University of Baghdad
College of Engineering
JE
JOURNAL OF ENGINEERING

# Journal of Engineering

journal homepage: www.joe.uobaghdad.edu.iq

Volume 29 Number 6 June 2023

# Automatic Spike Neural Technique for Slicing Bandwidth Estimated Virtual Buffer-Size in Network Environment

**Mohmmed Mousa Rashid Al-Yasari[1,*], Nadia Adnan Shiltagh Al-Jamali[2]**

[1]Information Institute for Postgraduate Studies (IIPS), Iraqi Commission for Computers and Informatics (ICCI), Baghdad, Iraq

[2] Department of Computer Engineering, College of Engineering, University of Baghdad, Baghdad, Iraq

phd202020555@iips.icci.edu.iq[1], nadia.aljamali@coeng.uobaghdad.edu.iq[2]

## ABSTRACT

The Next-generation networks, such as 5G and 6G, need capacity and requirements for low latency and high dependability. According to experts, network slicing is one of the most important features of (5 and 6) G networks. To enhance the Quality of Service (QoS), network operators may now operate many instances on the same infrastructure due to configuring able slicing QoS. Each virtualized network resource, such as connection bandwidth, buffer size, and computing functions, may have various virtualized network resources. Because network resources are limited, virtual resources of the slices must be carefully coordinated to meet the different QoS requirements of users and services. To achieve QoS, these networks may be modified using Artificial Intelligence (AI) and machine learning (ML). Developing an intelligent decision-making system for network management and reducing network slice failures requires reconfigurable wireless network solutions with machine learning capabilities. In this paper, using Spiking Neural Network (SNN) and prediction, we have developed a 'Buffer-Size Management' model for controlling network load efficiency by managing the slice's buffer size. To analyze incoming traffic and predict the network slice buffer size, our proposed Buffer-Size Management model can intelligently choose the best buffer size for each slice to reduce packet loss ratio, increase throughput to 95% and reduce network failure by about 97%.

**Keywords:** 5G, Bandwidth slicing, SDN, Spiking Neural Network

*Corresponding author
Peer review under the responsibility of University of Baghdad.
https://doi.org/10.31026/j.eng.2023.06.07

Article received: 17/12/2022
Article accepted: 07/04/2023
Article published: 01/06/2023

# تقنية الشبكة العصبية المتصاعدة التلقائية في تقسيم حزمة البيانات لتخمين حجم المخزن المؤقت الظاهري في بيئة الشبكة

محمد موسى رشيد الياسري[1,*] , نادية عدنان شلتاغ الجمالي[2]

[1] الهيئة العراقية للحاسبات والمعلوماتية، معهد المعلوماتية للدراسات العليا

[2] قسم هندسة الحاسبات، كلية الهندسة، جامعة بغداد

**الخلاصة**

تحتاج شـبكة الجيل الخامس والسـادس الى متطلبات معينة مثل تقليل زمن انتقال البيانات وزيادة السـعة وذلك لكثرة البيانات والحاجة الماسـة الى السرعة العالية ولتحقيق هذا فان أفضـل تقنية حالية هي تقطيع الشبكة الى شرائح متعددة. قد يكون لكل شريحة موارد خاصـة بها، مثل حزمة البيانات للاتصـال وحجم المخزن المؤقت ووظائف الحوسبة. ونظرًا لأن موارد الشبكة او الشريحة الواحدة محدودة، فانه يجب تنسيق الموارد الافتراضية للشرائح بعناية لتلبية متطلبات جودة الخدمة المختلفة للمستخدمين والخدمات. وعلى هذا النحو فانه يتطلب تطوير نظام لاتخاذ قرار ذكي لإدارة الشبكة وتقليل حالات فشل الشريحة وجعل الشبكة قابلة لإعادة التكوين مع إمكانات التعلم الآلي. باسـتخدام تقنية "Spiking Neural Network" , والتعلم الالي (ML) للتحكم في كفاءة تحميل الشـبكة من خلال إدارة حجم المخزن المؤقت للشـريحة. لتحليل حركة تدفق البيانات الواردة والتنبؤ بحجم المخزن المؤقت لشـريحة الشبكة؛ يمكن لنموذج إدارة حجم المخزن المؤقت'Buffer-Size Management' أن يختار بذكاء أفضـل كمية حجم المخزن المؤقت لكل شـريحة لتقليل نسـبة فقدان البيانات، وزيادة إنتاجية الشـبكة إلى 95% و تقليل فشـل الشبكة الى 97%.

**الكلمات الرئيسية:** الجيل الخامس، تقسيم حزمة البيانات، الشبكات العصبية المتصاعدة

## 1. INTRODUCTION

Due to the importance of mobile communication in today's technology era **(Gupta, 2015),** there is a problem with the number of communication devices that is increasing exponentially; to meet the demands of the next generation of communication. These devices require high bandwidth, mobility, low latency, and enhanced quality QoS **(Saad, 2019).** Communication technology has rapidly evolved from 2G to 4G and is preparing for 5G and 6G. Furthermore, future communication systems must operate reliably and seamlessly in diverse wireless networks and handle reconfiguration **(Thantharate, 2019).** User needs and communication reliability are constantly challenging for companies offering these services. Expanding Long Term Evolution (LTE) networks to increase bandwidth, throughput, and service quality is the best way to meet the demands of 5G networks **(Addad, 2019)**. 5G networks can enhance mobility in services, reconfiguration, infrastructure, and various activities. It would give numerous potentials for mobilizing a variety of application fields such as seamless mobility, traffic monitoring, healthcare services, etc. One of the key components of the 5G network, according to the definition of the third-generation partnership project (3GPP), is network slicing. The operator can increase QoS by reconfiguring and supplying portions of their network to their customers' needs through network slicing **(Mayoral, 2016)**. The providers may save many resources by reconfiguring and slicing their networks while no one uses them. Configuration and network slicing can reduce latency, increase bandwidth, allow seamless mobility, and improve QoS. For example, the same network infrastructure may offer various services to consumers with better QoS. Critical monitoring of the devices and their associated traffic is necessary for ensuring QoS and resource efficiency **(Oladejo, 2017)**. Machine learning has demonstrated its ability to make vital decisions in various settings. To make predictions and key decisions, machine learning would use reconfigurable network environments to keep track of various devices and evaluate network slices and a massive quantity of data created during communication **(Afolabi, 2019)**.

A 5G network slicing resource allocation is proposed and studied extensively **(Liang, 2018)**. They suggested QoS-aware resource allocation. In the proposed system, a virtualized infrastructure management provides virtual resources to VMs by monitoring resource use. A scheme is presented for buffers based on the quality-of-service sensitivity of buffer allocation in network slices with varying quality-of-service needs **(Ponulak, 2011)**. **(Thantharate et al., 2019)** provide a 'DeepSlice' model for managing network load efficiency and availability using in-network deep learning and prediction using Deep Learning (DL) Neural Network **(Wang, 2014)**.

The primary goal of the proposed work is to provide a reconfigurable wireless network slicing model for 5G networks based on machine learning and Spiking Neural Networks. Network service providers have several challenges, including determining an accurate slice assignment **(Chahlaoui, 2019)**. For some of the services in a normal network, having a larger buffer size might be beneficial, while it could be deadly for others. This depends on the specific traffic situation that the network is experiencing at the time. We call these throughputs, delay, and loss trade-offs. Maximum buffer size is excellent for file transfer systems that demand high throughput, whereas real-time applications want minimal and constant latency. Since virtual network slices provide various services with varied QoS needs, buffer sizes may be acceptable. In a restricted buffer size situation, effective buffer management is needed to optimize the overall QoS satisfaction of slices with varying QoS criteria **(Salman, 2020)**. Network slices with varying QoS needs have varied sensitivities that reveal buffer sizing-related QoS changes. They presented a buffer-sizing approach leveraging QoS sensitivity to increase QoS satisfaction in a restricted buffer size situation **(Chergui, 2019)**. This study involves the construction of an ideal SNN-based model that assures no slice failure condition. Network traffic is routed to other slices in case of a slice failure, ensuring regular operations for these requests. When a slice fails or is overloaded, the master slice will be a backup **(AlQahtani, 2020)**. This study aims to allocate the correct buffer size slice based on incoming new traffic requests and optimal network resource utilization slice in the overflow situation.

## 2. MACHINE LEARNING AND SPIKING NEURAL NETWORK

Due to its restrictive foundation, the existing LTE architecture cannot scale to meet the needs of various applications. When serving a company's unique needs, it frequently falls short of expectations because it lacks personalization **(Thantharate, 2019)**. Businesses have greater connection and throughput requirements than the current 4G LTE network can provide due to rising mobile data and customer demands **(Khan, 2020)**. Using network slicing, 5G may efficiently provide many virtual networks over a single physical one **(Li,**

**2020)**. Much effort is put into optimizing and effectively scheduling radio and network resources, but in 5G networks, resource allocation based on the service being provided is a must-have. The increasing number of devices and new services offered by 5G networks would add to the already massive quantity of data traffic that operators now deal with **(Shiltagh, 2015)**. With ML in place, gaps in the knowledge can be examined, and any required adjustments can be made. Machine learning offers a network analysis of the massive data set, which may be researched further to adjust any given slice swiftly and cost-effectively. The Spiking Neural Network consists of different layers that are named; H, I, and J, as they are shown in **Fig. 1A.** The input data is first encoded in the encoding process; the real information is encoded $t_{input}$ calculated based on Eq. (1)

$$t_{input} = t_{max} - round\left(t_{min} + \frac{(input - input_{\min})(t_{max} - t_{min})}{(input_{max} - input_{min})}\right) \quad (1)$$

$input_{max}$ and $input_{min}$ that represent the maximum and minimum values of the real input information. The $t_{max}$ and $t_{min}$ represent the largest and minimum interval and $t_{input}$ is defined as the input information in the time domain. **Fig. 1B**, SNN may dynamically activate network automation to adjust resource allocation; where $w_{ij}^{k}$ represent as a weight for each sub-connection and $t_i$ represent the first firing times of particular neurons in the respective layers and $d^k$ defined as the delay of synaptic terminals, which remains constant during a simulation but is updated by Spike Prop afterward. The unweight contribution $y_i^k(t)$ of a single synaptic terminal to the state variable is given by Eq. (2).

$$y_i^k(t) = \varepsilon(t - t_i^f - d^k) \quad (2)$$

where $t_i^f$ is the firing time for neuron (i) and $d^k$ is delay related to the synaptic terminal $k$.

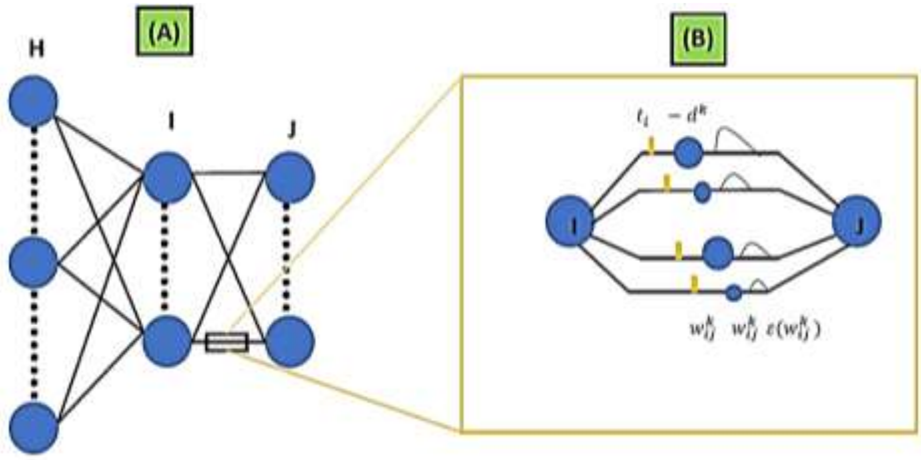


**Figure 1.** Spiking neural network architecture **(Bohte, 2002)**

The membrane potential (internal state) of a neuron is $m_j(t)$, equals the weighted sum of incoming $y_i^k$ delay response functions from the previous layer by Eq. (3).

$$m_j(t) = \sum_{i=1}^{NI} \sum_{k=1}^{d} w_{ij}^{K}(t) y_i^k(t) \quad (3)$$

Without human involvement, SNN will be accountable for delivering, processing, and making intelligent choices for network resource adaption. To make the best judgments, it will also weigh some different variables, maybe more than a single person could consider quickly **(Walter, 2015)**.
Whenever a new slice is added to the network, SNN will analyze it in real-time to determine how well it is doing. Where it stands compared to other networks, what problems could arise, what network sections are functioning normally, and if anything seems out of the ordinary **(Khan, 2020)**. Organizational challenges now prevent the widespread use of network slicing. This is because several hardware and organizations inside a service provider's network must be interacted with for a single modification to be implemented **(Meneses, 2019)**.
5G's programmability features will enable a customized end-to-end solution for every use case. Network slicing considers various factors, such as the kind of slice, buffer capacity, bandwidth, throughput, latency, equipment type, portability, reliability, isolation, power, and many more. Since 5G enables the collection of such large datasets, big data analytics need machine learning. One of the most important and useful ML-based applications in the wireless sector is the detection and revival of dormant cellular cells **(Saqib, 2019)**. Other relevant and useful ML-based applications in the wireless industry include optimizing mobile tower operations, accelerating wireless channel adoption, facilitating targeted marketing, autonomous decision-making in IoT networks, real-time data analysis, predictive maintenance, customer churn, sentiment analysis by social networking, fraud detection, e-commerce, and many more **(Paropkari, 2017)**. Since Uber employs real-time differential pricing depending on demand, the number of available vehicles, the weather, the time of day, and other factors, using ML in apps comparable to Uber will have several advantages. Better accuracy and prediction in the future may be achieved by using a platform built on machine learning to analyze and process massive amounts of historical and real-time data. Uber automatically adjusts to real-time price differences **(Abhishek, 2018)**.

## 3. BENEFITS OF SPIKING NEURAL NETWORKS

- Speed: Spiking Neural Networks are capable of transferring much information by using a few numbers of spikes. That initiates the possibility of the creation of high-speed operations.
- Real-time Action: spiked networks can use time-based information and readily integrate into “real” dynamic environments **(Ahmed, 2022)**.
- Complexity: spike networks, as a third generation, can also perform and has access to process the second generation computing with less complexity **(Khan, 2020)**.
- Biological Fidelity: Those networks can be applied in biological fields because of their similarity with biological neuronal networks **(Ahmed, 2022)**.

## 4. THE PROPOSED MODEL

The proliferation of 5G-connected devices will increase the volume of data that can be processed by neural networks, which are already in widespread use. An accurate analysis must be performed to make quick, effective decisions, which is impossible for a human brain to do in a reasonable amount of time.
To choose which slice's virtual buffer size to employ based on the input data, establish an ML model and then construct an SNN. In the first stage, aggregate traffic data, then in the

second stage, extract features of traffic to input to the SNN for encoding and convert the input information to the time domain. The third stage of SNN estimates the optimal virtual buffer's size for each slice and, in the final stage, distributes the results of the 'Buffer-Size Management on the slices.

The subsequent 'Buffer-Size Management' is used in managing network load, identifying the best virtual buffer size for each slice in the network, and so on. Our ML and SNN models utilize the same dataset, with over 65,000 distinct input combinations. Our dataset contains the most important network and device features, such as the packet delay budget, maximum packet loss, and simulation time. The number of input devices trying to connect to our system is growing. Smartphones, Internet of things devices, augmented reality and virtual reality devices, traffic from Industry or public safety communication, healthcare, smart city or smart house traffic, etc., are all examples represented in **Fig. 2**. Our SNN will keep track of everything that happens and utilize that data to make well-informed decisions in estimate network resource reservation is needed with high efficiency in the future.

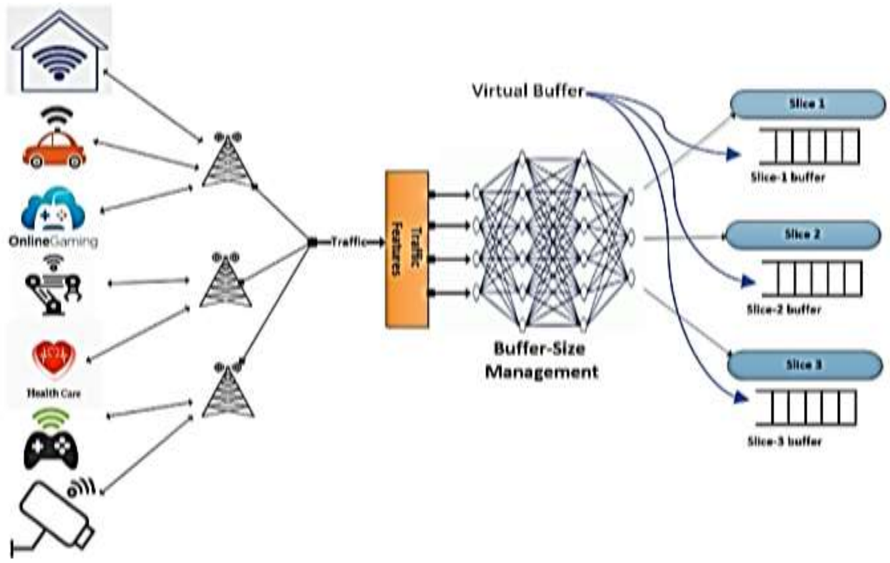


**Figure 2.** Generalized network slice representation of our buffer-size management model

## 5. PERFORMANCE EVALUATION

Here, we test the Buffer-Size Management model to see whether it meets our requirements for slice buffer prediction. To prove the efficacy of our method, a simulated network in the Mininet emulator is developed, and software-defined networking and Python 3.11 is utilized. Our approach will automatically estimate any new connections to the slice virtual buffer size needed to carry the device if any individual slice's usage rises over a particular threshold of its total available resources. Mininet with Python may be operated on a workstation running Ubuntu 14.04 LTS with a 2.2 GHz Intel CPU and 16GB of RAM. POX is set up on three different virtual machines and serves as the SDN controller for each slice.

### 5.1 Experimental Environment

Python was used to design a network with a grid topology and establish three virtual networks that share all nodes and connections. Each connection has a bandwidth of 150

MB/s, and the total buffer capacity is restricted to 250 packets. The network nodes are randomly paired with the host. The TCP packet size is 1,000 bytes, but the UDP packet size is 200. All senders create flows every second for the duration of the 50-second experiment. Assuming Slice 1 is utilized for file transfers, the average transmission rate of individual TCP flows to 50 MB/s, and UDP flows to 2 MB/s is set. Slice 2 is for real-time services. The transmission speeds for TCP and UDP are 5 and 2 MB/s, respectively. 70% describe the sets of QoS requirements that each slice must meet. Slice 3 is used for the alternative slice when slice 1 or slice 2 fails; the TCP and UDP transmission rates are 5 and 2 MB/s, respectively. Table 1 displays the experimental parameters. While each input to our model in our training dataset has six to eight parameters, our model only needs two or three features to identify the requested services and assign the appropriate fraction. This is crucial because many devices with varying capacities constantly and intermittently demand new services.

**Table 1.** Experimental parameters

| Parameters | Slice with static buffer size | Slice -1- | Slice -2- | Slice -3- |
|---|---|---|---|---|
| Link bandwidth | 150 MB/s | 150 MB/s | 150 MB/s | 150 MB/s |
| Throughput requirement | 70% | 95% | 80% | 70% |
| Data loss requirement | 10% | 5% | 10% | 14% |
| Total buffer size | 450 packets | change buffer | change buffer | change buffer |
| UDP packet size | 200 bytes | 2 MB/s | 2 MB/s | 2 MB/s |
| TCP packet size | 1,000 bytes | 50 MB/s | 50 MB/s | 50 MB/s |
| Total experiment time | 50 ms | 50 ms | 50 ms | 50 ms |

### 5.2 Experimental Results

In this section, the evaluation will be based on three criteria for evaluating the performance of the proposed work: the packet loss ratio and throughput. Finally, the amount of buffer size needed by each slice during the simulation shows the results of a study comparing the consequence rates of control slicing and non-control slicing methods using several Quality-of-Service (QoS) options so that the evaluation is very accurate.

#### 5.2.1 Packet loss ratio

The Packet Loss Ratio (PLR) for the network when three slices based on a spiking neural network are implemented. These proposed models are compared using the PLR parameter of the same network; the comparison also includes networks without the controller and determines the curve of the average proposed slices. These proposed models are compared using the PLR parameter of the same network; the comparison also includes networks without the controller and determines the curve of the average proposed slices. It is noted in **Fig. 3** that the loss of data in the slice without a controller is much more than in the control slices, where the rate of data loss is approximately 60%, while the maximum loss of the control slice did not exceed 20%, as indicated by the black line, which represents the average of the three slices.

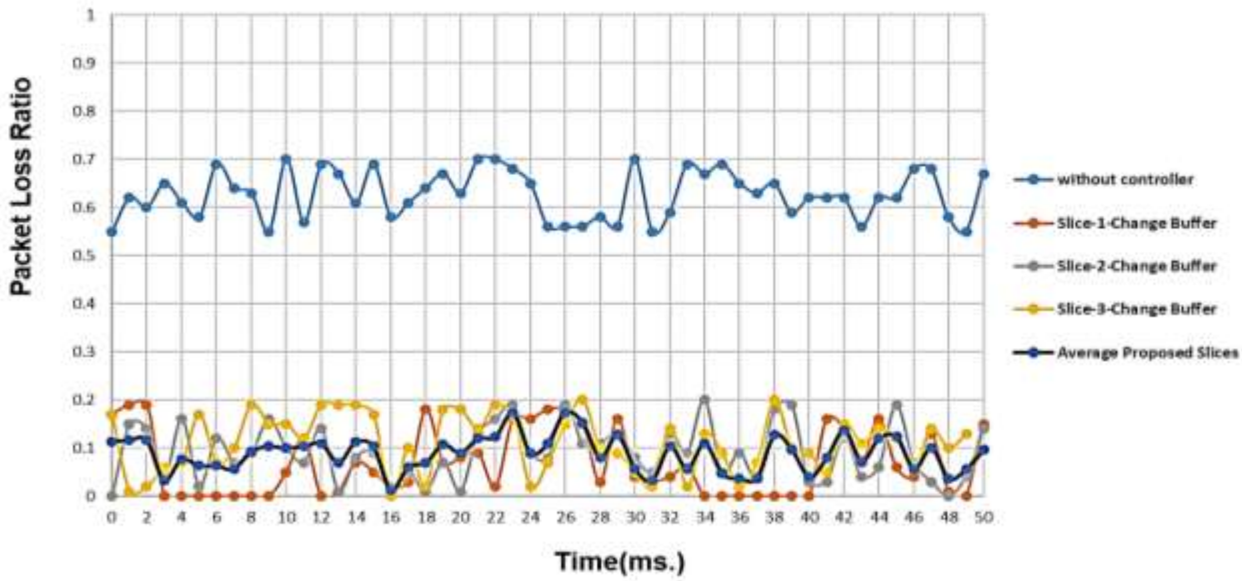


**Figure 3.** Packet loss ratios of a network slice and non-controller slice

### 5.2.2 Throughput

To prevent data loss and increase throughput in the network, Buffer-size management will now estimate the virtual buffer's slice size the next time. We carried out experiments to examine how to adjust the buffer size in each slice. **Fig. 4** displays the outcomes of comparing slicing and non-slicing approaches using varied buffer sizes. It is clear that the throughput of the control slices is more than the throughput of the non-control slice, and here the black line indicates the average control slice, which reaches more than 94%, compared to the without controller slice, whose throughput reaches less than 45%.

### 5.2.3 Buffer Capacity Amount

Slices 1, 2, and 3 were held constant while we tested different buffer capacity values. The slice's name is shown against the x-axis and virtual buffers' growth or shrinkage. The black line shows that the average slicing approach was more successful than the non-slicing approach. Performance gains or losses may be shown in **Fig. 5**, depending on the buffer sizes. At the 45.7 percentile, a performance improvement is reported. Up to 12%, more performance is possible with the correct buffer size; the maximum buffer size (i.e., 450 packets) was always used for the non-slicing method. In the equal-sized buffer technique, each slice has its buffer of the same size. The buffer widths of the three slices are optimized by our system, which examines the available buffer space at 50 different times. According to our traffic situation, the overall satisfaction with the proposed approach is higher than usual. To acquire results that are as similar as possible to real-time, we ran our simulation for 50 seconds. The model's slicing prediction accuracy over various unknown devices is also tested.

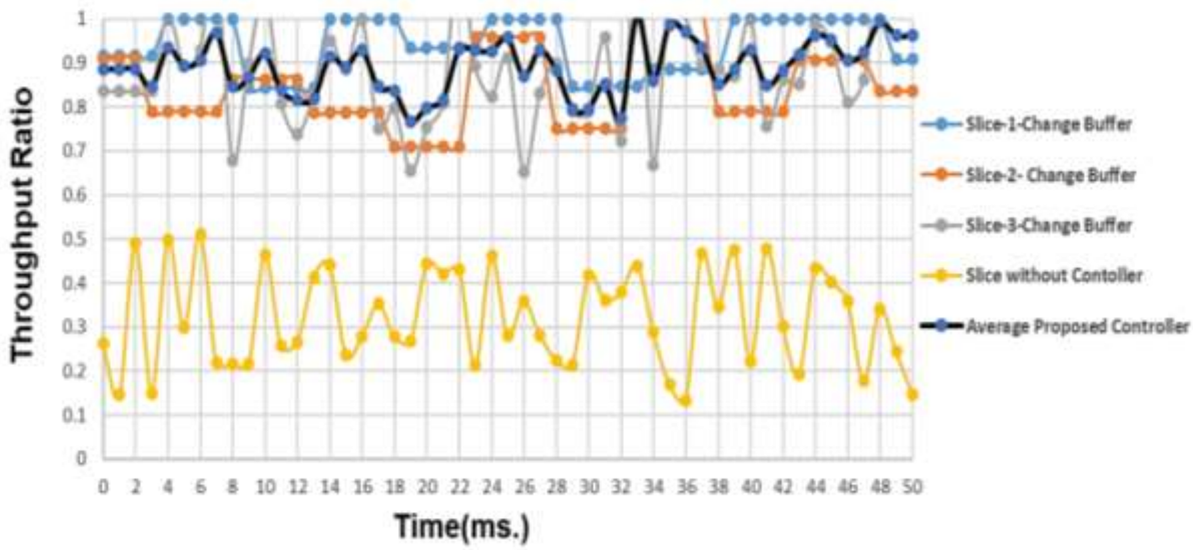


**Figure 4.** Throughout a network slice and non-controller slice

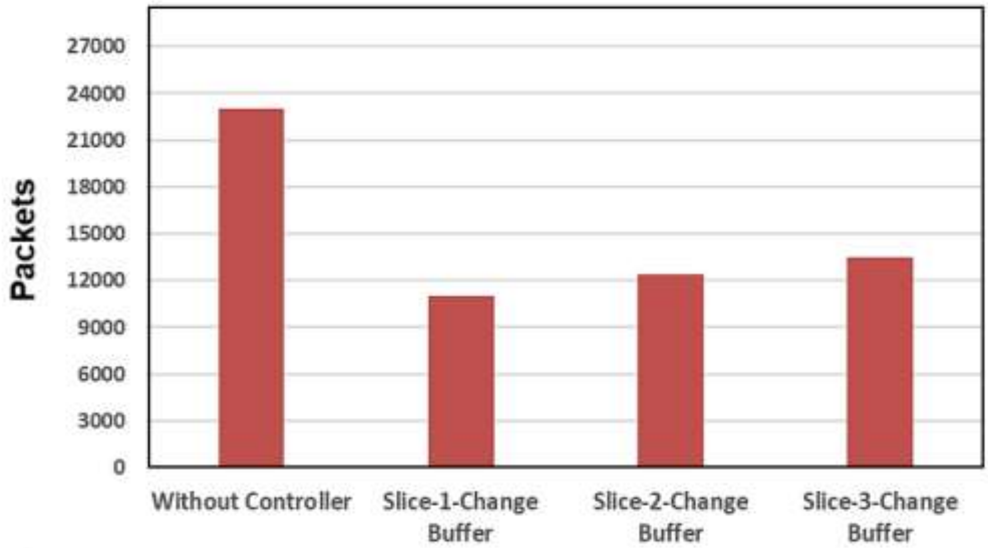


**Figure 5.** Buffer capacity for each slice during the simulation time

## 6. CONCLUSIONS

A significant challenge facing the research community is creating a smart decision-making system for incoming network traffic to guarantee load balancing, limit network slice failure, and give alternative slice buffer size in case of slice failure or overloaded situations. To solve this issue, we offer a 'Buffer-Size Management' technique that uses essential characteristics of connected devices to anticipate the most optimal virtual buffer size for every incoming network traffic. According to the findings of the experiments, the performance of the network slicing scheme combined with the proposed Buffer-Size Management is superior to that of the non-slicing scheme and the equalized buffer approach. And Indicative of the generalizability of the presented method, the Buffer-Size Management model achieved an overall recognition rate of 95%. Future work will improve the model with additional details, such as network function scheduling, that must be considered when embedding slices into a common physical network.

## REFERENCES

Abhishek, R., Tipper, D., and Medhi, D., 2018. Network virtualization and survivability of 5g networks: Framework, optimization model, and performance. In *IEEE Globecom Workshops (GC Wkshps)* pp. 1-6. doi:10.1109/GLOCOMW.2018.8644092.

Addad, R. A., Bagaa, M., Taleb, T., Dutra, D. L. C., and Flinck, H., 2019. Optimization model for cross-domain network slices in 5G networks. *IEEE Transactions on Mobile Computing*, *19*(5), pp. 1156-1169. doi:10.1109/TMC.2019.2905599.

Afolabi, I., Prados-Garzon, J., Bagaa, M., Taleb, T., and Ameigeiras, P., 2019. Dynamic resource provisioning of a scalable E2E network slicing orchestration system. IEEE Transactions on Mobile Computing, 19(11), pp. 2594-2608. doi:10.1109/TMC.2019.2930059.

Ahmed, N., Ngadi, A. B., Sharif, J. M., Hussain, S., Uddin, M., Rathore, M. S., and Zuhra, F. T., 2022. Network Threat Detection Using Machine/Deep Learning in SDN-Based Platforms: A Comprehensive Analysis of State-of-the-Art Solutions, Discussion, Challenges, and Future Research Direction. *Sensors*, *22*(20), pp. 1-34. doi:10.3390/s22207896

AlQahtani, S. A., and Alhomiqani, W. A., 2020. A multi-stage analysis of network slicing architecture for 5G mobile networks. Telecommunication Systems, 73(2), pp. 205-221. doi:10.1007/s11235-019-00607-2

Bohte, S. M., Kok, J. N., and La Poutre, H., 2002. Error-backpropagation in temporally encoded networks of spiking neurons. Neurocomputing, 48(1-4), pp. 17-37. doi:10.1016/S0925-2312(01)00658-0

Chahlaoui, F., El-Fenni, M. R., and Dahmouni, H., 2019. Performance analysis of load balancing mechanisms in SDN networks. Proceedings of the 2nd International Conference on Networking, Information Systems and Security (NISS19), P. 36, pp. 1-8. doi:10.1145/3320326.3320368

Chergui, H., and Verikoukis, C., 2019. Offline SLA-constrained deep learning for 5G networks reliable and dynamic end-to-end slicing. IEEE Journal on Selected Areas in Communications, 38(2), pp. 350-360. doi:10.1109/JSAC.2019.2959186.

Gupta, A., and Jha, R. K., 2015. A survey of 5G network: Architecture and emerging technologies. IEEE Access, (3), pp. 1206-1232, doi:10.1109/ACCESS.2015.2461602.

Khan, T. A., Mehmood, A., Ravera, J. J. D., Muhammad, A., Abbas, K., and Song, W. C., 2020. Intent-based orchestration of network slices and resource assurance using machine learning. In NOMS 2020-2020 IEEE/IFIP Network Operations and Management Symposium. pp. 1-2. doi:10.1109/NOMS47738.2020.9110408.

Li, X., Ni, R., Chen, J., Lyu, Y., Rong, Z., and Du, R., 2020. End-to-end network slicing in radio access network, transport network and core network domains. IEEE Access, (8), pp. 29525-29537. doi:10.1109/ACCESS.2020.2972105.

Liang, L., Hao Y., and Geoffrey Y. L., 2018. Toward intelligent vehicular networks: A machine learning framework. IEEE Internet of Things Journal, 6(1), pp. 124-135, doi:10.1109/JIOT.2018.2872122.

Mayoral, A., Vilalta, R., Casellas, R., Martínez, R., and Muñoz, R., 2016. Multi-tenant 5G network slicing architecture with dynamic deployment of virtualized tenant management and orchestration (MANO) instances. ECOC 2016; 42nd European Conference on Optical Communication, Dusseldorf, Germany, pp. 1-3.

Meneses, F., Fernandes, M., Corujo, D., and Aguiar, R. L., 2019. SliMANO: An expandable framework for the management and orchestration of end-to-end network slices. *IEEE 8th International Conference on Cloud Networking (CloudNet)* pp. 1-6. doi:10.1109/CloudNet47604.2019.9064072.

Oladejo, S. O., and Falowo, O. E., 2017. 5G network slicing: A multi-tenancy scenario. *Global Wireless Summit (GWS)*, pp. 88-92. doi:10.1109/GWS.2017.8300476.

Paropkari, R. A., Beard, C., and Van De Liefvoort, A., 2017. Handover performance prioritization for public safety and emergency networks. In 2017 IEEE 38th Sarnoff Symposium. 8(17), pp. 1-6. doi:10.1109/SARNOF.2017.8080381 .

Ponulak, F., and Kasinski, A., 2011. Introduction to spiking neural networks: Information processing, learning, and applications. Acta neurobiologiae experimentalis, 71(4), pp. 409-433.

Saad, W., Bennis, M., and Chen, M., 2019. A vision of 6G wireless systems: Applications, trends, technologies, and open research problems. *IEEE Network*, *34*(3), pp. 134-142. doi:10.1109/MNET.001.1900287 .

Salman, M. I., and Shaker, S. R., 2020. Link Failure Recovery for a Large-Scale Video Surveillance System using a Software-Defined Network. *Journal of Engineering*, *26*(1), pp. 104-120. doi:10.31026/j.eng.2020.01.09.

Saqib, M., Khan, F. Z., Ahmed, M., and Mehmood, R. M., 2019. A critical review on security approaches to software-defined wireless sensor networking. *International Journal of Distributed Sensor Networks*, *15*(12), pp. 1-17. doi:10.1177/1550147719889906.

Shiltagh, N. A., and Naser, M. T., 2015. A Spike Neural Controller for Traffic Load Parameter with Priority-Based Rate in Wireless Multimedia Sensor Networks. Journal of Engineering, 21(11), pp. 192-211. doi:10.31026/j.eng.2015.11.12.

Thantharate, A., Paropkari, R., Walunj, V., and Beard, C., 2019. DeepSlice: A deep learning approach towards an efficient and reliable network slicing in 5G networks. IEEE 10th Annual Ubiquitous Computing, Electronics and Mobile Communication Conference (UEMCON), 24(5), pp. 0762-0767. doi:10.1109/UEMCON47517.2019.8993066.

Wang, P., Lan, J., and Chen, S., 2014. OpenFlow based flow slice load balancing. China Communications, 11(12), pp. 72-82. doi:10.1109/CC.2014.7019842 .

Walter, F., Röhrbein, F., and Knoll, A., 2015. Neuromorphic implementations of neurobiological learning algorithms for spiking neural networks. *Neural Networks*, *(72)*, pp. 152-167. doi:10.1016/j.neunet.2015.07.004