuzzification process. The center-of-gravity method is a standard defuzzification method, as expressed in Eq. (1). The final result was determined by calculating the center points of all overlapping areas, ensuring that each

overlapping area was calculated only once. This defuzzification method accurately reflected the output characteristics of the system. In Eq. (1), x_i is the value of the output variable, $x_i\mu$ is the membership degree corresponding to x_i , and N is the number of values of the output variable.

$$\text{Defuzzification}(x) = \frac{\sum_{i=1}^N x_i \mu(x_i)}{\sum_{i=1}^N \mu(x_i)} \quad (1)$$

3.2.2 Comprehensive link score calculation

In the FLLB algorithm, the calculation of S_i is the key to determining the optimal traffic distribution strategy. The calculation process for the comprehensive score includes the following steps. The fuzzy logic system is based on four fuzzy inputs: queue length, link utilization, delay, and packet loss rate. Two fuzzy outputs are calculated: L_i and Q_i . L_i reflects the current load on the link. The lower the load, the lower the score. Q_i reflects the quality of links. In this study, to effectively combine link load and quality to determine the optimal link, we propose a comprehensive score equation as Eq. (2). L_i represents the load score of the i th link. Q_i denotes the quality score of the i th link. where $\frac{1}{L_i}$ represents the reciprocal of the link load. This design prioritizes lighter-loaded links because lighter-loaded links are more effective in reducing network congestion and higher-loaded links obtain a lower weight. Links of higher quality were considered suitable for selection. The advantage of this equation is that it can be adjusted dynamically. When the link load is light, $\frac{1}{L_i}$ will be larger; therefore, S_i of the link will be higher, and vice versa, to realize the optimization of low-load links.

$$S_i = \frac{1}{L_i} + Q_i \quad (2)$$

4. Evaluation and Experimental Results

4.1. Experimental environment

The simulation architecture for this experiment is shown in Fig. 3, and the parameters are listed in Table 1. The Ryu controller collects network status information and dynamically adjusts traffic forwarding using the FLLB algorithm to achieve the best network performance. The study uses a PC with a hardware performance of i7-12700h 4.7GHz and DDR5-4800Mhz 24GB RAM to install an Ubuntu 16.04 Linux environment. A Mininet simulator was used to build the SDN environment. There were five hosts and six switches. This study uses a fuzzy module to implement a fuzzy logic system and integrate it into an SDN controller.

4.2. Experimental Results

The FLLB algorithm was compared and analyzed with two different load-balancing algorithms: traditional equal-cost multipath routing (ECMP) [3] and fuzzy logic routing mechanism (FLRM) [4]. Fig. 4 shows the total

throughput of the different load-balancing algorithms in the simulation experiments. The figure compares the total throughput changes of the three load-balancing algorithms, FLLB, FLRM, and ECMP, within 20 s. From 15 to 20 s, FLLB has an advantage. At all time slots, the throughput of FLLB was slightly higher than those of FLRM and ECMP.

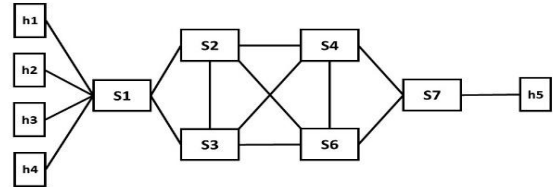


Fig. 3 Simulation topology

Table 1. Environment parameters

Attributes	Value
Simulator	Mininet 2.5
SDN Controller	Ryu Controller
SDN Protocol	OpenFlow V1.3
Switch	OpenFlow Switch
Packet generate tool	Iperf
Traffic type	UDP
Traffic flow	100Mb,150Mb,250Mb,300Mb
Link bandwidth	500Mbps
Packet size	1500byte(MTU)
Queue size	100 packets
monitoring period	1s
Simulation time	20s
(h1 to h5)	0s start
(h2 to h5)	5s start
(h3 to h5)	10s start
(h4 to h5)	15s start

Fig. 5 shows the packet delay changes of the different load-balancing algorithms within 20 s. Starting from the 5th second, the delays between the different algorithms begin to differ as the traffic increases. Under high load conditions from 15 to 20 s, FLLB's latency remains at about 1.4 ms, while FLRM's latency is approximately 1.6 ms. By contrast, the latency of ECMP increases significantly, which means that its ability to deal with sudden, large traffic is weak. FLLB has significant advantages under dynamic changes and high-load conditions such that it can manage network traffic. Fig. 6 shows the change in packet loss rate over time under different load-balancing mechanisms. ECMP allocates traffic based on the exact link cost and does not consider the real-time network status. Load imbalance can lead to increased packet loss rates in overloaded links. From 14 to 20 s, Fig. 6 shows the situation when high traffic is injected. This means that FLLB can respond quickly and stabilize the network performance when handling high traffic.

5. Conclusion

This study aimed to explore and improve the load-balancing mechanism in a network. The FLLB algorithm was proposed because of the shortcomings of traditional load-balancing methods. Under the SDN architecture, FLLB can dynamically adjust the traffic distribution according to the actual status of the link. In future, we will refer to [8] to implement the FLLB algorithm in B5G or 6G network environments, which will improve the overall QoS.

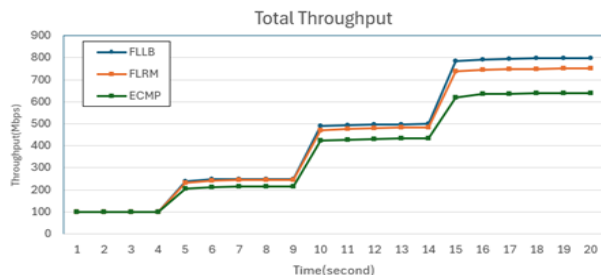


Fig. 4 Total throughput

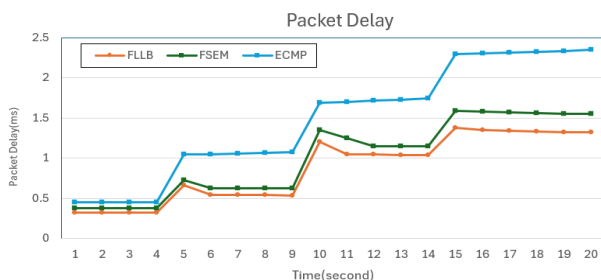


Fig. 5 Packet delay

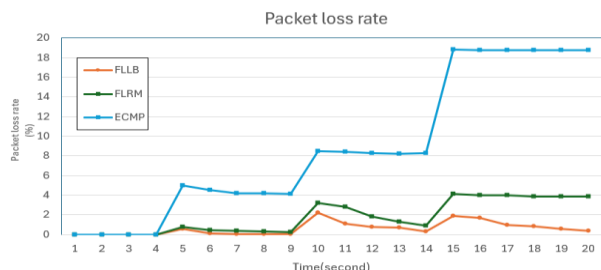


Fig. 6 Packet loss rate

Acknowledgments

This study was supported by the National Science and Technology Council (NSTC), Taiwan, under Contract No. NSTC 113-2221-E-242-001, NSTC 113-2221-E-992-001 and NSTC 113-2221-E-158-001.

References

1. Xabier Etxezarreta, Iñaki Garitano, Mikel Iturbe, Urko Zurutuza, "Software-Defined Networking approaches for intrusion response in Industrial Control Systems: A survey," *International Journal of Critical Infrastructure Protection*, vol 42, 100615, 2023.
2. Mamdani, Ebrahim H., "Application of fuzzy algorithms for control of simple dynamic plant," *Proceedings of the Institution of Electrical Engineers*, pp1585-1588, 1974.
3. C.Hopps, "Analysis of an Equal-Cost Multi-Path Algorithm", IETF RFC 2992, 2000.
4. Yang Xu, Wu Muqing, Yao Guohao, "An Effective Routing Mechanism Based on Fuzzy Logic for Software-Defined Data Center Networks," in *2020 IEEE 6th International Conference on Computer and Communications (ICCC)*, 2020.
5. O. Feuser, A. Wenzel, "On the effects of the IEEE 802.3x flow control in full-duplex Ethernet LANs", in *Proceedings 24th Conference on Local Computer Networks. LCN'99*, 1999.
6. ITU-T Y.1541, *Network performance objectives for IP-based services*, 2006.
7. ITU-T G.114, *One-way transmission time*, 2003.

8. S. H. A. Kazmi, F. Qamar, R. Hassan and K. Nisar, "Routing-Based Interference Mitigation in SDN Enabled Beyond 5G Communication Networks: A Comprehensive Survey," *IEEE Access*, vol. 11, pp. 4023-4041, 2023.

Authors Introduction

Wen-Shyang Hwang



He received B.S., M.S., and Ph.D. degrees in Electrical Engineering Department from National Cheng Kung University, Taiwan, in 1984, 1990, and 1996, respectively. He is currently a professor of Electrical Engineering Department of National Kaohsiung University of Science and Technology(NKUST). His current research interests are in the fields of multimedia wireless communication, wireless mesh networks, storage area networks, WDM metro-ring networks, software design for embedded systems, Internet QoS, and Internet applications.

Ming-Syuan Wu



He received M.S. degree (2021) from the Department of Electrical Engineering at NKUST. He is currently studying in the doctoral program at Department of Electrical Engineering of NKUST.

Cheng-Han Lin



He received his Ph.D. from the Department of Electrical Engineering at National Cheng Kung University in 2010 and his M.S. and B.S. degrees from the Department of Electrical Engineering at National Chung Cheng University in 2002 and 2004, respectively. He is currently an associate professor at the Department of Health-Business Administration at Fooyin University. His current research interests include wireless networks and SDN networks.

Yan-Jing Wu



She received a Ph.D. degree in computer engineering from the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung. She became an assistant professor in 2007, and was promoted to associate professor and full professor, respectively, in 2013 and 2023 at the Department of Information Technology and Communication, Shih Chien University, Kaohsiung Campus. Her current research interests are in the fields of mobile wireless communication networks and mobility management for multimedia traffic.

Ming-Hua Cheng



He received a Ph.D. degree in Electrical Engineering from NKUST in 2019. He has become an assistant professor since 2019 at the Department of Digital Media Design, Tzu-Hui Institute of Technology.

Sian-Fong Huang



He received M.S. degree (2024) from the Department of Electrical Engineering at the NKUST.