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

Re-Routable and Low-Latency All Optical Switching Algorithm for Next Generation DCN

Sougata Bera[1], Chandi Pani[2]

[1]Assistant Professor, Department of CSE, Meghnad Saha Institute of Technology
[2]Professor, Department of ECE, Meghnad Saha Institute of Technology

Abstract

Digital infrastructure is now indispensable to the function of society and quality of life of the citizen. All over the world there is leveraging of digital infrastructure that comprises use of data, computerized devices, methods and systems. During COVID-19 pandemic it comes more prominently in front. India being the most popular and populated country in the world, comprises nearly half billion internet users expected to transform the digital ecosystem. With the increasing demand of smartphone application, online activities gigantic amount of data has been generated, so evolution and development of data centers be the high importance not only for India but as well as for the world. Thus, it is the need to promote and create robust data center network infrastructure that supports emerging technologies such as 5G, IoT, AI, machine learning, adaptive manufacturing, smart agriculture, and automation and so on.

In this paper we propose an all Optical re-routable, low latency Data Center Network (DCN) architecture using Passive Optical Datacenter Switch (PODS). The PODS consists of an Arrayed Waveguide Grating Router (AWGR) in association with a controller unit. An efficient wavelength assignment algorithm is integrated in PODS, to reroute the packet for congestion avoidance and packet loss.. This feature makes the DCN architecture more scalable, proficient with high throughput, low latency, power efficient under high traffic and bursty traffic conditions. The performance of the proposed PODS-based DCN is compared with existing passive optical DCN architectures PODCA and finds improved characteristics in terms of delay, throughput and blocking probability. Simulation of the framework is done in Python platform and executed on google COLAB in the Windows environment and the result shows 46.3% improvement in latency and throughput in terms of 90% network load compared with PODCA architecture.

Keywords: AWGR architecture, Optical DCN, Blocking Probability

1. Introduction

Data centers are of utmost importance in today's technological landscape, serving critical functions in a wide range of domains including cloud services, scientific computing, and emerging big data applications. In addition to robust server infrastructure, the significance of high-bandwidth and energy-efficient interconnects and switches cannot be overstated within data centers. The datacenter related traffic has been growing annually by 25% and exceeded 20 Zettabytes in 2021 [1]. Advances in cloud computing and big-data analysis spurred by rapid progress in machine learning and artificial intelligence has accelerated the growth, which will yield the annual global datacenter traffic of 350 Zettabytes by 2030, giving a compound annual growth rate (CAGR) of 82% [2]. Total international bandwidth now stands at 997 Tbsp,

However, compared to the traditional networks (e.g., Local Area Networks and Wide Area Networks), the architectural and operational principle of DCN is highly shaped by a set of unique challenges and requirements that the DCN needs to tackle. A selected set of unique challenges and requirements for the DCN's design and operations are as follows:

- Large Scale: Modern Data Center contains hundreds of thousands of servers and is growing rapidly at an exponential rate. For example, Microsoft is hosting over 1 million servers in over 100 data centers globally. Such a huge system scale further increases the challenges on network design in terms of interconnection, cost, and robustness.
- Wide Variety of Applications: Modern DCs host various services and applications, including not only online applications such as Web search, Web mail, and interactive games, but also infrastructure services such as distributed file systems and distributed execution engines. The diversified services and applications in DCs define a variety of different traffic characteristics. Measurement results have indicated contrasting traffic characteristics when hosting different applications.
- High Energy Consumption: According to the ICA report, Global data centre electricity use in 2021 was 220-320 TWh2 [3], or around 0.9-1.3% of global final electricity demand. This excludes energy used for cryptocurrency mining, which was 100-140 TWh in 2021. The percentage of energy consumption from the network can even increase to 50% in the future when the servers themselves become more energy-proportional to their workloads.
- Strict Service Requirement: A basic requirement for service hosting in data centers is to provide 24/365 services, which demands high system robustness. However, network failures from hardware, software, and human errors can be inevitable, constant monitoring and agile failure recovery are required.

The electronic switches typically boost the bandwidth per port by aggregating multiple pins. The data rate of each pin is limited by the high-speed serializer/deserializer (SERDES), which is located at the interface of the electronic switches. Since the number of I/O pins on a chip package is also limited by physical constraints, there exists a trade-off between the bandwidth per port and the number of ports. Although a large switching capacity can be achieved by interconnecting numerous electronic switches in a Clos or Fat-tree topology [4], it results in quite complex multi-stage control and multi-hop switching, and thus increases the switching latency and energy consumption of the switch network. Also, such a kind of switched network always occupies a large volume, leading to an increase in cost. Fortunately, the recent advancement of silicon photonics, including device and packaging technologies, promises attractive solutions in the form of integrated optical switches. Some initial explorations have shown that integrated optical switches can achieve low latency, high bandwidth, and high energy efficiency to alleviate rack/board/chip-level communication bottlenecks. The latency, throughput and energy consumption of optical wavelength switches and optical space switches, including strictly non-blocking, rearrangeable non-blocking and blocking switches, have been systematically explored for the intra-rack application. It shows that the switching in the wavelength domain generally achieves a high saturation throughput but results in a large latency at light load. In contrast, the switching in the space domain typically achieves a small latency at light load but leads to an inferior maximal throughput, especially for the blocking switches. Additionally, the optical wavelength switch usually outperforms the space switch when the packet size is small. However, the number of available wavelengths and scalability of (Arrayed Waveguide Grating Routers) AWGR are the main constraints of such a category of integrated switches. Furthermore, the maximal throughput of DRAGON [4] and AWGR with conventional control strategies as discussed in [5] is limited to 50 Tbps.

However, power consumption, latency, and throughput on large clusters are critical challenges for electrical switches. Some existing optical DCNs are also commonly based on AWGR. DOS [6] and Petabit [7] represent the state-of-the-art in terms of AWGR optical switching in high-performance data center interconnects. The DOS topology consists of an array of (TWCs), an AWGR, and a loopback shared buffer. Each node can access any other node through the AWGR by configuring the transmitting wavelength of the TWC. The system is configured by the control plane that controls the TWC and the label extractors (LEs). The control plane is used for contention resolution and TWC tuning. The scalability of DOS depends on the scalability of the AWGR and the tuning range of the TWC.

Existing optical data center networks commonly rely on various optical switching technologies, such as the Semiconductor Optical Amplifier (SOA)-based switch, Micro-Electro-Mechanical Systems (MEMS) switches, and Arrayed Waveguide Grating Routers (AWGR). In particular, the MEMS switch, employed in systems like c-through [8] and Helios [9], is a reconfigurable optical switch driven by power. However, its reconfiguration time is relatively long, typically measured in milliseconds, which makes it less suitable for fast packet switching in data center network (DCN) applications.

Several traditional electrical data center networks, such as Fat-Tree, VL2, Flattened Butterfly and Bcube [10-12], employ a multi-layer architecture. These networks consist of numerous identical switches at the lower level, facilitating connectivity with end nodes like servers or racks. At the upper layers, a small number of costly and high-capacity switches are strategically placed to aggregate and distribute network traffic.

The Arrayed Waveguide Grating Router (AWGR) is an optical device that operates passively and does not require reconfiguration. It offers a unique capability for resolving packet contention in the wavelength domain. The AWGR leverages its cyclic routing characteristic, enabling multiple inputs to reach the same output simultaneously through the utilization of distinct wavelengths. Several AWGR-based data center network (DCN) architectures have been introduced in recent literature, including DOS and Petabit. These architectures incorporate tunable wavelength converters (TWCs) that facilitate flexible wavelength management. However, it is important to note that TWCs are known to be power-hungry devices, consuming a significant amount of electrical power in their operation.

Petabit employs a three-stage Clos network architecture in which each stage comprises an array of AWGRs for passive packet routing. In the first stage, tunable lasers are utilized to direct the packets through the AWGRs. In the second and third stages, tunable wavelength converters (TWCs) are employed to adjust the wavelengths as necessary and route the packets to their respective destinations. Unlike DOS, Petabit does not utilize any buffers in order to avoid the power consumption associated with electrical-to-optical (E-O) and optical-to-electrical (O-E) conversions.

Nevertheless, both DOS and Petabit optical switches suffer from certain limitations. The primary drawback of DOS lies in its reliance on electrical buffers for congestion management, which entails power-hungry electrical-to-optical (E-O) and optical-to-electrical (O-E) converters, leading to increased overall power consumption. Additionally, the DOS architecture utilizes tunable wavelength converters, which are considerably more expensive compared to the commodity optical transceivers commonly used in current switches. Similarly, Petabit also necessitates a substantial number of tunable wavelength converters (TWCs) to establish interconnections among each stage of AWGRs.

Passive Optical Data Center Network Architecture (PODCA) [13] is developed with AWGR and TWC and its functionality is controlled by the Control Unit that assigns the desired wavelengths. This architecture calculates the wavelength without level extraction of the packet. According to the results

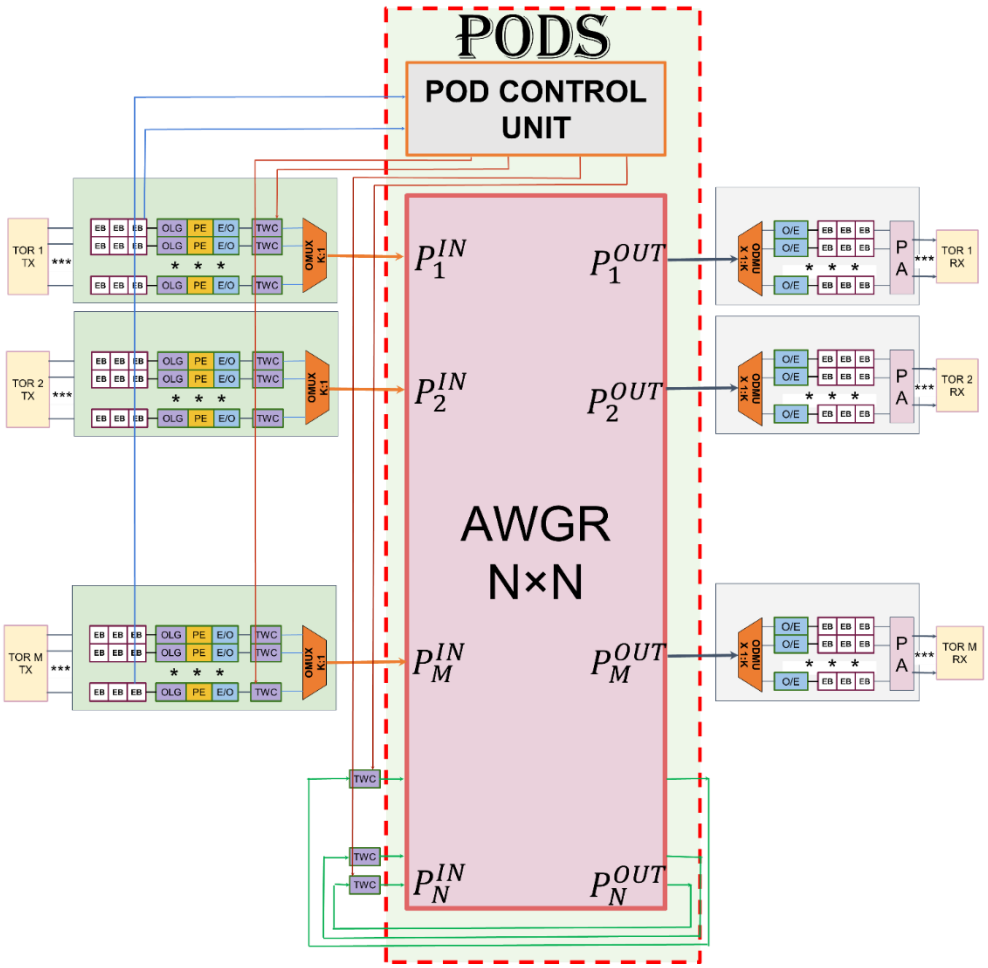
mentioned in reference [13-14], the packet latency of PODCA is below 9 μs with respect to other existing passive optical DCN architecture [DOS, LIONS]. The main disadvantage of PODCA architecture is low throughput w.r.t DOS and LIONS architecture.

In this paper, we integrate the best feature of PODCA with DOS and LIONS architecture to get high throughput and low latency. Here the packets from the servers are stored in priority buffer before forwarding and a loopback method is used to reroute the packet if the wavelength is not available for forwarding the packet to proper destination. This loopback methodology makes the model more scalable, robust with low blocking probability and increases the data rate up to 533 Tbps.

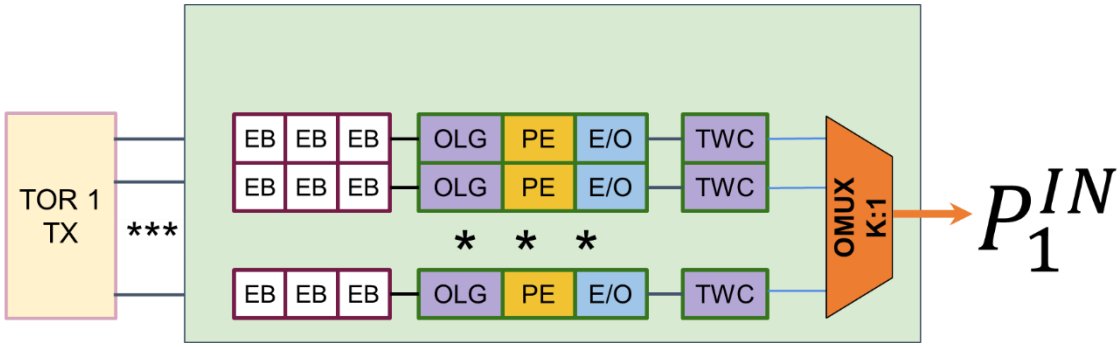
The rest of the paper is organized as follows: Section II describes the system model and its description. The functional description of the control panel is described in section III. Section IV describes the simulation process and result analysis. Finally section V is the conclusion of the paper.

2. PODS based DCN System Model

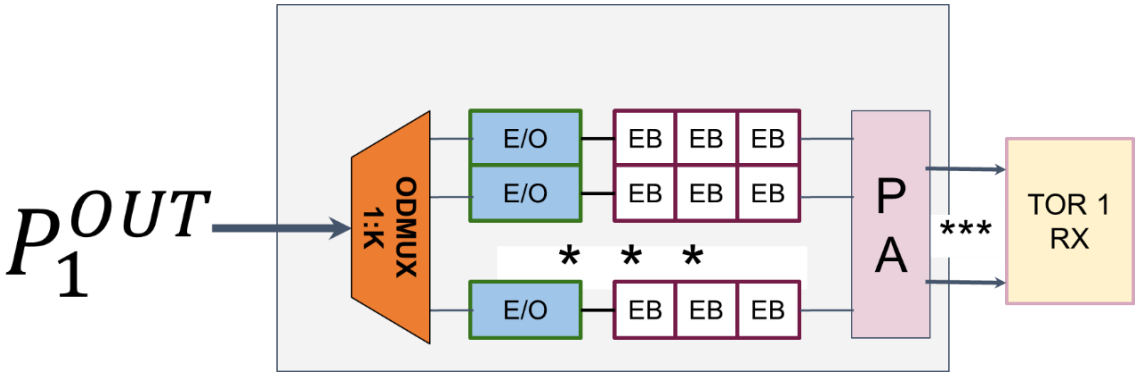
Proposed Passive Optical Datacenter Switch (PODS) base DCN architecture with POD control unit is shown in Fig.1(a). Dotted block is shown as PODS internal architecture that consists of AWGR and POD control unit. The proposed model consists of TOR, AWGR, TX and RX modules. All the ports of the TOR are configured as input-output ports. Out of total ports of TOR, some ports are connected to the end users or servers and the rest are connected to AWGR through TX or RX modules as shown in Fig 1(b) and Fig 1(c).



(a) PODS Based DCN Architecture



(b) TX module of PODS Based DCN Architecture



(c) RX module of PODS Based DCN Architecture

Fig 1: PODS System Model

The TX module of each ToR consists of electrical buffer (EB), optical channel adapter, optical label generator (OLG), packet encapsulator (PE), and electro-optic converter (lasers, typically) to send the incoming packets to the AWGR through tunable wavelength converter (TWC). After passing through the TWC, wavelengths are combined by Optical Multiplexer (OMUX) and finally reached to the input port of the AWGR. The RX module consists of an Optical Demultiplexer (ODMUX) followed by an optical receiver called optical to electrical converter (OE converter), electrical buffer (EB), and packet adapter (PA).

In this architecture the generated packets from the server, first arrived to any input port of the TOR (ToR_{IN}), and classify the packet as per the service class, then placed it to the shared buffer of TX module, marked as EB as shown in Fig.1(b), similarly when the packets are out from the ToR_{OUT} port of AWGR, after demultiplexing the packets they are converted to optical to electrical and stored in a shared buffer of TOR receiver (RX) module. As per the service class of the packet the priority level of the buffer is set. The buffers 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 flow chart for the function of the Proposed PODS based DCN architecture.

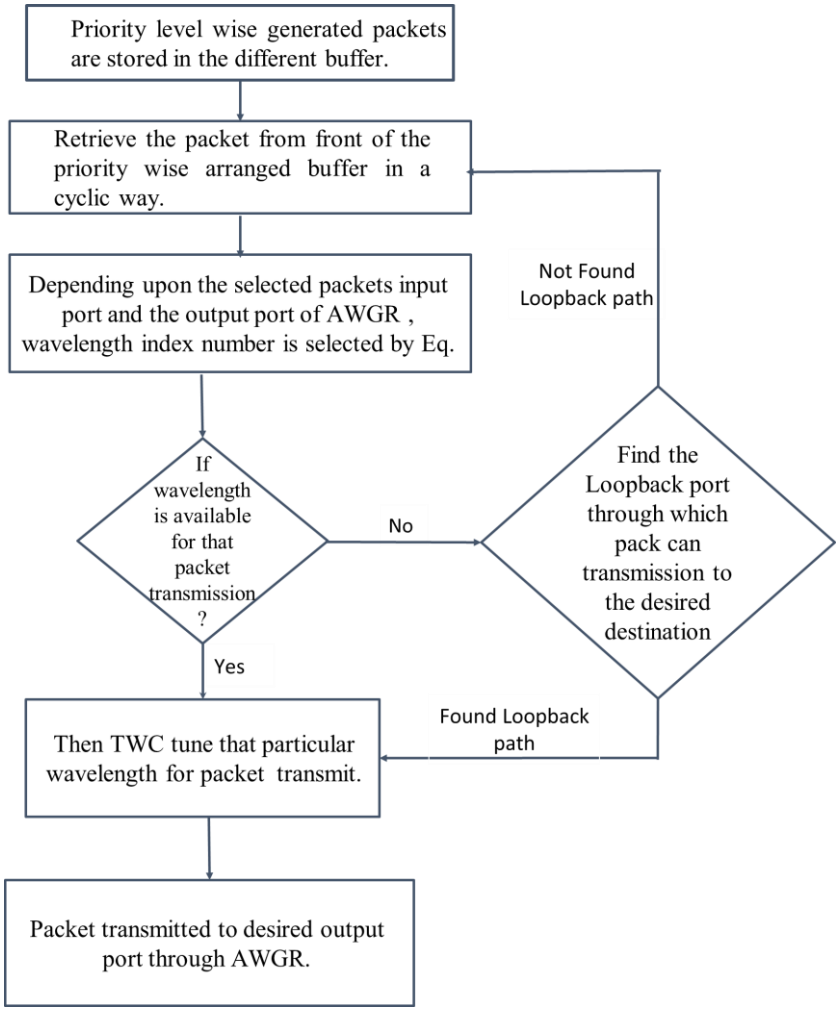


Fig. 2. Flowchart of PODS Based DCN Architecture

3. Analytical Modeling

To describe the performance of the system model we consider:

P is the total number of ports of AWGR. Out of P number of ports some ports are connected to the TOR and rest of the ports are used for the loopback path for rerouting. We denote W as the number of available wavelengths and let W=P× F where F ≥ 1 is an integer. Wavelength w is denoted as $\boldsymbol{\lambda}_W$, where w is the wavelength index. The AWGR routes #wavelengths from an #input port to a specific #output port in a cyclic way and is denoted as λ_w^S means it is routed from input port S with wavelength w to reach the desired output port.

wavelength = (# output port + # input port) % no of port + 1 + f × no of port , where f ∈ F (1) [6]

Table 1 shows the variables used to develop the analytical modeling of proposed architecture.

Let B is a number of shared buffers in each TOR. B number of shared buffers are further subdivided into B_1, B_2, B_3 and B_4. Where B_1 number of buffers assigned for HRT, B_2 number of buffers assigned for SRT, B_3 number of buffers assigned for EDEL and B_4 number of buffers assigned for FCFS. Each Packet has the same priority level placed in the same priority buffer in the round-robin manner. Depending upon the set of priority, the packets are selected from the front of each buffer for transmission.

TABLE 1. LIST OF VARIABLES USED IN THE ARCHITECTURE

Notation	Corresponding Meaning
P	Number of port in AWGR
W	Number of available wavelength in the system
F	Number of wavelength used for every pair of input output port
$\boldsymbol{\lambda}_w$	w is the wavelength index
$\lambda_w^{S,D}$	$\lambda_w^{S,D}$ the wavelength w is selected for packet transmission from Sth input port P_S^{IN} to Dth output port P_D^{OUT}
B	Number of shared buffers are subdivided into B_1, B_2, B_3 and B_4
P_S^{IN}	Sth input port in AWGR
P_D^{OUT}	Dth output port in AWGR
$Pk_{pid}^{S,D,m}$	pid Packet transmit from mth buffer of Sth input port P_S^{IN} to Dth output port P_D^{OUT}
Pr_b[i]	Packet stored in the priority buffer Pr_b[i] $i \in B$.

If the wavelength is not available for transmitting the packet to any particular port, the data is sent to the loopback port so that the data can transfer to the particular output port with different wavelength.

$Pk_{pid}^{S,D,m}$ indicates the pid packet transmit from mth buffer of Sth input port P_S^{IN} to Dth output port P_D^{OUT} , $\lambda_w^{S,D}$ indicates as the wavelength w is selected for packet transmission from Sth input port P_S^{IN} to Dth output port P_D^{OUT} and $\lambda_w^{S,L}$ indicates as the wavelength w is selected for packet transmission from Sth input port P_S^{IN} to Lth loopback output port P_L^{OUT} and $\lambda_w^{L,D}$ indicates as the wavelength w is selected for packet transmission from Lth loopback input port P_L^{IN} to Dth output port P_D^{OUT}.

To maximize packet transmission Eq. 2 should be maximized subject to the condition described in Eq. 3, Eq.4 and Eq.5.

Maximize:

$$\sum_{S,D,m} Pk_{pid}^{S,D,m} \quad (2)$$

s.t.:

$$\forall\, \lambda_w^{S,D} = 0 \quad for\ the\ packet \quad Pk_{pid}^{S,D,m} \quad (3)$$

or,

$$\lambda_w^{S,L} = 0 \quad for\ the\ packet \quad Pk_{pid}^{S,D,m} \quad (4)$$

$$\text{and } \lambda_w^{L,D} = 0 \quad for\ the\ packet \quad Pk_{pid}^{S,D,m} \quad (5)$$

Algorithm of Packet Scheduling through Control Panel

PROCESS 1: [PACKET CREATE AND STORE IN SELECTED BUFFER]

In simulation all the created packets are assigned with a destination address which is generated randomly and also the interarrival rate between the packets follow poisson distribution.

Generated packets are placed in different buffers as per their priority as mentioned here B_1, B_2, B_3 and B_4.

Priority level wise generated packets $Pk_{pid}^{s,d}$[pid − packet id, $s \in \#\, input\ port$ and $d \subset \#\, output\ port$] are stored in the buffer Pr_b[i] $i \in B$.

PROCESS 2: [ASSIGN WAVELENGTH FOR ROUTING]

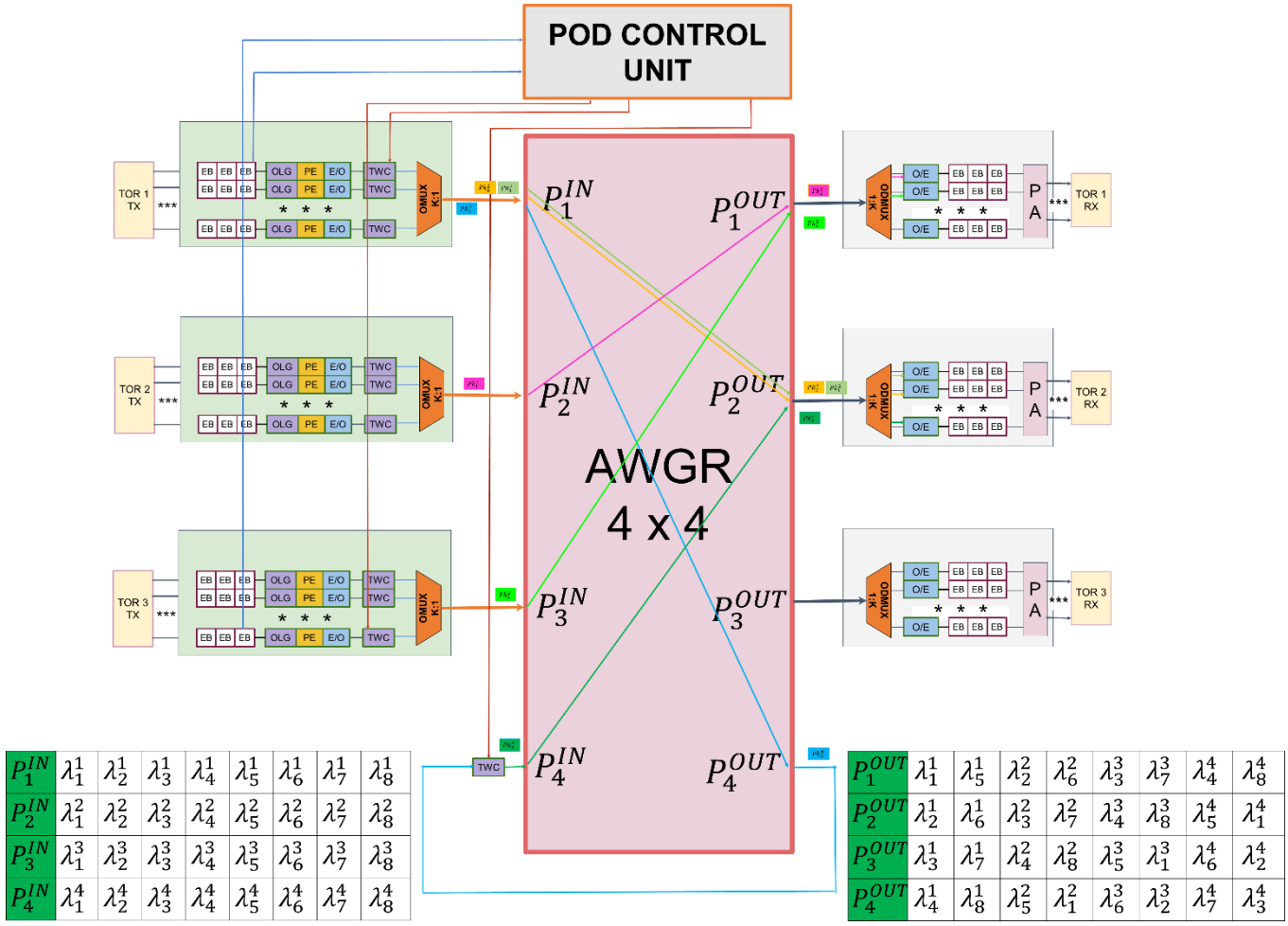
1. Set the retrieval sequence of packets from the buffer Pr_b[i] $i \in B$
2. $F = \frac{no\ of\ wavelength}{no\ of\ port}$

$$where\ no\ of\ wavelength = (integer\ factor) \times no\ of\ port$$

3. For i in range B :
   a. retrieve front packet of the priority buffer Pr_b[i]
   b. depending upon 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 tuned to that particular wavelength and the packet is ready for transmission.
   d. If # wavelength is unavailable for that # output port then the packet find the # Loopback port and destined to particular output port with different wavelength.
   e. For each # Loopback port find the availability of each pair of # wavelength for transmit the packet from # input port to # Loopback port and # Loopback port to # output port.

   i If found a pair of # wavelength then tune the corresponding TWC to the particular wavelength and assign that packet for transmission.

   If not Found then again we used Loopback path for delay otherwise, the will not be transmitted and it's called # Blocking.



P_1^{IN}	λ_1^1	λ_2^1	λ_3^1	λ_4^1	λ_5^1	λ_6^1	λ_7^1	λ_8^1
P_2^{IN}	λ_1^2	λ_2^2	λ_3^2	λ_4^2	λ_5^2	λ_6^2	λ_7^2	λ_8^2
P_3^{IN}	λ_1^3	λ_2^3	λ_3^3	λ_4^3	λ_5^3	λ_6^3	λ_7^3	λ_8^3
P_4^{IN}	λ_1^4	λ_2^4	λ_3^4	λ_4^4	λ_5^4	λ_6^4	λ_7^4	λ_8^4

P_1^{OUT}	λ_1^1	λ_5^1	λ_2^2	λ_6^2	λ_3^3	λ_7^3	λ_4^4	λ_8^4
P_2^{OUT}	λ_2^1	λ_6^1	λ_3^2	λ_7^2	λ_4^3	λ_8^3	λ_5^4	λ_1^4
P_3^{OUT}	λ_3^1	λ_7^1	λ_4^2	λ_8^2	λ_5^3	λ_1^3	λ_6^4	λ_2^4
P_4^{OUT}	λ_4^1	λ_8^1	λ_5^2	λ_1^2	λ_6^3	λ_2^3	λ_7^4	λ_3^4

Fig 3: Operational model of the proposed architecture

For explaining the operation we consider Wavelength [W]=8, No of AWGR port [P] used to connect the TOR =4 and Loopback port [L] = [Port no 4]. So, here $F = \frac{W}{P} = 2$ and the number of priority buffers B = 4.

Fig. 3 shows the operational model of the proposed architecture using PODS. Here, simultaneously three packets $[Pk_3^{1,2}, Pk_2^{1,2}, Pk_1^{1,2}]$ arrived in ToR_1 which is connected to the P_1^{IN} input port of AWGR, One packets $[Pk_4^{2,1}]$ arrived in ToR_2 which is connected to the P_2^{IN} input port of AWGR and one packets $[Pk_5^{3,1}]$ arrived in ToR_3 which is connected to the P_3^{IN} input port of AWGR depending upon the switching criteria Control panel assigns the wavelength $[\lambda_3^3, \lambda_2^2, \lambda_6^1, \lambda_2^1]$ for the packet $[Pk_5^{3,1}, Pk_4^{2,1}, Pk_2^{1,2}, Pk_1^{1,2}]$ transmission from input ToR to output ToR. Since, packet$[Pk_3^{1,2}]$ can not be transmitted simultaneously to the output port due to unavailability of the wavelength from AWGR port 1 to AWGR port 2, Loopback path is used for transmission. Control panel assigns the wavelength $[\lambda_4^1]$ for the packet $[Pk_3^{1,2}]$ to transmission from input ToR 1 to Loopback port 3 and Loopback port 3 to output port 2 assigns the wavelength $[\lambda_1^4]$ and reach the desired destination as shown in wavelength assignment in Fig 3.

4. SIMULATION RESULTS

We use Integer linear programming (ILP) as a mathematical tool to maximize the wavelength allocation for packet transmissions in a slot. We coded the algorithms in Python and executed on google

COLAB in the Windows environment to find the performance of the proposed architecture and compared it with PODCA architecture. For simulation purposes we have assigned the transmission rate of a tunable transmitter (and wavelength capacity) to be 40 Gbps as per reference [13]. The tuning time of tunable transmitters is 8 ns. The size of a packet is 1500 bytes and inter-arrival follows Poisson distribution. For simulating the network we have considered each rack has a 256MB buffer memory. This full buffer is partitioned into 116 no. of virtual buffers. These 116 no. of virtual buffers are distributed into 4 service classes as B1-high-priority real-time (HRT), B2-standard-priority real-time (SRT), B3-Earliest Deadline First (EDEL) and B4-First-Come-First-Served (FCFS), having B1=44, B2=34, B3=23 and B4=14 respectively. The traffic arrival rate per rack is 40 Gbps, The size of AWGR used for simulation is 128 × 128 and the number of available wavelengths is 256. 10% of the AWGR port is used for the Loopback path, mean's 13 ports are used for loopback connections. Number of ToR connected with the AWGR port is 115. So, for data transmission speed for this PODS model is (115 × 116 × 40 Gbps) ≈ 533 Tbps.

Total latency and network load of the framework is calculated as

Latency = Transmission time + Average queuing delay in the buffer.

$$\textbf{Network Load} = \frac{\textbf{NO OF PACKET CURRENTLY TRANSMIT}}{\textbf{MAX NO OF PACKET CAN TRANSMIT THROUGH AWGR PORT}}$$

The Simulation Snap of PODS based DCN Architecture is shown in Fig 4, executed in the Google Colab Environment. Table 2, Table 3 and Table 4 show the Network Load with respect to packet Arrival Rate, Latency with respect to packet Arrival Rate and Blocking probability with Network Load for PODS and PODCA-S (one TOR associated with one AWGR port of PODCA) architecture respectively. From Table 2 and Fig 5 it is observed that network load becomes 100% for 36 Gbps Arrival Rate for PODCA-S architecture whereas in our PODS architecture network load becomes 95.4%. The data shown in Fig. 4 is highlighted in Table 2. Table 3 and Fig 6 shows the comparison of latency performance between PODS and PODCA-S, it shows that at 36 Gbps Arrival Rate PODS latency time is approx 46.3% less than the PODCA-S. From Table 4 and Fig 7 it also observed that due to the loopback port, blocking probability is improved by 92% under full load condition.

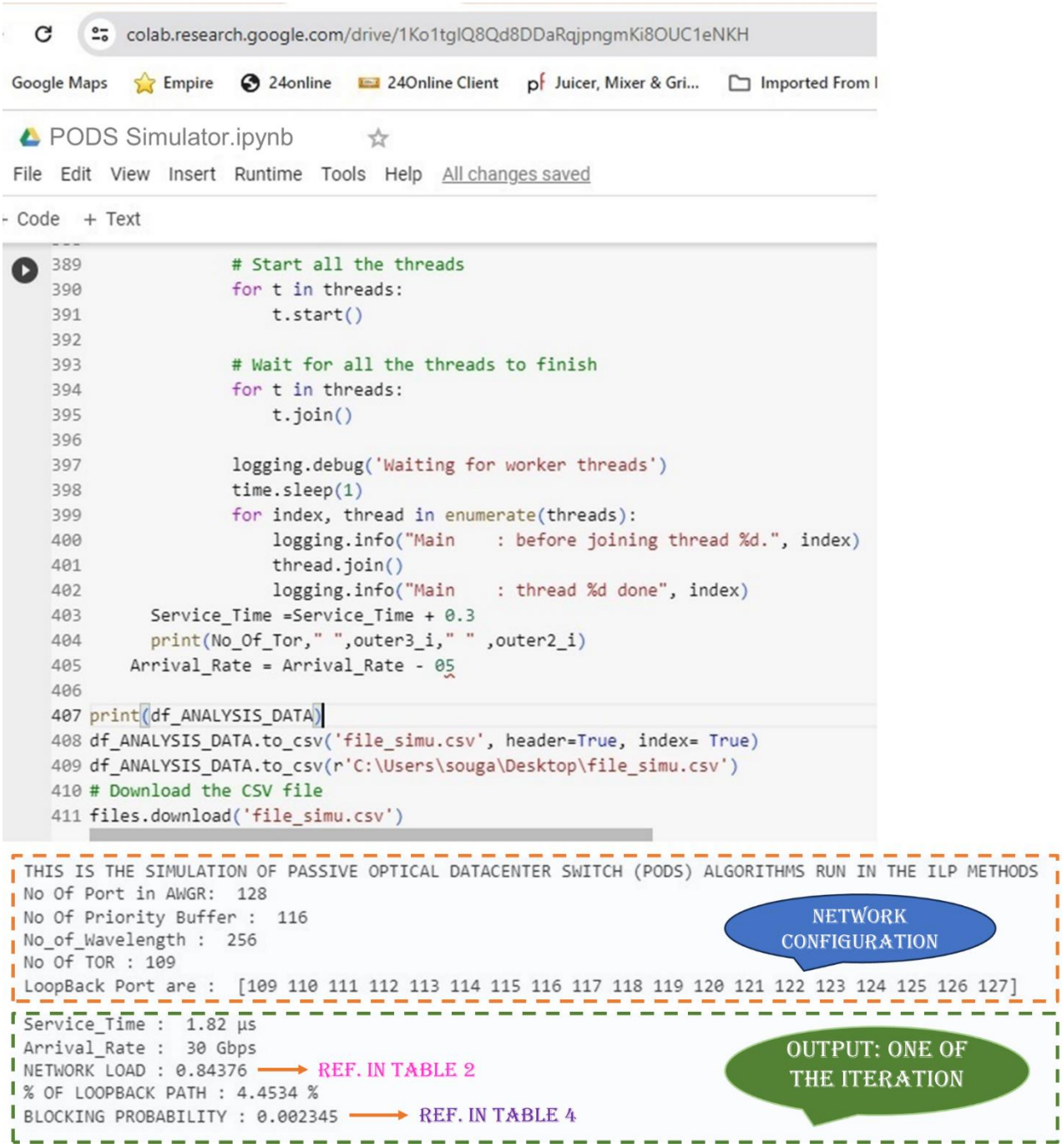


Fig 4: Snap of PODS based DCN Architecture execute in the Google Colab Environment

Table:2. Network Load with respect to packet arrival rate

Arrival Rate(Gbps)	0.1	6	12	18	24	30	36
N/W Load in PODS(%)	16.12	25.37	37.14	53.2	68.24	83.87	95.4
N/W Load in PODCA-S(%)	17.21	27.25	45.24	65.2	82.03	95.12	100

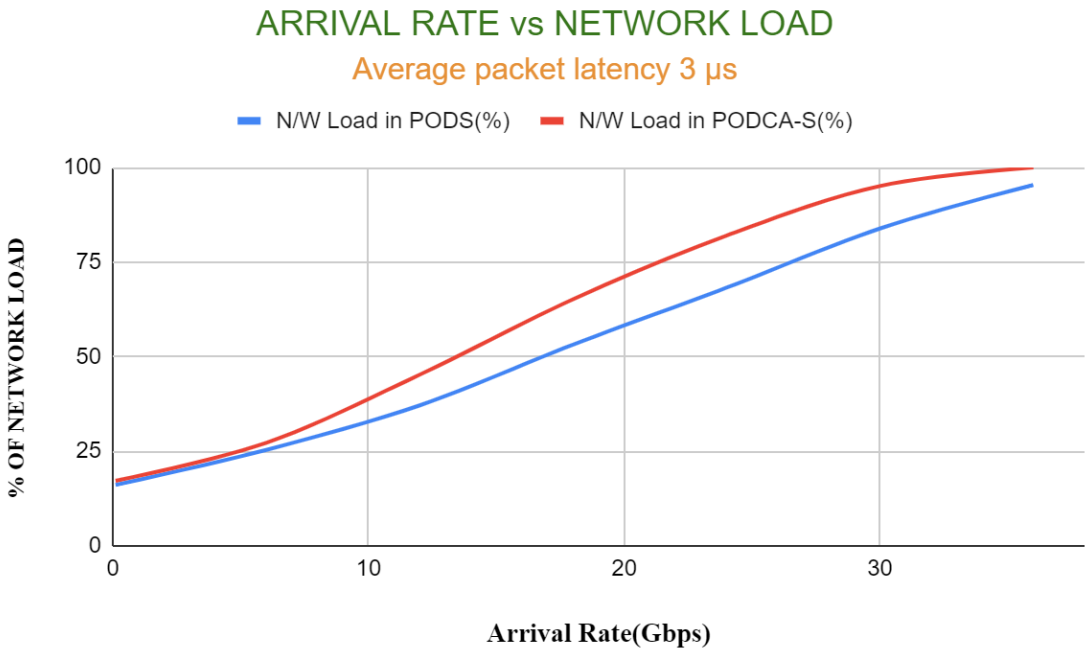


Fig 5: Plot for Arrival Rate vs Network Load

Table: 3. Latency with respect to packet arrival rate

Arrival Rate(Gbps)	0.1	6	12	18	24	30	36
PODS Latency (µs)	2.1	2.27	2.45	2.28	3.35	3.82	4.78
PODCA-S Latency (µs)	2.1	2.4	3.2	3.9	4.65	6.3	8.9

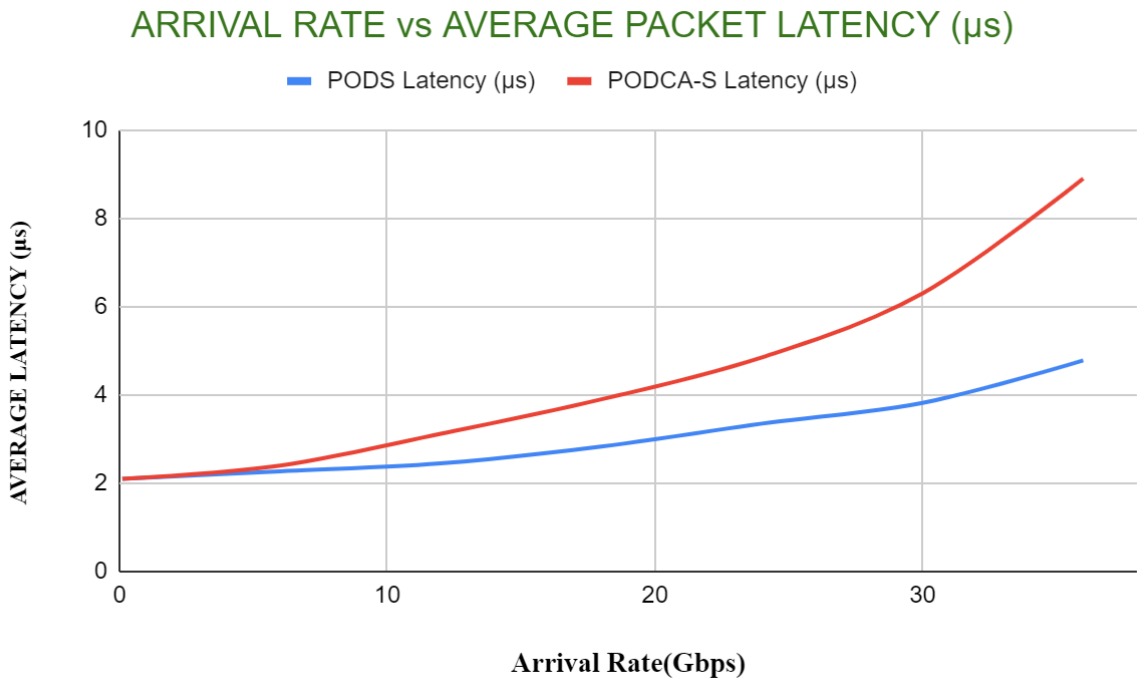


Fig 6: Plot for Arrival Rate vs Average packet latency

Table: 4. Blocking Probability respect to Network Load

NETWORK LOAD	0.7	0.75	0.8	0.85	0.9	0.95	0.99
Blocking Probability in PODS with loopback	0	0	0.001	0.002	0.004	0.0069	0.009
Blocking Probability in PODS without loopback	0.0535	0.061	0.068	0.076	0.084	0.095	0.113

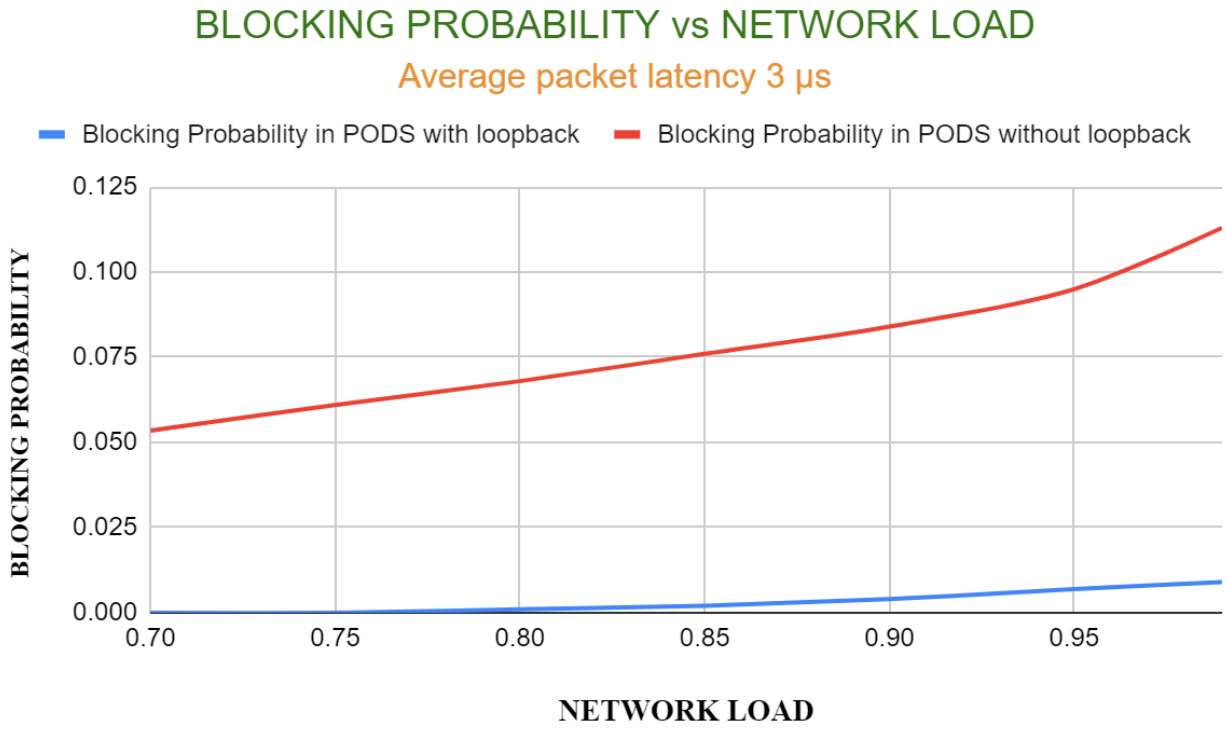


Fig 7: Plot for Blocking Probability vs Network Load

5. Conclusions

In this paper, we presented Low-latency Switching Architecture for DCN using Passive Optical Datacenter Switch (PODS). An efficient wavelength assignment algorithm is applied in the architecture to reroute the packet through AWGR for congestion control. From the simulation results it is observed that latency of the proposed framework is improved by 46.3% and blocking probability is improved by 92% under full load condition. These simulation results ensure that our proposed framework can accommodate more traffic that in turn improves the scalability of the framework. So we can claim that our proposed framework surely will be an efficient solution for next generation DCN.

6. References

1. Cisco System Inc., “Cisco visual networking index: Forecast and methodology, 2016–2021,” San Jose, CA, USA, White Paper, 2017.
2. International Business Strategies (IBS) Inc., “Impact of AI on electronics and semiconductor industries,” Technical Reports, 2020. https://www.ibsinc.net/editorial-calendar

3. IEA (2022), Global Energy Review: CO2 Emissions in 2021, IEA, Paris https://www.iea.org/reports/global-energy-review-co2-emissions-in-2021-2, License: CC BY 4.0
4. Z. Wang et al., “High-radix nonblocking integrated optical switching fabric for data center,” J. Lightw. Technol., vol. 35, no. 19, pp. 4268– 4281, 2017
5. Z. Wang et al., “Systematic exploration of high-radix integrated silicon photonic switches for datacenters,” in IEEE/ACM International Conference on Computer-Aided Design (ICCAD), pp. 1–8, Nov 2019.
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. 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.
9. 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.
10. J. Kim, W. J. Dally, and D. Abts, “Flattened butterfly: a cost efficient topology for high-radix networks,” ACM SIGARCH Comput. Archit. News, vol. 35, no. 2, pp. 126–137, 2007.
11. A. Greenberg, J. R. Hamilton, N. Jain, S. Kandula, C. Kim, P. Lahiri, D. A. Maltz, P. Patel, and S. Sengupta, “VL2: a scalable and flexible data center network,” ACM SIGCOMM Comp. Commun. Rev., vol. 39, no. 4, pp. 51–62, 2009.
12. C. Guo,G.Lu, D. Li, H. Wu, X. Zhang, Y. Shi, C.Tian, Y. Zhang, and S. Lu, “BCube: a high performance, server-centric network architecture for modular data centers,” ACM SIGCOMM Comp. Commun. Rev., vol. 39, no. 4, pp. 63–74, 2009.
13. 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.
14. R. Matsumoto et al., "Performance Analysis of Scalable Optical Circuit Switch Employing Fast-Tunable AMZI Filters for Coherent Detection," in Journal of Lightwave Technology, vol. 41, no. 3, pp. 934-948, 1 Feb.1, 2023, doi: 10.1109/JLT.2022.3200722.