International Journal for Multidisciplinary Research (IJFMR)
E-ISSN: 2582-2160 ● Website: www.ijfmr.com ● Email: editor@ijfmr.com

Scalable and QoS Aware Hierarchical AWGR-Based Solution for Hybrid Electro-Optic DCN Architecture

Sougata Bera[1], Chandi Pani[2]

[1]Department of CSE, Meghnad Saha Institute of Technology, Kolkata, WB, India
[2]Department of ECE, Meghnad Saha Institute of Technology, Kolkata, WB, India

Abstract:
A Hybrid Electro-Optic Data Centre Network (DCN) Architecture enormously benefits network performance, scalability, and energy efficiency for cloud-centric applications in DCN. Application of Arrayed Waveguide Router (AWGR) provides a high-performance optical routing and Wavelength multiplexing capability in DCN. In this paper, our objective is to design a cost-effective, scalable, and QoS-aware hierarchical AWGR-based solution for real-time cloud-centric applications. Our proposed model enhances connectivity, reduces blocking probability, and ensures Quality of Service (QoS) compared with the limitations of single-layer AWGR implementations. To study the performance of the proposed model, an experimental setup is developed in the laboratory using the 5 Raspberry Pi module with 24 Top of Rack (ToR) switches and found 18.7% further improvement in blocking probability to single layer model.

Keywords: Hybrid Electro-Optic architecture, QoS, AWGR, Blocking Probability.

Introduction

As the demand for higher bandwidth, lower latency, and energy efficiency grows in modern data centers, optical networking solutions have emerged as a promising alternative to traditional electronic networks. Optical DCN relies on various optical switching technologies, including SOA-based switches, Micro-Electro-Mechanical Systems (MEMS) switches, and Arrayed Waveguide Grating Routers (AWGR). MEMS switches, used in systems like c-through [2] and Helios [3], are reconfigurable optical switches driven by power but have a long reconfiguration time, making them less suitable for fast packet switching in data center networks. Several hybrid electro/optical interconnecting RotorNet [5]. It has a predefined scheduler, but it is not sensitive to traffic dynamics. AWGR is an optical device that resolves packet contention in the wavelength domain through its cyclic routing characteristic. It allows multiple inputs to reach the same output simultaneously. Several AWGR-based data center network architectures, such as DOS [6] and Petabit [7], use tunable wavelength converters (TWCs) for flexible wavelength management. However, TWCs are power-hungry and consume significant electrical power during operation. architectures have been proposed for data center networks [4]. Most of the architecture required a centralized scheduler to reconfigure the entire architecture in response to traffic dynamics except RotorNet [5].

It has a predefined scheduler, but it is not sensitive to traffic dynamics. AWGR is an optical device that re-

solves packet contention in the wavelength domain through its cyclic routing characteristic. It allows multiple inputs to reach the same output simultaneously. Several AWGR-based data center network architectures, such as DOS [6] and Petabit [7], use tunable wavelength converters (TWCs) for flexible wavelength management. However, TWCs are power-hungry and consume significant electrical power during operation. In this domain, contention resolution is achieved by wavelength conversion, fiber optic delay lines (FDLs) and rarely used deflection routing as mentioned in reference [8]. The Passive Optical Data Center Network Architecture (PODCA) [9] is a passive optical DCN architecture developed using AWGR and TWC, controlled by a Control Unit that assigns desired wavelengths. It calculates wavelength without level extraction of packets and has a packet latency below 9 µs compared to other passive optical DCN architectures like DOS and LIONS [10]. Recently proposed and investigated architecture OPSquare [11, 12] and ROTOS [13], an optical DCN architecture, utilizing fast optical switches and optical flow control, offering low latency, high connectivity, low cost, and power consumption. Single-layer PODS-based architecture [14] describes a reroutable, low-latency DCN framework for next-generation networks. In this paper, we integrate the best feature of PODCA with DOS and PODS-based architecture to get high throughput and low latency in hierarchical architecture. The packets from the servers are stored in a dynamically allocated buffer before forwarding and a loopback method is used to reroute the packet if the wavelength is not available for forwarding the packet to the proper destination. This loopback methodology and dynamic buffering make the model more scalable, and robust with low blocking probability and increase the data rate.

The rest of the paper is organized as follows: Section II describes the proposed model and its description. Section III describes the mathematical analysis of the proposed model. The experimental results are shown in Section IV. And finally, section V is the conclusion of the paper.

Proposed Hierarchical AWGR-Based DCN Architecture

Fig.1 shows the detailed architecture of the proposed hierarchical AWGR-based DCN framework.

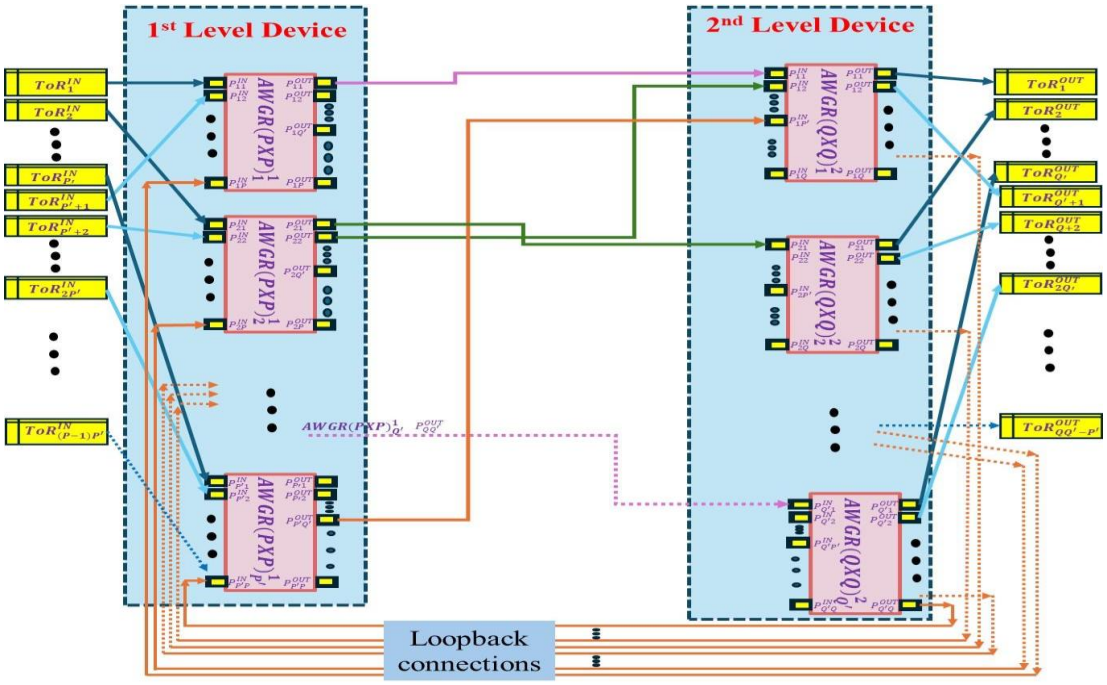

Fig. 1. Proposed Hierarchical AWGR-Based DCN Architecture

In the architecture, the total 'N' number of ToRs is connected to form the NxN hierarchical AWGR-based DCN architecture. The connections between the ToRs are established by the switching matrix designed by hierarchical AWGR. In this proposed model the two-layer hierarchical architecture is considered to avoid wavelength overlapping and optimized buffering algorithm [14]. In the proposed model, some ports of the AWGR are used for the loopback connection to avoid the unavailability of the wavelength during packet transmission. When a packet comes from the ToR switch with a particular destination address, AWGR assigns a wavelength to transfer the packet to the particular destination, with a particular value of wavelength as per AWGR routing algorithm. If the particular wavelength is not available for particular output port then the packet is forwarded to loopback port and destinated to the particular output port with a different wavelength. Also, when the packets come from the ToR switches, the packets are classified according to the service class and stored in the buffer as per priority.

Here only four types of buffers are considered, and they are classified as B1-high-priority real-time (HRT), B2-standard-priority real-time (SRT), B3-Earliest Deadline First (EDEL), B4-First-Come-First-Served (FCFS). Packets are processed from the buffer in a round-robin manner. Fig.2 shows the flowchart for the working method of the proposed model.

III. Mathematical analysis of the proposed model

To design NxN hierarchical AWGR-based Hierarchical module following are the considerations:
Table 1 shows the list of notations used for mathematical modelling.

Table1: List of notations

Symbol	Description
D_a^1	1st-level AWGR device D at position a, where a =1,2,…,P′.
D_k^2	2nd-level AWGR device D at position k, where k =1,2,…,Q′.
ToR_n^{IN}	nth ToR used for Input, where n=1,2,…,N.
ToR_n^{OUT}	nth ToR used for Output, where n=1,2,…,N.
$D^1_{aP_b^{IN}}$	b-th input pin of the 1st-level device D_a^1.
$D^1_{iP_j^{OUT}}$	j-th output pin of the 1st-level device D_i^1.
$D^2_{kQ_l^{IN}}$	l-th input pin of the 2nd-level device D_k^2.

Design Requirements

Switch Size: N×N

1st-Layer AWGR: No. of AWGR is P′ each AWGR unit having P no. of input port and P number of output ports. So that P′×P=N

2nd-Layer AWGR: No. of AWGR is Q′ each AWGR unit having Q no. of input port and Q number of output ports. So that Q′×Q =N

Lambda (λ): Lambda must be a multiple of N

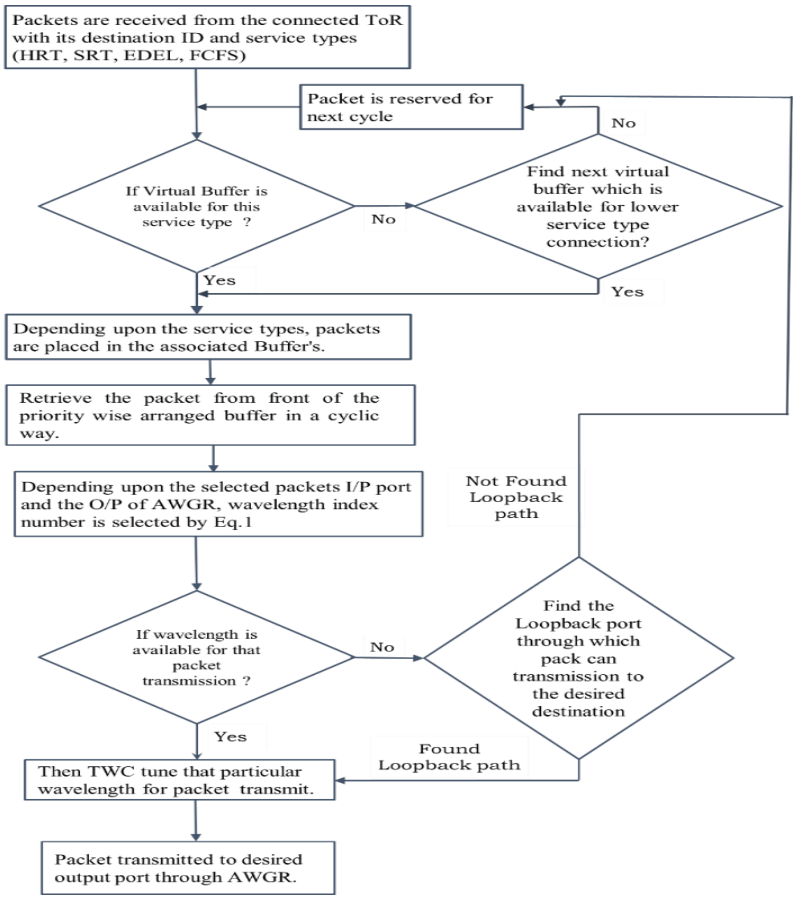

Fig. 2 Flowchart for the working method of the proposed model

Steps for Connection establishment between two layers

Step 1: Mapping Input Ports to 1st-Layer AWGR

Each input port ToR_n^{IN} is mapped to a specific 1st-level P×P AWGR device and its input pin.

1st-Level AWGR Device Sequence Number (a):

$$a = (ToR_n^{IN} - 1) mod\ P' + 1$$

Input PIN Number (b) on the D_a^1 device:

$$b = \lceil ToR_n^{IN} / P' \rceil$$

Where, ToR_n^{IN} corresponds to n.

This ensures that all input ports are distributed evenly across the P′1st-level devices.

Step 2: Connecting 1st-Level AWGR Devices to 2nd-Layer AWGR

The outputs of the 1st-level AWGR devices ($D^1_{iP_j^{OUT}}$) are routed to the inputs of the 2nd-level Q×Q AWGR devices. The logic is as follows:

2nd-Level AWGR Sequence Number (k):

$$k = \left\lceil \left(\left\{ \left(D_{iP_j^{out}}^1 - D_i^1 + P' \right) mod\ P' \right\} \times P + D_{iP_j^{out}}^1 \right) / Q \right\rceil$$

Input PIN Number (*l*) on the D_k^2 device

$$l = \left[\left\{ \left(D_{iP_j^{out}}^1 - D_i^1 + P' \right) mod\ P' \right\} \times P + D_{iP_j^{out}}^1 \right] mod\ Q + 1$$

Where, $D_{iP_j^{out}}^1$ corresponds to j and D_i^1 corresponds to i.

This connection logic ensures that the outputs from the 1st-layer AWGR devices are systematically forwarded to the appropriate 2nd-layer AWGR device inputs.

Wavelength and Dynamic Buffer Assignment for Routing and QoS Enhancement

To minimize the blocking probability (P_b),

maximize Pr_b[i] i ∈ B

P_b is calculated as the ratio of the number of packets not successfully placed to the total number of packets generated:

$$P_b = \frac{total\_packets\_generated\ -\ successful\_placement\_counter}{total\_packets\_generated}$$

Buffer Selection Algorithm [14]

Set:

1. Packet stored in the priority buffer Pr_b[i] i ∈ B (In our case $1 \leq i \leq 4$)
2. Ni is the number of buffers under each buffer type b[i].
3. $index_i$ is the index representing the next available position in buffer type i.
4. ***The optimized approach for dynamic packet allocation with buffer indexing is as follows:***
5. Initialize counters for successful placement of packets and total packets generated.
6. ***For each packet that arrives:***
7. Determine the service type of the packet.
8. Select b[i] the buffer type based on the service type and buffer priorities.
9. If the corresponding buffer type is available (i.e., index $_{buffer\_type}$<N_{buffer_type}):
10. Place the packet in the buffer at the next available position for the buffer type.
11. Increment the indexing variable for the buffer type (index $_{buffer\_type}$).
12. Increment the counter for successful placement of packets.
13. If the corresponding buffer type is not available:
14. Iterate over lower priority buffer types (b[i]≠step 6) until an available buffer is found.
15. Place the packet in the first available buffer.
16. Go to step 8
17. Increment the counter for total packets generated.

Follow the same process for all ports.
Calculate the overall blocking probability P_b as the ratio of the number of packets not successfully placed to the total number of packets generated:

Wavelength Routing Algorithm [14]
1. Set the retrieval sequence of packets from the buffer Pr_b[i] $i \in B$

$$F = \frac{no\ of\ wavelength}{no\ of\ port}$$
$$where\ \ no\ of\ wavelength\ = (integer\ factor) \times\ no\ of\ port$$

2. For i in range B:
a. retrieve the front packet of the priority buffer Pr_b[i]
b. depending upon the input port number (# input port) and the output port number (# output port) of AWGR wavelength index number (# wavelength) is selected.
c. For fc in range (F):
1. # wavelength = (# output port + # input port) % $no\ of\ port$ - 1 + $f \times\ no\ of\ port$ where $f \in F$
2. If: # wavelength available then TWC is tuned to that wavelength and the packet is ready for transmission.
d. If # wavelength is unavailable for that # output port, the packet finds the # Loopback port and is destined to the output port with a different wavelength.
e. For each # Loopback port find the availability of each pair of # wavelength for transmitting the packet from # input port to # Loopback port and # Loopback port to # output port.
i) If find a pair of # wavelengths then tune the corresponding TWC to the wavelength and assign that packet for transmission.
ii) If not found then again use the Loopback path for delay otherwise, the packet will not be transmitted and termed # Blocking.

IV. Experimental Results

For experimental validation, the AWGR-based Hierarchical framework is designed in the laboratory using 5 Raspberry-Pi modules with 24 ToR switches. Fig.3 shows the experimental set up for 24x24 DCN architecture. In layer 1 Raspberry-Pi modules are configured as 8x8 AWGR and in layer 2 Raspberry-Pi modules are configured as 12x12 AWGR. Fig.4 shows the actual implementation setup in the laboratory.

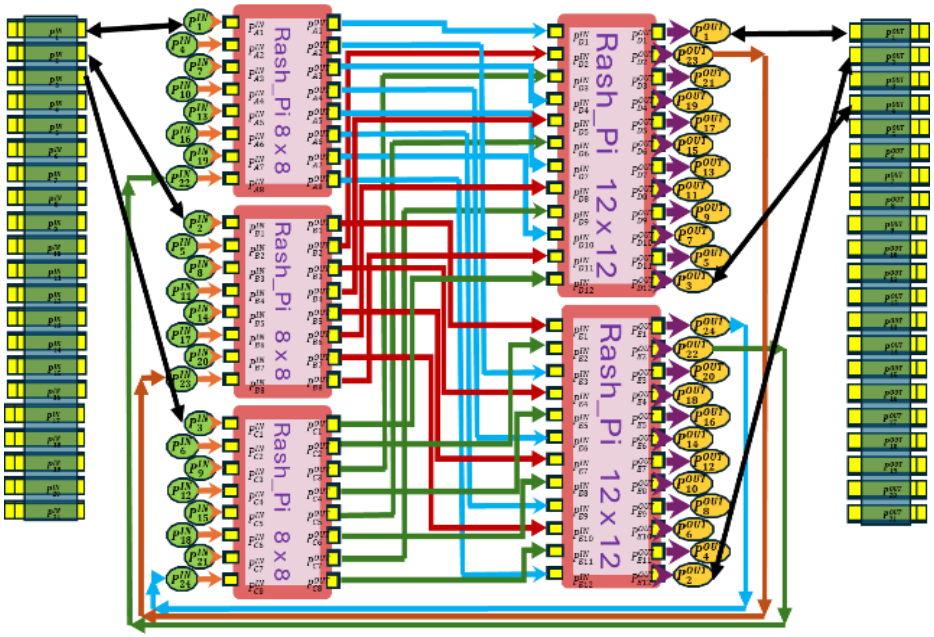

Fig. 3 Experimental set up for Two layer Hierarchical 24x24 DCN Architecture using five Raspberry Pi Units

Following are the hardware used to perform the experiment:

Hardware Configuration:

The Raspberry Pi (model Broadcom BCM2711, Quad-core Cortex-A72) was used to implement an 8x8 and 12x12 Arrayed Waveguide Grating Router (AWGR) and its control unit.

21 ervers acted as Top-of-Rack (ToR) switches.

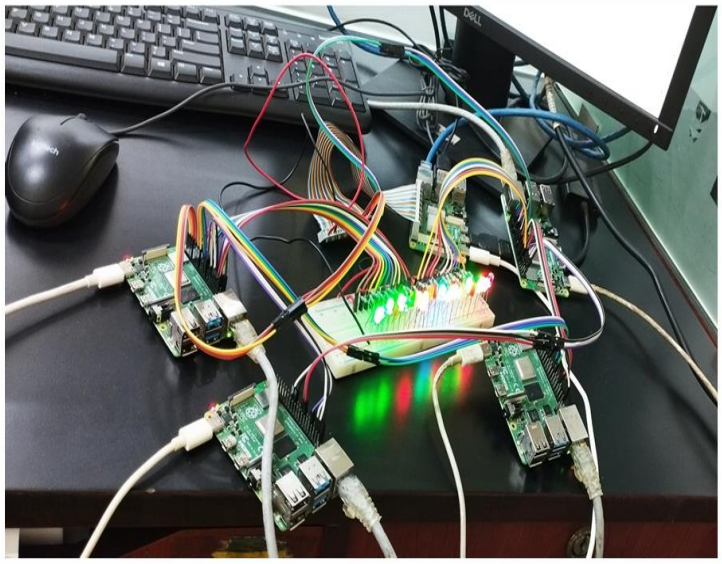

Fig.4 Physical implementation setup in the laboratory for 24x24 DCN architecture using Raspberry-Pi module

Network Load and Performance:

Each ToR switch used 24 wavelengths for packet transmission, resulting in 504 possible communication links (21 ToRs × 24 wavelengths).

Buffers were allocated per ToR to manage traffic from four service classes: High-Real-Time (HRT), Soft-Real-Time (SRT), Earliest Deadline First (EDF), and First-Come-First-Served (FCFS).

Traffic distribution chosen as: 35% for HRT, 30% for SRT, 20% for EDF, and 15% for FCFS.
Fig. 5 shows the snapshot of the program execution and wavelength assignment for ToR5. At a specific instant, it is also observed that out of 294 connections, only 239 were successfully established and 55 were blocked due to insufficient buffer availability and wavelength allocation conflicts, resulting in a blocking probability of 0.18707.

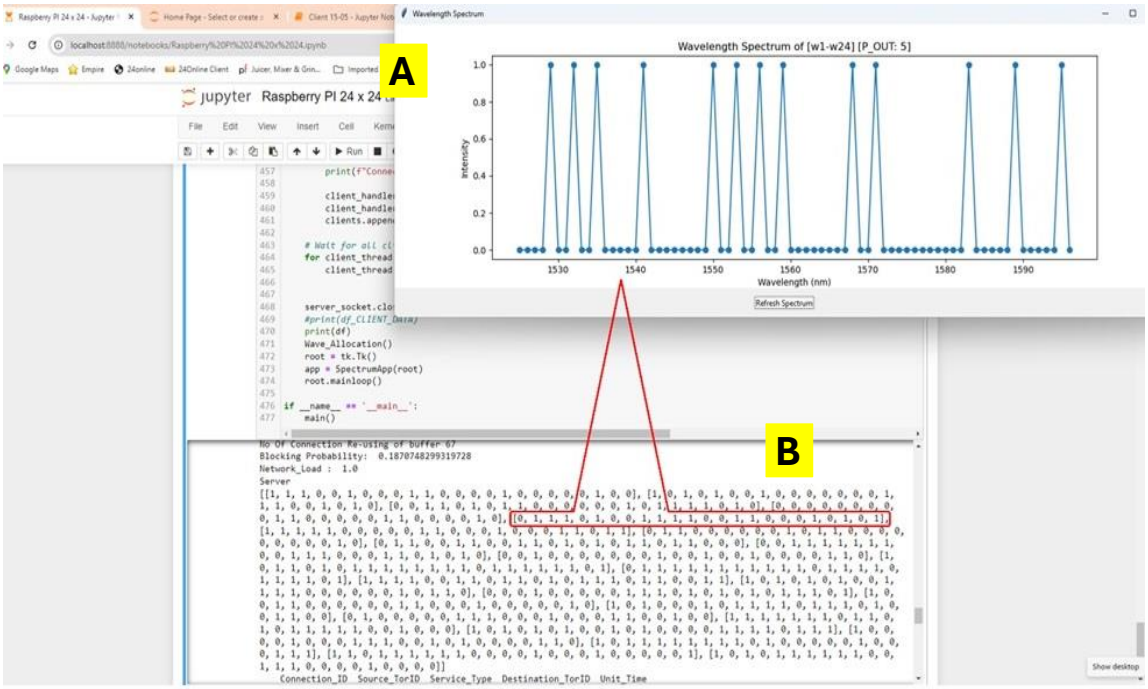

Fig. 5 Snapshot of execution of program and wavelength assignment for ToR5 for Hierarchical Model

Block A of Fig. 5 shows the wavelength allocation, where the sequence [0,1,1,1,0,1,0,0,1,1,1,1,0,0,1,1,0,0,0,1,0,1,0,1] represents the assignment in port 5. Wavelengths w_2,w_3,w_4,w_6,w_9,w_{10},w_{11},w_{12},w_{15},w_{16},w_{20},w_{22} and w_{24} are set to 1, indicating successful connection establishment between the source and destination, while wavelengths w_1,w_5,w_7,w_8,w_{13},w_{14},w_{17},w_{18},w_{19},w_{21} and w_{23} remain unused.
Block B of Fig. 5 displays the specific wavelengths utilized for successful communication. From this result, it is observed that only 18.7% of blocking is there in the framework. Fig. 6 represents the blocking probability of the entire network with network load with and without buffer reuse. It is observed that the blocking probability is further decreased to 46.7% with buffer reuses. The results indicate that the implementation of buffer reuse improves blocking probability by reducing network congestion and jamming, ensuring enhanced scalability.

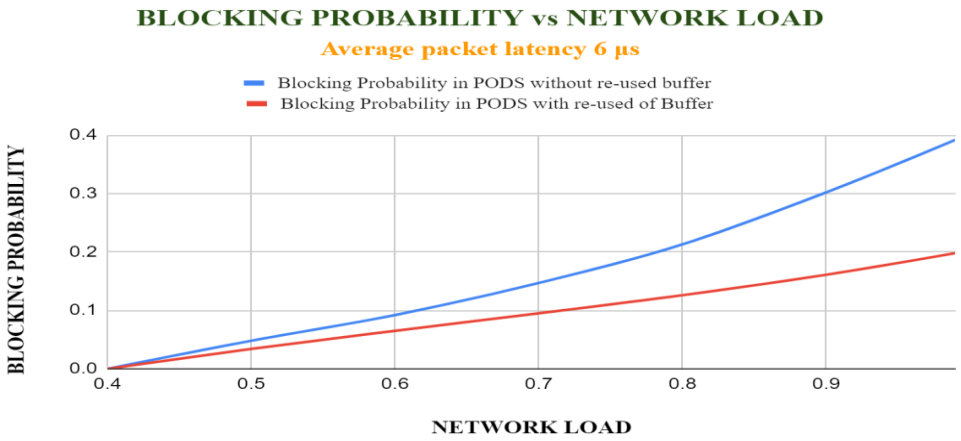

Fig. 6 Blocking probability of the framework with respect to network load for the proposed AWGR-based Hierarchical Model

Fig.7 shows the blocking probability of the proposed model with and without using the loopback methodology and it is observed that the blocking probability is further improved to 10% with the application of buffer reused with loopback methodology.

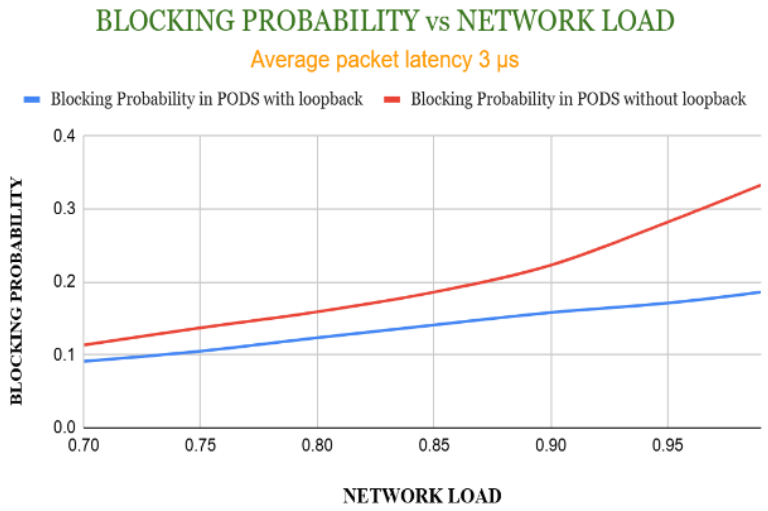

Fig. 7 Blocking probability with respect to network load with and without loopback method for Hierarchical Model

V. CONCLUSION

Therefore, by utilizing the benefits of optical switching, lower latency, scalability, and QoS by further lowering the blocking probability, we may conclude that our suggested hierarchical AWGR-based DCN design offers a potential solution for high-performance data centers. Therefore, we can say that our suggested model may be a novel way to scale the DCN architecture while satisfying the demands of workloads involving AI, big data analytics, and cloud computing.

REFERENCE:

1. IEA (2024), Global data center electricity use to double by 2026, IEA, Paris

https://www.datacenterdynamics.com/en/news/global-data-center-electricity-use-to-double-by-2026-report/

2. Guohui Wang, David G. Andersen, Michael Kaminsky, Konstantina Papagiannaki, T.S. Eugene Ng, Michael Kozuch, and Michael Ryan. 2010. C-Through: part-time optics in data centers. SIGCOMM Comput. Commun. Rev. 40, 4 (October 2010), 327–338.
3. Nathan Farrington, George Porter, Sivasankar Radhakrishnan, Hamid Hajabdolali Bazzaz, Vikram Subramanya, Yeshaiahu Fainman, George Papen, and Amin Vahdat. 2010. Helios: a hybrid electrical/optical switch architecture for modular data centers. SIGCOMM Comput. Commun. Rev. 40, 4 (October 2010), 339–350.
4. Balanici, M.; Pachnicke, S. Hybrid Electro-Optical Intra-Data Center Networks Tailored for Different Traffic Classes. J. Opt. Commun. Netw. 2018, 10, 889–901, https://doi.org/10.1364/jocn.10.000889.
5. Mellette, W.M.; McGuinness, R.; Roy, A.; Forencich, A.; Papen, G.; Snoeren, A.C.; Porter, G. Rotornet: A scalable, low-complexity, optical datacenter network. In Proceedings of the Conference of the ACM Special Interest Group on Data Communication, New York, NY, USA, 07 August. 2017, https://doi.org/10.1145/3098822. 3098838.
6. X. Ye, Y. Yin, S. B. Yoo, P. Mejia, R. Proietti, and V. Akella, “DOS: a scalable optical switch for datacenters,” in 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems, ACM, 2010, p. 24.
7. K. Xi, Y.-H. Kao, M. Yang, and H. Chao, “A petabit bufferless optical switch for data center networks,” in Optical Interconnects for Future Data Center Networks, Springer, 2013, pp. 135–154.
8. Singh, A.; Tiwari, A.K. Analysis of Hybrid Buffer Based Optical Data Center Switch. J. Opt. Commun. 2018, 42, 415–424, https://doi.org/10.1515/joc-2018-0121
9. Xu, Maotong & Liu, Chong & Subramaniam, S.. (2018). PODCA: A passive optical data center network architecture. Journal of Optical Communications and Networking. 10. 409. 10.1364/JOCN.10.000409.
10. Y. Yin, R. Proietti, X. Ye, C. J. Nitta, V. Akella and S. J. B. Yoo, "LIONS: An AWGR-Based Low-Latency Optical Switch for High-Performance Computing and Data Centers," in IEEE Journal of Selected Topics in Quantum Electronics, vol. 19, no. 2, pp. 3600409-3600409, March-April 2013, Art no. 3600409, doi: 10.1109/JSTQE.2012.2209174.
11. F. Yan, W. Miao, O. Raz and N. Calabretta, "Opsquare: A flat DCN architecture based on flow-controlled optical packet switches," in Journal of Optical Communications and Networking, vol. 9, no. 4, pp. 291-303, April 2017, doi: 10.1364/JOCN.9.000291.
12. Xue, X.; Wang, F.; Agraz, F.; Pages, A.; Pan, B.; Yan, F.; Guo, X.; Spadaro, S.; Calabretta, N. SDN-Controlled and Orchestrated OPSquare DCN Enabling Automatic Network Slicing With Differentiated QoS Provisioning. J. Light. Technol. 2020, 38, 1103–1112, https://doi.org/10.1109/jlt.2020.2965640.
13. Xue, X.; Yan, F.; Prifti, K.; Wang, F.; Pan, B.; Guo, X.; Zhang, S.; Calabretta, N. ROTOS: A Reconfigurable and Cost-Effective Architecture for High-Performance Optical Data Center Networks. J. Light. Technol. 2020, 38, 3485–3494, https://doi.org/10.1109/jlt.2020.3002735
14. Sougata Bera, Chandi Pani, “Re-routable and Low-latency All Optical Switching Algorithm for Next Generation DCN” - IJFMR Volume 5, Issue 6, November-December 2023. DOI 10.36948/ijfmr.2023.v05i06.9965