IEEE Access
Multidisciplinary ⋮ Rapid Review ⋮ Open Access Journal

Received January 27, 2021, accepted February 5, 2021, date of publication February 9, 2021, date of current version March 23, 2021.

Digital Object Identifier 10.1109/ACCESS.2021.3058135

Segment Switching: A New Switching Strategy for Optical HPC Networks

JOSÉ DURO, SALVADOR PETIT, MARÍA E. GÓMEZ, AND JULIO SAHUQUILLO, (Member, IEEE)

Departamento de Ingeniería de Sistemas y Computadores, Universitat Politècnica de València, 46022 Valencia, Spain

Corresponding author: José Duro (jodugo1@gap.upv.es)

This work was supported in part by the Ministerio de Ciencia, Innovación y Universidades and in part by the European ERDF under Grant RTI2018-098156-B-C51.

ABSTRACT Photonics are becoming realistic technologies for implementing interconnection networks in near future Exascale supercomputer systems. Photonics present key features to design high-performance and scalable supercomputer networks, such as higher bandwidth and lower latencies than their electronic supercomputer networks counterparts. Some research work is focused on conventional network topologies built with photonic technologies, with the aim of taking advantage of photonic characteristics. Nevertheless, these approaches fail in that they keep low the network utilization. We looked into this downside and we found that circuit switching was the main performance limitation. In this article we propose a new switching mechanism, called *Segment Switching*, to address this constraint and improve the network utilization. Segment Switching splits the circuit in *segments* of the whole path, and uses buffering on selected nodes on the network. Experimental results show that the devised approach significantly outperforms photonic circuit switching in conventional torus and fat tree networks by 70% and 90%, respectively.

INDEX TERMS Interconnection networks, simulation, photonic technology, exascale supercomputers.

I. INTRODUCTION

In the last few years different projects around the world have focused on the design of a single supercomputer able to reach exascale computing performance. Exascale computing will improve the performance of important scientific applications like weather forecasting and medicine. This has motivated countries around the world compete with each other in the race to design supercomputers to reach exascale computing. These systems, however, need to face the performance of the major system components including, computational power, interconnects, distributed storage, and energy management. This article focuses on the interconnection network.

Designing efficient and high-performance interconnection networks for exascale supercomputers is a major challenge because communications requirements exponentially grow with the number of nodes in the network. To accomplish this challenge, photonics is a promising technology mainly due to the huge network bandwidth this technology provides compared to electronics. Some attempts have been made in this direction. For example, the Tofu Interconnect D (TofuD) [1] implemented in the Fugaku supercomputer (the new top in the Top500 list in June 2020 [2]) is partly implemented with photonic links. Further attempts need to be made in the design of fully photonic interconnection networks. To this end, the designer of the interconnection network must consider photonic technological constraints to implement the topology, the routing algorithm and the switching strategy. Most research efforts have focused on the two former while paying little attention to the switching strategy. In contrast, this article focuses on new switching mechanisms for future photonic exascale networks.

As photonics technologies maturate, it is expected that implementing buffers in photonic routers will become feasible in the next coming years. Consequently, existing photonic network approaches commonly assume circuit switching as this mechanism does not require buffers in the routers. Instead, circuit switching reserves resources along the path of the message before it is sent, so the message is not blocked on its way to the destination. In contrast, other works such as Data Vortex [3] implement packet switching for a fully optical *bufferless* network, jointly with a new topology and routing algorithm. Nevertheless, traditional packet switching is not implemented but a bufferless variant, and packet contention is resolved by a distributed deflection routing control scheme. This scheme works by miss-routing packets when

The associate editor coordinating the review of this manuscript and approving it for publication was Zeev Zalevsky.

they cannot follow a minimal path due to conflicts in the network resources. Deflection-based approaches work well when the network presents a low utilization and the paths are not long; however, when the traffic is not low enough they accelerate the network towards saturation [4].

Regarding the switching mechanism, recent studies [5], [6] claim that circuit switching is a better option than packet switching for optical networks since it improves *switching time*, which refers to the time needed for electronic components to establish a new optical configuration. Circuit switching, which does not require buffering support, works well with short message paths because they minimize the time resources are reserved, which reduces reservation conflicts. Note that reservation conflicts should be avoided because they lower the network utilization, and consequently decrease its performance. More precisely, improving the network utilization translates into higher network performance, which translates into overall system performance gains [7]. In future exascale networks, however, paths are expected to be longer due to huge number of nodes in the network. This means that new switching techniques need to be devised. Note, that if *pure* circuit switching is used then network resources are blocked for a longer time, which causes this switching technique to present a poor utilization of the network resources mainly due to the high number of reservation conflicts. We identified this problem in a previous work [8], where we showed that photonics can provide important performance improvements when applied to conventional topologies like torus and fat tree networks. Nevertheless, we found that photonic links were underused, which indicates that there is a high potential to improve performance.

The aim of this article is to improve the network performance of photonic networks for exascale computing. With this aim, we focus on improving the network utilization by proposing a new switching mechanism that better exploits the main features of photonics (i.e. number of channels and aggregated bandwidth). In particular, we propose *Segment Switching*, a novel switching strategy which allows to improve network utilization and, therefore, network performance in torus and fat tree topologies. Segment Switching improves network utilization by splitting the circuit from message source to destination into smaller *segments*. More precisely, instead of reserving the entire route from source to destination, we reserve a segment or fraction (defined by the availability of network resources) or the entire route. In this way, unlike circuit switching, long paths are not reserved at the same time, so reducing reservation conflicts and allowing the transmission to proceed even if the path is not fully reserved.

To be effective, Segment Switching requires a limited number of buffering. Nevertheless, we show that with current photonics technologies, which allow a small number of buffers to be implemented in a photonic switch, the proposal is feasible and provides significant performance benefits. Experimental results show that Segment Switching with only 1MB buffer on a quarter of the network switches improves speedup by 25% on a torus network, and when buffers are included at the top level of a fat tree network, performance improves by 30%.

The remainder of this article is organized as follows. Sections II and III presents the related work and photonics background. Section IV describes the main contribution of this work. Section V introduces the experimental setup. Section VI discusses the obtained results and finally, Section VII presents some concluding remarks.

II. RELATED WORK

Over the last years photonics have become a major topic of interest to the scientific community, mainly due to the advantages offers compared to electronics.

Some approaches have focused on hybrid opto-electrical architectures. Helios [9] and HydRa [10] implement a network consisting of two layers. In Helios, the top layer (Top-of-Rack) consists of electronic switches based on packet switching, and the lower layer is composed of optical switches used for all-to-all communication with each other. In HydRa the main focus is on the design of a software-optimized low-cost network controller. Tofu Interconnect D (TofuD) [1] developed by Fujitsu is the last version of the Tofu Interconnect [11], [12], based on an irregular 6-D torus network that uses photonic links to interconnect neighbouring nodes. Each TofuD router connects four core-memory groups through six internal interfaces to access the torus. SUDOI [13] also proposes a hybrid network consisting of a multi-layer architecture for ubiquitous data center, that implement optical communications in the intermediate layer, interconnecting users accesses with the data center.

Other approaches are based on all-optical networks, DOS architecture [14] propose an optical switch architecture based on Arrayed Waveguide Division Multiplexing (AWGR) with a loopback shared buffer system. An enhanced approach called LIONS [15] presents different loopback buffers for DOS architecture. The TOR of the OSA architecture [16] is composed of Wavelength Selective Switches (WSS) and Optical Switching Matrices based on MEMS (MicroElectroMechanical System). Another proposal with this scheme is PROTEUS [17] that proposes reconfigurable topology in the MEMS. Elastic optical networks [18], based on Space Division Multiplexing (SDM), provide a variable bandwidth per switch depending on network demands or the resource assignment policy [19].

P-Torus [20] an architecture based on a two layers switching. In the lower layer, TORs are interconnected among them through a 4×4 torus. TORs also connected to an Interconnection Passive Aggregation (IPA) switch. In the upper layer, IPAs are interconnected in a all-to-all network. OPSquare [21] is an architecture based on two inter- and intra-cluster networks, where TORs have two Wavelength Division Multiplexing (WDM) bi-directional optical links, one connected to the inter-cluster switch and the other to the intra-cluster switch. Intra-cluster connections take a single hop, while at most two hops are needed to reach a node in a rack of another clusters. In the HiFOST [22] architecture, the TORs of each

cluster are interconnected each other by a flow controlled Fast Optical Switch (FOS), which provides both intra- and inter-cluster connectivity.

Data Vortex [3] proposes a new topology and routing algorithm that implements packet switching and resolves packet contention with a distributed deflection routing control scheme. It is based on a multistage network architecture based on a banyan network [23] with a structure of concentric cylinders (routing stages) through which, the package, once injected, can only proceed from inner stages to destination, despite not following a minimal path.

III. BACKGROUND

This section provides the background on the basics of circuit switching and wavelength-division multiplexing (WDM). Then, we explain how circuit switching works on photonics using WDM identifying the main disadvantages.

1) CIRCUIT SWITCHING

To send a message trough a circuit-switched network, a path (or circuit) is established for the message from source to destination. The selection of the path is performed by reserving the links that compose the circuit. Once all the links of the circuit are reserved, the message can be injected from the source to reach the destination without any contention along the path.

One of the main drawbacks of conventional circuit switching is that if any link of the path cannot be reserved (for instance, it is being reserved for another circuit), the transmission of the message must wait until the link is released.

This is one of the main causes why circuit switching is not usually used in current electronic networks, where packet switching is the preferred approach. Nevertheless, circuit switching has been recently applied to optical networks since the integration of a high number of buffers (enough to efficiently support packet switching) is not trivial in photonics. Moreover, link reservation contention can be mitigated with WDM.

2) WDM

WDM allows multiplexing different wavelengths of light onto the same optical fiber or link, thus, physically enabling several transmissions (e.g., as much as the number of multiplexed wavelengths) per link. The maximum amount of wavelengths that can be multiplexed onto a link is limited by the optical communication band [24] and depends on the minimum spacing that can be implemented between two consecutive wavelength frequency values without causing interference. Nowadays, a 100GHz channel spacing is typically used, which gives 40 wavelengths per link [25], but this spacing can be reduced in order to populate a link with more wavelengths. For instance, 50GHz spacing allows populating the link with 80 wavelengths, and, as shown more recently in [26], 160 wavelengths can be achieved with 25GHz spacing.

Each wavelength provides a given bandwidth and the bandwidth of a link, known as aggregated bandwidth, is computed as the sum of the bandwidths provided by all the wavelengths multiplexed onto the link. This aggregated bandwidth is distributed among several *channels*, and each channel is used for a different transmission. A previous study [8] found that five 320-Gbps channels provide the best performance for an aggregated link bandwidth of 1.6 Tbps. Henceforth, this will be the number of channels assumed in this work.

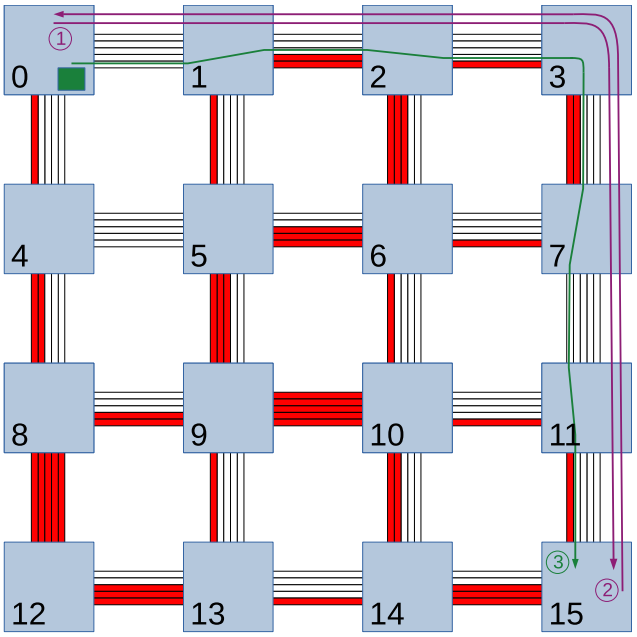


FIGURE 1. Message transmission example with circuit switching and WDM.

3) PUTTING THEM ALL TOGETHER: CIRCUIT SWITCHING WITH WDM

Let us explain how circuit switching works jointly with WDM through a working example.

Suppose the network topology is a 16-node 2D-mesh network like that shown in Figure 1. Consider that a message is sent from node 0 to node 15 using X-Y routing. Assume that each link of the mesh implements five channels and some of the channels (highlighted in red) are already reserved. To send the message, three steps are required. In the step ①, a *reservation* message is sent from node 0 to node 15. As the reservation message advances through the path, it reserves free channels to perform the transmission. Once the reservation message arrives to destination, it is sent back to node 0 (step ②). The second step has two main purposes: i) to configure the optical switches along the path to use the reserved channels for the circuit, and ii) to notify node 0 whether the circuit has been established or not. In case the circuit is established successfully, in step ③, the reserved channels (highlighted in green) are used to send the message through the circuit.

Note that if one of the links in the path does not have free channels, the reservation will fail, similarly as it occurs in conventional circuit switching. This means that a single link without available channels in enough to prevent the

transmission and, consequently, the use of the remaining links along the path. Moreover, the problem may affect several transmissions contending for the same link, causing, as experimental results will show (Section VI), an overall reduction of link utilization and thus of network performance. This issue is tackled by the proposed approach presented in the next section.

IV. OPTICAL SEGMENT SWITCHING

This section discusses the devised approach to improve photonic circuit switching on classical network topologies.

As discussed above, due to contention at the channel reservation stage, circuit switching does not leverage the WDM's huge potential on performance, resulting on underutilization of the network.

WDM improves network utilization by allowing several transmission channels per physical link, which enables the reservation of one channel for a circuit without precluding the reservation of the remaining channels in the same link for additional circuits. However, when all the channels in any link of the message path are already reserved, the circuit cannot be established. Therefore, circuit switching constraints still remain in WDM-based photonic networks.

The main shortcoming of circuit switching is that the chance of finding a free channel diminishes as the length of the path increases, thus aggravating in exascale networks. The previous reasoning means that there is a need of enabling the capability of establishing *circuit segments* instead of the entire circuit in order to address this shortcoming. With a segment of the circuit we refer to a fraction of the entire message path. By deploying this capability we pursue that if the entire circuit cannot be allocated, at least the first segment is reserved, allowing the message to advance while improving the utilization of the first links of the circuit. Once the first segment is used for transmission, consecutive segments will be reserved until destination is reached.

Note that this approach imposes several requirements on the network design. First, buffering support is required to store data at the end of each established segment. This involves including buffering in a subset of the network switches.

Second, messages should be packetized (divided into small packets of the same size) to allow packet storage when the buffer does not present enough capacity to store the full message.

Finally, an algorithm to reserve the segments that compose a circuit is required. Below, these design requirements and the proposed design options are discussed.

A. BUFFERING SUPPORT

Although it is technically feasible to implement buffering in current photonic switches, this technology is not mature enough to be applied in a production and commercial system. However, its cost suggest to reduce buffering when possible. In this sense, packet switching techniques would be costly design choice option, since they require a high amount of

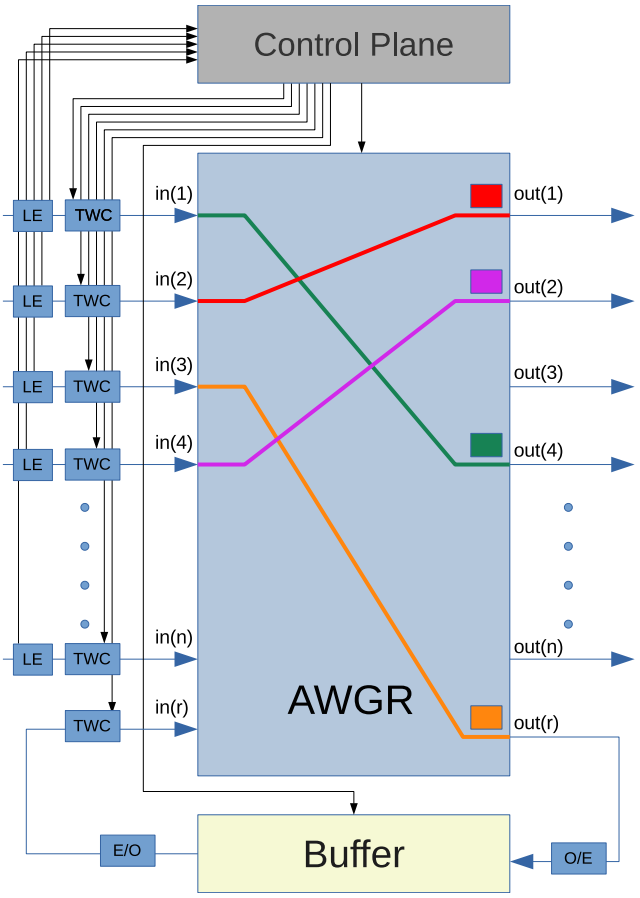


FIGURE 2. Photonic switch block diagram with *n* photonic inputs and outputs and a dedicated input and output to communicate with an associated buffer.

buffers (e.g. one buffer per input and output for each switch). Nevertheless, some recent proposals of photonic switches that can be leveraged by our approach integrate a small amount of buffers [14], [15].

In particular, we leverage the *Lions* design [15], and, based on this design, we devise an optical switch especially tailored for our proposal. Figure 2 shows the block diagram of the devised optical switch. The central module is an Arrayed Waveguide Grating Router (AWGR) [27], [28], a passive device that interconnects inputs and outputs at photonic channel level. At the input ports, the Label Extractors (LE) and Tunable Wavelength Converters (TWCs) [29], [30] control the wavelengths used to transmit data, thus the wavelengths of the input and output channel can differ.

The proposed switch includes an associated buffer connected to the AWGR through an input channel and an output channel. The buffer can be implemented either with electronic or photonic technology.

Although electronic buffers require signal conversions (opto-electrical and electro-optical), this time is negligible (in the order of picoseconds) [31], [32] compared to the network cycle (nanoseconds) [33], [34], thus we use electronic buffers due to the storage capacity they provide. Note that by just adding a single buffer connected to one input and output

port of the AWGR, we do not modify the timing behavior of the original Lions design.

Finally, communications of the AWGR with external links and the buffer are orchestrated by the control plane. This component is an electronic component that configures the TWCs to establish the circuits. Note that the devised design allows the circuit to be partitioned into segments. With the exception of the last segment, whose destination matches the destination of the circuit, each segment ends in a buffer. Conversely, there is a buffer at the source of every segment, except for the first segment, which begins at the source of the circuit.

B. MESSAGE PACKETIZATION

Without message packetization, buffers along the path should be oversized to allow storing messages of any length. To overcome this problem and make an efficient use of the buffer, messages are split into small fixed-size packets. Each packet is handled as an independent unit regarding routing and storage in the buffers.

As it will be shown in Section VI-A, message packetization improves network utilization. This is mainly due to two reasons: i) due the small packet size, it is more likely to find a buffer with enough space to hold it, so packetization enables establishing segments that could not be established when considering the whole message; and ii) the time that a channel is busy when transmitting a packet is usually much shorter than the time required for the transmission of the whole message, therefore reducing link contention when reserving channels, and thus improving network utilization.

C. SEGMENT RESERVATION

Algorithm 1 presents the pseudo-code for reserving a segment of the circuit. The algorithm has two main parts. In the first part, the route of the transmission is traversed reserving the required channels for the segment (lines 3–6). If the whole route is traversed and all channel reservations succeeded until destination, then the whole circuit can be established and no buffering is needed along the path (lines 7–11).

However, if one of the channel reservations fails, then the second part of the algorithm, which reserves a buffer entry for the segment, is performed. In this part, the route is traversed back from the node where the channel reservation failed to the source node (lines 14–20). The backward traversal ends as soon as a node with a free buffer entry is found. In this way, we ensure that the end of the segment is as close as possible to the destination of the entire route.

After the backward traversal has been performed, in case the buffer entry reservation succeeded, the reserved segment spans from the source node to the node with the buffer entry (lines 21–23). Otherwise, similarly to as done in conventional circuit switching, the segment reservation is retried (lines 24–26).

Once a segment has been reserved, the segment is used to perform the transmission. After that, a new segment must be reserved from the intermediate node to the destination (or other intermediate node). This implies additional calls to the algorithm until the destination is reached.

Algorithm 1: Segment Reservation Algorithm

```
1  /*** CHANNEL RESERVATION ***/
2  node_i ← source node;
3  repeat
4      try to reserve a free channel in node_i to next node;
5      node_i ← next node;
6  until link reservation fails or node_i = destination node;
7  if node_i = destination node then
8      /* The segment is the whole circuit:
9      no buffer reservation is needed. */
10     return segment [source node ... destination node];
11 end
12
13 /*** BUFFER RESERVATION ***/
14 repeat
15     try to reserve a free buffer entry in node_i;
16     if buffer reservation fails then
17         release previously reserved channel;
18     end
19     node_i ← prev node;
20 until buffer reservation succeeds or node_i = source
     node;
21 if buffer reservation succeeded then
22     /* Return the reserved segment */
23     return segment [source node ... node_i];
24 else
25     /* A segment cannot be reserved, retry */
26     retry segment reservation algorithm;
27 end
```

Although potentially any switch may include a buffer, in this work, we study the impact on performance of placing buffers in only a subset of the network nodes. In this way, savings can be achieved both in energy consumption and implementation complexity. To perform this study, we focus on torus and fat tree topologies and, for each topology, we devise distinct buffer layouts across nodes. For the fat tree topology, buffers are only deployed on the switches of specific network levels, prioritizing top levels as they are more prone to become contention points of the reservation algorithm. For the torus, we ensure that when advancing in a given dimension there is a buffer every n switches. For instance, for an $8 \times 8 \times 8$ torus, with a configuration where a quarter of nodes implement buffers, if there is a buffer in node 0 (0,0,0), next buffer in X dimension will be in node 4 (3,0,0), in Y dimension will be in node 32 (0,3,0) and in Z dimension will be in node 256 (0,0,3).

Figure 3 plots an example of segment reservation and transmission. As in Figure 1, a message is sent from node 0 to node 15 following X-Y routing. Notice that both buffer distribution and buffer availability is included in the figure. In this case, no free channels are available after node 11.

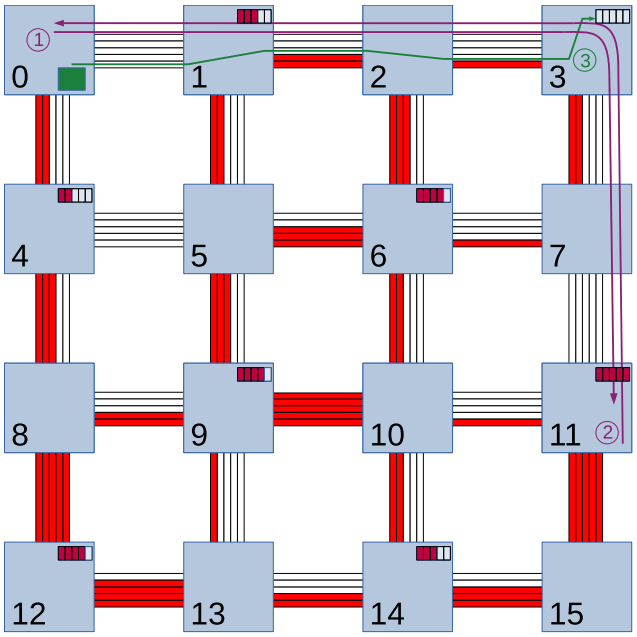


FIGURE 3. Example of a segment reservation and transmission in a 2D-mesh network with a buffer every 2 nodes in each dimension.

In step ①, which corresponds to the first part of Algorithm 1, the reservation message tentatively reserves the channels it crosses until node 11 is reached. Then, the message is sent back (second part of the algorithm) to reserve a free buffer entry and establish the optical circuit ②. Since the buffer in node 11 is full, the entry is reserved in buffer 3 (the closest that has an available slot). Note that this message also releases the channels tentatively reserved in the first step between buffers 3 and 11 since they will be not take part by the segment. Once the reservation message notifies node 0 that the segment destination is at node 3, node 0 performs the transmission ③. After that, a new reservation and transmission must be made from source node 11 to destination node 15.

V. EXPERIMENTAL SETUP

This section describes phINRFlow (photonic Interconnection Network for Research Flow-level Simulation Framework), the simulation framework used to model the proposed approach, presenting the optical and network configurations used for the performed experiments, as well as the traffic patterns considered in these experiments.

phINRFlow is a flow-level simulator for photonic interconnects that inherits functionality from INRFlow [35], originally developed with the aim of modeling electrical networks. It implement multiple, direct and indirect, network topologies (e.g. cubes, dragonfly or trees) and multiple traffic generation methods (e.g. synthetic or traces). It is highly scalable and includes the main components necessary for modelling photonic interconnects. These capabilities enable us to evaluate the system under realistic loads, giving insights to its viability as a candidate for exascale systems.

In this work we model and evaluate two classical network topologies consisting of 1728 nodes, a 3D-torus of $12\times12\times12$ nodes and a 12-ary 3-tree under a synthetic traffic where each node sends 100 messages to random destinations. Both networks are evaluated with loads composed of messages of two lengths: 80% of short messages (4KB) and 20% of long messages (512KB). In order to obtain more accurate results, for each network configuration, 20 simulations have been performed with different seeds.

The latencies of the components of the photonic switch depicted in Figure 2 are taken into account according to the literature. In particular, we have considered the time that LEs require to route the messages [36]–[38], the switching time of TWCs [39], and the delays of opto-electrical (O/E) and electro-optical (E/O) converters [31], [32]. Recall that, as stated in Section IV-A, the overall delay added by these components is in the order of picoseconds, which can be considered negligible, considering that optical network switching time is in the order of nanoseconds [33], [34]. Moreover, these times can be much shorter according to novel research regarding different materials such as graphene, reaching the order of femptoseconds [40].

Finally, circuit reservation time is also taken into account in the experimental setup. This time is not fixed for all routes, but it depends on the route length. This time takes a simulation cycle per hop, both forward to reserve the segment and backward to establish it.

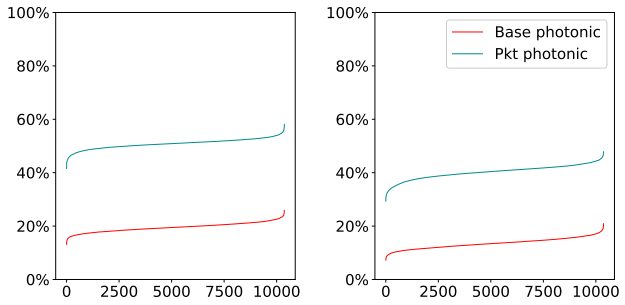
VI. EXPERIMENTAL RESULTS

This section evaluates the mechanisms proposed in Section IV. First, we explore the impact of message packetization based on a given MTU (Maximum Transmission Unit). The circuit is established only during the transmission of a packet, and released after sending the packet. This will allow to block the network resources for a shorter time. After analyzing the impact on performance, we analyze the effect of adding a small amount of buffers in the network to allow splitting the entire path, which translates into shorter *segments*, thus reducing the time the network resources are blocked by the transmission of a packet.

A. IMPACT OF PACKETIZATION VS PURE CIRCUIT SWITCHING

The first approach of this work is aimed at maximizing the network utilization by limiting the maximum amount of information that is sent at the same time in classical topologies using optic technology. We define an MTU of 4 KB (based in the size of short messages) in the network, but unlike typical circuit switching, we reserve network resources only for the transmission of only one packet. Once the packet is sent, the used resources are released. This way will allow other nodes to reserve and use resources earlier, therefore, improving the utilization of the network resources.

Figure 4 shows the link utilization, arranged in increasing order, in the studied network configurations.



(a) *Torus* (b) *Fat tree*

FIGURE 4. Channel utilization in bufferless photonic networks. An MTU is not defined (in red) and an MTU is defined (blue).

This metric has been calculates with Equation 1:

$$U_{link} = \frac{T_{used}}{T_{total}} \tag{1}$$

It can be appreciated that in the torus topology, on average, link utilization rises from 20% in the base photonic network (red color) to 50% packetized photonic network (blue), which reaches 60% in the most used links. Regarding the fat tree topology, utilization rises on average from 15% to 40%. This means that in both topologies, links are being used to transmit data for a larger fraction of the time, and therefore, more data are transmitted in less time.

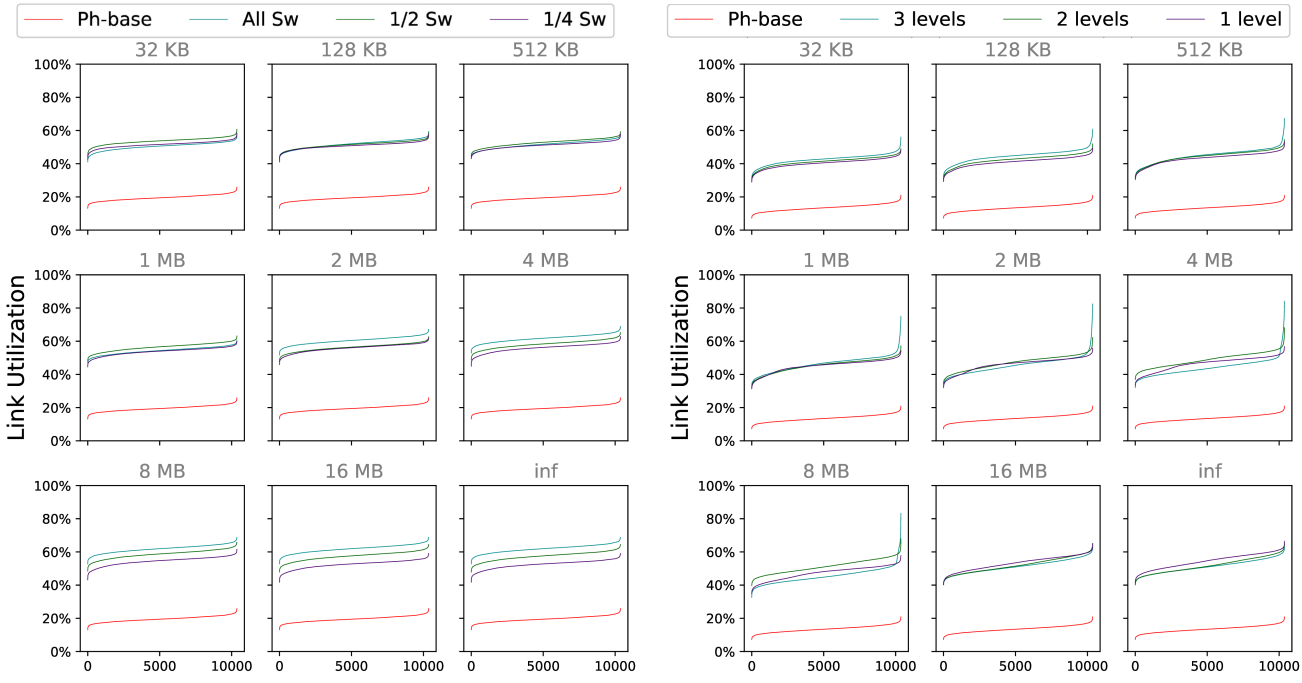
B. IMPACT OF BUFFERS ON THE NETWORK PERFORMANCE

1) LINK UTILIZATION

In the previous section we have shown that message packetization helps the photonic network to increase link utilization and network throughput. Next, we analyze the impact of introducing a small amount of buffers in the network, shortening the paths and reducing the time that network resources are reserved for sending a packet. These buffers can be located in all the switches or only in a subset of them. Buffer sizes have been assumed to be multiples of the defined MTU (4KB). For instance in a 32KB buffer, 8 messages can be placed.

Figure 5 shows that link utilization significantly increases when buffers are included in the network, both in the torus and in the fat tree, even with a small quantity of buffers. The bufferless baseline networks (colored in red), achieve an average utilization around 20%, while the buffered networks obtain an average link utilization over 50%. As commented before, buffers are included in some of the networks switches but not necessarily in all of them. Notice that in the switches where we introduce buffers there is just one buffer in the switch as shown in Section IV-A. Moreover, different buffer sizes (from 32KB to unlimited) have been evaluated. In the torus topology, buffers are deployed in every switch (labelled as *All Sw* in the figure), in a half of the switches (1/2 Sw in the figure) and in a quarter of the switches (1/4 Sw in the figure). As can be observed in Figure 5a, buffering significantly increases link utilization regardless of the number and size of buffers. Increasing the number of buffers provides a marginal improvement of link utilization, which becomes more evident when the buffer size increases. The link utilization for a 32KB buffer is on average by 50%, while for the infinite buffers is around 60%.

In the fat tree, buffers are deployed in the last level (labelled as 1 level in Figure 5b), in the two upper levels (2 levels) and in all the 3 levels of the fat tree. Again, buffering provides a



(a) *Torus* (b) *Fat tree*

FIGURE 5. Link utilization when buffers are included in (a) the torus topology and (b) in the fat tree.

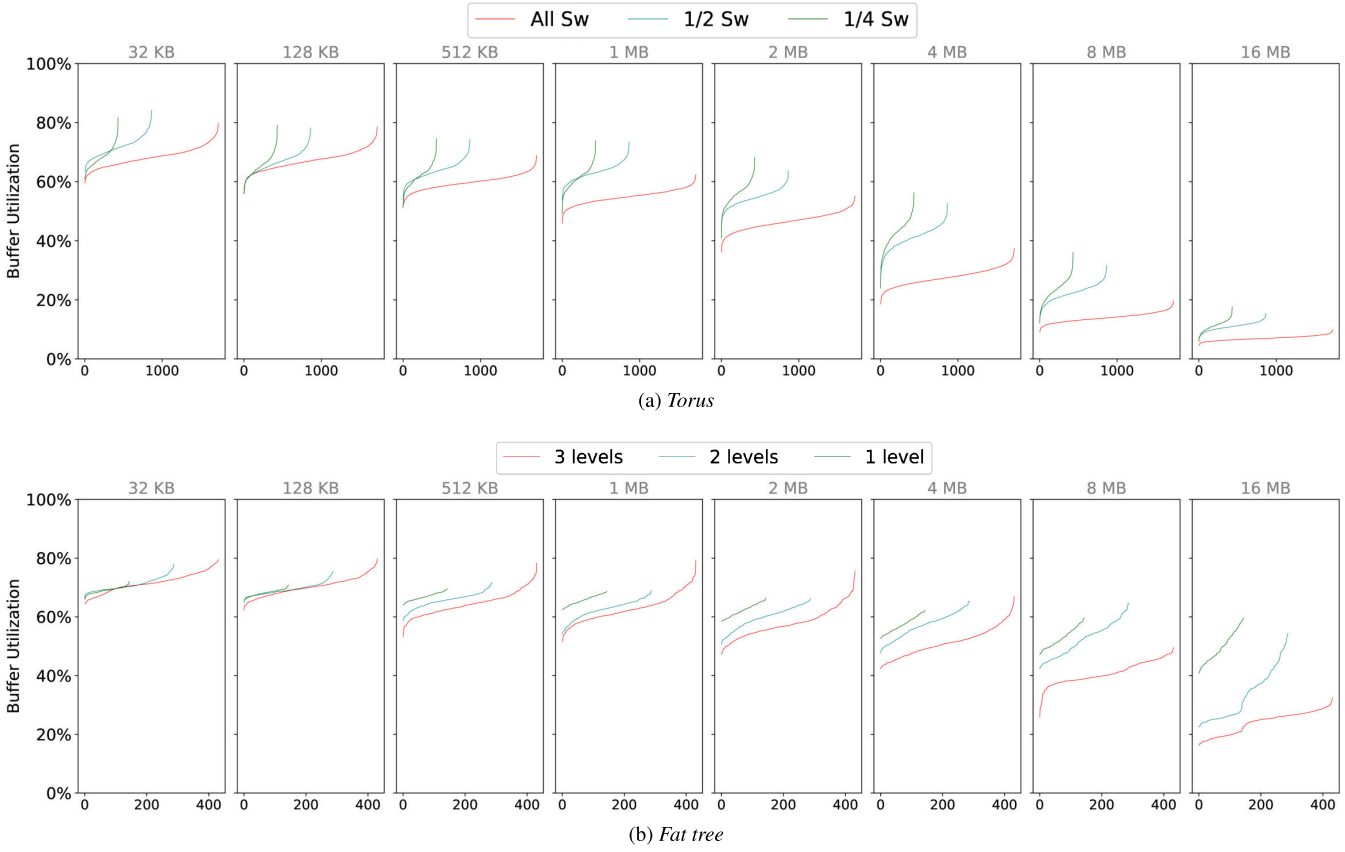


FIGURE 6. Buffer Utilization in (a) the torus topology and (b) in the fat tree topology.

significant link utilization increase, even with just one level. If buffers are introduced in more levels, marginal improvements are obtained on average, being the utilization gain remarkable only in a small number of links, which reaches by 80% utilization when buffers are in all the 3 levels of the fat tree. The buffer size has a greater impact on link utilization improvement. The average utilization for a 32KB buffer is by 40%, and for hypothetical unlimited buffers by 50%. These increases in the link utilization will translate into network performance enhancement as shown in Section VI-C.

2) BUFFER UTILIZATION

So far we have studied the impact of the number of buffers on link utilization, this section analyzes the utilization of the buffers. Figure 6 presents the results for the studied designs, which has been calculated with Equation 2. Like in previous study, each point of the line corresponds to one buffer. Notice that the length of the 3 lines of the same plot differ, this happens because the number of buffers also do that.

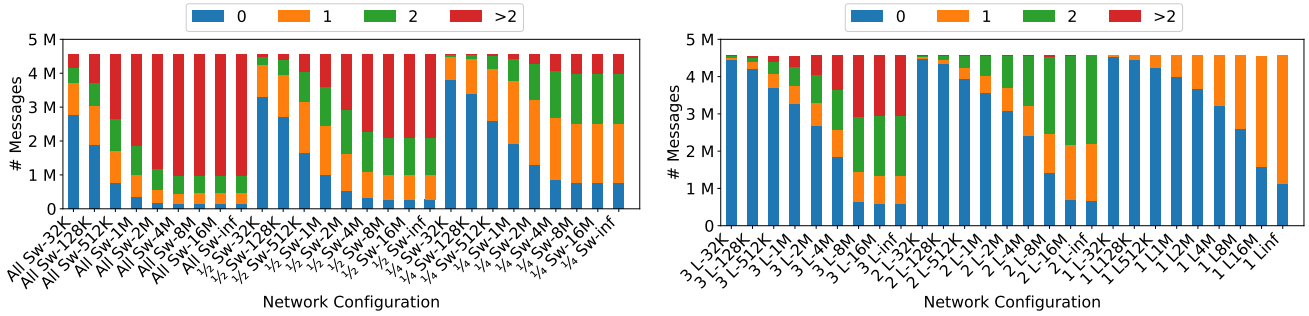
$$U_{buffer} = \frac{\sum_{1}^{n_{cycles}} \frac{Slots_{occupied}}{Slots_{total}}}{n_{cycles}} \quad (2)$$

In the case of the torus topology, it can be observed in Figure 6a that the larger the buffers the lower the utilization. The utilization starts by 70% in the smallest 32KB buffer and goes down to around 10% in largest 16 MB buffer.

This means that large buffers are underutilized and may suggest that they are a waste of resources. This will be corroborated in Section VI-C where we analyze the network performance. Comparing the lines of the same plot, we can see that the larger the buffers the higher the differences among the utilizations drawn in the same plot. This suggests that putting a large amount of resources in the network is not a good policy after a certain amount since they are barely used.

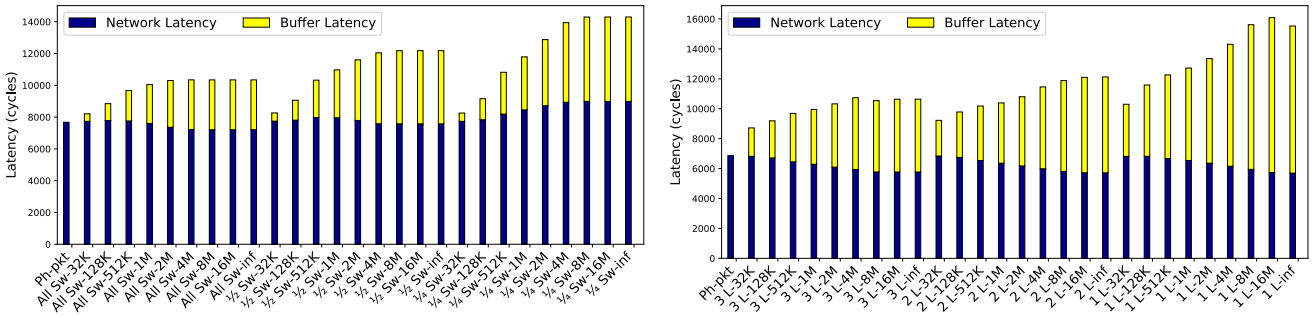
Regarding the fat tree topology, Figure 6b shows that the general utilization trend is similar to the torus, using larger buffers results in less buffer utilization. Nevertheless, there are two differences that should be emphasized. On the one hand, it can be appreciated (looking all the plots of the figure from left to right) that the buffer utilization when buffers are only placed in the last level (red lines) of the fat tree goes down slower than in the torus as the buffer size increases. This suggests that the most used buffers are the ones located in the last level of the network. On the other hand, it can be seen that in this topology the distance among the lines of the same plot widens as the buffer size increases, more than in the torus topology. This also confirms that the last level of the fat tree is the one that most contributes to performance.

After analyzing the buffer utilization, we focus on the path length followed by packets. We analyze how many times packets are buffered before reaching their destination. This information is shown in Figure 7a for each studied network configuration. In the torus topology, the term N Sw-S refers



(a) *Torus* (b) *Fat tree*

FIGURE 7. Re-stored Messages. Number of times that messages are stored over its path from source to destination.



(a) *Torus* (b) *Fat tree*

FIGURE 8. Packet latency in network cycles split in network and buffer latency.

to the buffer layout and the buffer size S. For instance, 1/2 Sw-32K, means that there is a 32KB buffer in half of the nodes. In the fat tree, N L-S indicates that buffers of S size are present in N levels, that is, 2L-1M means that there are buffers of 1MB in the two last levels of the fat tree.

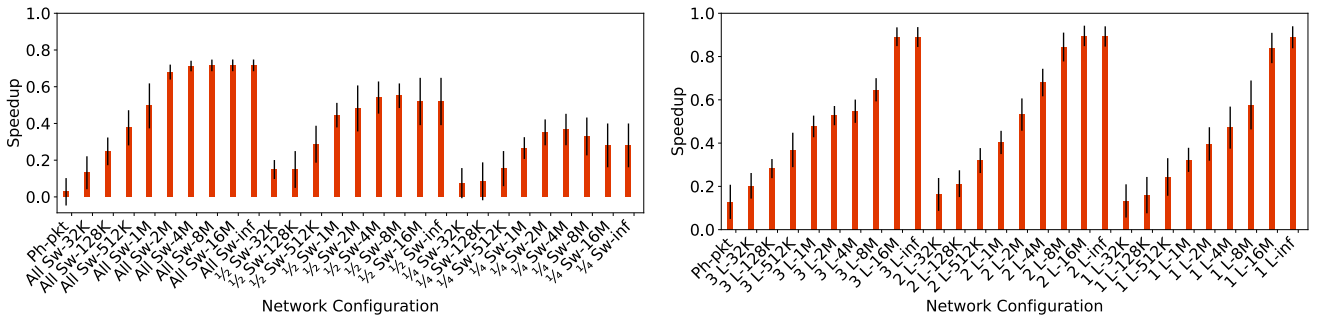
As expected, the larger the number of buffers the higher the number of times packets are buffered. In the torus topology, three main groups of bars can be appreciated in each plot of Figure 7a, corresponding to *All* (buffers in all the switches), *1/2* (buffers in half of the switches) and *1/4* (a quarter of the switches), respectively. In the case of 32KB buffers, 85% of the packets are never stored before arriving to their destination. On the other hand, as the buffer size increases, as more storage resources are available in the network, more times packets are stored. Nevertheless, from a buffer size of around 4 MBs, for a number of buffers in the network, the number of times packets are stored gets stable.

The number of times packets are stored is related to the length of the paths. As more times packets are stored, shorter paths are used. Figure 7a is also related to Figure 6a, as more times packets are stored, the buffers are more used. If we compare on the basis of the same buffering capacity, for instance, 32KB in all the switches versus 128KB in only in a quarter of the switches, it can be appreciated that distributing the buffering capacity among more switches performs better as a higher amount of packets is never stored in intermediate buffers. In the first case (32KB in all the switches), 65% of the packets are sent directly to destination and, in the second (128KB in a quarter), 75% of them are never stored in intermediate buffers.

Figure 7b shows the results for the fat tree. As can be seen, the fat tree presents a different behavior, adding more buffers does not increase significantly the number of times packets are stored. The number of packets that are never stored in intermediate buffers is similar for one, two or three levels with buffers. This can be seen for instance in 1 L-32K, 2 L-32K and 3 L-32K, where packets are sent directly to destination in more than the 95% of the packets. Doing the buffers larger makes increasing the number of times packets are stored. We can see how blue bars are reduced progressively up to a size of 8-16 MB. From this size it stabilizes. In case of having only buffers in the topmost level, packet can be stored only 0 times or once. If there are buffers in two levels then packets can be stored 0, 1 or twice, and so on.

3) PACKET LATENCY

In addition to utilization, a key factor to understand the behavior of the proposal is the impact on packet latency. Figure 8 plots the average latency per packet separated in buffer and network latency. The former latency corresponds to the time a packet waits in an intermediate buffer, while the latter represents the time that the packet takes through the network, which accounts the time from the packet generation to reach the destination (excluding the buffering time). As expected,



(a) *Torus* (b) *Fat tree*

FIGURE 9. Speedup of the network with buffers for (a) torus and (b) fat tree over the Photonic Baseline Network.

buffer latency increases with the buffer size, since message packets can spend additional time in the buffers before reaching the destination. This, in turn, increases the overall latency. Nevertheless, in addition to the latency increase, as mentioned in Section VI-B1, there is also an increase in link utilization that translates into network throughput improvements. Therefore, more messages can be transmitted at the same time, shortening the time needed to transmit the data over the network and thus providing a better performance, as shown in the next section.

C. NETWORK PERFORMANCE

Figure 9 shows the average speedup over the photonic baseline network achieved by the proposed network mechanism with synthetic traces (see Section V). Results are plotted with a 95% confidence interval through 20 executions of each trace with different seeds. The first bar corresponds to the speedup achieved when only a packet is transmitted through each established circuit, but considering no buffering in the network. In the other bars of the plot, buffers are in the network with different amounts and sizes. As studied above, the behavior of the studied topologies differs. Thus, results will be discussed for each topology.

Regarding the torus, as can be seen in Figure 9a, the best results are reached when buffers are deployed in more switches of the network. These results were expected since these networks present the highest link utilization (see Figure 5a). The buffer size is significant only for small buffers, and practically scarce o no additional performance benefits are reached over 4MB buffers. An important point is that given an amount of storage, it is better to distribute it over the network as much as possible than concentrating it on just a subset of switches. This can be observed comparing, for instance, the 1-1M configuration where a speedup by 50% is reached against the 2-2M and 4-4M configurations where 45% and 35% speedups are obtained, respectively.

In the fat tree the trend is different, as shown in Figure 9b. Looking at the *groups of bars* of the figure, it can be observed that similar speedups are obtained regardless of the number of levels where buffers are deployed in the switches. In contrast, when focusing on a given number of levels (e.g. 3L), it can be appreciated that the key parameter for performance in the fat tree is the buffer size, that is, the greater the total buffer size the higher the speedup. Therefore, it can be concluded that in the fat tree, unlike the torus, the best way to distribute a given buffering capacity is to have large buffers in the topmost level of the topology. In contrast, distributing buffers over more switches of the network has no strong impact on performance in the fat tree. This is in line with the link utilization results studied above (see Figure 5b), where it was shown that differences among the three lines of the same plot are marginal, but link utilization increases with the buffer size.

The highlighted differences among topologies appear both due to the topology properties and to the associated routing algorithms. Torus is a direct network topology, where all the switches have nodes injecting packets in the network, while fat tree is an indirect topology, where computing nodes that inject packets in the network are connected only to the first stage or level of the topology. Thus in torus, packets injected in the network by the computing nodes compete with packets in intermediate buffers while the fat tree does not have this inconvenient in all the network switches. Moreover, the organization of the fat tree topology makes that packets first follow an upward sub-path, they have a turnaround in a given level of the topology, and finally they follow a downward phase. The adaptive routing followed in the upward phase of the fat tree allows to avoid conflicts since any of the output ports in the current switch can be used. After the turnaround is performed, conflicts can appear (in the topmost level in most of the cases) so that storing packets in that last phase is highly convenient.

VII. CONCLUSION

Photonics have been shown to be promising technologies for future exascale networks, mainly due to the huge bandwidth they deploy. As a consequence, when using photonics in conventional HPC networks, the link utilization is rather low.

In this article we have proposed Segment Switching aimed at improving the link utilization in conventional networks topologies. Segment Switching relies on two main mechanisms: packetizing messages and buffering. With packetizing, we pursue to reduce the amount of information that is sent at the same time over the route. This way reduces the time

messages block the network resources, and thus the time other nodes wait before injecting messages. This mechanism increases link utilization by 30% on average in the studied topologies, torus and fat tree and, a result, also enhances network performance by 5% in the torus and 10% in the fat tree. With buffering, we pursue to reduce the time network resources are blocked by shortening the circuit established to send a message. Multiple circuit lengths have been analyzed by studying different layouts for allocating buffers to switches. Different number of buffers and buffer sizes have been studied.

Experimental results show that the studied topologies present different buffering demands and require distinct designs. Regarding the torus topology, performance improves when buffering is supported by more switches. The maximum performance is achieved with 4MB buffers. Larger buffers provide scarce or no performance benefits at all. This means that reducing the average circuit length in the torus is more critical for performance than increasing the buffer size. Regarding the fat tree topology, the key parameter is the buffer size. Deploying buffering only in the upwards stage, if it is large enough allows to achieve the best performance. This happens because contention is less frequent in the downwards phase in this topology.

To sum up, for a given buffering capacity, the best distribution is to give a fraction of storage to each switch in the torus and, to accumulate that capacity in the last-stage switches in a fat tree. Segment Switching improves network performance up to 70% and 90%, in the studied torus and fat tree topologies respectively.

REFERENCES

[1] Y. Ajima, T. Kawashima, T. Okamoto, N. Shida, K. Hirai, T. Shimizu, S. Hiramoto, Y. Ikeda, T. Yoshikawa, K. Uchida, and T. Inoue, "The tofu interconnect d," in *Proc. IEEE Int. Conf. Cluster Comput. (CLUSTER)*, Sep. 2018, pp. 646–654.

[2] (Jun. 2020). *Top500 Website*. [Online]. Available: http://www.top500.org/

[3] O. Liboiron-Ladouceur, A. Shacham, B. A. Small, B. G. Lee, H. Wang, C. P. Lai, A. Biberman, and K. Bergman, "The data vortex optical packet switched interconnection network," *J. Lightw. Technol.*, vol. 26, no. 13, pp. 1777–1789, Jul. 2008.

[4] C. Gómez, M. E. Gómez, P. López, and J. Duato, "How to reduce packet dropping in a bufferless NoC," *Concurrency Comput., Pract. Exper.*, vol. 23, no. 1, pp. 86–99, Jan. 2011.

[5] G. Porter, R. Strong, N. Farrington, A. Forencich, P. Chen-Sun, T. Rosing, Y. Fainman, G. Papen, and A. Vahdat, "Integrating microsecond circuit switching into the data center," *ACM SIGCOMM Comput. Commun. Rev.*, vol. 43, no. 4, pp. 447–458, Sep. 2013.

[6] N. Farrington, G. Porter, Y. Fainman, G. Papen, and A. Vahdat, "Hunting mice with microsecond circuit switches," in *Proc. 11th ACM Workshop Hot Topics Netw. HotNets-XI*, 2012, pp. 115–120.

[7] J. Duato, S. Yalamanchili, and L. Ni, *Interconnection Networks*. San Mateo, CA, USA: Morgan Kaufmann, 2003.

[8] J. Duro, J. A. Pascual, S. Petit, J. Sahuquillo, and M. E. Gómez, "Modeling and analysis of the performance of exascale photonic networks," *Concurrency Comput.,Pract. Exper.*, vol. 31, no. 21, Nov. 2019.

[9] N. Farrington, G. Porter, S. Radhakrishnan, H. H. Bazzaz, V. Subramanya, Y. Fainman, G. Papen, and A. Vahdat, "Helios: A hybrid electrical/optical switch architecture for modular data centers," in *Proc. ACM SIGCOMM Conf.*, 2010, pp. 339–350.

[10] K. Christodoulopoulos, D. Lugones, K. Katrinis, M. Ruffini, and D. O'Mahony, "Performance evaluation of a hybrid optical/electrical interconnect," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 7, no. 3, pp. 193–204, Mar. 2015.

[11] Y. Ajima, T. Inoue, S. Hiramoto, and T. Shimizu, "Tofu: Interconnect for the k computer," *Fujitsu Sci. Tech. J*, vol. 48, no. 3, pp. 280–285, 2012.

[12] Y. Ajima, T. Inoue, S. Hiramoto, S. Uno, S. Sumimoto, K. Miura, N. Shida, T. Kawashima, T. Okamoto, O. Moriyama, and Y. Ikeda, "Tofu interconnect 2: System-on-chip integration of high-performance interconnect," in *Proc. 29th Int. Conf. (ISC)*. Leipzig, Germany: Springer, Jun. 2014, pp. 498–507.

[13] H. Yang, J. Zhang, Y. Zhao, J. Han, Y. Lin, and Y. Lee, "SUDOI: Software defined networking for ubiquitous data center optical interconnection," *IEEE Commun. Mag.*, vol. 54, no. 2, pp. 86–95, Feb. 2016.

[14] X. Ye, Y. Yin, S. J. B. Yoo, P. Mejia, R. Proietti, and V. Akella, "DOS: A scalable optical switch for datacenters," in *Proc. 6th ACM/IEEE Symp. Archit. Netw. Commun. Syst. ANCS*, 2010, pp. 1–12.

[15] 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," *IEEE J. Sel. Topics Quantum Electron.*, vol. 19, no. 2, Mar. 2013, Art. no. 3600409.

[16] K. Chen, A. Singla, A. Singh, K. Ramachandran, L. Xu, Y. Zhang, X. Wen, and Y. Chen, "OSA: An optical switching architecture for data center networks with unprecedented flexibility," *IEEE/ACM Trans. Netw.*, vol. 22, no. 2, pp. 498–511, Apr. 2014.

[17] A. Singla, A. Singh, K. Ramachandran, L. Xu, and Y. Zhang, "Proteus: A topology malleable data center network," in *Proc. 9th ACM SIGCOMM Workshop Hot Topics Netw. Hotnets*, 2010, pp. 1–6.

[18] O. Gerstel, M. Jinno, A. Lord, and S. J. Yoo, "Elastic optical networking: A new dawn for the optical layer?" *IEEE Commun. Mag.*, vol. 50, no. 2, pp. s12–s20, Feb. 2012.

[19] H. Yang, Q. Yao, A. Yu, Y. Lee, and J. Zhang, "Resource assignment based on dynamic fuzzy clustering in elastic optical networks with multi-core fibers," *IEEE Trans. Commun.*, vol. 67, no. 5, pp. 3457–3469, May 2019.

[20] C. Chaintoutis, A. Bogris, and D. Syvridis, "P-torus: Torus-based optical packet switching architecture for intra-data centre networks," in *Proc. Photon. Switching Comput. (PSC)*, Sep. 2018, pp. 1–3.

[21] F. Yan, W. Miao, O. Raz, and N. Calabretta, "Opsquare: A flat DCN architecture based on flow-controlled optical packet switches," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 9, no. 4, pp. 291–303, Apr. 2017.

[22] F. Yan, X. Xue, and N. Calabretta, "HiFOST: A scalable and low-latency hybrid data center network architecture based on flow-controlled fast optical switches," *IEEE/OSA J. Opt. Commun. Netw.*, vol. 10, no. 7, pp. 1–14, Jul. 2018.

[23] L. R. Goke and G. J. Lipovski, "Banyan networks for partitioning multiprocessor systems," in *Proc. 25 Years Int. Symposia Comput. Archit. (Sel. Papers) ISCA*, 1998, pp. 21–28.

[24] V. Alwayn, *Optical Network Design and Implementation*. Indianapolis, IN, USA: Cisco Press, 2004.

[25] R.-J. Essiambre and R. W. Tkach, "Capacity trends and limits of optical communication networks," *Proc. IEEE*, vol. 100, no. 5, pp. 1035–1055, May 2012.

[26] E. Temprana, E. Myslivets, B. P.-P. Kuo, L. Liu, V. Ataie, N. Alic, and S. Radic, "Overcoming kerr-induced capacity limit in optical fiber transmission," *Science*, vol. 348, no. 6242, pp. 1445–1448, Jun. 2015.

[27] R. Ramaswami, K. Sivarajan, and G. Sasaki, *Optical Networks: A Practical Perspective*. San Mateo, CA, USA: Morgan Kaufmann, 2009.

[28] X. Ye, S. J. B. Yoo, and V. Akella, "AWGR-based optical topologies for scalable and efficient global communications in large-scale multiprocessor systems," *Opt. Commun. Netw., IEEE/OSA J.*, vol. 4, no. 9, pp. 651–662, Sep. 2012.

[29] K. K. Chow, C. Shu, C. Lin, and A. Bjarklev, "Polarization-insensitive widely tunable wavelength converter based on four-wave mixing in a dispersion-flattened nonlinear photonic crystal fiber," *IEEE Photon. Technol. Lett.*, vol. 17, no. 3, pp. 624–626, Mar. 2005.

[30] K. Xi, Y.-H. Kao, and H. J. Chao, "A petabit bufferless optical switch for data center networks," in *Optical Interconnects for Future Data Center Networks* (Optical Networks). New York, NY, USA: Springer, 2013, pp. 135–154.

[31] S. Werner, J. Navaridas, and M. Lujan, "Designing low-power, low-latency Networks-on-Chip by optimally combining electrical and optical links," in *Proc. IEEE Int. Symp. High Perform. Comput. Archit. (HPCA)*, Feb. 2017, pp. 265–276.

[32] S. Bahirat and S. Pasricha, "METEOR: Hybrid photonic ring-mesh network-on-chip for multicore architectures," *ACM Trans. Embedded Comput. Syst.*, vol. 13, no. 3s, pp. 1–33, Mar. 2014.

[33] S. Lange, A. S. Raja, K. Shi, M. Karpov, R. Behrendt, D. Cletheroe, I. Haller, F. Karinou, X. Fu, J. Liu, and A. Lukashchuk, "Sub-nanosecond optical switching using chip-based soliton microcombs," in *Proc. Opt. Fiber Commun. Conf.*, Washington, DC, USA: Optical Society of America, 2020.

[34] H. Ballani, P. Costa, R. Behrendt, D. Cletheroe, I. Haller, K. Jozwik, F. Karinou, S. Lange, K. Shi, B. Thomsen, and H. Williams, "Sirius: A flat datacenter network with nanosecond optical switching," in *Proc. Annu. Conf. ACM Special Interest Group Data Commun. Appl., Technol., Archit., Protocols Comput. Commun.*, Jul. 2020, pp. 782–797.
[35] J. Navaridas, J. A. Pascual, A. Erickson, I. A. Stewart, and M. Luján, "INRFlow: An interconnection networks research flow-level simulation framework," *J. Parallel Distrib. Comput.*, vol. 130, pp. 140–152, Aug. 2019.
[36] N. Calabretta and H. Dorren, "All-optical label processing in optical packet switched networks," in *Proc. Opt. Fiber Commun. Conf.*, 2010, pp. 1–3.
[37] C. Bintjas, N. Pleros, K. Yiannopoulos, G. Theophilopoulos, M. Kalyvas, H. Avramopoulos, and G. Guekos, "All-optical packet address and payload separation," *IEEE Photon. Technol. Lett.*, vol. 14, no. 12, pp. 1728–1730, Dec. 2002.
[38] F. Ramos, E. Kehayas, J. M. Martinez, R. Clavero, J. Marti, L. Stampoulidis, D. Tsiokos, H. Avramopoulos, J. Zhang, P. V. Holm-Nielsen, and N. Chi, "IST-LASAGNE: Towards all-optical label swapping employing optical logic gates and optical flip-flops," *J. Lightw. Technol.*, vol. 23, no. 10, p. 2993, 2005.
[39] S. J. B. Yoo, H. Jae Lee, Z. Pan, J. Cao, Z. Yanda, K. Okamoto, and S. Kamei, "Rapidly switching all-optical packet routing system with optical-label swapping incorporating tunable wavelength conversion and a uniform-loss cyclic frequency AWGR," *IEEE Photon. Technol. Lett.*, vol. 14, no. 8, pp. 1211–1213, Aug. 2002.
[40] M. Ono, M. Hata, M. Tsunekawa, K. Nozaki, H. Sumikura, H. Chiba, and M. Notomi, "Ultrafast and energy-efficient all-optical switching with graphene-loaded deep-subwavelength plasmonic waveguides," *Nature Photon.*, vol. 14, no. 1, pp. 37–43, Jan. 2020.

JOSÉ DURO received the B.S. degree from UCLM, in 2014, and the M.S. degree in computer engineering from Universitat Politècnica de València (UPV), Spain, 2015, where he is currently pursuing the Ph.D. degree with the Parallel Architecture Group (GAP).

His research interests include computer architecture, interconnection networks, and photonic technology.

SALVADOR PETIT received the Ph.D. degree in computer engineering from Universitat Politècnica de València (UPV), Spain. Since 2009, he has been an Associate Professor with the Computer Engineering Department, UPV, where he has taught several courses on computer organization. He has published more than 100 refereed conference and journal articles. His research interests include multithreaded and multicore processors, memory hierarchy design, resource management, and HPC photonic networks. In 2013, he received the Intel Early Career Faculty Honor Program Award.

MARÍA E. GÓMEZ received the B.S., M.S., and Ph.D. degrees in computer engineering from Universitat Politècnica de València (UPV), Spain, in 1996 and 2000, respectively. She joined the Department of Computer Engineering (DISCA), UPV, in 1996, where she is currently a Full Professor. She has published more than 80 conference and journal articles. She has served on program committees for several major conferences. Her research interests include processor architecture and interconnection networks.

JULIO SAHUQUILLO (Member, IEEE) received the B.S., M.S., and Ph.D. degrees from Universitat Politècnica de València (UPV), Spain, all in computer engineering. He is currently a Full Professor with DISCA Department, UPV. He has taught several courses on computer architecture. He has authored more than 150 refereed conference and journal articles. His research interests include processor microarchitecture, memory hierarchy design, interconnects, GPU architecture, and system resource management.