UNIVERSITAT
POLITÈCNICA
DE VALÈNCIA

Photonic Interconnection Networks for Exascale Computers

January 2021

Author: José Duro Gómez

Advisors: María Engracia Gómez Requena

Julio Sahuquillo Borrás

Doctoral Committee

- Prof. Antonio Robles Martínez
  Universidad Politécnica de Valencia

- Prof. Carlos Reaño González
  Queen's University of Belfast

- Prof. Pedro Javier García García
  Universidad de Castilla-La Mancha

Agradecimientos

Esta tesis doctoral marca el final del capítulo académico de mi vida.

Empezó como algo inesperado, cuando en el máster tanto a José Puche y a mí nos ofrecieron esta posibilidad. De las dos opciones, al final ambos nos decidimos por "los mejores congresos del mundo" que Julio nos ofreció, y aún con los contratiempos, las largas reuniones y tantas horas, aquí estoy escribiendo estas palabras.

Sin embargo, no he llegado hasta este punto yo solo, y aquí es donde puedo aprovechar para dar visibilidad a esas personas que me han acompañado, ayudado y (probablemente lo más importante) soportado.

En primer lugar, me gustaría agradecer a mis directores, tanto a María Engracia como a Julio, por confiar en mi tanto en la realización de la tesis como en todo lo relacionado con el proyecto europeo. En este punto también quiero mencionar a Salva, que ha sido como mi "tercer" director. Quería agradecerles por su guía en el doctorado, por esas discusiones en las reuniones (que echaré de menos) en las que siempre aprendía algo y gracias por su paciencia en todo el proceso.

También quería agradecer a mis compañeros del Grupo de Arquitecturas Paralelas, tanto los del laboratorio "bueno" como los del "malo", que me han soportado y ayudado a partes iguales. Al mencionado José Puche, con quien llevo compartiendo penas desde el grado; a Javier Prades y José Manuel Rocher, entre los ratos del café, los de la "bilis" y los de intentar ayudarnos entre nosotros todo es mucho más llevadero; a Leito Chancay, el gran apoyo en el piso de toda la vida, que siempre tenía una sonrisa al llegar. Y al resto de compañeros y amigos que me

han apoyado y me han sufrido, a Fran, José María, Santi, Jaime, Tomás, Carlos, Josué, Danilo, William, Andrés, Adrián, Vicent, Xisco, Escamilla, Ricardo, ..., gracias.

Por último, quiero agradecer a mi familia por el apoyo que me han dado siempre desde el inicio, y especialmente a mi pareja, que me siguió en esta aventura a Valencia, me ha apoyado en todo y que se le ha hecho el doctorado casi más largo que a mí. A mis abuelos, que me ayudan desde arriba y mi abuela, que aunque se le olvide que he estudiado, siempre me pregunta cómo me va todo.

Muchas gracias a todos.

Abstract

In the last recent years, multiple research projects around the world have focused on the design of supercomputers able to reach the exascale computing barrier, with the aim of supporting the execution of important applications for our society, such as health, artificial intelligence, meteorology, etc. According to the growing trend in the computational power in each supercomputer generation, this objective is expected to be reached in the coming years. However, achieving this goal requires addressing distinct major challenges in the design and development of the system. One of the main ones is to achieve fast and efficient communications between the huge number of computational nodes and the memory systems.

Photonics technology provides several advantages over current electrical networks, such as higher bandwidth in the links, greater network parallelism thanks to DWDM, or better cable management due to its much smaller size. In this thesis, a feasibility study and development of interconnection networks have been developed using photonics technology for future exascale systems within the European project ExaNeSt.

First, a characterization study of exascale applications from the network requirements has been carried out. The results of the analysis helps understand the network requirements of exascale applications, and thereby guide us in the design of the system network. This analysis considers three main parameters: the distribution of the messages based on their size and type, the required bandwidth consumption throughout the execution, and the spatial communication patterns between the nodes. The study reveals the need for a fast and efficient interconnection network, since most communications consist of bursts of transmissions, each with an average message size of 50 KB.

Next, this dissertation concentrates on identifying the main elements that differentiate photonic networks from electrical ones. We identify a sequence of steps in the design and implementation of a simulator either i) dealing with photonic technology from scratch or ii) to extend an existing electrical network simulator in order to model photonics.

After that, two main performance comparison studies between electrical networks and different configurations of photonic networks are presented using classical topologies. In the former study, carried out with both synthetic traffic and traces of ExaNeSt in a torus, fat tree and dragonfly, we found that photonic technology represents a noticeable improvement over electrical technology. Furthermore, the study shows that the parameter that most affects the performance is the bandwidth of the photonic channel. The latter study analyzes performance of real applications in large-scale simulations in a jellyfish topology. The results of this study corroborates the conclusions obtained in the previous, also revealing that photonic technology allows reducing the complexity of some topologies, and therefore, the cost of the network.

In the previous studies we realize that the network was underutilized mainly because the studied topologies for electrical networks do not take advantage of the features provided by photonic technology. For this reason, we propose Segment Switching, a switching strategy aimed at reducing the length of the routes by implementing buffers at intermediate nodes along the path. Experimental results show that each of the studied topologies presents different buffering requirements. For the torus, the higher the number of buffers in the network, the higher the performance. For the fat tree, the key parameter is the buffer size, achieving similar performance a configuration with buffers on all switches that locating buffers only at the top level.

In summary, this thesis studies the use of photonic technology for networks of exascale systems, and proposes to take advantage of the characteristics of this technology in current electrical network topologies.

Resumen

En los últimos años, distintos proyectos alrededor del mundo se han centrado en el diseño de supercomputadores capaces de alcanzar la meta de la computación a exascala, con el objetivo de soportar la ejecución de aplicaciones de gran importancia para la sociedad en diversos campos como el de la salud, la inteligencia artificial, la meteorología, etc. Teniendo en cuenta la creciente tendencia de la potencia computacional en cada generación de supercomputadores, este objetivo se prevee accesible en los próximos años. Sin embargo, alcanzar esta meta requiere abordar diversos retos en el diseño y desarrollo del sistema. Uno de los principales es conseguir unas comunicaciones rápidas y eficientes entre el inmenso número de nodos de computo y los sistemas de memoria.

La tecnología fotónica proporciona ciertas ventajas frente a las actuales redes eléctricas, como un mayor ancho de banda en los enlaces, un mayor paralelismo a nivel de comunicaciones gracias al DWDM o una mejor gestión del cableado gracias a su reducido tamaño. En esta tesis se ha desarrollado un estudio de viabilidad y desarrollo de redes de interconexión haciendo uso de la tecnología fotónica para los futuros sistemas a exaescala dentro del proyecto europeo ExaNeSt.

En primer lugar, se ha realizado un análisis y caracterización de aplicaciones exaescala. Este análisis se ha utilizado para conocer el comportamiento y requisitos de red que presentan las aplicaciones, y con ello guiarnos en el diseño de la red del sistema. Este análisis considera tres parámetros: la distribución de mensajes en base a su tamaño y su tipo, el consumo de ancho de banda requerido a lo largo de la ejecución y la matriz de comunicación espacial entre los nodos. El estudio revela la necesidad de una red eficiente y rápida, debido a que la mayoría de las comunicaciones se realizan en burst y con mensajes de un tamaño medio inferior a 50KB.

A continuación, esta tesis se centra en identificar los principales elementos que diferencian las redes fotónicas de las eléctricas. Identificamos una secuencia de pasos en el diseño de un simulador, ya sea i) haciéndolo desde cero con tecnología fotónica o ii) adaptando un simulador de redes eléctricas existente para modelar la fotónica.

Después se han realizado dos estudios de rendimiento y comparativas entre las actuales redes eléctricas y distintas configuraciones de redes fotónicas utilizando topologías clásicas. En el primer estudio, realizado tanto con tráfico sintético como con trazas de ExaNeSt en un toro, fat tree y dragonfly, se observa como la tecnología fotónica supone una clara mejora respecto a la eléctrica. Además, el estudio muestra que el parámetro que más afecta al rendimiento es el ancho de banda del canal fotónico. El segundo estudio muestra el comportamiento y rendimiento de aplicaciones reales en simulaciones a gran escala en una topología jellyfish. En este estudio se confirman las conclusiones obtenidas en el anterior, revelando además que la tecnología fotónica permite reducir la complejidad de algunas topologías, y por ende, el coste de la red.

En los estudios realizados se ha observado una baja utilización de la red debido a que las topologías utilizadas para redes eléctricas no aprovechan las características que proporciona la tecnología fotónica. Por ello, se ha propuesto Segment Switching, una estrategia de conmutación orientada a reducir la longitud de las rutas mediante el uso de buffers intermedios. Los resultados experimentales muestran que cada topología tiene sus propios requerimientos. En el caso del toro, el mayor rendimiento se obtiene con un mayor número de buffers en la red. En el fat tree el parámetro más importante es el tamaño del buffer, obteniendo unas prestaciones similares una configuración con buffers en todos los switches que la que los ubica solo en el nivel superior

En resumen, esta tesis estudia el uso de la tecnología fotónica para las redes de sistemas a exascala y propone aprovechar las características de esta tecnología en las actuales topologías de redes eléctricas.

Resum

Els darrers anys, múltiples projectes de recerca a tot el món s'han centrat en el disseny de superordinadors capaços d'assolir la barrera de computació exascala, amb l'objectiu de donar suport a l'execució d'aplicacions importants per a la nostra societat, com ara salut, intel·ligència artificial, meteorologia, etc. Segons la tendència creixent en la potència de càlcul en cada generació de superordinadors, es preveu assolir aquest objectiu en els propers anys. No obstant això, assolir aquest objectiu requereix abordar diferents reptes importants en el disseny i desenvolupament del sistema. Un dels principals és aconseguir comunicacions ràpides i eficients entre l'enorme nombre de nodes computacionals i els sistemes de memòria.

La tecnologia fotònica proporciona diversos avantatges respecte a les xarxes elèctriques actuals, com ara un major ample de banda als enllaços, un major paral·lelisme de la xarxa gràcies a DWDM o una millor gestió del cable a causa de la seva mida molt més xicoteta. En aquesta tesi, s'ha desenvolupat un estudi de viabilitat i desenvolupament de xarxes d'interconnexió mitjançant tecnologia fotònica per a futurs sistemes exascala dins del projecte europeu ExaNeSt.

En primer lloc, s'ha dut a terme un estudi de caracterització d'aplicacions exascala dels requisits de xarxa. Els resultats de l'anàlisi ajuden a entendre els requisits de xarxa de les aplicacions exascale i, per tant, ens guien en el disseny de la xarxa del sistema. Aquesta anàlisi considera tres paràmetres principals: la distribució dels missatges en funció de la seva mida i tipus, el consum d'ample de banda requerit durant tota l'execució i els patrons de comunicació espacial entre els nodes. L'estudi revela la necessitat d'una xarxa d'interconnexió ràpida i eficient, ja que la majoria de comunicacions consisteixen en ràfegues de transmissions, cadascuna amb una mida mitjana de missatge de 50 KB.

A continuació, aquesta tesi se centra a identificar els principals elements que diferencien les xarxes fotòniques de les elèctriques. Identifiquem una seqüència de passos en el disseny i implementació d'un simulador: i) tractar la tecnologia fotònica des de zero o ii) per ampliar un simulador de xarxa elèctrica existent per modelar la fotònica.

Després, es presenten dos estudis principals de comparació de rendiment entre xarxes elèctriques i diferents configuracions de xarxes fotòniques mitjançant topologies clàssiques. En el primer estudi, realitzat tant amb trànsit sintètic com amb traces d'ExaNeSt en un toro, fat tree i dragonfly, vam trobar que la tecnologia fotònica representa una millora notable respecte a la tecnologia elèctrica. A més, l'estudi mostra que el paràmetre que més afecta el rendiment és l'amplada de banda del canal fotònic. Aquest darrer estudi analitza el rendiment d'aplicacions reals en simulacions a gran escala en una topologia jellyfish. Els resultats d'aquest estudi corroboren les conclusions obtingudes en l'anterior, revelant també que la tecnologia fotònica permet reduir la complexitat d'algunes topologies i, per tant, el cost de la xarxa.

En els estudis anteriors ens adonem que la xarxa estava infrautilitzada principalment perquè les topologies estudiades per a xarxes elèctriques no aprofiten les característiques proporcionades per la tecnologia fotònica. Per aquest motiu, proposem Segment Switching, una estratègia de commutació destinada a reduir la longitud de les rutes mitjançant la implementació de memòries intermèdies en nodes intermedis al llarg de la ruta. Els resultats experimentals mostren que cadascuna de les topologies estudiades presenta diferents requisits de memòria intermèdia. Per al toro, com més gran siga el nombre de memòries intermèdies a la xarxa, major serà el rendiment. Per al fat tree, el paràmetre clau és la mida de la memòria intermèdia, aconseguint un rendiment similar tant amb una configuració amb memòria intermèdia en tots els commutadors com amb una que només localitzen memòries intermèdies al nivell superior.

En resum, aquesta tesi estudia l'ús de la tecnologia fotònica per a xarxes de sistemes exascala i proposa aprofitar les característiques d'aquesta tecnologia en les topologies actuals de xarxes elèctriques.

Contents

List of Figures

List of Tables

Chapter 1

Introduction

This chapter presents the main reasons that led us to develop this dissertation. We also summarize major exascale computing challenges, and present the project where this dissertation has been carried out, the European Exascale System Interconnect and Storage (ExaNeSt) project. After that, we expose the main features and advantages provided by photonics technologies, fundamental photonics concepts and its components to help understand the decision behind the selection of this technology. Then, we introduce the main steps that have been followed to design the photonics interconnects devised in this dissertation. After that, we describe the objectives of this work, and summarize the main contributions made throughout the development of this work. Finally, the thesis outline is presented.

1.1 Introduction and Motivation

This thesis has been conceived from the work carried out by Polytechnic University of Valencia in the ExaNeSt European project. This project focuses on the development of an interconnection network and storage system that help achieve the challenge of exascale computing. In addition to focus on classical electronic network, this project has also addressed the study of an interconnection network for future exascale systems using photonics technologies since these technologies feature, among others, much higher network bandwidth with much lower energy consumption.

1.1.1 Exascale Computing

The most powerful supercomputers in the world are ranked in the Top500 list [73] by their computational power in terms of floating-point operations executed per second (FLOPS). This list is updated every six months by adding new supercomputers or updating those already included. In this classification, it can be seen the computational trend over years from its creation in June 30, 1993 led by a 1024-core system from Los Alamos in United States, that reached 131 GFlops.

In the last decade, computational power has grown from 2.331 TeraFlops, reached by Jaguar supercomputer with 224.162 cores in June 2010, to 54.902 TeraFlops, reached by Tianhe-2A with 3,12 million cores in June 2015. Following the computational trend observed over years, it was expected that supercomputers would break the ExaFlop (10^{18}) barrier by 2020. However, the most powerful supercomputer listed in November 2020, the Supercomputer Fugaku, has reached 537 PetaFlops peak computational power with 7,6 million cores.

In this context, great efforts have been made to accomplish this objective in a single supercomputer. Other attempts have focused on a different direction by using distributed machines. In this context, ExaFlop computational power has been reached in June 2020 by the initiative folding@home[1] [83]. In this initiative, volunteer people give up computing power from their devices for specific purposes in a *distributed system* that allows scientists to carry out their experimental executions. This *distributed system* aggregates as much as 280,000 GPUs and 4.8 million CPU cores around the world, reaching 1.01 ExaFlops peak computational power in a project trying to face up the SARS-CoV-2 virus.

[1] http://foldingathome.org

On the other hand, energy efficiency represents an important perspective when building new supercomputers. The Green500 [37] list ranks the Top500 supercomputers from this axis. This list sorts supercomputers according to the *power per watt* metric. In the most recent list, November 2020, the top 1 position in Green500, NVIDIA DGX SuperPOD, corresponds to the 170 in Top500 list, with a peak performance of 2,3 TFlops and power efficiency of 26,2 GFlops/watt. Fugaku supercomputer, top 1 mentioned above, is number 10 in Green500 which delivers 15,4 GFlops/watt.

Reaching the exascale computing objective in the context of High-Performance Computing (HPC) is challenging and requires from multiple combined solutions addressing, among others, computational power at chip-level (nodes of the system), data movement across the system, distributed storage, energy management, etc. Among these challenges, the design of high performance and efficient interconnection networks is one of the key challenges to efficiently face the data movement in order to keep the over system performance high. This is one of the most critical challenges in the design of exascale supercomputers due to the incredibly high number of compute nodes, the distributed memory system, and, ultimately, the increasing communication requirements.

1.1.2 European Exascale System Interconnect and Storage (ExaNeSt)

Among the initiatives to reach the exascale computing barrier led by the European Union, Horizon 2020 (H2020)[39] has been the biggest European Union research and innovation program with the main objective of ensuring the global competitiveness of Europe. This dissertation has been developed within the ExaNeSt project. ExaNeSt [43, 30] is a European project under the H2020 program, aimed at developing an exascale system that can be scaled up to tens of millions of interconnected low-power consumption ARM cores to solve large-scale scientific and big data problems. In Figure 1.1 we can see the prototype of a rack developed in the project. As part of H2020, ExaNeSt has been one of the European projects that support a ground-breaking computing architecture for exascale-class systems in collaboration with two other contemporary and one subsequent projects. ExaNoDe (European Exascale Processor & Memory Node Design) [31], focused on computer node and memory concerns. ECOSCALE (Energy-Efficient Heterogeneous Computing at ExaScale) [26], focused on heterogeneous architectures and especially the efficient use of FPGA-base accelerators. EuroEXA (Co-designed Innovation and System for Resilient Exascale Computing in Europe: From Applications to Silicon) [28], focused to provide a petascale-level prototype by innovating both on technology and applications.

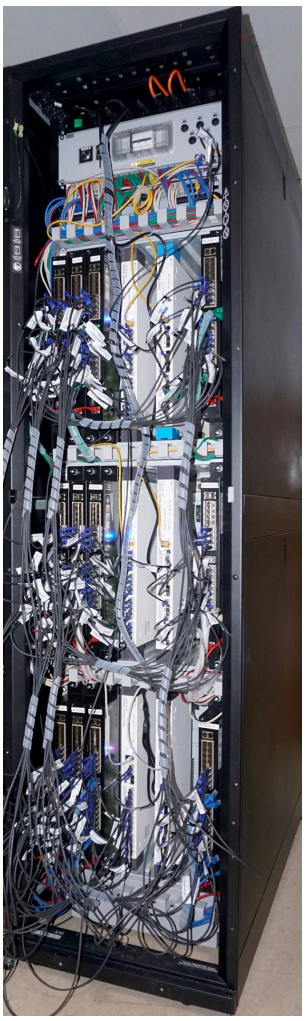


Figure 1.1: Rack prototype. 12 Blade HPC Testbed of ExaNeSt in Liquid-Cooling Rack. Source: ExaNeSt project.

The main objectives of the ExaNeSt project were the design and implementation of:

- *A physical rack prototype and its liquid-cooling subsystem providing ultra-dense compute packaging.*

  ExaNeSt adopts the packaging and cooling technology of Icetope partner, leader in Totally Liquid Cooled (TLC) technology for computing infrastructures. In the prototype, each chassis can accommodate insertion of up to 9 blades. Each blade is a sealed, self-contained entity, immersed in a sophisticated and non-conductive coolant liquid.

- *A storage architecture with distributed (in-node) non-volatile memory (NVM) devices.*

  Aiming to avoid excessive latency and energy consumption, in ExaNeSt these storage devices have been placed close to the compute nodes. This decision makes them accessible through fast custom-made interconnects. In contrast, the storage devices of traditional supercomputers are located in a central location.

- *An unified low-latency interconnect network, designed to efficiently uphold desired Quality-of-Service guarantees for a mix of proposed storage with the inter-processor flows.*

  In ExaNeSt there are addressed the different levels of the interconnect, examining suitable low-power electrical and optical technologies and appropriate topologies. Interconnection network is severely constrained by system packaging and topology selection, solved by multi-tier interconnects to address the disparate needs and requirements at separate building blocks inside the rack.

- *An efficient rack-level memory sharing based on Unimem, developed in EuroServer project [25, 29].*

  Original technology offers the ability to access areas of memory located in remote nodes, where each physical memory page can be cached at only a single node. In ExaNeSt the plan was to enable a virtual global address space rather than a physical one and test with the proposed alternative storage and interconnect options on actual hardware, using real-world HPC and exascale applications.

To provide support for a system of this large size, ExaNeSt has been confronted with the huge challenge of designing an interconnection network able to meet very strict performance, resilience, and cost constraints.

The ExaNeSt interconnection network consists of a multi-tier interconnect which can be divided into two distinct parts. On the one hand, the lower tiers, which are physically fixed employing boards and back-planes into the supercomputer racks. On the other hand, the higher tiers or the top of racks, which are fully reconfigurable using custom-made FPGA-based routers [16]. This flexibility allows building any network topology, i.e. direct, indirect or hybrid, or even the use of standard off-the-shelf commodity switches. In order to meet the requirements of such demanding interconnect, in the project it has been explored the use of photonic technology as an alternative within the higher tier due to the advantages that this technology provides compared to current electrical technology.

Regarding the development of the project, we have been responsible together with the Polytechnic University of Valencia's Nanophotonics Technology Center (NTC) [51] of carrying out proof of concepts, viability study and development of this technology, from which this doctoral thesis has been developed.

1.2 Photonics Features

Photonics interconnects are envisaged as a promising technology to overcome the so-called communication bottleneck of its electronic counterparts. The technology features, among others, huge network bandwidth, low energy consumption, better trade-off in speed/distance or a greatly cable management make this technology an option to consider in the development of new HPC or exascale systems.

Silicon photonics-based interconnects are being deployed in data communication systems due to their potential to achieve large scale and low cost integration together with a low power operation. This potential relies on some advantages like the higher bandwidth capability, where achieving more than 10 Gbps with conventional copper wires remains a challenge, while a single optic fiber can offer bandwidths in the Terabit range. Other potential improvement is network parallelization, provided by the Dense Wavelength Division Multiplexing (DWDM) [6] technique that allows splitting up the optical signal of the link into multiple independent wavelengths. Optic fibers provide a better distance/speed trade-off, able to transmit data along several kilometers without any bandwidth penalties. Finally, due to their inherent lightness and thinness, using optic fiber cables instead of copper ones highly reduces cable density, which greatly eases cable management, as can we see in Figure 1.2.

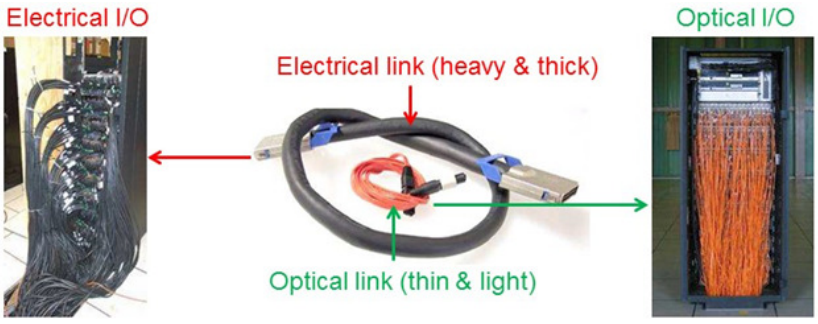


Figure 1.2: Comparison between electrical and optical links. Source: ExaNeSt project.

Current state-of-the-art photonics technologies require from pluggable transceivers to transform electrical signals to optical signals and vice versa. The bandwidth limit of current Quad Small

Form-factor Pluggable (QSFP) transceivers based on Vertical-Cavity Surface-Emitting Laser (VCSEL) technology is by 40 Gbps. Nevertheless, silicon photonic interconnects are expected to reach the 100 Gbps mark and beyond in the near future. For instance, Intel Corporation and other big companies such as IBM or Cisco Systems have moved their silicon photonics efforts beyond research and development, and have produced engineering samples that run at speeds of up to 100 Gbps. Moreover, Intel and Corning are currently developing the MXC connector, which supports up to 64 fibers communicating at 25 Gbps, reaching data transmission capacity by 1.6 Tbps over a 300 meters distance.

It is well known that integrated elements in silicon photonics-based have become significantly faster in the last decades [72, 20], enabling the implementation of silicon photonic routers, which are a key development for inter-rack and intra-rack full-optical networks based only on optical components (i.e. all-optical networks) for Exascale systems. In this regard, current efforts have concentrated on the realization of reliable hybrid silicon lasers, electro-optic modulators, ring resonators and receivers; the most critical building components of photonic circuits.

Laser sources inject light into the chip's waveguides. Laser sources are probably the most difficult devices to be integrated on silicon. Duan et al. [23, 24] have developed hybrid silicon/III-V lasers with less power consumption than previous works [21, 22], although not yet achieving ultra-low power consumption, which will significantly reduce packaging costs. The electro-optical modulators establish the switching capacity, that is, the operation bandwidth of any photonic integrated circuit. High bandwidth modulation can be realized in silicon with free-carrier induced index change [67], using biased pn structures (carrier depletion) achieving up to 30-50 Gbps data rates [47, 71]. Optical ring resonators are the key component to leverage mentioned DWDM technology. A ring resonator captures specific optical wavelengths; thus, it can redirect these wavelengths to other waveguides and receivers, so enabling the implementation of complex optical on-chip networks and photonic routers. Finally, optical coherent receivers (also known as photo-detectors) convert the amplitude, phase, and polarization of an optical signal into the electrical domain have already been integrated providing very high data conversion rates [19, 18].

In this context, some researching are focused on developing new switch architectures incorporating photonic technology together with different network organizations, both in hybrid approaches like Helios [32] or Hydra [15] or in all-optical approaches like DOS [80] or OSA [13].

Novel researches are focused on the use of new materials to improve both the data transmission capacity and the switching and conversion time of the all optical elements, such as photonics

based on graphene, which, despite having different limitations, current investigations reach times of the order of femtoseconds (10^{-15} seconds) [8, 54].

Although this technology is still not mature enough to be integrated in the processor die, some attempts have been done in silicon chips, what is referred to as on-chip silicon photonics. In the ExaNeSt project we measured potential bandwidth in off-chip silicon photonics performing proves of concepts. The gathered information was used to parametrize and setup our simulation framework. Unlike existing work, we propose new mechanisms based on modify the switching technique of the network to take advantage of the mentioned photonic features.

1.3 Designing a photonic network for exascale computing: Major steps

As mentioned above, dealing with efficient data movements in exascale networks is a key challenge that must to be faced to deal with system performance and energy. To deal with this issue, we have followed an iterative process composed of several steps, which are further explained below.

The first step to be done, before starting the design of a new network, is to analyze the network requirements of representative workloads. This step is required in order to properly dimension the network performance according to the traffic characteristics. In this context, we have performed an analysis and characterization of the exascale workloads that ExaNeSt partners like Istituto Nazionale di Fisica Nucleare (INFN) provided to us as representative of this type of workloads. The workload characteristics can be used to properly dimension the required link bandwidth, so that the proper technology for photonics links can be analyzed. Different technologies have been studied and compared in this dissertation.

Once the target technology is used, then and in order to explore the performance of different photonics networks, a simulation framework needs to be chosen. In this regard, we developed a simulation platform to faithfully model supercomputer networks using photonic technology. This platform allows modeling and exploring various photonic configurations. This has been done with the help of people from Nanophotonics Technology Center (NTC), which is a world leading company in silicon photonics located at our university. These people provided us detailed experimental features and values gathered at proves of concepts.

Once the simulation tools were ready and the applications analyzed, in a subsequent step we performed some studies and analysis to determine the best network configuration to leverage the photonics interconnects or the behavior of the technology in networks with a high amount of computation nodes. This is possible because photonics technology with DWDM provides the capability of, with a given photonic system (i.e. aggregate bandwidth), modifying the organization of the system to prioritize bandwidth or network parallelism. Finally, with the obtained results from the performed studies, the aim is to improve the network to face the challenges posed by exascale computing by proposing new network strategies that take advantage of the photonic technology.

1.4 Thesis Contributions

This thesis has been developed with the objective of face up to the challenge that presents the development of the interconnection networks for future exascale systems. We are focused in the study and development of photonic interconnects, providing the contributions described below.

- **Workload Characterization for exascale networks.** As mentioned above, we studied network requirements of exascale applications. The results of this study are aimed to guide us in the design of exascale networks. For this purpose, we have analyzed bandwidth and latency requirements of real HPC exascale applications provided by partners of the ExaNeSt project.

- **Modeling photonic networks.** Photonic technology has become a promising and viable alternative for both on-chip and off-chip networks, nevertheless, this technology is not mature enough yet and the development of new simulation frameworks and tools is required. In this dissertation we provide guidelines for extending some existing electrical network simulation frameworks to implement and develop networks based on photonic technologies.

- **Comparative studies between state-of-the-art photonics technologies and classical electrical networks.** Once the environment is defined, we have made a study to compare latencies and performance between networks with both technologies, electrical and photonics, using synthetic traffic and real applications provided by ExaNeSt partners. Additionally, we have performed an analysis of performance in large scale systems.

- **Development of new mechanisms for improving the performance of networks that use photonic technology.** Finally, we have proposed new mechanisms to take advantage of the features provided by photonics interconnects and that allow, in this way, reaching exascale computing.

1.5 Outline

The outline of this thesis is the following:

- Chapter 2 discusses previous work on simulation frameworks and photonic-based architectures.
- Chapter 3 describes the workload characterization performed for some ExaNeSt applications.
- Chapter 4 proposes the fundamentals to take into account to adapt current electrical network simulators to photonic technology.
- Chapter 5 analyzes the major design parameters considered for photonic networks.
- Chapter 6 discusses the study performed for large-scale simulations in an exascale system with photonic technology.
- Chapter 7 proposes new mechanisms to take advantage of the photonic technology. In particular, segment switching strategy is proposed to improve the utilization and performance of photonic interconnection networks.
- Chapter 8 draws the main conclusion obtained in the thesis, presents future directions and enumerate the publications that have been made in the development of the thesis.

Chapter 2

Related Work

This chapter discusses previous work related to this thesis. The state-of-the-art presented is divided into two main sections. First, simulation frameworks mainly oriented to interconnection networks modeling photonic technologies are presented. After that, we discuss related work on optical interconnection technologies focusing on switch and network architectures.

2.1 Simulation Frameworks

The interest of the academia and industry communities in the development of Exascale systems, and in improving on-chip architectures, has fostered the research on photonics technology. In order to study these systems, novel simulation and estimation tools are required.

Most of the current network simulators [5, 40, 41] focus on packet-switching electrical networks. These tools can be adapted to model packet-switching hybrid electro-optical networks. However, in order to adapt them to model the circuit switching capabilities of all-optical networks, a significant amount of programming effort is required. Due to this fact, some tools have been proposed designed from the ground up to support all-optical networks. In this regard, a well known simulator is PhoenixSim [12, 63]. This framework models multiprocessor systems that use electrical networks, optical networks, and hybrid networks. PhoenixSim is based on the OMNeT++ simulation environment [74] and allows the analysis of interconnection networks from both the physical level (e.g. optical insertion loss, crosstalk, energy dissipation) and the system level (e.g. latency, performance, execution time).

The Design Space Exploration of Networks Tool (DSENT) [69] improves the PhoenixSim model of electro-optical interface circuitry such as modulators, receivers, and thermal tuning, capturing trade-offs among photonic devices and modulator/receiver specifications that can be exploited to reach optimal configurations in terms of area and power. DSENT is designed to enable fast area and power evaluation of multiple optical network configurations and, when coupled with an architectural simulator, to obtain power and area estimations for the simulated network. However, DSENT does not model photonic switches so it cannot be used to simulate circuit-switched networks. In addition, DSENT does not support traffic patterns and workload traces, so it cannot provide the details of a system-level simulation.

LioeSim [49] is an electrical and optical network simulator that uses Orion [42] for modeling electrical routers and links. Unlike DSENT, it models photonic switches and allows analyzing both physical level (optical insertion loss, crosstalk, optical power budget, energy dissipation) and system level (latency, energy delay product) performance metrics of interconnection networks. Unfortunately, LioeSim is focused on on-chip networks.

Finally, there is also a need for aiding designers in layout tasks such as visually placing photonic devices, connecting waveguides, etc. To this end, Hendry et al. introduce the Visual

Automated Nanophotonic Design And Layout (VANDAL) in [11], which also can be interfaced with industry-standard software tools for chip fabrication processes.

2.2 Switch and Network Architectures with Photonic Technologies

Over the last years, photonic technology has become a research line of interest to the scientific community mainly due to the improvements in bandwidth and parallelism that it offers compared to electrical ones. Below we discuss some works focusing on switch and network architectures.

2.2.1 Switch Architectures

Some effort has been focused on the improvement of existing switch architectures based on photonics technology.

There are approaches based on hybrid optical-electrical switch architectures like Helios [32], with two-layer network. One with electrical Top-of-Rack (TOR) switches based on packet switching and other layer with optical switches used to all-to-all communication between them. In HydRa [15] are focusing mainly on its network controller with low cost components.

Other approaches are based on all-optics networks. In this context, DOS architecture [80] proposes an optical switch architecture based on Arrayed Waveguide Division Multiplexing (AWGR) with a loopback shared buffer system. An enhanced scheme called LIONS [81] presents different loopback buffer in scheme for DOS architecture.

The OSA architecture [13] is based on Wavelength Selective Switch (WSS) and an Optical Switching Matrix based on MEMS (Microelectronical system) where TORs are connected. Another proposal with this scheme is PROTEUS [66] which proposes reconfigurable topology in the MEMS.

2.2.2 Network Architectures

In addition to researching switch architectures with photonic technology, some works are focused on network architecture and topology. P-Torus [10] is an architecture based on a two-layer switching. In the lower layer, TORs are interconnected with neighbors in a 4x4 torus and connected to an Interconnection Passive Aggregation (IPA) switch. In the upper layer IPAs are interconnected in an all-to-all network.

Another structure based on two parallel inter- and intra-cluster networks is OPSquare [78]. TORs have two DWDM bi-directional optical links, one connected to the inter-cluster switch and the other to the intra-cluster switch. This structure provides a single-hop connection for intra-cluster while at most two hops are needed to interconnect racks of different clusters.

Another approach has performed in HiFOST [77] architecture that is based on that n TORs in each cluster are interconnected by a flow-controlled Fast Optical Switch (FOS) through its intra-cluster DWDM optical interface. Moreover, the modified TORs have another bi-directional DWDM optical port for inter-cluster connectivity.

The Tofu Interconnect D (TofuD) [1] developed by Fujitsu and used in the Fugaku supercomputer is a network topology partly implemented with photonic links. TofuD is based in an *irregular torus* based on classical torus/mesh topologies where each switch connects four core-memory groups through six internal interfaces to access a 6D-mesh/torus network, where half of the links are photonic.

Opposite, Data Vortex [46], a fully optical network based on packet-switching technique, proposes a new topology and routing algorithm. Data Vortex implements packet switching even though it avoids the use of buffers, by resolving packet contentions via a distributed deflection routing control scheme. This deflection scheme works by miss-routing packets when 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 too long; however, when the traffic is not low enough they accelerate the saturation of the network [36].

2.3 Summary

This chapter has discussed important research works regarding photonics technologies from three main perspectives.

First, with respect to existing simulation frameworks, different functionalities can be appreciated. We have identified the main features characteristics that photonic networks should feature. Instead of implementing a photonic network simulator from scratch, we can model them in an electrical network simulator. We followed this approach in the ExaNeSt simulation frameworks, Insee [62] and INRFlow [52]. In the developed simulation frameworks we modeled precise photonic features experimentally checked in the NTC lab within the ExaNeSt project, used in this dissertation for conducting the work presented in chapters 4, 5, 6 and 7.

Regarding switch architectures, and according to the underlying technologies, two main design choices can be considered: opto-electric which combines both optical and electrical technologies, and fully photonic, which as stated in its name only the photonic technology is considered. Among the existing architectures, in this dissertation, we use the discussed Lions architecture to develop our approaches, due to the scalability that provide us in the number of switch ports and the possibility of adding buffers in the network.

The final point of view pays attention to the network architecture. Unlike these works, that focus either on the architecture or the topology, we go a step away by considering applying DWDM techniques or the switching technique limitations in the development of our proposal.

Chapter 3

Workload Characterization for Exascale Computing Networks

This chapter presents a workload characterization study in the context of the European Project ExaNeSt, introduced in Section 1.1.2, which focuses, among others, on studying photonics network technologies to implement future exascale systems. For this purpose, we characterized relevant ExaNeSt applications from the computer network perspective by analyzing the distribution of messages, the dynamic bandwidth consumption, and the spatial communication patterns among cores. These studies provide a sound knowledge of the network requirements and to guide network designers in decision making.

To characterize the network requirements of the ExaNeSt workloads, they were profiled to collect their execution time and both point-to-point and MPI (Message Passing Interface) collective messages The profiled information is used to build traces based on openMPI [34] that are analyzed to obtain insights into the distribution of message sizes, the temporal evolution of bandwidth utilization, and spatial communication patterns. The characterization results will be considered to select and develop suitable topologies and network technologies for a feasible and efficient exascale network implementation.

This chapter is organized as follows. First we provide a background about MPI collectives. Next, we present an overview of the studied workloads. After that, we analyze results of

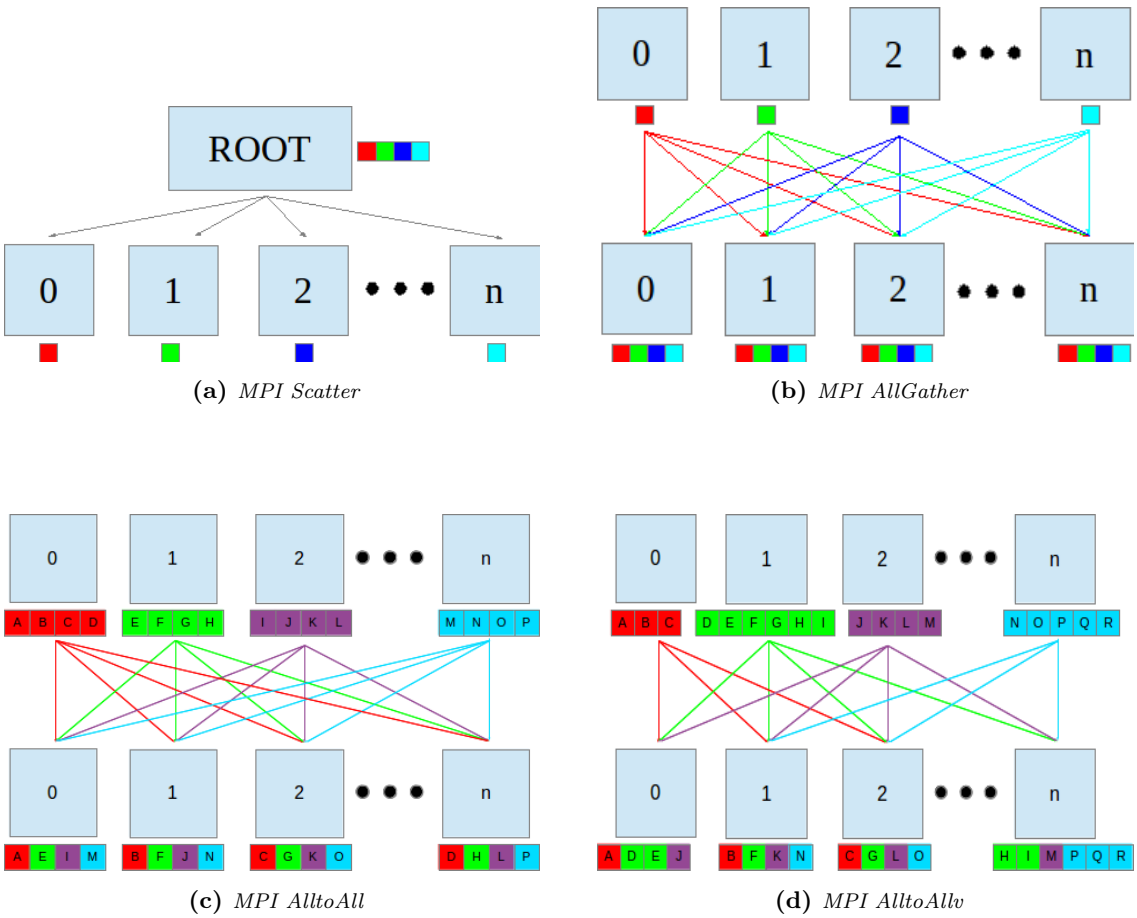
the distribution of message sizes, temporal evolution of bandwidth consumption, and spatial communication patterns of the studied workloads.

3.1 Background on MPI Collectives

ExaNeSt applications consist of thousands of threads which have been coded with Message Passing Interface (MPI) collectives[38]. In order to help understand the characterization study, this section identifies the MPI collectives used by the studied workloads and describes how they work.

When profiling the ExaNeSt workloads, we found the following set of primitives: *MPI_Bcast*, *MPI_Scatter*, *MPI_Scatterv*, *MPI_Gather*, *MPI_Allgather*, *MPI_Reduce*, *MPI_Allreduce*, *MPI_Alltoall*, *MPI_Alltoallv*, and *MPI_Scan*.

- *MPI_Bcast*: This primitive allows a process (i.e. the root) to send an array chunk to all processes in a communicator (i.e. the set of MPI processes involved in the collective).



(a) *MPI Scatter* **(b)** *MPI AllGather*

(c) *MPI AlltoAll* **(d)** *MPI AlltoAllv*

Figure 3.1: MPI collectives.

- *MPI_Scatter*: This collective is similar to *MPI_Bcast*. The main difference is that *MPI_Scatter* sends different equally-sized chunks of an array to different processes (see Figure 3.1a).

- *MPI_Scatterv*: A variation of the *MPI_Scatter* collective where chunks can present different sizes.

- *MPI_Gather*: It implements the opposite behaviour of *MPI_Scatter*. Instead of spreading elements from one process (root) to many processes, *MPI_Gather* takes elements from many processes and sends them to the same single process (root).

- *MPI_Gatherv*: As *MPI_Gather*, but with chunks of different sizes.

- *MPI_Allgather*: Given a set of data elements distributed across all processes, this collective sends all the elements to all the processes. One implementation of this collective is based in two stages. On the first stage, *MPI_Allgather* forwards the elements to the root process, and then, on a second stage, this process sends the collected data to all processes (see Figure 3.1b).

- *MPI_Reduce*: This primitive is similar to *MPI_Gather*. *MPI_Reduce* takes an array of elements on each process and returns a processed array of elements to the root process. Thus, this primitive implies a computation with the data of each element.

- *MPI_Allreduce*: As *MPI_Reduce* but the result of the computation is distributed among all the processes.

- *MPI_Barrier*: This primitive implements a barrier. Thus, a process calling it stalls until all the processes in the communicator have also called it. One implementation of this primitive is based on a *MPI_Bcast* but with very short messages (i.e. tokens).

- *MPI_Alltoall*: It is similar to *MPI_Scatter* but in this case, all processes divide input arrays with the same size into equal chunks and send each chunk to all processes in the communicator (see Figure 3.1c). With this primitive, each process sends and receives the same amount of data.

- *MPI_Alltoallv*: This primitive presents two main differences with respect to *MPI_Alltoall*. On the one hand, the input arrays can have different size and, on the other hand, a

process can receive differently-sized chunks from each sender (or not receive anything from a particular core) (see Figure 3.1d).

- *MPI_Scan*: It calculates an incremental partial reduction among participant processes. This means that each process i calculates the reduction from process 0 to itself. That is, the last process n will obtain the total reduction among all the processes.

3.2 ExaNeSt Workloads

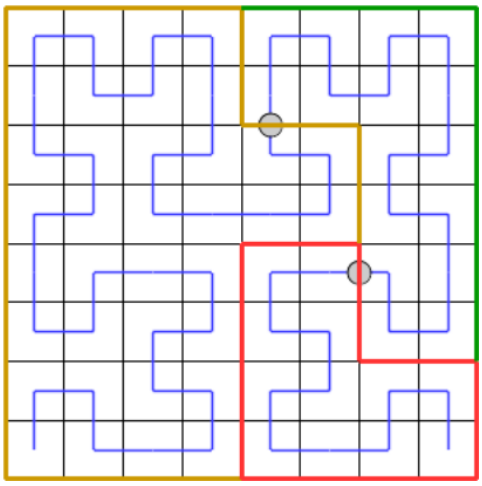
Analyzed ExaNeSt workloads consist of four main applications existing or developed by the ExaNeSt partners, namely *Gadget*, *Lammps*, *DPSNN* and *RegCM*.

3.2.1 GADGET

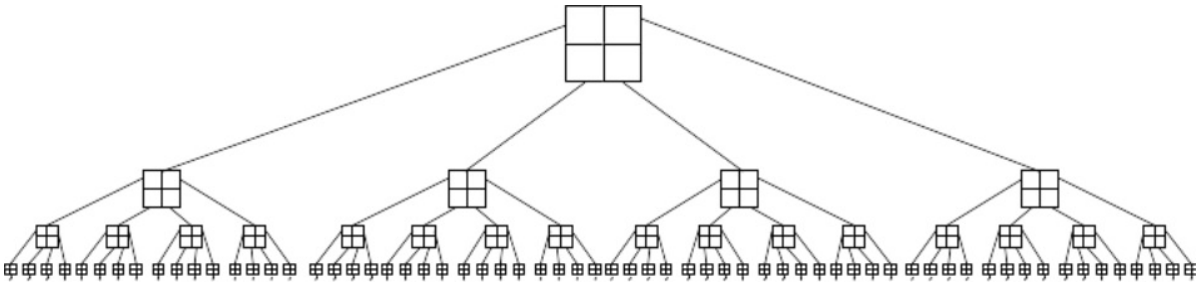
GADGET (GAlaxies with Dark matter and Gas intEracT) is a scientific code aimed at solving the gravitational and hydrodynamical equations that rule the formation and evolution of cosmic structures. The scientific problem can be divided into two main parts: gravity, a long-range component affecting all the computational elements of the chosen domain, and hydrodynamics, which is almost local and only affects ordinary matter (in astrophysics, called "baryonic" matter). For the sake of simplicity, we have developed a model that only considers gravitational effects. GADGET computes gravitational forces using a TreePM technique; this means that a mean-field approximation is used for large scales - called Particle-Mesh, PM - while at smaller scales, a tree-code is used.

GADGET-2 [68] employs a Peano-Hilbert curve as a method to construct PM and the corresponding oct-tree method as depicted in Figure3.2.

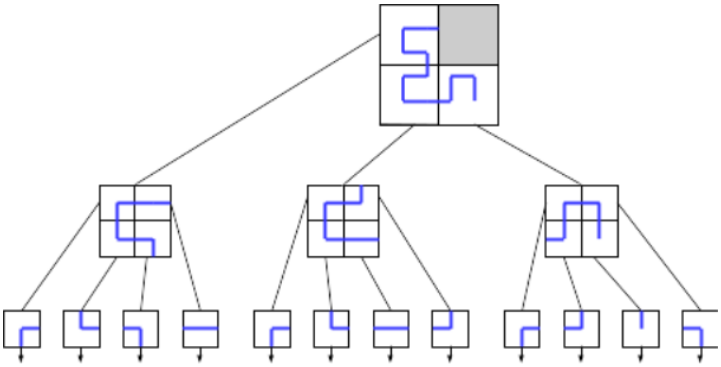
Peano-Hilbert presents three main properties that provide a better approach than a simple mesh implementation. The first property of these space-filling curves is their self-similarity. Hence, we can quickly 'contract' or 'expand' a given Peano-Hilbert curve and again obtain one of lower or higher order. The second property is that the points that are close along the 1D Peano-Hilbert curve are, in general, also close in 3D space; that is, the mapping preserves locality. If we cut a Peano curve into segments of a certain length, we obtain a decomposition which has the property that the spatial domains are connected and 'compact'; that is, they tend to have small surface-to-volume ratios and low aspect ratios. The third property is a close correspondence between the spatial decomposition obtained by a hierarchical oct-tree and that



(a) *Peano-Hilbert curve split in three partitions.*



(b) *Oct-tree of the entire Peano-Hilbert curve.*



(c) *Oct-tree of the red partition of the Peano-Hilbert curve.*

Figure 3.2: Peano-Hilbert decomposition. Note that modeling a 3D space would require a 3D curve decomposed in sub boxes and an oct-tree data structure. A 2D representation is shown here for simplicity.

obtained from segmenting a Peano-Hilbert curve. Therefore, we can cut the curve by a given point and the resulting oct-tree can be assigned to a compute node. Because of this property, we obtain a tree whose geometry is not affected by the parallelization method, and the internal results of the tree become strictly independent of the number of processors used.

Figure 3.2a shows an example of a particle-matrix mapped in a Peano-Hilbert curve. The matrix is split in three segments (colored in red, green and yellow). Figure 3.2b shows the tree

representation corresponding to the entire Peano-Hilbert curve. Each segment is delivered to a given processor that internally orders the segment as an oct-tree as depicted in Figure 3.2c for the segment colored in red.

Once the data of the particles have been distributed among the processors, the parallel computation begins. However, due to the initial distribution of particles, it is possible that a processor does not have all the information needed for the computation of forces between them and can find nodes of the tree that do not have information (these nodes are called pseudo-particles). In the example of Figure 3.2c pseudo-particles have been colored in grey.

The information of these particles is located in another processor or set of processors. Therefore, the processor first checks the target processor storing the pseudo-particle and then delivers a point-to-point message, asking for the required information. After that, the target processor responds with the pseudo-particle information, which allows the local processor to continue with the force calculations.

3.2.2 LAMMPS

LAMMPS [57, 58] is a classical molecular dynamics code that models an ensemble of particles in a liquid, solid, or gaseous state. It can model atomic, polymeric, biological, metallic, granular, and coarse-grained systems using various force fields and boundary conditions.

From a scientific point of view, LAMMPS integrates Newton's equations of motion for collections of atoms, molecules, or macroscopic particles that interact via short- or long-range forces with various initial and boundary conditions. For computational efficiency, LAMMPS uses neighbor lists to keep track of nearby particles. The lists are optimized for systems with repulsive particles at short distances so that the local density of particles never becomes too large.

On parallel machines, LAMMPS uses spatial-decomposition (SD) techniques to partition the simulation domain into small 3D sub-domains, one of which is assigned to each processor. Each processor computes local forces and updates the positions and velocities of all atoms within each box at each timestep. Atoms are reassigned to new processors as they move through the physical domain. To compute forces on its atoms, a processor needs only to know the positions of atoms in nearby boxes, thus making the required communications by MD local. In essence, processors communicate and store "ghost" atom information for atoms that border their sub-domain.

The size and shape of the box assigned to each processor depend on N (number of atoms), P (number of processors), and the aspect ratio of the physical domain, which the algorithm assumes to be a 3D rectangular parallelepiped. As a general rule, the number of processors will be chosen trying to make each processor's box as "cubic" as possible. The following algorithm describes the tasks performed in a single timestep by the spatial-decomposition algorithm on each processor.

1. Move necessary atoms to new boxes.
2. Make lists of all atoms that will need to be exchanged.
3. Construct neighbor lists of interaction pairs in box z.
4. Compute forces of atoms in the box.
5. Update atom positions in box z.
6. Exchange atom positions across box boundaries with neighboring processors.

To analyze the communication patterns employed in the algorithm, we will only focus on steps 1, 2 and 6, where the communication exchange happens. This exchange involves six steps in which the communication happens between adjacent processors following a 3D pattern:

1. Processors exchange information in the east dimension (-X).
2. The processors send information in the west dimension, and, if required, data acquired in the previous step is sent. In this way, all positions of atoms in the west-east dimension have been acquired by each processor (see Figure 3.3). If information of more distant processors is required, steps 1 and 2 must be repeated.
3. The process is repeated in the north-south and in the up-down dimension.

For example, thanks to this pattern information, 26 boxes (processors) are acquired in just 6 data exchanges or with a few extra more if additional distant information is required. The communication pattern of LAMMPS is most efficient (in a parallel sense) for systems whose particles fill a 3D rectangular box with roughly uniform density. These communications will be fast as they generate little contention for the use of resources.

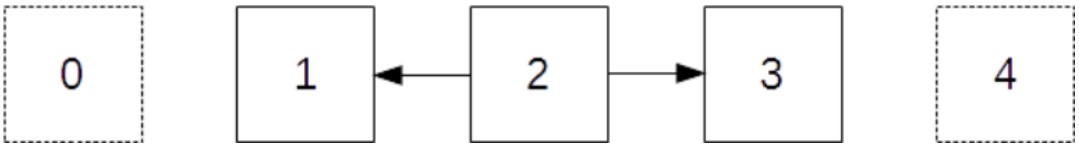


Figure 3.3: Communication pattern of LAMMPS in one of the dimensions.

3.2.3 DPSNN

Distributed Simulation of Polychronous Spiking Neural Networks (DPSNN) [56] is a natively distributed and parallel mini-application benchmark designed to capture major key features needed by large cortical simulations.

DPSNN consists of two main execution phases: i) The creation of the network of axonal polychronous arborizations and synapses which interconnect the system, and ii) the simulation of the synaptic and neural dynamics. Our interest relies on the communication pattern used in DPSNN to exchange information. Below, we discuss both phases emphasizing the communication perspective.

i) **Phase 1. Initial construction of the connectivity infrastructure.** In this phase, each target process is informed of which sources need to communicate with it and, using this information, a database is created representing locally incoming "axons" and "synapses". This process is carried out in two main steps.

   - In the former step, each process informs other processes about the existence of incoming synapses to be established. This is performed by sending a single message (containing the number of connections to be created) between pairs of processors and involves the use of the MPI_AlltoAll directive, which creates a network load proportional to the square of processes. This step contributes to reducing the cost of the construction of the synapses to be created in the next step and the simulation time because the application will have knowledge about the non-existence of connections between pairs of processes.

   - In the latter step, the target processes are informed about the identities of the synapses that need to be created. This exchange is carried out using the MPI_alltoallv directive, where the network load is proportional to the number of processes and the subset of target processes reached by each source.

ii) **Phase 2. Delivery of spiking messages during the simulation phase.** In this phase, DPSNN performs the delivery of spiking messages. This exchange of information (spikes are delivered to target processes) is performed using a synchronous approach before starting each time iteration. This phase is also carried out in two steps.

- In the former step, single word messages representing the spike counters are delivered to subsets of potentially connected target processes. On each pair of source-target processes, the counter informs about the spike or absence of it to be transmitted in the following step.

- In the latter step, the spiking counter is used to establish a communication between processes that need to communicate. Both steps are implemented using the MPI_Alltoallv directive but among sets of processes of decreasing size, as shown in Figure 3.4.

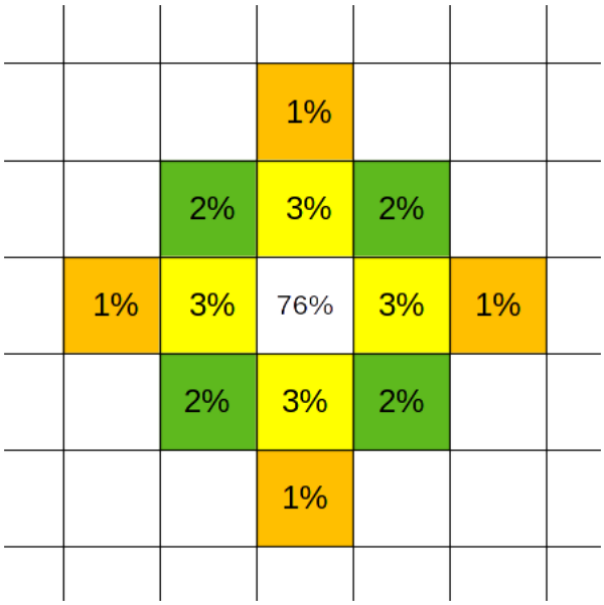


Figure 3.4: Distribution of the amount of information exchanged by MPI_Alltoallv in DPSNN

3.2.4 RegCM

Regional climate models (RCMs) are widely used tools to produce high-resolution climate simulations at regional scales. The ICTP regional climate modeling system, RegCM, is one of the most used RCMs worldwide, with applications ranging from regional process studies to paleoclimate, climate change, chemistry climate and biosphere-atmosphere interactions. Model improvements include developing a new microphysical cloud scheme, coupling with a regional

ocean model, inclusion of full gas-phase chemistry, upgrades of some physics schemes (convection, PBL, cloud microphysics) and development of a non-hydrostatic dynamical core.

RegCM4 [35] is a regional climate model based on the concept of one-way nesting, in which large scale meteorological fields from a Global Climatic Model (GCM) run, providing initial and time-dependent meteorological boundary conditions for high-resolution simulations without any active feedback. The RegCM4 is a hydrostatic, compressible and sigma-p vertical coordinate model, running on an Arakawa B-grid n which wind and thermodynamic variables are horizontally staggered. A time-splitting explicit integration scheme is used, in which the two fastest gravity modes are separated from the model solution and then integrated with smaller time steps.

Regarding the communication pattern, RegCM performs a 2D Cartesian domain decomposition of the space (a 2D domain mapped onto a 1D array) where the simulation is located. On each step, each process needs to communicate with all its neighbors; for the process that is not on the border of the region, this means four communications (two in the North-South direction and two in the East-West direction, represented with arrows in Figure 3.5). If certain climatic conditions are triggered, it is also necessary to transfer the value on the corners of the diagonal cells (dotted arrows in the figure), creating minimal communications (one floating-point number) among the cells located in the diagonal directions.

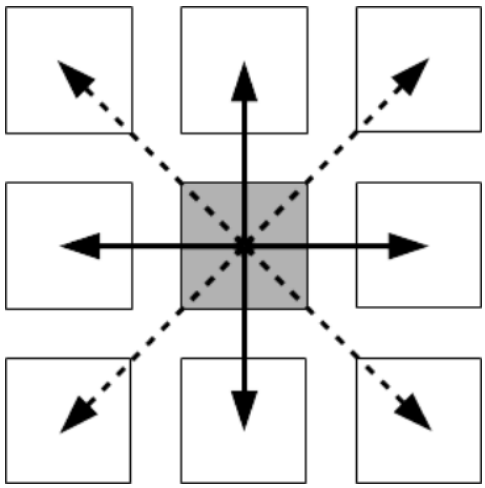


Figure 3.5: Four main communications (solid arrows) and four secondary communications (dashed arrows) in RegCM.

When the output must be written on the disk (once every several time steps), one process (zero process) collects all data from the other processes and writes them on the disk. For this reason, there are some massive communications from all processes to zero process, which is in charge

of saving the current status of the simulation. The effect of this massive transmission is a huge congestion point in the network.

3.3 Trace Analysis Methodology

Different workload sizes have been considered to provide insights on the relationship between the bandwidth requirements and the network core count. More precisely, ExaNeSt partners provided 19 traces generated with real-hardware using different network statistic collection tools such as *scalasca* [64], which provides MPI primitives information in addition to computation times and message timestamps.

These traces correspond to six different executions (varying the core count from 24 to 768) from Lammps, five from RegCM (from 24 to 192 cores), five from DSPNN (from 32 to 128 cores in the 32x32 neural columns configuration and from 64 to 128 cores in the 64x64 neural columns configuration), and three from Gadget (from 24 to 72 cores). In this work we present the analysis of a representative subset of the provided traces.

The first step in the characterization study is to provide a general overview of the traces from the network bandwidth perspective. For this purpose, we gathered two key parameters: the execution time and the overall transferred data. From them, we computed the average bandwidth consumed by each application.

We consider all the transferred data, both header and payload. The messages are split into packets of 72 bytes to be injected on the network, where 64 bytes correspond to payload and the remaining 8 bytes left are for the header. More precisely, a 1KB-message is split into 16 packets as done in some modern machines [17], incurring a 128B overhead for the headers.

Table 3.1 shows the results. As observed, the obtained values widely vary across the studied variables. The execution time presents differences ranging from around 6 up to 23839 billion cycles. Bandwidth requirements exhibit high variations regardless of the execution time. Since the traces do not include the processor frequency, we have assumed a 2 GHz processor clock to calculate the average bandwidth in seconds. The results show that there are applications with relatively few bandwidth requirements (about 8 MBps) and applications with huge bandwidth requirements (e.g., around 18.5GBps). Note that when the number of cores exceeds 192 in Lammps, the execution time increases. The reason is that the original system used to generate

these traces has fewer cores than parallel threads are generated in these configurations, which affects the execution time.

Table 3.1: Average bandwidth requirements for each trace.

Application		**Execution Time (cycles)**	MB **Transf (Total)**	**MB/s (average)**
Lammps	24 Cores	43,754,790,287	24,644	1,126
	48 Cores	21,713,810,259	31,141	2,868
	96 Cores	10,887,071,229	40,934	7,520
	192 Cores	5,983,794,523	55,338	18,496
	384 Cores	152,820,600,368	71,924	941
	768 Cores	322,882,228,358	97,993	607
RegCM	24 Cores	139,543.000,917	22,976	329
	48 Cores	80,112,643,804	33,157	828
	96 Cores	49,580,028,588	47,213	1,905
	144 Cores	51,640,032,689	58,138	2,252
	192 Cores	40,411,947,679	69,131	3,421
Gadget	24 Cores	316,651,890,866	152,567	964
	48 Cores	257,497,854,652	267,104	2,075
	72 Cores	207,637,515,308	415,640	4,004
DPSNN 32x32	32 Cores	8,795,221,526,254	34,533	8
	64 Cores	4,568,949,694,490	45,690	20
	128 Cores	2,744,786,342,258	65,125	47
DPSNN 64x64	64 Cores	23,829,773,201,115	170,572	14
	128 Cores	12,287,105,198,156	199,951	33

3.4 Analysis of Message Sizes

The message size analysis was performed to discern if message delivery could be improved by prioritizing either latency or bandwidth. For instance, if there are many short messages, we could prioritize latency over bandwidth; however, we should prioritize bandwidth to improve network and hence application performance with many large message sizes. Although 19 traces have been analyzed, only the identified representative patterns are presented and discussed for illustrative purposes.

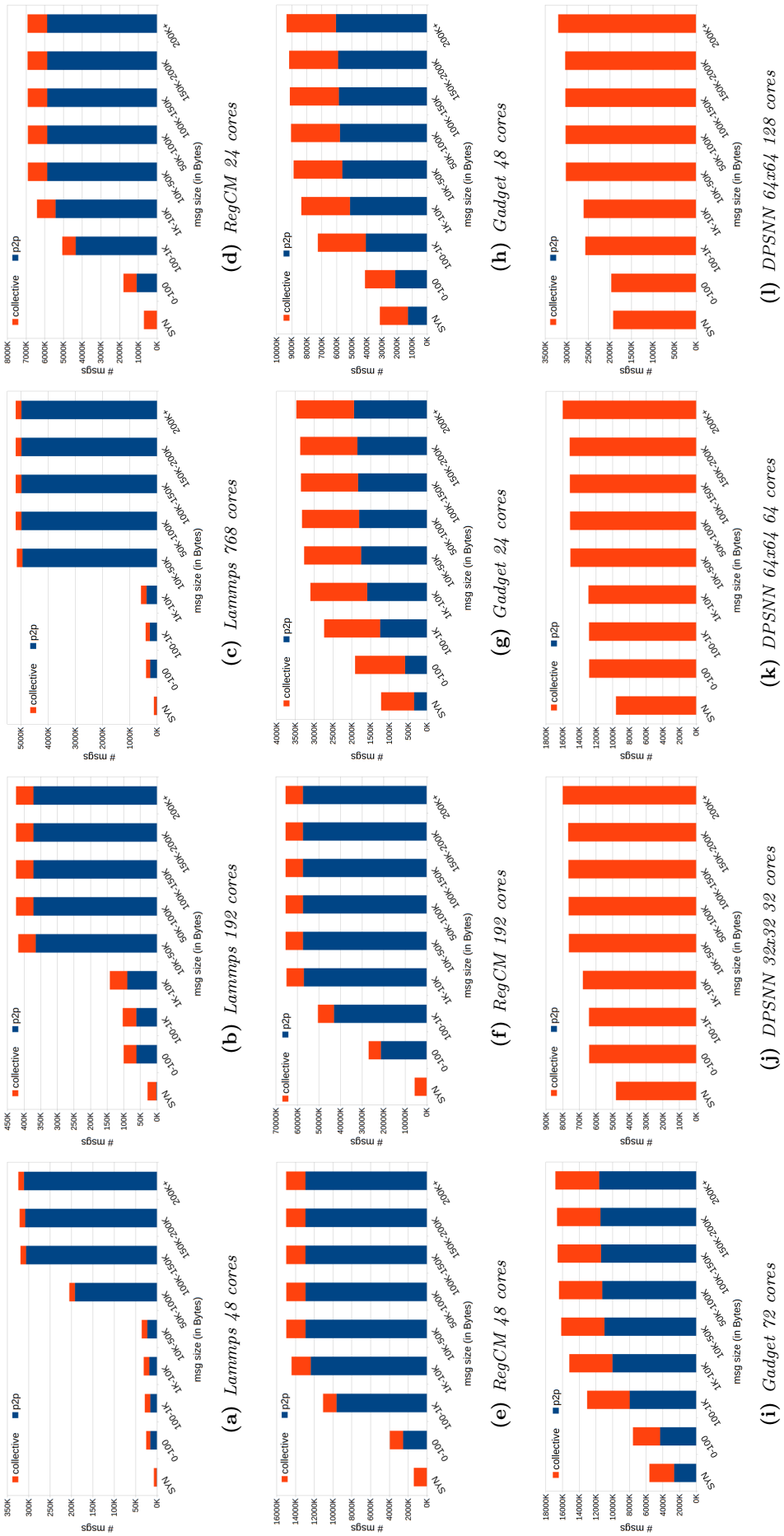
Figure 3.6 shows the cumulative message size distribution varying the core count across the studied traces. The Y-axis indicates the amount of messages (both due to collective and point-to-point MPI primitives) transferred and the X-axis the message size distributed in ranges. The first column (labeled as SYN) refers to the number of synchronization messages, whose main characteristic is that they do not include payload; however, they are also analyzed because, as results will show, they can generate a considerable amount of traffic in some applications.

Regarding message size distribution in Lammps (see Figures 3.6a-3.6c, it can be appreciated that most of the message sizes fall in between 10KB and 50KB for a core count larger or equal than 192. In contrast, for 48 cores, the dominating message size ranges from 50KB to 100KB. In summary, on average, the lower the core count, the larger message sizes are used.

Figures 3.6d-3.6f plot the message size distribution in the traces from the RegCM application. The message size in this application is quite homogeneous when varying the core count. The dominant size ranges from 100B to 1KB in all traces, although the 192-core trace also presents a significant amount of smaller messages (e.g., from 0 to 10KB).

Gadget (Figures 3.6g-3.6i) shows a high percentage of synchronization messages regardless of the number of cores. As observed, a significant amount of messages (ranging from 30% to about 50%) present a size smaller than 1KB, although around 20% of messages are bigger.

Finally, Figures 3.6j-3.6l depicts the message size distribution in the traces from DPSNN. The first plot (Figure 3.6j) corresponds to the application working with 32x32 neural columns (about 1.2M neurons and 2.6G synapses), while the two remaining plots refer to the application with 64x64 neural columns (about 5M neurons and 10G synapses). On average, the traffic generated by synchronization messages represents as high as by 60% of the total amount. Unlike the previous analyzed traces, the traffic in this application is due to MPI collectives instead of point-to-point messages.



(a) *Lammps 48 cores* (b) *Lammps 192 cores* (c) *Lammps 768 cores* (d) *RegCM 24 cores*

(e) *RegCM 48 cores* (f) *RegCM 192 cores* (g) *Gadget 24 cores* (h) *Gadget 48 cores*

(i) *Gadget 72 cores* (j) *DPSNN 32x32 32 cores* (k) *DPSNN 64x64 64 cores* (l) *DPSNN 64x64 128 cores*

Figure 3.6: Message size distribution.

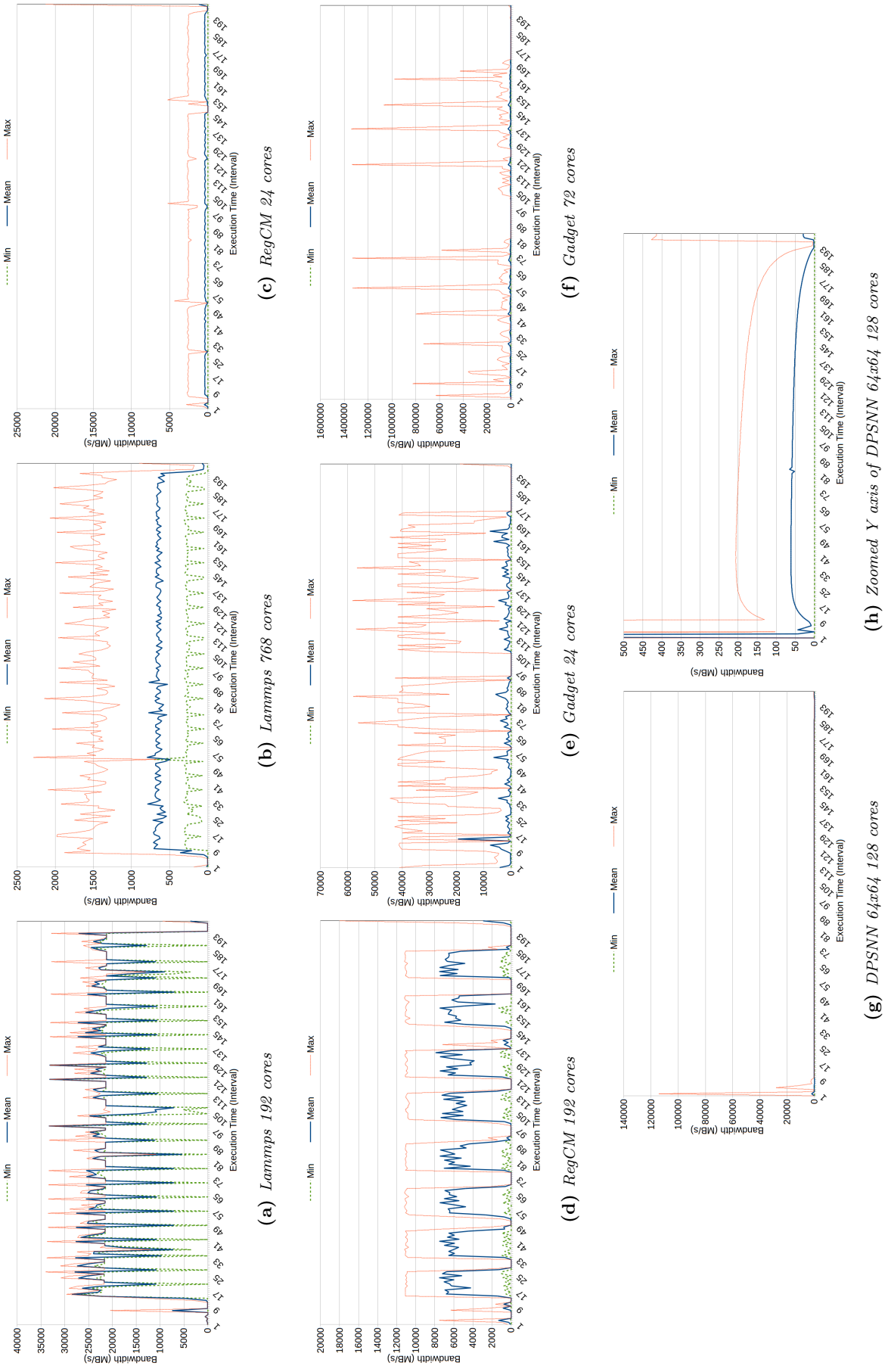
3.5 Analysis of Temporal Evolution

In Section 3.3 (Table 3.1) we showed the bandwidth consumption for each studied trace averaged along the application execution time. This section aims to analyze the dynamic bandwidth requirements in order to explore how requirements evolve with time.

We analyze all traces of each application and we found that the behavior of some applications changes when varying the number of cores. For illustrative purposes, we show an example of each of the multiple patterns exhibited by the studied applications when varying the core count. To ease the visual analysis and homogenize the representation of the plots, we divided the execution time of each application in 200 intervals of the same length. Then, we divided each interval in 10M-cycle subintervals and calculated the bandwidth consumption of each subinterval. These values show the average of the bandwidth consumption of each interval, which is labeled as Mean in the figures. The maximum (Max) and minimum (Min) bandwidth consumptions among the subintervals are also plotted to discern bursts communication patterns easily.

Figure 3.7a and Figure 3.7b show the bandwidth patterns of Lammps for 192 and 768 cores respectively, which are representative of all the patterns of this application. Figure 3.7a shows the behavior exhibited for a core count less or equal than 192, while the other figure shows the behavior exhibited when the number of cores rises over 192. It can be appreciated that in the former case, the studied variables many times overlap each other. However, when the number of cores rises over 192, there is a clear differentiation among the three plotted variables. As expected (see Table 3.1), the network traffic is much less important in 768 cores than in 192. Since the difference between the maximum and the average is relatively large, we made a further refinement by focusing on the most critical interval, that is, the interval with the highest difference.

The RegCM application exhibits similar bandwidth patterns regardless of the core count except for the 24-core trace. Figure 3.7c and Figure 3.7d show both patterns. We chose the 192-core trace, which is the one presenting the highest average bandwidth.



(a) *Lammps 192 cores* **(b)** *Lammps 768 cores* **(c)** *RegCM 24 cores*

(d) *RegCM 192 cores* **(e)** *Gadget 24 cores* **(f)** *Gadget 72 cores*

(g) *DPSNN 64x64 128 cores* **(h)** *Zoomed Y axis of DPSNN 64x64 128 cores*

Figure 3.7: Temporal evolution of min, mean and max bandwidth.

Compared to Lammps, RegCM presents much higher bandwidth requirements, more than 8x on average and a similar factor for the maximum bandwidth. Interestingly, bandwidth experiences a sharp rise at the end of the execution in both exhibited behaviors. Regarding Figure 3.7d, RegCM shows on average fewer bandwidth requirements than Lammps (by 40%) and similar maximum bandwidth requirements.

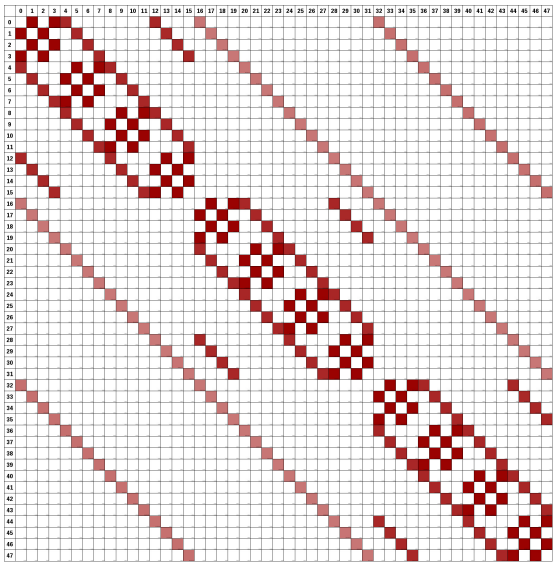
Figure 3.7e and Figure 3.7f show the temporal evolution of Gadget. This application presents a huge difference between the average and the maximum, which implies a bursty communication pattern; that is, there are sub-intervals with high communication requirements and others presenting very low traffic.

DPSNN exhibits a homogeneous behavior across all the studied traces. Because of this reason, we chose the trace presenting the highest bandwidth requirements, that is, a 64x64 neural matrix with 128 cores. Figure 3.7g shows the results. Compared to the previous applications, it can be observed that DPSNN has low bandwidth requirements during almost all the execution. However, in the first intervals, DPSNN presents huge bandwidth consumption. Because of this fact, we had to reduce the Y-axis in Figure 3.7g to appreciate the average, as shown in Figure 3.7h.

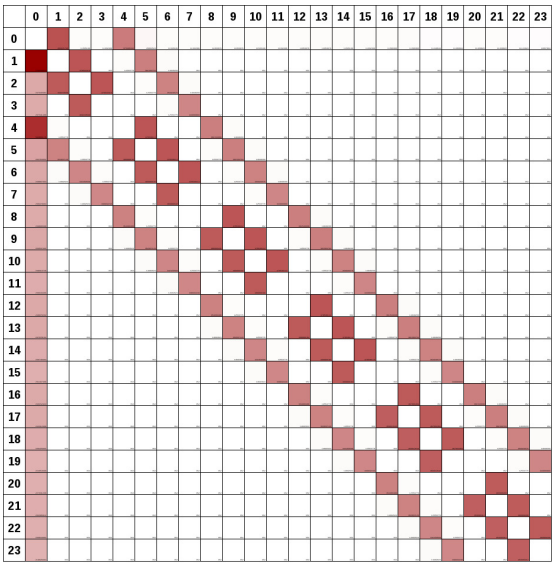
3.6 Analysis of spatial communication

Once the applications across the execution time have been analyzed, this section studies the amount of traffic that each core sends/receives to/from each other. In other words, the spatial distribution of communication among cores (matrix source-destination).

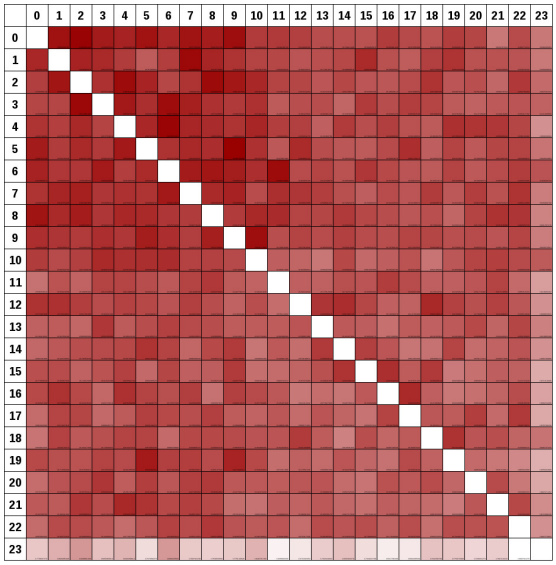
Figure 3.8 shows the resulting matrix of communications for the four studied applications. The darker the color, the higher the amount of transferred bytes. It can be seen that the traffic concentrates on a small percentage of cores in Lammps and RegCM, whereas it spreads among all the cores in DSPNN and Gadget. In DSPNN the traffic follows a regular pattern , darker in the central cores and lighter in the top and bottom cores, which means that cores trend is to communicate more with neighbors, while in Gadget cores randomly communicate all-to-all.



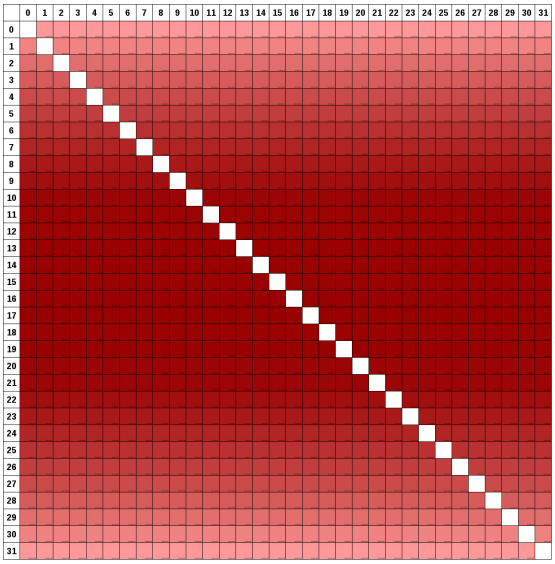
(a) *Lammps 48 cores*



(b) *RegCM 24 cores*



(c) *Gadget 24 cores*



(d) *DPSNN 32x32 32 cores*

Figure 3.8: Communication matrix among cores.

3.7 Summary

Workload characterizations are required to guide researchers in the design of new systems. In this chapter we have analyzed real traces of applications used in the European project ExaNeSt, which are being used to design and implement the interconnection network for an exascale system.

The analysis has been performed considering three main characteristics: the distribution of message types and sizes, the bandwidth consumed during the execution time and the spatial communication among cores.

Regarding the analysis of messages distribution, most applications (three out of the four studied) present a higher amount of point-to-point messages, although one of the applications (DPSNN) is entirely dominated by MPI collectives. In general, most messages are below 50KB regardless of the workload size.

The analysis of bandwidth consumed during execution time indicates that applications present a wide range of average bandwidth requirements; however, most applications present bursty communications patterns that can stress the interconnection network at given points of time.

Finally, the spatial communications matrix analysis for the different applications shows diverse spatial communication patterns among applications. For instance, in some applications, the traffic is spread among all the cores, whereas in others, bandwidth consumption is concentrated in hot spots. This means that, in order to support communication bursts and unclog congested network links, a suitable exascale network must provide higher-than-average bandwidth in the surroundings of key cores at specific points of time.

Chapter 4

Modeling a Photonic Network for Exascale Computing

System-level photonic network simulators can help guide researchers and designers to explore the design space by assessing multiple model choices. Nevertheless most current research is done on electrical network simulators, whose components work widely different from photonics components and, moreover, photonics technology adds new components that are not present in electrical networks.

This chapter discusses the critical steps that have been carried out to model photonic interconnects by extending currents electrical simulation frameworks. More precisely, we extended the INSEE simulator [62], one of the simulators used in the ExaNeSt project.

Modeling photonics networks on an electrical simulation framework is not a straightforward process, but it requires a sound knowledge of the basics of photonics and electrical networks. Many aspects are widely different since photonics offers new possibilities and prevents from some others. In this chapter, we summarize and compare how both technologies work and discuss the rationale behind the proposed extensions. The devised extensions model, among others, optical routers, wavelength-division multiplexing, circuit switching, and specific routing algorithms.

This chapter is organized as follows. First, we introduce the INSEE simulation environment that has been extended to model photonics technologies. Second, we present the main photonic components and features that have been model. Finally, the developed simulator is used to illustrate how the implemented photonics features impact on the network performance compared to electrical networks.

4.1 The INSEE Simulation Framework

This section summarizes the main characteristics of the Interconnection Network Simulation and Evaluation Environment (INSEE) [62] that we extended to model photonic networks. The INSEE simulator was originally developed with the aim of modeling electrical networks. This framework, originally developed at the University of Manchester, is publicly available and was selected as one of the main ExaNeSt project [43] simulation platforms.

INSEE implements multiple topologies (e.g. mesh-like, tree-like or dragonfly) and allows multiple traffic generation methods (e.g. synthetic, traces or architectural simulators); such a flexibility normally is not provided by other existing simulators [53]. INSEE achieves this flexibility with a light use of system resources in terms of memory and CPU computing power, allowing the simulation of large systems in a few days at most.

The core of the simulation environment consists of a functional system simulator and a network traffic generator. Both components feature a modular design that can be expanded with new modules or extending and modifying the capabilities of the existing ones. We leveraged this modular design to carry out multiple extensions that allow INSEE to support all-optical networks.

As a simulator originally designed to study electrical networks at a low level of abstraction, INSEE implements electrical network components, most of them can be reused to simulate photonics based networks. Nevertheless, key components need to be highly modified to mimic the behavior both of photonics technology and photonics network components, as is the case of the routers, the switching technique and links. In addition, other techniques only apply to photonics technology, for instance, current photonics technology allows to populate a single fiber link with multiple channels.

4.2 Proposed Photonics Extensions

This section presents the extensions and upgrades of components proposed to model photonics networks, mainly focused on the INSEE simulator. For this purpose, we briefly discuss and compare the working behavior of the major components of a high-performance network. This analysis is carried out from the underlying technology (electrical vs photonics) perspective.

4.2.1 Optical Routers versus Electrical Routers

Typical electrical routers implement internal buffers at input ports that provide local temporal storage for in-transit packets (or a smaller data unit, depending on the switching technique). Packets are kept in these buffers in case they cannot advance due to traffic or network constraints.

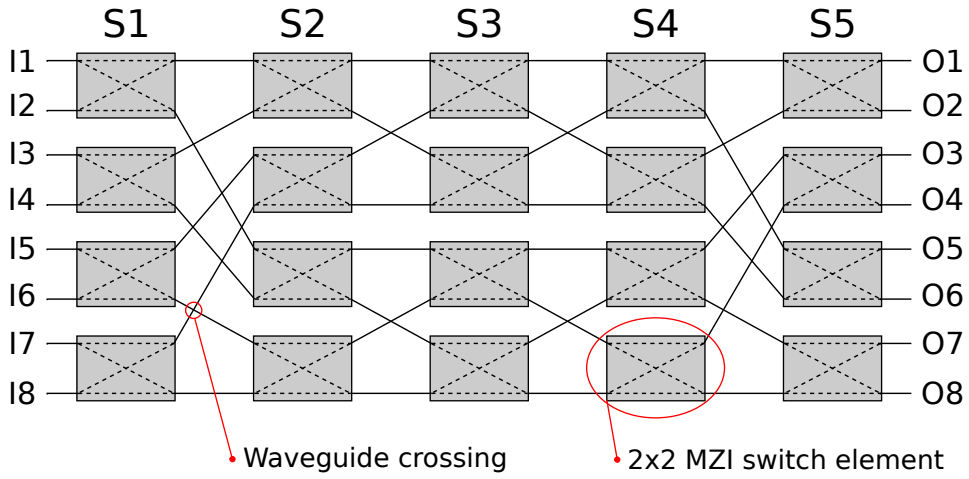


Figure 4.1: 8x8 optical switch based on a Benes architecture.

On the contrary, all-optical networks normally do not provide buffering capacity at network routers. This fact is illustrated in Figure 4.1, which presents an scheme of an optical switch with no buffering support. In particular, it is shown an 8x8 (8 inputs and 8 outputs) reconfigurable optical switch based on a Benes architecture. The minimal building block is a 2x2 Mach-Zehnder Interferometer switch element and each optical path goes through 5 stages of those elements. As we can see, waveguide crossings are required for the two-dimensional connection of the 20 switching elements [48].

The lack of buffering capacities means that once the data is injected in the network it must be delivered without waits, that is, without being blocked all along its path until destination. An

interesting attempt to deal with this drawback is the use of hybrid electro-optical routers but this approach, requires electro-optical and opto-electrical conversions to write and read data into and from electric buffers. As discussed below, the fact that most all-optical networks do not provide buffering support has important consequences for the switching technique used in all-optical networks.

4.2.2 Circuit Switching versus Packet Switching

Two main switching techniques, circuit switching and packet switching, can be implemented in electrical networks.

In the former, to send a message trough a network, a path (or circuit) needs to be selected from the source node of the message to the destination node. The path selection 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 node to reach the destination node without any contention along the path. One of the main drawbacks of circuit switching is that if any of the links that compose the path cannot be reserved (for instance, it is being used for another circuit), the message transmission must wait until the link is released and can be reserved, which leads highly wasteful of scarce network resources. This is the main reason because circuit switching is barely used in current high-performance electrical networks.

In the latter, depending on the routing mechanism used, wormhole or virtual cut-through, arbitration is performed on every switch on a per-packet basis. That is, the message is sent to the buffer of the next switch, from where it goes to the next one or remains stored in the buffer until it can be routed again. Notice that classical packet switching cannot be supported by design by all-optical networks since buffering is not mature enough to implement buffers in all ports of the optical routers.

4.2.3 Dense Wavelength-Division Multiplexing

Although circuit switching can be considered an outdated switching method, it can bring important advantages in optical networks combined with dense wavelength-division multiplexing (DWDM). In addition, recent studies claim that circuit switching is a better option than packet switching for optical networks since it improves *switching time*, that is, the time required for electronic components to establish a new optical configuration [59, 33].

DWDM multiplexes the optical spectrum in different wavelengths within the same link, thus physically allowing several transmissions (e.g., as much as the number of wavelengths) per link. The maximum amount of wavelengths per link is limited by the optic communication band [2] and depends on the minimum spacing between wavelengths' frequency values without causing interference between transmissions on two consecutive wavelengths. Nowadays, a 100GHz channel spacing is typically used, which gives 40 wavelengths per link [27], 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 [70], 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.

To model both DWDM and circuit switching together in the baseline simulator, several design issues have been considered. First, since there are multiple wavelengths in the same link, circuit switching needs to be adapted since more than one path per link can be defined at the same time; that is, each channel (i.e., a set of wavelengths) can be part of an eligible path. Therefore, when a message is ready to be injected into the network, the number of possible paths that it can reserve is much higher than in electric networks.

Figure 4.2 shows an example illustrating how circuit switching works with DWDM in a 2D-mesh network. 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 using minimal path. 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 route does not have any free channel, the reservation will fail, similarly as it occurs in conventional circuit switching.

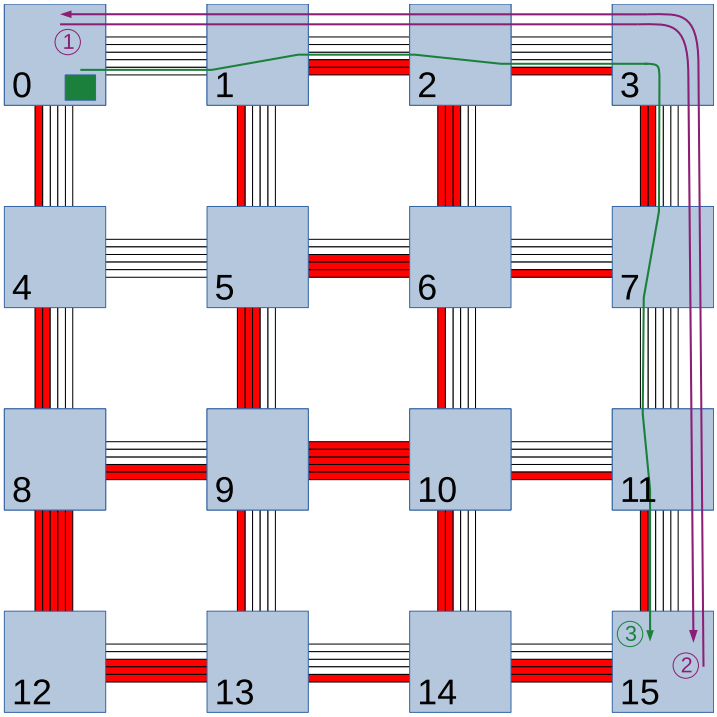


Figure 4.2: Message transmission example with circuit switching and DWDM.

4.2.4 Photonic Links versus Electrical Links

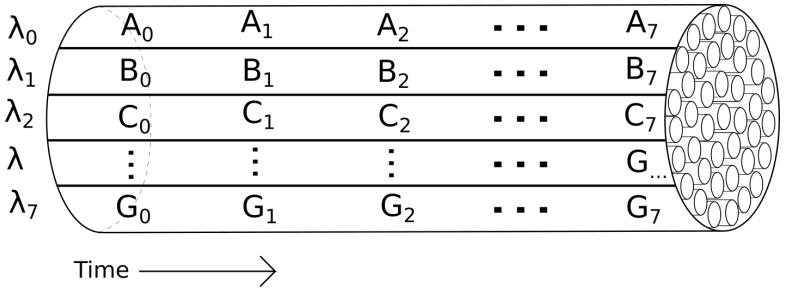
As explained above, electrical links only allow to send information of a single message or packet on a network cycle. In contrast, optical links are split in channels each one using a different set of wavelengths.[1]

To exploit this issue, in the simulator we need to provide support to configure the amount of wavelengths per optical link and group this wavelengths in independent channels, handling the transmission of a different message in each one. In short, the configuration options regarding optical link can be provided by the aggregated bandwidth or the channel bandwidth and the number of channels per optical link.

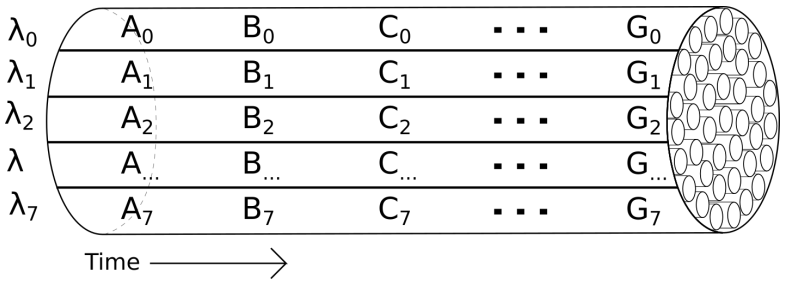
[1]Note: The term channel has been used in the literature also to refer to a single wavelength in optical technology. In this dissertation we use this term from a *computer perspective* to refer to a set of wavelengths used to transfer the same message.

4.2.5 Transmission units: Phits versus Bits

Virtual cut-through and wormhole are the most widely used packet switching techniques in electrical networks. These techniques split the packet in small *flits*, which are the units of flow control. Flits are divided in *phits* (physical units). A phit is the amount of bits that can be transferred in a single electrical network cycle. In contrast, optical networks only can transfer one single bit per wavelength and per network cycle (note that optical cycles are much smaller than electrical cycles).



(a) Coarse-grain cycle approach.



(b) Packed wavelengths approach.

Figure 4.3: Example of using eight wavelengths to transmit eight phits, referred to from A to G, with the studied transmission approaches.

In general, electrical network simulators define the phit size as an integer in amount of bytes (8 bits). Therefore, when such a kind of electrical simulator is used as a basis of an optical simulator, the byte is kept as the minimal transference unit per cycle. However, as mentioned above, optical links transfer one bit per wavelength in a given network cycle. Therefore, a new approach is required to fulfill this mismatch.

We provide two main approaches, *coarse-grain cycles* and *packed wavelengths*, to model the transmission of bits instead of phits in INSEE. The former approach defines coarse-grain cycles, consisting of 8 *small* simulation cycles (i.e., photonic cycles) working as the cycle unit, which allows submitting 8 bits (the minimum phit size in the baseline simulator) per cycle using the

same wavelength. The latter groups 8 wavelengths, which acts as a single transmission unit; that is, 8 wavelengths are used to transmit 8 bits of the same message in a single photonic cycle, which implies that the minimum channel size is 8 wavelengths. Figure 4.3 presents an example where 8 phits (labeled as A to G) are transmitted in both approaches. In the coarse-grain cycles approach, each phit is transmitted in a different wavelength while in the packed wavelengths approach several wavelengths cooperate to transmit the same phit in parallel.

As shown in Table 4.1, choosing between both approaches presents a trade-off in the network features. For an optical link composed of 40 wavelengths, the coarse-grain cycle approach can provide up to 40 parallel channels, offering in each channel the bandwidth of a single wavelength (40 Gbps per channel). In contrast, the packed approach provides a limited maximum number of 5 channels, but each one aggregates the bandwidth of 8 wavelengths ($40 \times 8 = 320$ Gbps per channel). Note that between both approaches there are several possibilities for *hybrid* approaches. For instance, the amount of channels can be halved with respect to the coarse-grain cycles approach (second and third row of Table 4.1). Then, instead of transmitting 1 byte in 1 network cycle using 1 wavelength, the byte can be divided in 2 nibbles that are transmitted by 2 wavelengths (doubling the network frequency).

# **Approach**	# **Channels**	**Channel Bandwidth**
Coarse-grain cycles	40	40 Gbps
Hybrid	20	80 Gbps
Hybrid	10	160 Gbps
Packed wavelengths	5	320 Gbps

Table 4.1: Trade-off between the studied transmission approaches for an optical link populated with 40 wavelengths.

4.2.6 Topologies and Routing

Network topologies implemented in electrical simulators need to be tailored to work under photonics technology. In particular, the routing algorithm must be adapted to circuit switching with DWDM approach.

To this end, we explain some approaches to adapt the three most widely used network topologies for HPC together with the most used routing policies:

- Torus. The modeled topology uses Dimension Order Routing (DOR) in which in order to establish a path between each source-destination pair only paths using the minimum

number of hops are considered, crossing the network dimensions always in the same order, in particular XYZ.

- Fat tree. This topology uses an adaptive version of the Up/Down routing. In this case, among the possible alternatives in the Up stage, we select the links which have the lowest channel utilization with the aim of balancing the use of the network links.

- Dragonfly. In this topology, the most used routing algorithm is Valiant [44]. This policy selects randomly an intermediate switch (named "proxy") and performs minimal routing from source to proxy, and from proxy to destination. Although the maximum length path of Valiant is longer than when using minimal routing (from 5 to 7 hops), it is known that it provides higher load balancing and avoids bottlenecks for specific types of traffic. However, DWDM provides advantages regarding parallelism. Thus, minimal routing can be implemented with photonic technology.

4.3 Performance Evaluation

The aim of this section is to illustrate how a photonic network with features discussed in Section 4.2 performs in terms of network bandwidth and latency with respect to an electrical network. In other words, we compare the results obtained with the devised simulator extensions with those provided by the baseline INSEE network simulator.

Next, we discuss the design options that have been selected to carry out the experiments presented in this chapter.

4.3.1 System Details

This section specifies the network bandwidth, the number of channels per link and the network topology that have been considered to obtain the results.

In the experiments we consider a 10 Gigabit Ethernet electrical network. We choose this bandwidth because an important set (by 35.6%) of the supercomputers ranked in Top500 [73] implement this network technology.

In the case of the modeled photonics network, a 40 Gbps is considered as the bandwidth provided by each individual wavelength. This bandwidth is selected according to the actual VCSEL technology exposed in Section 1.2.

Approach	# **Channels**	**Phit Size (bytes)**	**Channel BW (Gbps)**
Electrical	-	4	10
Packed-wavelength	5	1	320
Hybrid	10	1	160
Hybrid	10	2	160
Hybrid	10	4	160
Hybrid	20	1	80
Hybrid	20	2	80
Coarse-Grain	40	1	40

Table 4.2: Studied network configurations considering a link populated with 40 wavelengths.

Table 4.2 summarizes the main design choices of the studied configurations for a photonic link multiplexed in 40 optical wavelengths. In addition to the two main transmission approaches, labeled as *packing wavelength* and *coarse grain*, hybrid intermediate schemes that combine both approaches have been studied.

Taking into account that each wavelength provides a bandwidth of 40 Gbps, the aggregate bandwidth of the link for all configurations is 1.6 Tbsp. This bandwidth is split among the channels for each configuration depending on the amount of wavelengths that compose them. In this chapter, we also study future optical links with 80 wavelengths, which gives 3.2 Tbps as aggregate bandwidth per single fiber. To carry out this study, we devised new configurations obtained by doubling the number of channels of the configurations depicted in Table 4.2.

4.3.2 Experimental Results

To carry out this experiment we consider two excerpts of traces provided by partners from the execution of two ExaNeSt workloads, `Gadget` and `Lammps` discussed in Chapter3. These traces have different number of nodes; thus, for `Gadget`, which has 72 nodes, we have used a 3D torus with 4x6x3 nodes and for `Lammps`, which has 192 nodes, the 3D torus is configured with 4x8x6 nodes.

Figure 4.4 and Figure 4.5 present the execution time obtained with the proposed electrical network and the photonic network configurations with the aforementioned traces. The first bar corresponding to the electrical network. Subsequent bars correspond to photonic configurations, which are labeled (X axis) using four variables, referring to (from left to right) the amount of wavelengths per link, the number of channels per link, the size of the phit and the bandwidth per channel.

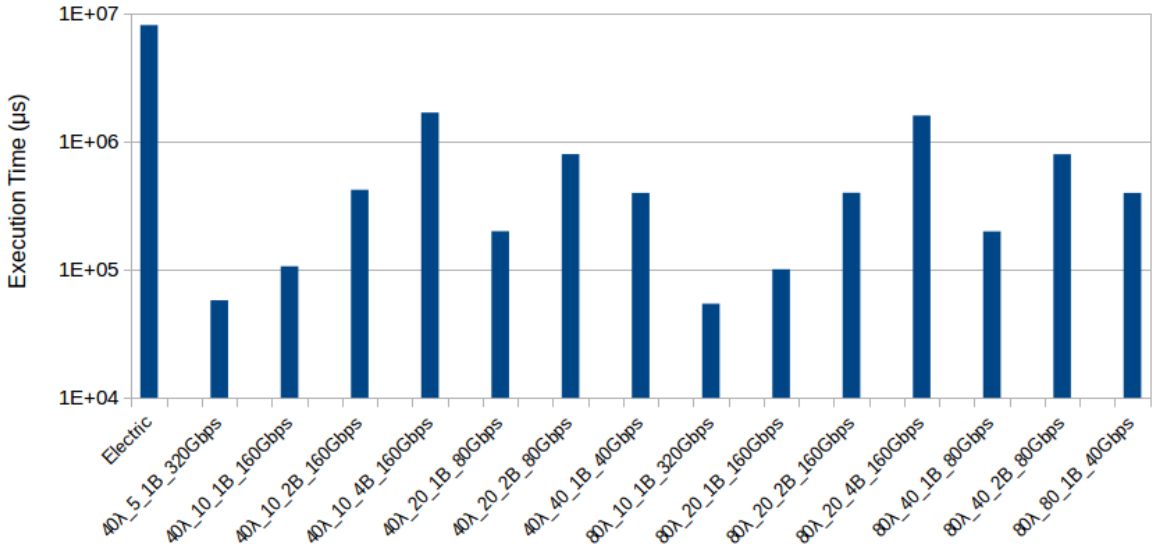


Figure 4.4: Execution time (in us) for *Gadget*.

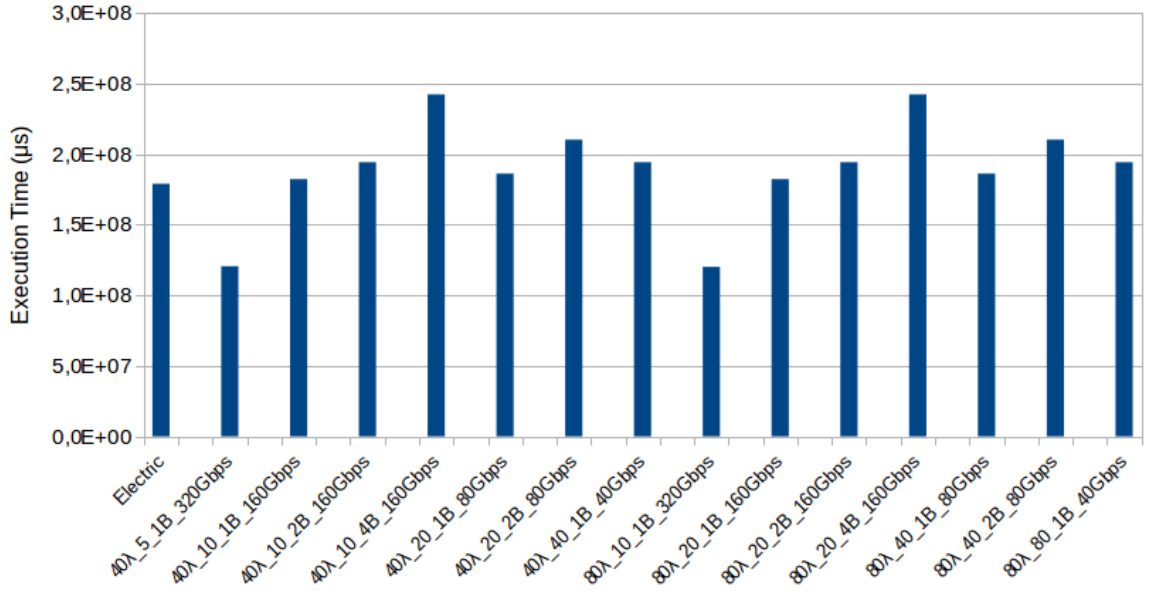


Figure 4.5: Execution time (in us) for *Lammps*.

As it can be observed, configurations with 40 wavelengths per link provide the same results as those with 80 wavelengths regardless of the workload. This means that an aggregate link bandwidth of 1.6 Tbps, achievable with current photonics technology, suffices to achieve the best performance. Thus, from now on, we focus on configurations with 40 wavelengths per link. Among these, the configuration that provides the best results in both studied workloads is *packed-wavelength* (see Table 4.2). This is because it features the highest bandwidth per channel (320 Gbps), which means that this parameter is the one that most impacts on performance.

On the other hand, for a given bandwidth per channel (e.g. 160 Gbps), the best strategy is to reduce the phit size. For instance, in both traces, the best configuration for 80 Gbps per channel is the one with 1-byte phit size (i.e. 40λ_20_1B_80Gbps). This is because large phit sizes waste bandwidth when they are underused. This effect is so significant that sometimes it is better to reduce the phit size even if the bandwidth per channel is reduced (e.g. comparing 40λ_10_2B_160Gbps with 40λ_20_1B_80Gbps in Figure 4.4).

It can be seen that while in Figure 4.4 the photonic network widely improves the performance against the electric network (ranging the benefits from 1 to 2 orders of magnitude – pay attention to Y axis scale–), in Figure 4.5 the execution time is similar or better in the electrical network except in the optimal configurations with 320 Gbps per channel. The reason behind these results is that the former excerpt belongs to a phase of the execution dominated by network transactions with scarce computation time while the latter has a large amount of computation time interleaved with network usage. Thus, the benefits of photonics are only significant with the highest bandwidths.

Figure 4.6 presents the average transmission delay, which includes the injection delay and the transit delay of packets for *Gadget*. As it can be observed, almost all the delay is caused by injection. Although in this figure the electric network shows a low delay, it must be taken into account that in the electric network the delay is computed for each 64-byte packet while in the photonic configurations the delay is calculated for the transmission of a whole message, thus they cannot be directly compared. Nevertheless, the results confirm a strong correlation between network delay and execution time for the optical configurations.

In `Lammps`, as shown in Figure 4.7, the distribution of delays widely differs between electrical and photonic networks. While in the former most of the delay is caused by injection, in the latter the average delay is evenly distributed between injection and transit delays. This is because the average message size in `Lammps` is longer, as analyzed in Section 3.4. Thus, a low

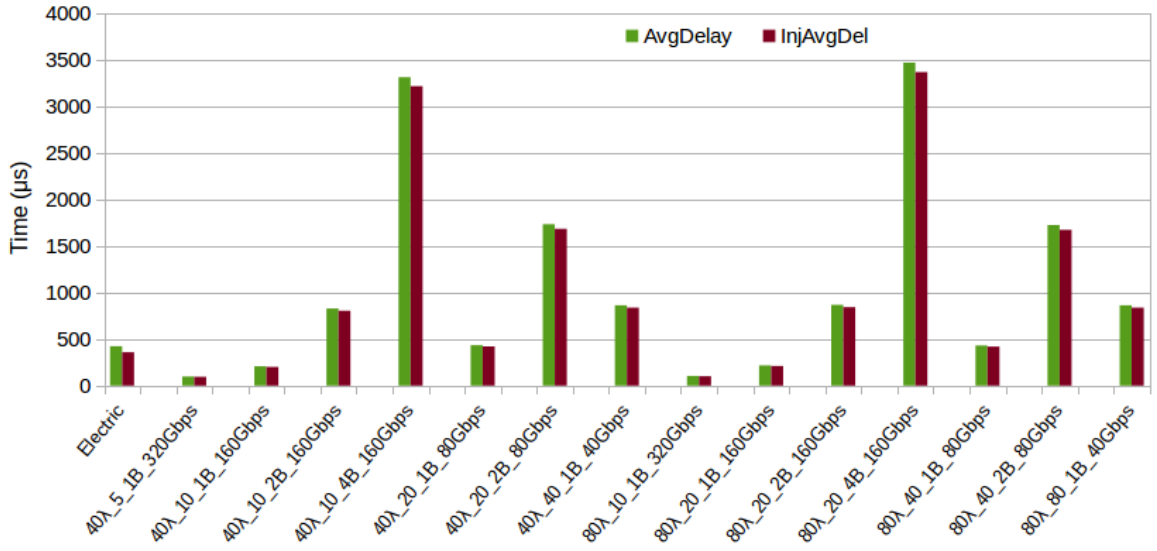


Figure 4.6: Average network delay (in us) for *Gadget*.

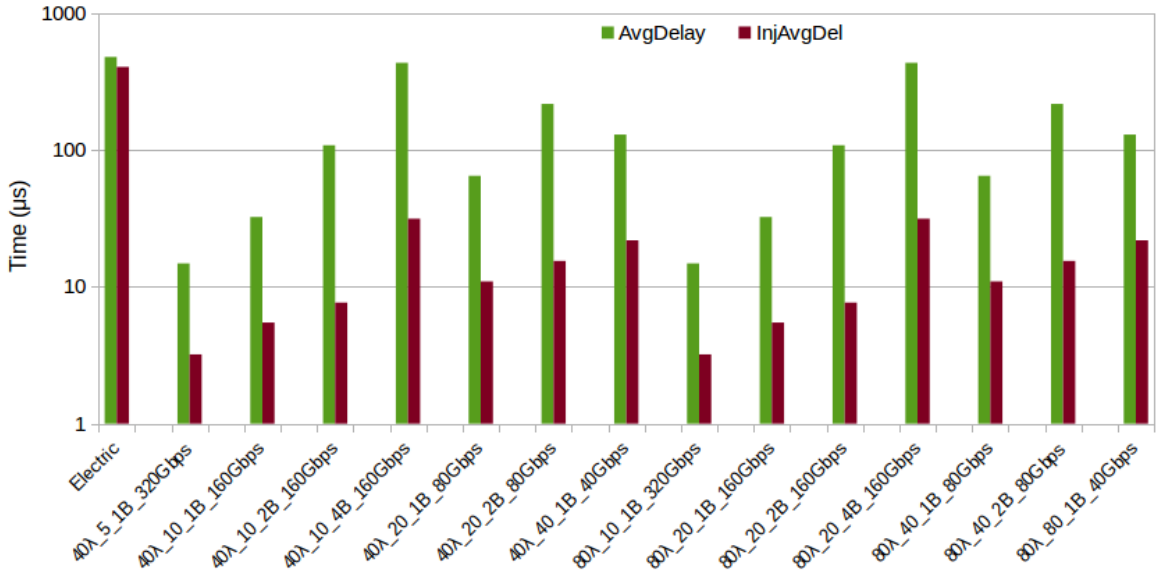


Figure 4.7: Average network delay (in us) for *Lammps*.

photonic bandwidth per channel has a high impact on the total delay, affecting execution time accordingly.

4.4 Summary

Due to Exascale networks will count on thousands of nodes, and photonics has emerged as a promising network technology to face data movements, photonic networks simulation tools are required to guide designers in decision taking.

This chapter has discussed the process required to build a photonic network simulator on top of the INSEE electrical network simulation framework. We have identified the key components that are involved, described the major working differences between such components depending on the underlying technology (electrical and optical), and studied optical-only features of some components (e.g. DWDM). It has been presented major simulator extensions in the router and the link architecture, DWDM implementation together with circuit switching, and specific routing methods.

For illustrative purposes, some experiments have been conducted aimed at comparing photonic networks with electrical networks in a 3D torus topology. Multiple configurations have been evaluated varying four main parameters: the amount of wavelengths, the number of channels per link, the phit size and the bandwidth per channel. Experimental results, obtained with excerpts of real applications provided by ExaNeSt partners, show that depending on the optical network configuration the execution time of the application can widely differ even with two optical network technologies (e.g. 1.6 and 3.2 Tbps aggregate link bandwidth, that is, 40 and 80 wavelengths per link respectively). In general, the future 3.2 Tbps aggregate link bandwidth will not provide additional performance benefits for the studied workloads, but 1.6 Tbps and 320 Gbps per channel is enough to obtain the best results across the studied configurations. Moreover, we found that the parameter that most impacts on performance is the bandwidth per channel, achieving the best results with 320 Gbps channels. Finally, for lower bandwidths per channel (e.g. 160 and 80 Gbps), reducing the phit size provides the best trade-off.

Chapter 5

Analysis of the Performance in Photonic Networks

After implementing photonic technology into a well-known simulation framework, which includes the use of optical routers, wavelength-division multiplexing and circuit switching, among others, and perform an initial evaluation of photonic technologies, in this chapter we present an extensive simulation study using realistic photonic network configurations with synthetic and realistic traffic applications. In particular, we have used applications' traces obtained by ExaNeSt partners discussed in chapter 3, which have been used to design and evaluate the ExaNest network architecture. These traces perform a high amount of communications compared to computation, which is key to evaluate the network.

The chapter first discusses the setup considered to perform simulations, and after that, we present and analyze the obtained results.

5.1 Experimental Setup

This section discusses the experimental setup performed to evaluate the performance of the photonic interconnect. After describing the system configurations for both electrical and photonic networks, we discuss the characteristics of the network traffic.

5.1.1 System Configurations

The network configuration includes the number of wavelengths per link, the bandwidth, the network topologies and the kinds of traffic that have been considered to obtain the results.

The experiments consider the same configurations exposed in Section 4.3.1, an electrical network 10 Gigabit Ethernet and a photonic network with an aggregate bandwidth of 1.6 Tbps, 40 wavelengths per link, and different number of channels, as shown in Table 5.1. This table shows the characteristics of the seven studied photonic configurations, one packed-wavelength, one coarse-grain, and five hybrid configurations.

Technique	# **Channels**	**Phit Size (bytes)**	**Channel BW (Gbps)**
Electrical	-	4	10
Packed-wavelength	5	1	320
Hybrid	10	1	160
Hybrid	10	2	160
Hybrid	10	4	160
Hybrid	20	1	80
Hybrid	20	2	80
Coarse-Grain	40	1	40

Table 5.1: Studied network configurations considering a link populated with 40 wavelengths.

5.1.2 Network Traffic

To evaluate the impact of varying the amount of photonic channels on the execution time, two types of traffic have been taken into account, synthetic and based on traces extracted from ExaNeSt applications.

Regarding synthetic traffic, we analyze the impact of the packet length on the execution time using uniform traffic, which is related to the ability to establish a photonic route from source

to destination, in particular we perform several experiments sending 10 GB of data varying the average length of the packets, from 1 KB to 128 KB. These experiments have been carried out using an (8x8x8) 3D torus, an 8-ary 3-tree and a (4,6,4) dragonfly topology.

With respect to more real traffic, two traces collected by partners from the execution of two ExaNeSt MPI based applications analyzed in Chapter 3, Gadget and Lammps, have been studied. Note that these applications are composed of different number of tasks, thus, different network configurations have been used to simulate each trace.

On the one hand, for Gadget application, composed of 72 tasks, we have used a (4x6x3) 3D torus, a 3-ary 4-tree and a (3,4,3) dragonfly. Gadget performs short computations, mainly using barriers for synchronization and broadcasts for data exchanges.

On the other hand, to simulate Lammps, composed of 192 tasks, we have used a (4x8x6) 3D torus, a 6-ary 3-tree and a (6,3,4) dragonfly. In this case, several kinds of collectives are used to perform the data exchange but mainly point-to-point messages. As in the previous case, the amount of computation is negligible compared to the amount of data sent through the network. Notice that it is not possible to match the number of nodes in the topologies with the number of tasks of the traces. For that reason, the size of the topologies is as close as possible to the number of tasks of the applications.

5.2 Experimental Results

In this section, we analyze the results obtained for distinct photonic and electric network configurations with both our modified INSEE [50] and the original one [55]. First, we present a detailed analysis of the results obtained using synthetic traffic and several photonic configurations. Then, we analyze the performance achieved by the two traces gathered with the applications from the ExaNeSt project in both electric and photonic networks. Section5.2.3 concludes remarking the configuration parameters that most affect the performance considering the two kinds of studied traffic.

5.2.1 Synthetic traffic

First, we analyze the execution time obtained with synthetic traffic using the three aforementioned topologies with the four channel splitting techniques described in Table 5.1, that is, taking into account the number of channels, from 40 to 5.

From the results depicted in Figure 5.1, the parameter that most impact on performance is the number of channels in which is split the photonic link. In fact, the results show an almost linear relationship between the number of channels and the execution time. Each time we reduce by half the number of channels, the execution time is also reduced almost by half. The rationale behind this behavior is that photonic channels are underused as discussed below, and maximizing the bandwidth per channel is the most effective way to take advantage of the high bandwidth provided by the photonic interconnect. Following this rationale it could be inferred that a single channel would be the best design choice, this configuration will be studied in Chapter 6.

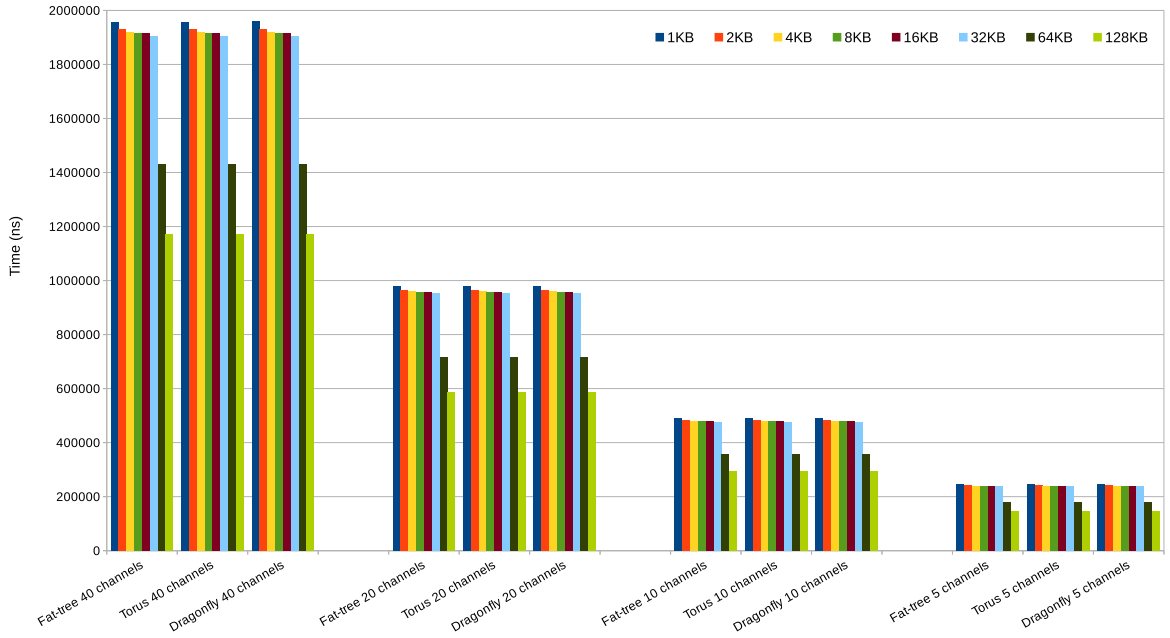


Figure 5.1: Execution time (in ns) sending 10 GB of synthetic traffic using 8-packet lengths from 1 KB to 128 KB in the studied network topologies. Photonic links are configured using 5, 10, 20 and 40 channels.

Regarding the message size, the results show that the longer the message the shorter the execution time. The reason for this behavior is that the incurred overhead in a photonic network significantly rises with the reservation of the paths, hence it reduces when long messages are used because this approach reduces the amount of messages sent. Surprisingly, the network

topology employed does not affect the results, which indicates that the network is underused, mainly due to the high amount of bandwidth provided by the photonic technology.

As shown, the best performance is achieved using the photonic configuration with a fewer number of channels. Figure 5.2 depicts the average time required by a packet to travel from source to destination (injection + transmission). As expected, results show that the 5-channel configuration is the most efficient for all topologies and packet sizes due to the higher bandwidth utilization. Furthermore, we can also see that as the packet size increases, for a given size, the average time required to send a packet slightly decreases at reducing channels per link. Notice that, when all packets sizes are considered, using this configuration has a higher impact on performance. The overhead, depicted in Figure 5.3, is mainly caused in the route reservation stage and the injection delay which is up to 30% of the total time for 1KB messages and just by 0.3% for 128KB messages.

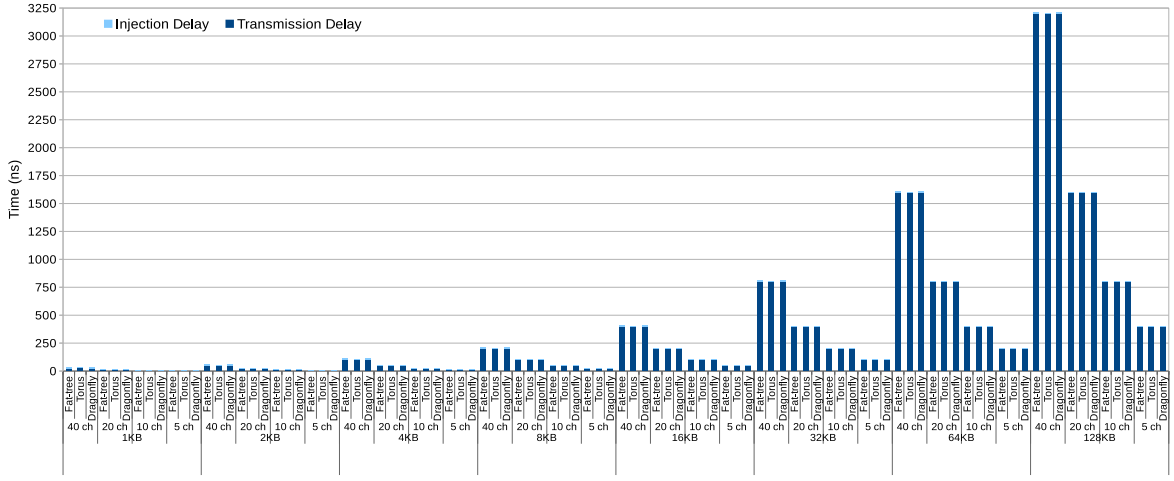


Figure 5.2: Average sending time (in ns) per packet showing the injection delay and transmission time varying the packet length from 1 KB to 128 KB in the studied network topologies. Photonic links are configured using 5, 10, 20 and 40 channels.

The injection delays shown in Figure 5.3 rise as a consequence of the reservation of the channel to perform the transmission. As photonic interconnects use circuit switching, the network is congestion-free, meaning that no interference will occur once a packet is injected. To analyze the impact of the network topology on the time required to reserve a channel, Figures 5.4 and 5.5 represent the number of retries (failed attempts to establish a route for a given message). Results for the 5-channel photonic configuration are only shown because no retries appear with a greater number of channels.

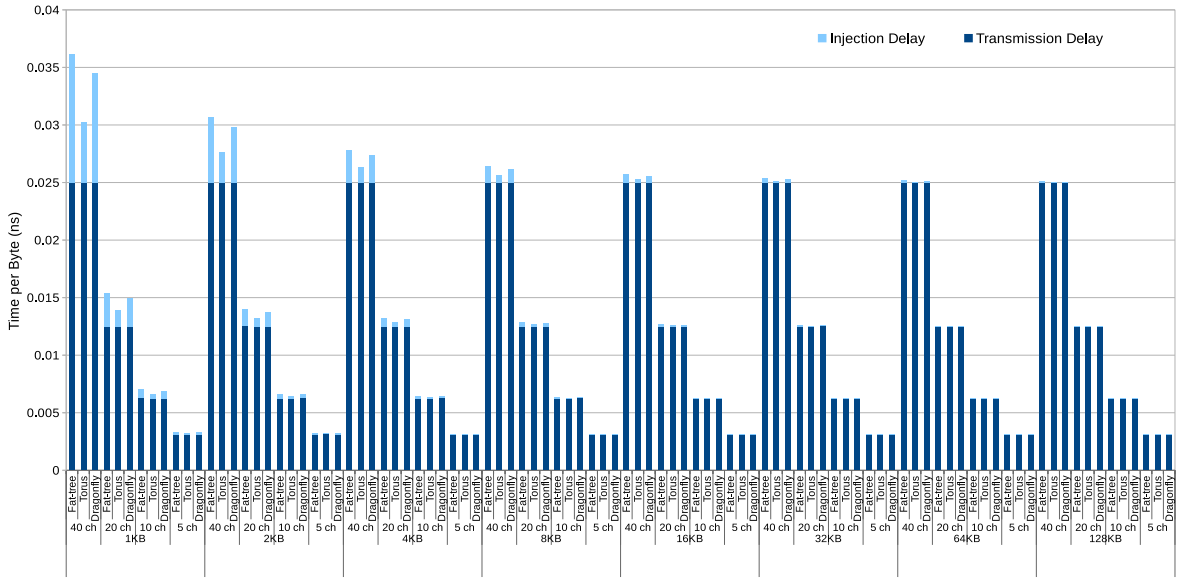


Figure 5.3: Average sending time (in ns) per byte showing the injection delay and transmission time varying the packet length from 1 KB to 128 KB in the studied network topologies. Photonic links are configured using 5, 10, 20 and 40 channels.

Figure 5.4 plots the total number of retries. If we focus on the fat tree topology we can see that the number of retries is negligible regardless of the packet size. The reason is the high number of alternative paths provided by this network, which makes the reservation of photonic paths to success almost always. In the case of the torus topology, the number of retries slightly decreases as the packet size increases. The reason for this behavior is the use of static routing, that even when we reduce the number of packets sent, these packets will use the same set of paths to reach destination. A completely different picture happens for the dragonfly topology in which the increase of the packet size reduces remarkably the number of retries. In this case, reducing the number of packets sent also reduces the utilization of links that connect the local group to intermediate proxies mainly due to Valiant routing, thus reducing the number of failures to reserve a photonic path.

The obtained reduction in the number of retries was expected because it is sent the same amount of data using longer packets. In order to check the real impact of the use of longer messages we represent the number of retries per packet in Figure 5.5. In this case, it can be observed that the number of retries increases with the packet length. This is the expected behavior because the transmission of packets requires longer time, thus bigger packets maintain the photonics channels occupied for longer, avoiding others nodes reserve channels for their transmissions. As a result, subsequent transmission attempts will be delayed more often. In any case, we have to consider that, although the maximum number of retries per packet is obtained with the longest

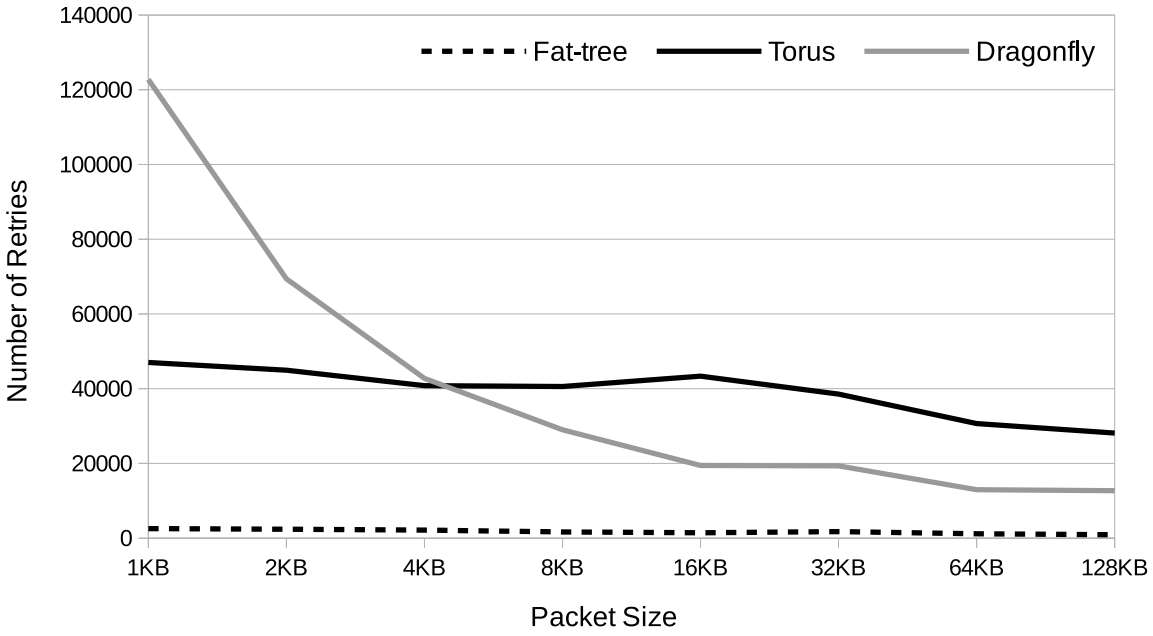


Figure 5.4: Total number of retries to establish the photonic route with 5 photonic channels.

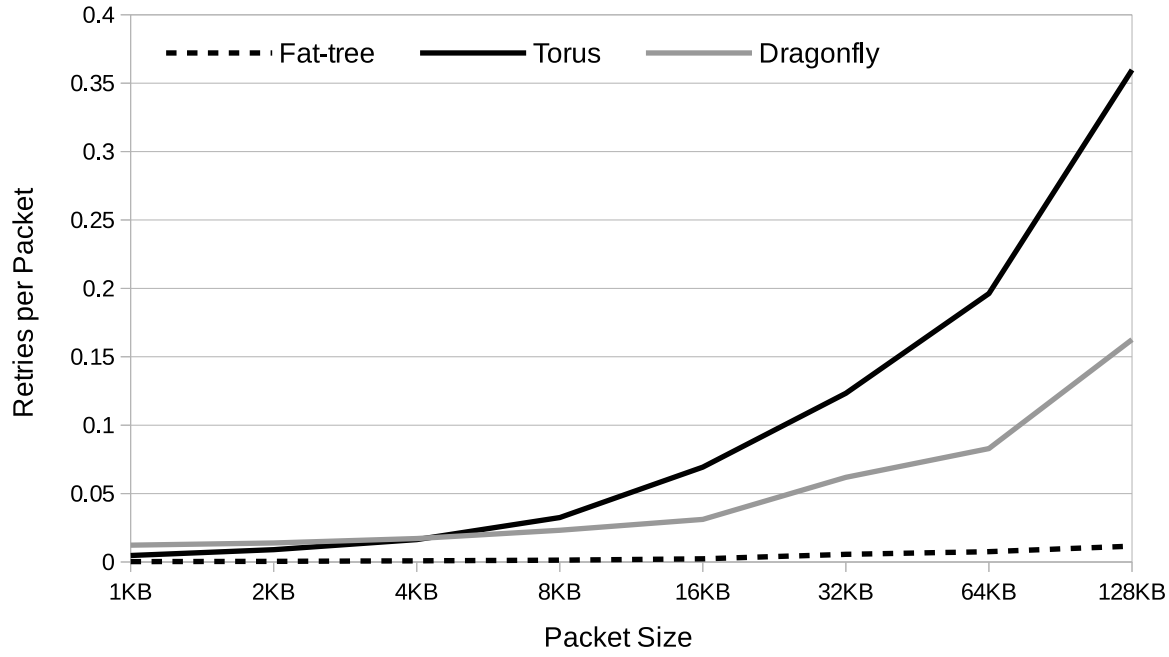


Figure 5.5: Retries per packet to establish the photonic route with 5 photonic channels.

packets, the overhead introduced is negligible with respect to the overall transmission time (see Figure 5.3).

5.2.2 Real application traffic

We have analyzed the photonic configurations using synthetic traffic, however, this kind of traffic does not model interactions between diverse nodes in real applications such as the causality between messages. For that reason, we evaluate the performance that real parallel applications can achieve when executed over photonic interconnects. Furthermore, we also compare the performance of those applications when executed in traditional electrical networks. Obtained results are depicted in Figure 5.6. For the sake of clarity, due to the huge differences in terms of execution time, we represent the results using two axes, the left one for electrical network and the right one for photonic configurations.

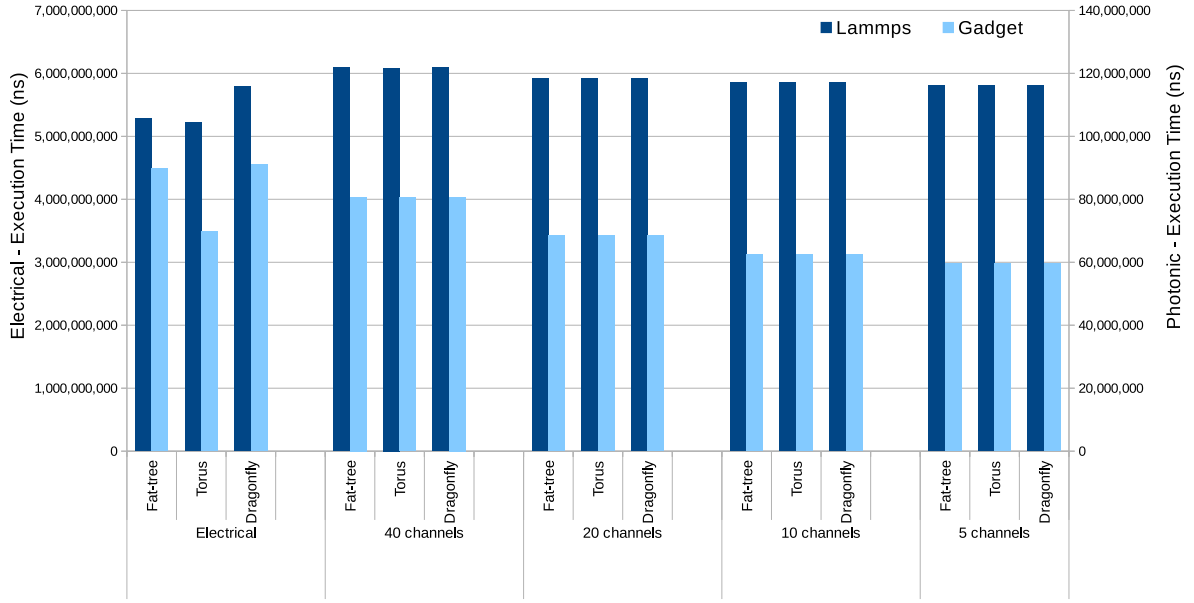


Figure 5.6: Execution time (in ns) for 10 Gigabit Ethernet (electrical) and photonic network configured using 5, 10, 20 and 40 channels with two ExaNeSt traces.

First, let us analyze the time required to execute both applications in the electrical network. The results, unlike synthetic traffic, show the impact of the topology on the execution time. We would like to remark the higher performance achieved by *Gadget* application when executed in torus topology due to the match between the communication pattern of the application and the underlying physical arrangement of nodes. When we compare these results with those obtained using photonic configurations, we can see that, as it happens with synthetic traffic, the network topology does not impact on performance. However, if we compare both network technologies, we can see how the applications executed in photonic configurations deliver executions times around one order of magnitude lower.

Regarding the photonic configurations, we can see that the less the number of photonic channels the higher the performance, especially for Gadget. These results corroborate the findings achieved with synthetic traffic, clearly showing that the assigned bandwidth per channel is key to take advantage in the photonic interconnect.

5.2.3 Final remarks

This section summarizes the finding using both synthetic and applications traffic. The results with photonic interconnect have shown that the most important performance factor is the number of channels used to split the photonic link, and the implications of assigning more bandwidth to channels make applications perform better. This conclusion reveals that current topologies, routing and channel reservation strategies are not able to take advantage of the high amount of resources (i.e. bandwidth or parallelization) provided by photonic technologies.

In photonic interconnects, due to the use of circuit switching, once the sequence of photonic channels has been reserved, the transmission is performed without any interference. For this reason, the main performance drawback occurs in the injection phase when there is no available channel to perform the transmission, situation that is less frequent with long packets.

Regarding the performance delivered by the electrical and photonic interconnects, it is clear that the new technology outperforms, around one order of magnitude in terms of execution time, to traditional electrical networks.

5.3 Summary

In this chapter we perform an exhaustive study of the behavior and performance of photonic networks with both synthetic and real-application traces and three well-known topologies: 3D torus, fat tree and dragonfly. These results have been also compared to a traditional electrical network.

Experimental results, obtained with synthetic traffic and excerpts of real applications from the ExaNeSt project, show that the optical network configuration has a great impact on the execution time of the applications, even when the same optical network technology is used (i.e. an aggregated bandwidth of 1.6 Tbps). In general, the parameter that impacts on performance the most is the bandwidth per channel, achieving the best results with 320 Gbps per channel (i.e. 5 channels). In addition, we also found that applications using long messages require up to

40% less time to deliver the same amount of data than using shorter messages. Regarding the performance achieved using electrical networks with real applications traffic, the use of photonic interconnects greatly reduces the execution time, one order of magnitude for the configurations evaluated in this work.

A surprising result is the lack of impact on the results of the network topology, which mainly rises due to the under-utilization of the photonic network capacity which leaves open a possible line of future work. Traditional network topologies are not designed to deal with the possibilities that offer the photonic technology. For this reason, specific topologies or new mechanisms should be designed to take advantage of the characteristics of photonics such as the high bandwidth and the lack of congestion.

Chapter 6

Study of Performance for Large-Scale Simulations

This chapter analyzes the results obtained from ExaNeSt workloads running on an exascale system implemented with photonic interconnects. Once studied the performance of photonic networks varying the photonic configuration, this chapter is focused on future exascale systems. We show how photonic technologies are able to reduce the complexity of existing state-of-the-art topologies, in particular a jellyfish topology.

6.1 Experimental Setup

To perform this study, we have implemented a jellyfish topology[65]. Jellyfish topology is a high-capacity network interconnect based on random graph connections, defined by three parameters (n, k, r). The former refers to the number of switches and the two latter to the switch configuration as follows:

- **n:** number of switches.
- **k:** switch radix.
- **r:** switch ports connected to the network.

Thus, we will use the pair (k, r) to refer to the switch configuration. Note that the number of simulated nodes is equal to the number of switches multiplied by the number of switch ports connected to nodes, as shown in Equation 6.1

$$N_{nodes} = (k - r) \times n \tag{6.1}$$

Experiments have been carried out assuming a photonic configuration with 40 wavelengths that provides an aggregated bandwidth of 1.6 Tbps per link. As mentioned in previous chapters, the link is split in multiple channels each one grouping a subset of the total wavelengths. For instance, in the 20-channel configuration, the link is split into 20 channels each one grouping 2 wavelengths, which gives 80 Gbps per channel.

In order to identify the design requirements of future exascale systems, among others, the platform developed within the ExaNeSt project, and several technical and scientific applications, have been used. These applications, explained in Chapter 3 belong to different fields such as Astrophysics, Neuroscience, Climatology, Material Science, etc. Despite we have gathered traces from the real executions of applications provided by ExaNeSt partners, these traces are constrained to a predetermined size in terms of number of processors and amount of communications. In other words, traces do not provide flexibility enough to evaluate exascale interconnection networks. For this reason, and taking into account the mentioned characterization, we have extracted the main communication patterns and developed kernels that mimic the behavior of real applications. These kernels allow us to explore different sizes of the application while essentially maintaining the behavior and the causality in the communications.

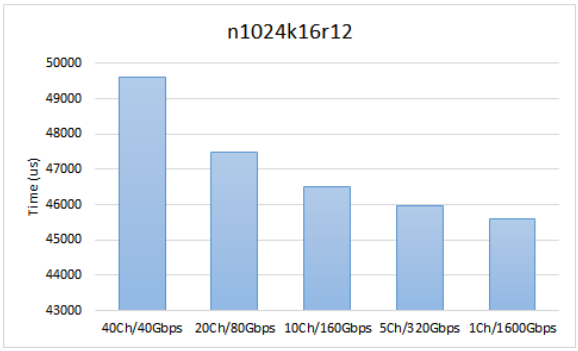
6.2 Experimental Results

For each application, we depict two kinds of graphs presenting execution times corresponding to one application. First, the photonic technology is evaluated varying the amount of photonics channels and bandwidth per channel. Different photonic configurations have been modeled and evaluated for comparison purposes. Second, the impact on performance varying the jellyfish topology for a given photonic configuration is explored. More precisely, we reduced the number of ports per switch connected to the network in order to study the impact of reducing the network connectivity, i.e., the network complexity is simplified by reducing the network cost. The main aim is to take advantage of the higher bandwidth and channel parallelism in a single photonic link.

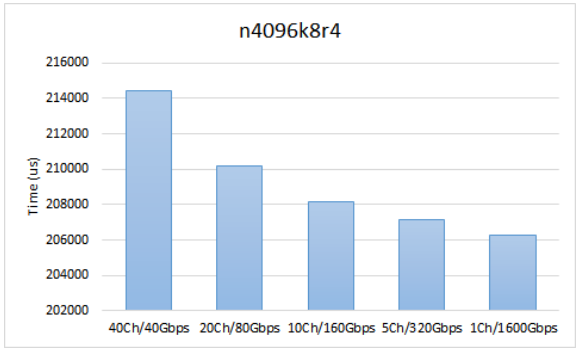
6.2.1 Gadget

Figure 6.1a shows the results of the Gadget application obtained with the jellyfish topology based on point-to-point messages, with 4,096 nodes and k16r12 switches. It can be observed that decreasing the amount of channels, and thus increasing the channel bandwidth (i.e. moving from left to right in the plot), from 40 channels at 40 Gbps to 20 channels at 80 Gbps, the execution time improves. Thus, we can conclude that the channel bandwidth is key for performance and the channel parallelism does not help improve the performance for this application. In fact, the highest performance is achieved with just one channel. However, it should be noticed that similar performance (e.g. by 1%) is achieved with 5 channels. We have also explored different nodes and switch configurations and found similar trends for 16,384 nodes with k8r4 routers, depicted in Figure 6.1b.

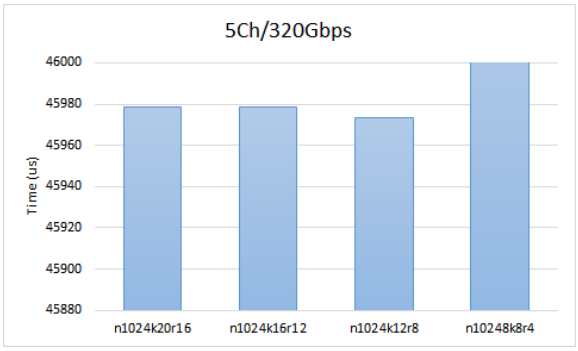
To carry out the second study with Gadget, we selected photonic configuration with 5 channels at 320 Gbps per channel to explore the best jellyfish configuration. Across the different applications, using low channel parallelism (e.g. 5 or 10 channels) seems to be the sweet spot in terms of performance. Therefore, although this application offers similar results with a single channel, 5 channels have been selected for a more realistic scenario. In this experiment, we study the impact on performance of reducing complexity by reducing the number of network links per switch (parameter `r`). The study concentrates on finding out which is the least complex jellyfish network that sustains the performance. Figure 6.1c shows the results for 4,096 nodes. As observed, reducing the number of network links per switch (parameter `r`) from 16 to 8 does not affect the execution time; however, a minimal impact rises when dropping to 4



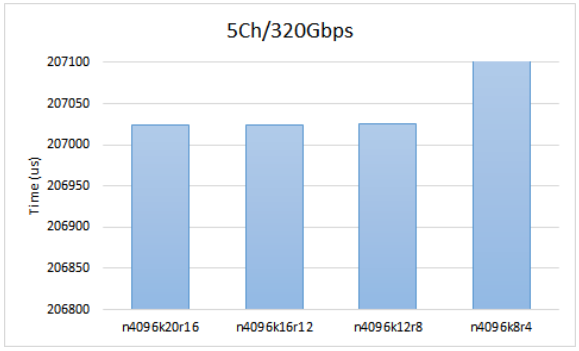
(a) Varying the number of channels from 1 to 40 for the k16r12 switch and 4,096 nodes



(b) Varying the number of channels from 1 to 40 for the k8r4 switch and 16,384 nodes



(c) Varying the switch configuration for 4,096 nodes



(d) Varying the switch configuration for 16,384 nodes

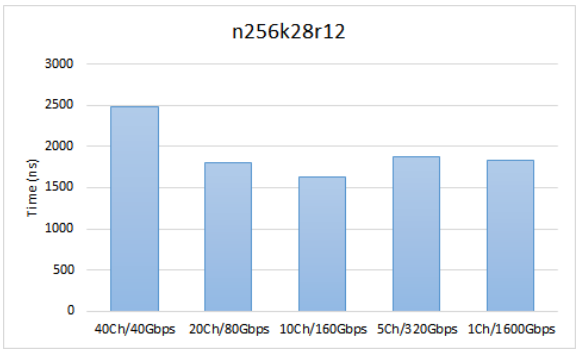
Figure 6.1: Execution time for Gadget.

links. The same trend is exhibited when increasing the number of nodes to 16,384 as shown in Figure 6.1d.

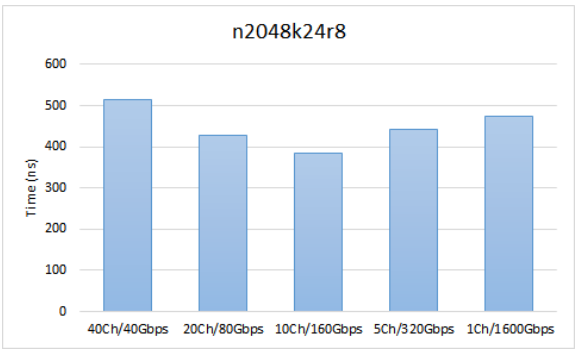
6.2.2 Lammps

Figure 6.2a shows the results of the Lammps application with 4,096 nodes and k28r12, and Figure 6.2b with 32,768 nodes and k24r8. As observed, both plots show the same trend. It can be appreciated that there is a trade-off between bandwidth per channel and channel level parallelism. Performance improves when the number of channels rises up to 10; however, adding more channels does not bring added performance; instead, performance drops mainly due to the fact that bandwidth per channel worsens. To study the impact on performance of reducing jellyfish network complexity, the number of photonics channels has been fixed to 10, since this photonic configuration provides the best performance for Lammps. Then, parameter `r` is reduced from 16 to 4. Figure 6.2c and 6.2d present the performance results of these experiments. As observed, the performance decreases as `r` is reduced down to 8. However, the impact on

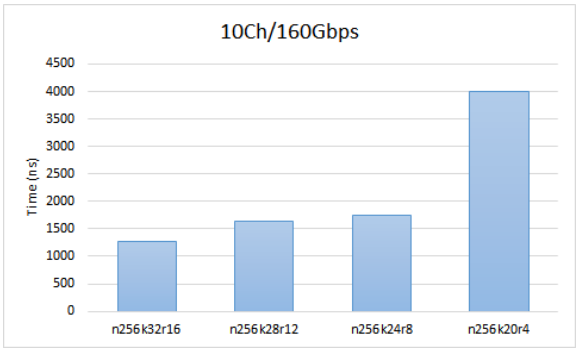
performance can be considered acceptable since the performance results keep close to those achieved in the previous experiment. Only when `r` is set to 4, the execution time grows by a factor greater than 2 ×.



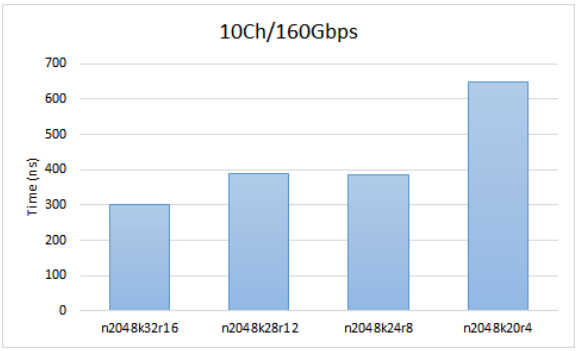
(a) *Varying the number of channels from 1 to 40 for the k28r12 switch and 4,096 nodes*



(b) *Varying the number of channels from 1 to 40 for the k24r8 switch and 32,768 nodes*



(c) *Varying the switch configuration for 4,096 nodes*

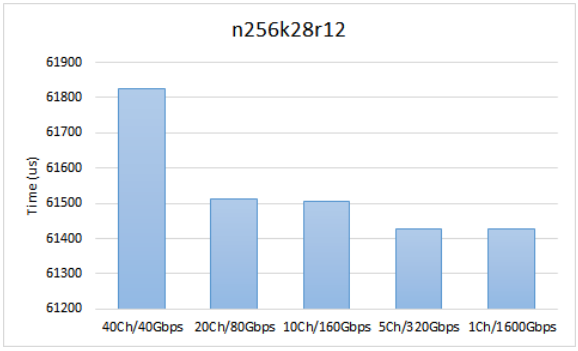


(d) *Varying the switch configuration for 32,768 nodes*

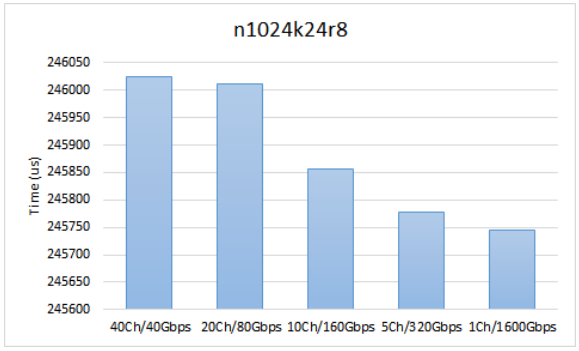
Figure 6.2: Execution time for Lammps.

6.2.3 RegCM

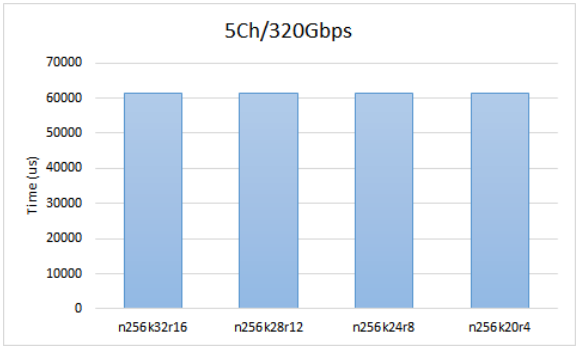
With respect to RegCM, we evaluate networks with 4,096 (Figure 6.3a) and 16,384 (Figure 6.3b) nodes using k28r12 and k24r8 jellyfish switches, respectively. As Gadget, RegCM performance grows with the channel bandwidth and it does not benefit from channel level parallelism. Nevertheless, the benefits are not significant when dropping the number of channels from 5 to 1. Thus, we assume 5-channel links when studying the effect of reducing the jellyfish network complexity (Figure 6.3c and 6.3d). In contrast to the previously studied applications, the performance of RegCM is not affected by the network configuration, enabling reducing the number of network ports per switch down to 4.



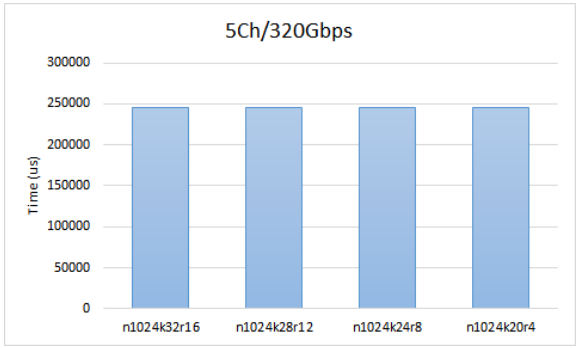
(a) *Varying the number of channels from 1 to 40 for the k28r12 switch and 4,096 nodes*



(b) *Varying the number of channels from 1 to 40 for the k24r8 switch and 16,384 nodes*



(c) *Varying the switch configuration for 4,096 nodes*



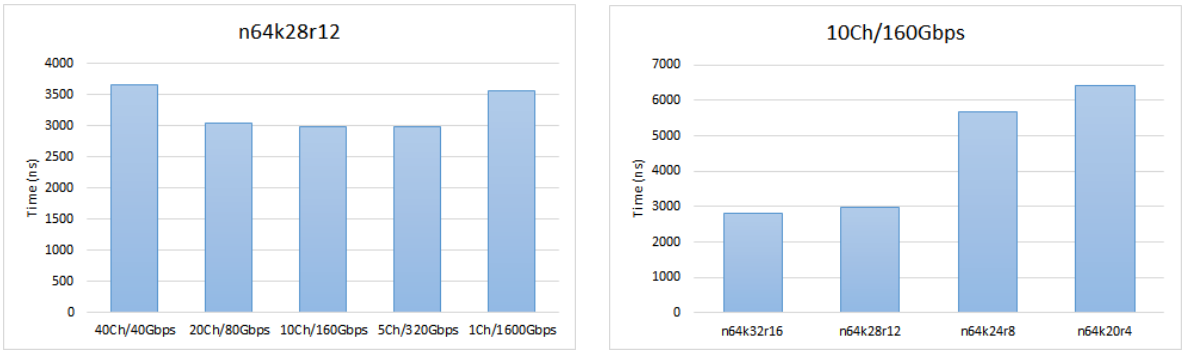
(d) *Varying the switch configuration for 16,384 nodes*

Figure 6.3: Execution time for RegCM.

6.2.4 DPSNN

In our experimental framework, DPSNN presents a high demand of simulation resources. Thus, we have simulated single workload with 1,024 nodes (connected with k28r12 switches). Note that although this number of nodes is smaller than the ones used in other applications, it is much higher than the amount of nodes that the traces provide us.

As Figure 6.4a shows, DPSNN execution time improves with channel parallelism. However, it does not get any significant benefit with more than 5 or 10 channels. In fact, the performance can heavily drop if the bandwidth per channel is severely reduced (e.g. 40Gbps). Given these results, we select a 10-channel photonic network to study the impact of reducing jellyfish network complexity (Figure 6.4b). The results show that when reducing the number of network links per switch from 16 to 12, the performance slightly decreases but this drop is not enough to cancel the benefits achieved by using multiple photonic channels. However, reducing the number of links below 12 (e.g. 8) presents a high impact on performance in this application.



(a) *Varying the number of channels from 1 to 40 for the k28r12 switch and 1,024 nodes*

(b) *Varying the switch configuration*

Figure 6.4: Execution time for DPSNN for 1,024 nodes.

6.3 Summary

In Chapter 5 we concluded that state-of-the-art photonics interconnects (with an aggregated bandwidth of 1.6 Tbps) achieve higher performance than electrical interconnects, even with a single photonic link splits in multiple channels. This chapter presents results obtained from ExaNeSt applications kernels developed from scratch based on the behavior of real applications in a scalable network simulator that incorporates the features of photonics technologies. This platform has been used to explore exascale photonics networks. For this purpose, we have explored multiple switch configurations in a jellyfish network topology, varying the number of photonics channels per link and bandwidth per channel. The study has analyzed a high number of nodes (reaching up to 32K in some cases).

From the analysis of the experimental results, two main conclusions can be drawn. First, almost all of the studied applications improve performance when using DWDM. These results confirm previous studies in this dissertation. This study confirms that employing a few channels per link (e.g., five or ten) provides the best results, even in large scale networks. In other words, expanding the number of channels beyond ten will improve the channel level parallelism but at expenses of dropping the channel bandwidth which can hurt the performance. Second, the complexity of the jellyfish topology can be largely reduced with photonics technologies, since the performance of the network is sustained even with a reduced number of network ports per switch. For a given application, if the amount of nodes connected to each switch is kept constant, then the number of network ports can be reduced, in general, from 16 down to 8 without a significant impact on performance.

Chapter 7

Segment Switching: A new Switching Strategy for Optical HPC Networks

Regarding the design of fully optical interconnection networks exposed in this dissertation, there are several key components that need to be addressed for exascale computing. Previous chapters have exposed that the parameters with higher impact on the network performance are the topology, the routing algorithm, and the switching mechanism.

In chapters 5 and 6 we studied how photonic technology affects to network performance. Results have revealed that network is underutilized because classical topologies and switching techniques are not taking advantage of photonic technology. For this reason, we propose *Segment Switching*, a switching strategy for optical circuit switching focused on reducing path length.

The chapter is organized as follow. First, we present the proposal and the different aspects that we have taken into account for the development. Then, we continue with the experimental setup of the performed experiments. Finally, we present an exhaustive evaluation of the network.

7.1 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 DWDM's huge potential on performance, resulting on underutilization of the network. DWDM 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 DWDM-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.

7.1.1 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 buffers (e.g. one buffer per input and output for each switch). Nevertheless, recent proposals of photonic switches [80, 81] integrate a small amount of buffers and can be leveraged by our approach.

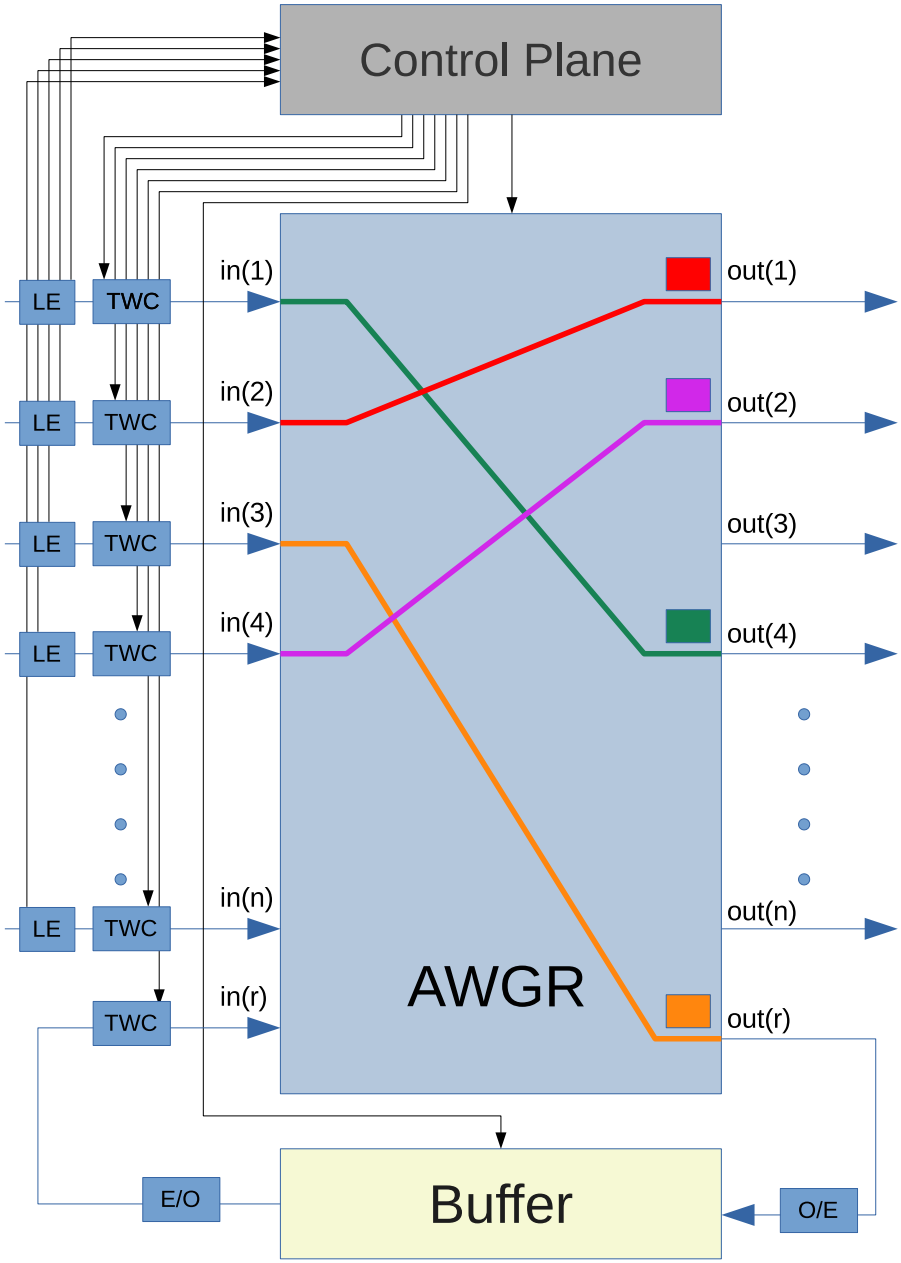


Figure 7.1: Photonic switch block diagram with n photonic inputs and outputs and a dedicated input and output to communicate with an associated buffer.

In particular, we leverage the *Lions* design [81], and, based on this design, we devise an optical switch especially tailored for our proposal. Figure 7.1 shows the block diagram of the devised optical switch. The central module is an Arrayed Waveguide Grating Router (AWGR) [60, 79], a passive device that interconnects inputs and outputs at photonic channel level. At the input ports, the Label Extractor (LE) and the Tunable Wavelength Converters (TWCs) [14, 76] 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) compared to the network cycle (nanoseconds) [75, 3], thus we use these buffers due to the storage capacity they provide.

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.

7.1.2 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 7.3.1, 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.

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
```

7.1.3 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.

Although potentially any switch may include a buffer, in this proposed scheme, 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 8x8x8 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 7.2 plots an example of segment reservation and transmission. As in Figure 4.2, 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. 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

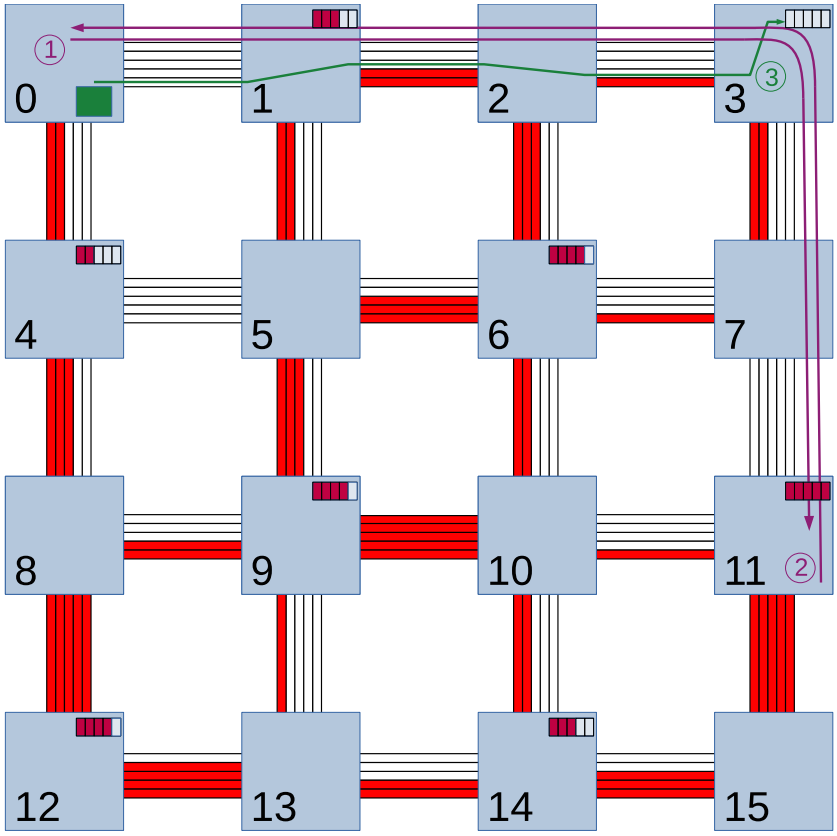


Figure 7.2: Example of a segment reservation and transmission in a 2D-torus network with a buffer every 2 nodes in each dimension.

the transmission ③. After that, a new reservation and transmission must be made from source node 11 to destination node 15.

7.2 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 [52], 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 modeling 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 12x12x12 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 7.1 are taken into account according to the literature. In particular, we have considered the time that LEs require to route the messages [9, 7, 61], the switching time of TWCs [82], and the delays of opto-electrical (O/E) and electro-optical (E/O) converters [75, 3]. 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 [45, 4]. Moreover, these times can be much shorter according to novel research regarding different materials such as graphene, reaching the order of femptoseconds [54]. Finally, circuit reservation time is also taken into account in the experimental setup. In particular, for each circuit established, reservation is assumed to take a number of network cycles that equals to the number of nodes traversed by the reservation message (both forward and backward).

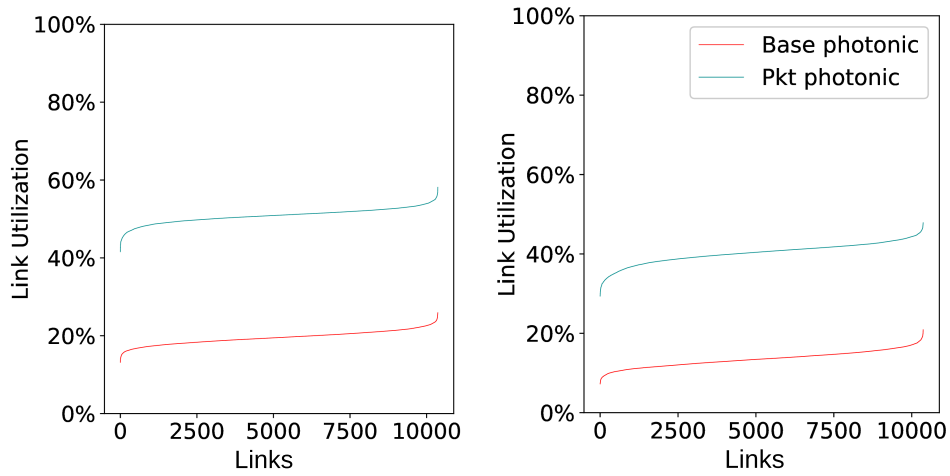
7.3 Experimental Results

This section evaluates the mechanisms proposed in Section 7.1. 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.

7.3.1 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 7.3 shows the link utilization, arranged in increasing order, in the studied network configurations.



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

Figure 7.3: Link 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 7.1:

$$U_{link} = \frac{T_{used}}{T_{total}} \tag{7.1}$$

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.

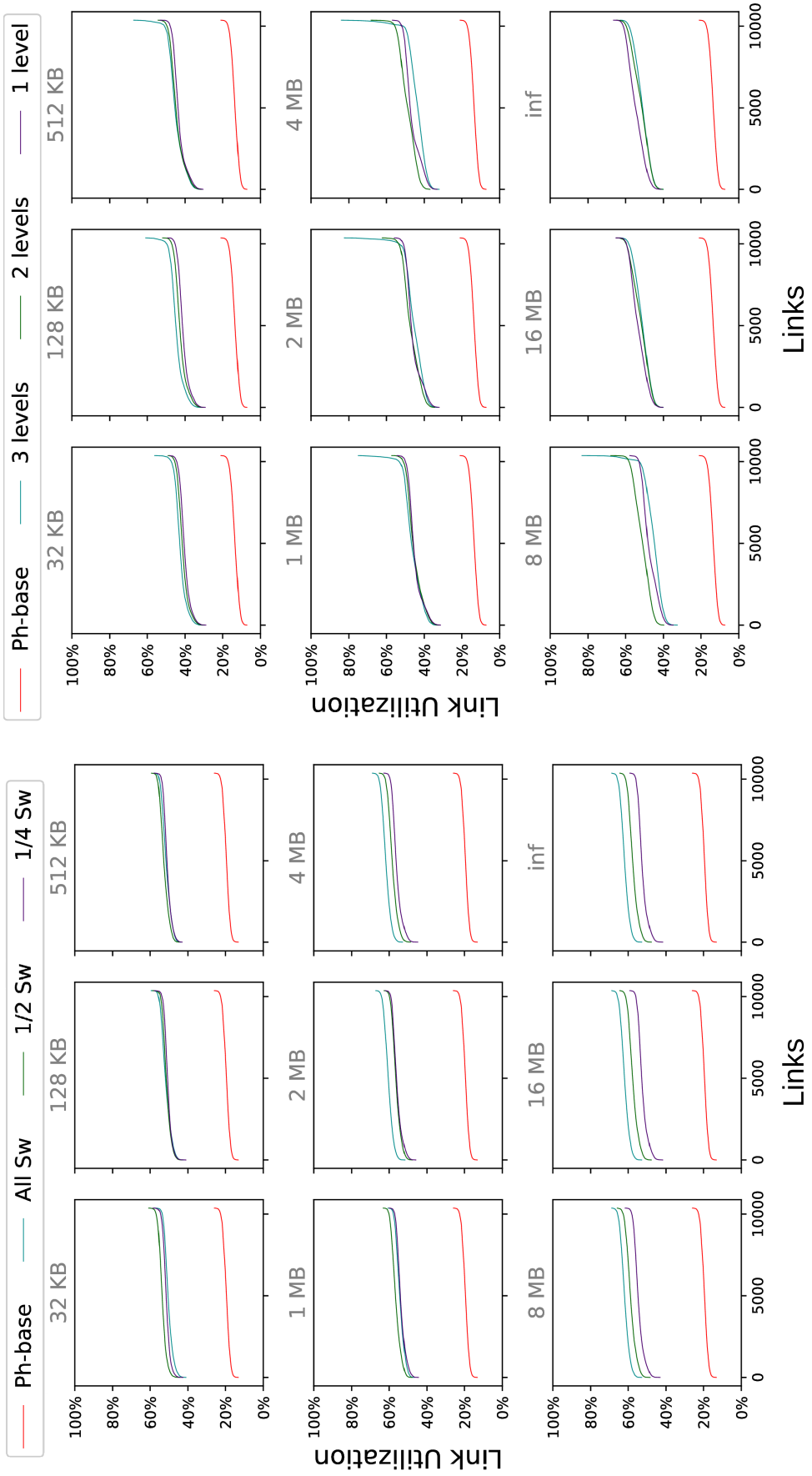
7.3.2 Impact of Buffers on the Network Performance

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 7.4 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 7.1.1. Moreover, different buffer sizes (from 32KB to unlimited) have been evaluated. In the torus topology, buffers are deployed in every switch (labeled 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 7.4a, 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 (labeled as 1 level in Figure 7.4b), in the two upper levels (2 levels) and in all the 3 levels of the fat tree. Again, buffering provides a 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 7.3.3.



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

Figure 7.4: Link utilization when buffers are included in (a) the torus topology and (b) in the fat tree.

Buffer Utilization

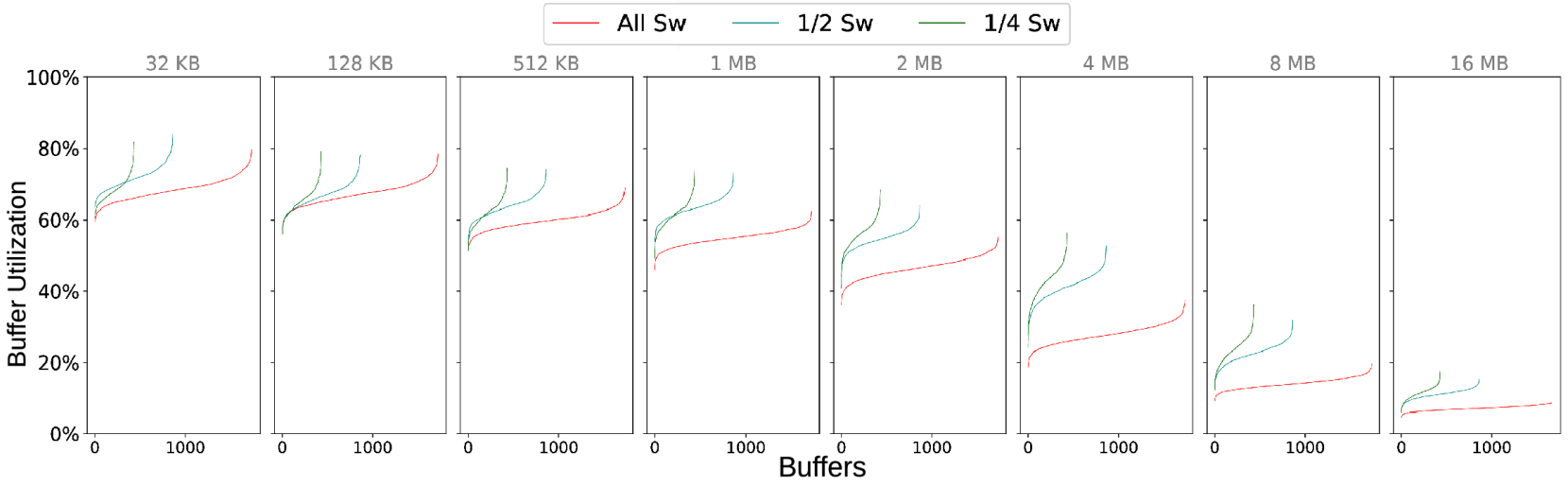
So far we have studied the impact of the number of buffers on link utilization, this section analyzes the utilization of the buffers. Figure 7.5 presents the results for the studied designs, which has been calculated with Equation 7.2. Like in previous study, each point of the line corresponds to one buffer. Notice that the length of the 3 lines of the same plot differ, this happens because the number of buffers also do that.

$$U_{buffer} = \frac{\sum_{1}^{n_{cycles}} \frac{Slots_{occupied}}{Slots_{total}}}{n_{cycles}} \tag{7.2}$$

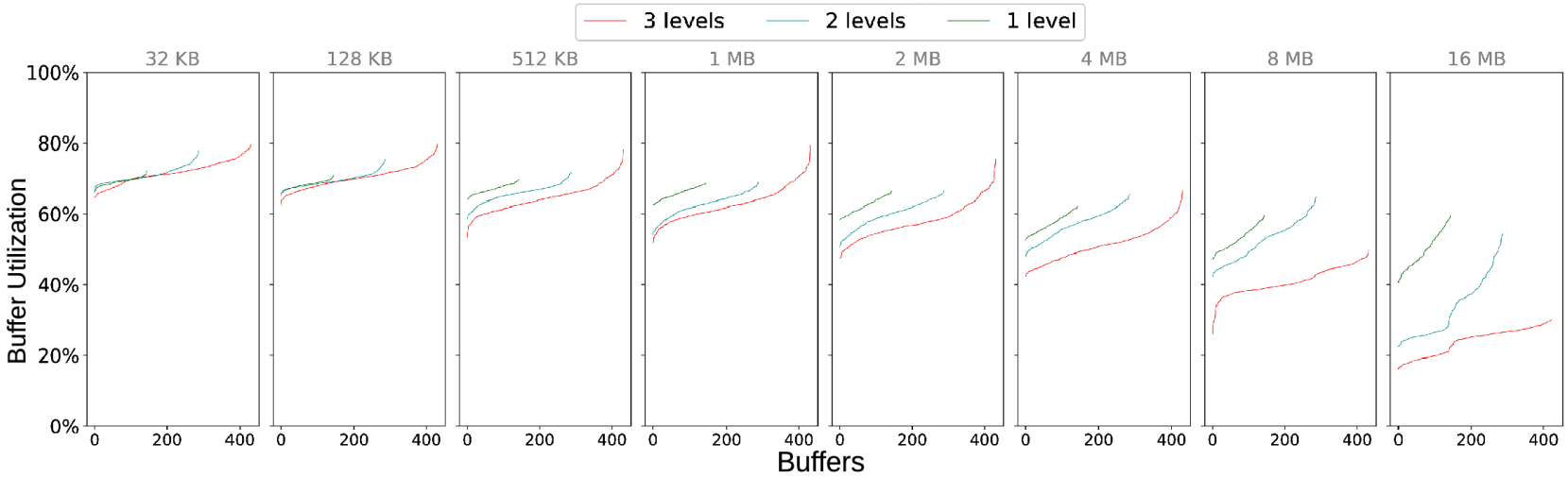
In the case of the torus topology, it can be observed in Figure 7.5a that the larger the buffers the lower the utilization. The utilization starts by 70% in the smallest 32KB buffer and goes down to around 10% in largest 16 MB buffer.

This means that large buffers are underutilized and may suggest that they are a waste of resources. This will be corroborated in Section 7.3.3 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 7.5b 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.



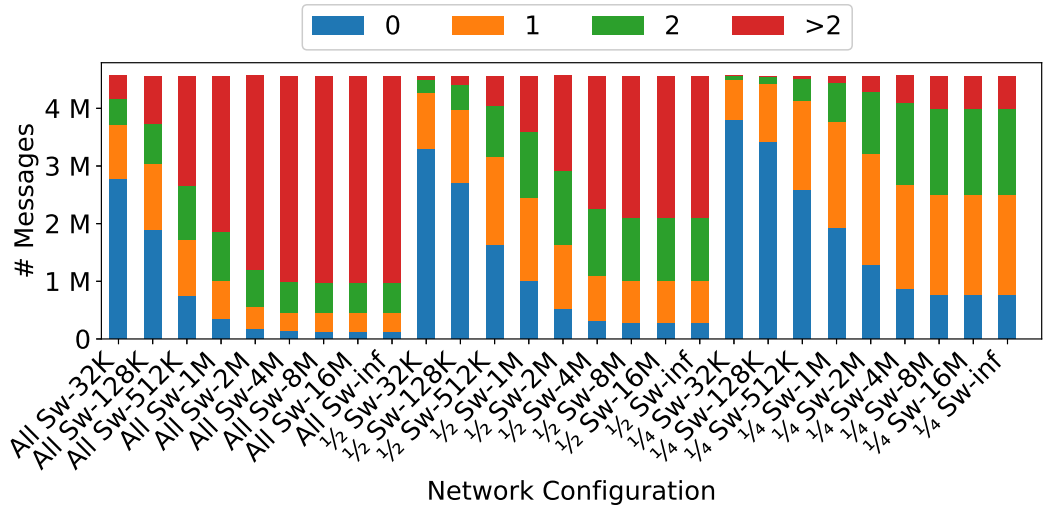
(a) *Torus*



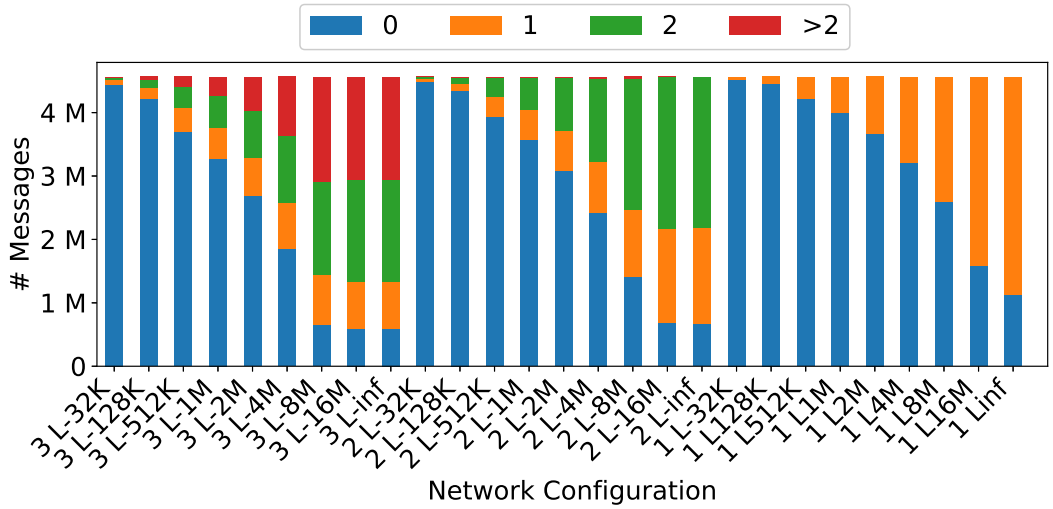
(b) *Fat tree*

Figure 7.5: Buffer Utilization in (a) the torus topology and (b) in the fat tree topology.

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 7.6a for each studied network configuration. In the torus topology, the term N Sw-S refers 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.



(a) *Torus*



(b) *Fat tree*

Figure 7.6: Re-stored Messages. Number of times that messages are stored over its path from source to destination.

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 7.6a, 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.

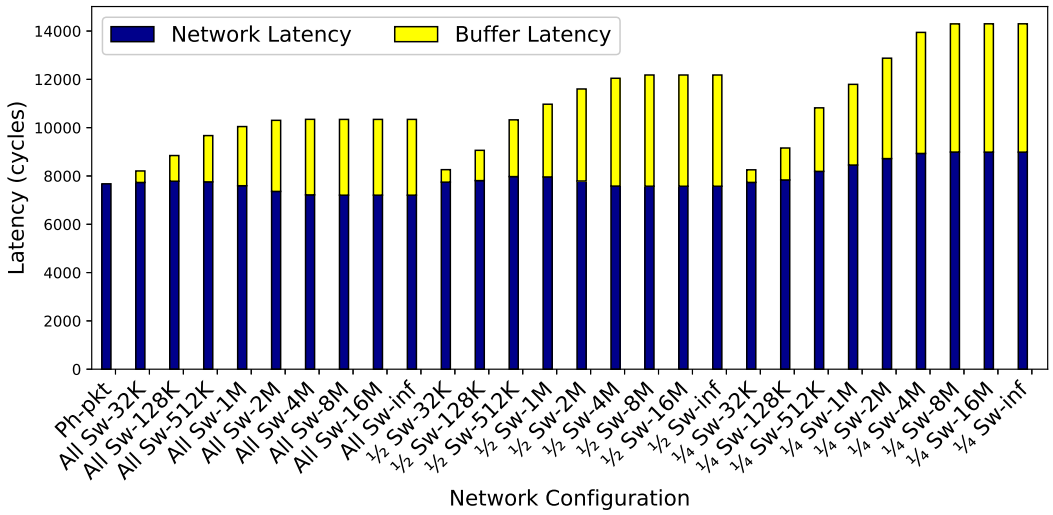
The number of times packets are stored is related to the length of the paths. As more times packets are stored, shorter paths are used. Figure 7.6a is also related to Figure 7.5a, as more times packets are stored, the buffers are more used. If we compare on the basis of the same buffering capacity, for 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 7.6b 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.

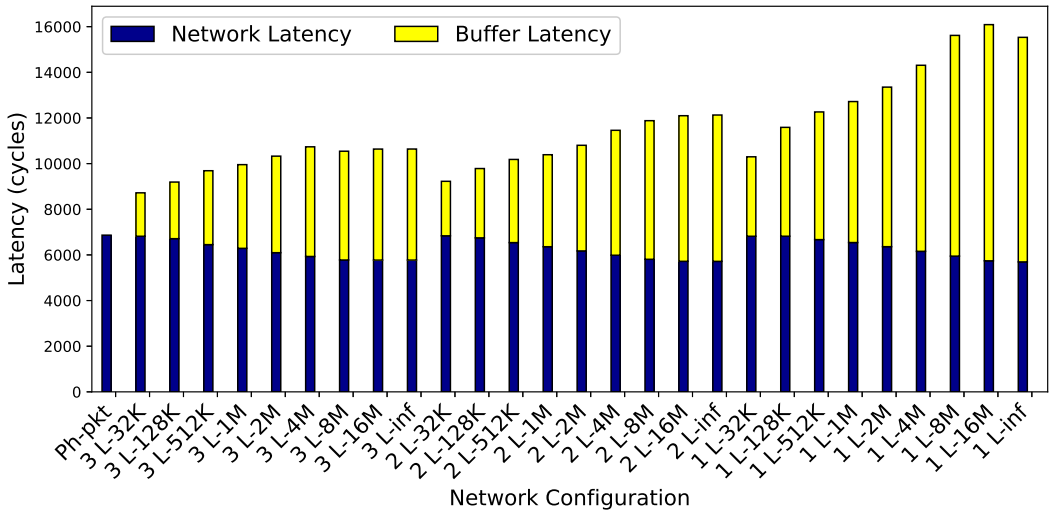
Packet Latency

In addition to utilization, a key factor to understand the behavior of the proposal is the impact on packet latency and the latency distribution. Figure 7.7 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 (excluding the time spent in intermediate buffers). As expected, 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 7.3.2, 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.



(a) *Torus*



(b) *Fat tree*

Figure 7.7: Packet latency separated in network and buffer latency.

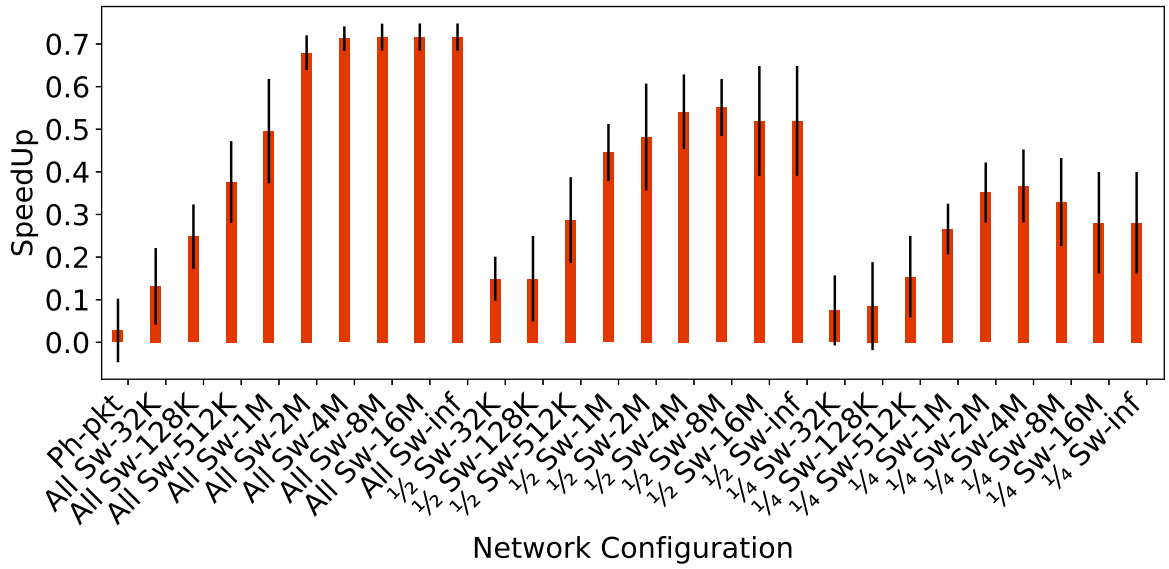
7.3.3 Network Performance

Finally, this section analyzes the impact of Segment Switching on network performance. As mentioned in Section 7.3.2, an increase in link utilization should translate into network performance improvements since link utilization is closely related to network throughput. Additionally, increasing the network throughput, shorten the time needed to transmit data over the network.

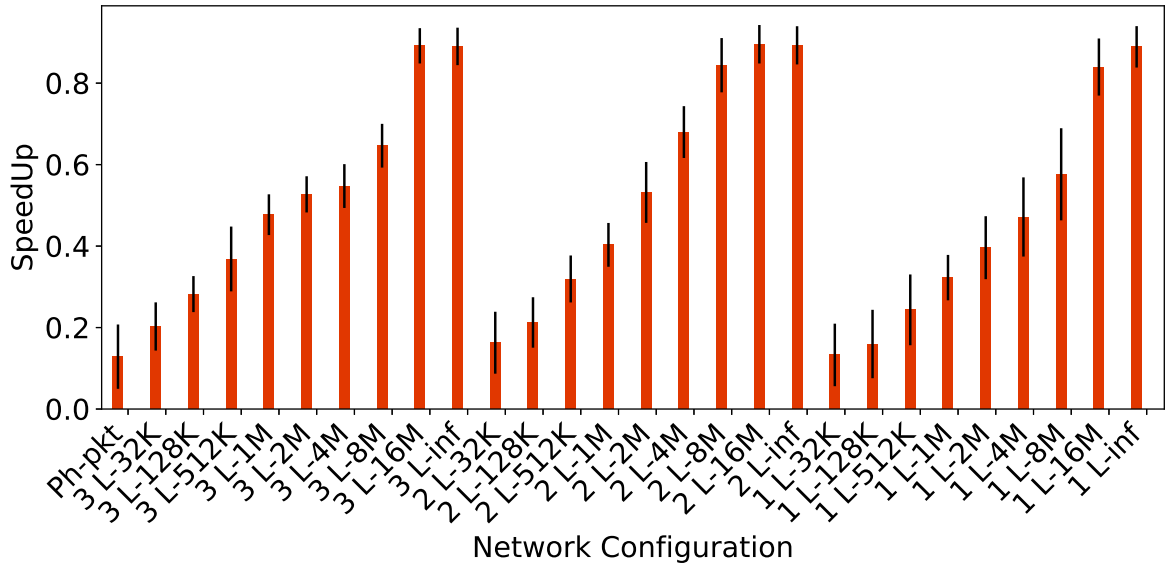
Figure 7.8 shows the average speedup over the photonic baseline network achieved by the proposed network mechanism with synthetic traces (see Section 5.1). Results are plotted with a 95% confidence interval through 20 executions of each trace with different seeds. The first bar corresponds to the speedup 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 7.8a, 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 7.4a). 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 7.8b. 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 7.4b),



(a) *Torus*



(b) *Fat tree*

Figure 7.8: Speedup of the network with buffers for (a) torus and (b) fat tree over the Photonic Baseline Network.

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.

7.4 Summary

In this chapter 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.

Chapter 8

Conclusions

This thesis has addressed a viability study of introducing photonic technology in the design of exascale networks in order to help achieve the challenge of exascale computing. We analyze the network requirements of representative workloads. We developed a simulation framework to model photonic networks. Using this framework we performed some studies to determine the best network configuration to leverage the photonic technology in networks with a high number of nodes. Finally, we improve the network by proposing new routing strategies that take advantage of photonic technology.

In this chapter we summarize the main contributions made in this dissertation based on the initially proposed objectives, the plans for future work, and an enumeration of the scientific publications made from this work.

8.1 Main Contributions

Important research in the HPC area, and more precisely in the context of supercomputers, is focused on reaching exascale computing. To accomplish this objective some challenges need to be solved in the whole system regarding computational nodes, storage systems or the interconnection network, among others. The European project ExaNeSt was focused in the development of a system for these purposes. Jointly with other partners, we have worked on the development of the interconnection network. More precisely, in cooperation with the Nanophotonics Technology Center (NTC) of the UPV we have focused on considering the photonic technology in the top-of-rack network level. While NTC has focused on the physical features we have focused on the network design.

In Chapter 3 we presented the first step in the research for the use of photonic technology to achieve a network able to meet the challenge that exascale computing presents, presenting a characterization study of the workloads provided by ExaNeSt partners to evaluate the devised networks. The characterization study has focused on analyzing three traffic parameters: the distribution of messages, the dynamic bandwidth consumption, and the spatial communication patterns among cores. From the point of view of message distribution, point-to-point communication messages are predominant in most applications, in general with message sizes below 50KB. The analysis of the consumed bandwidth presents a wide range of average bandwidth requirements, however, this consumption appears in bursty communications in most of the applications. Finally, a great variety of spatial communication patterns was shown from hot spot to spread communications.

Regarding the simulation tools, Chapter 4 identified and explained the main features to implement when adapting or developing tools implementing photonic technology. In particular, we focused on the INSEE Simulation Framework, a simulator developed originally to model electrical networks. In this chapter we raised the different network components of photonic interconnects and current electrical ones, how this should be considered when extending the simulator, and we provided different solutions to these issues. The main aspect that has been addressed is the implementation of the DWDM in the link structure that affects the entire simulation framework. This mechanism allows multiple messages in the same link to be transmitted at the same time. We also studied how this mechanism could be adapted to work on existing topologies and on the circuit switching technique, which is the one usually used in this technology. We also extended the photonic simulation framework to provide support for setting different configuration parameters such as the network aggregate bandwidth, the num-

ber of channels per optical link (that set the bandwidth per channel) or the phit size (defining the amount of information sent by time unit). Experiments varying the photonic interconnect configuration showed the importance of choosing the best setup regarding HPC applications.

Motivated by the results obtained in Chapter 4, in Chapter 5 and 6 we analyzed the behavior and performance of photonic networks. In Chapter 5, the analysis considered both synthetic traffic an ExaNeSt traces in three well-known topologies: 3D torus, fat tree and dragonfly. Results showed that the optical network configuration has a great impact on the execution time of the applications, even with the same aggregated bandwidth. In general, the parameter that most affects the network performance is the bandwidth per channel, achieving the best results the 5-channel per optical link configuration with a bandwidth of 320 Gbps per channel.

In Chapter 6 we analyzed photonic interconnects behavior in large-scale simulations using the ExaNeSt applications in a jellyfish topology. Considering the configuration options that the jellyfish topology brings, this study has been devised employing different topology settings per application. Also, even though we had traces extracted from real executions by ExaNeSt partners, there are constrained to the network size they were gathered. For this reason, and using the characterization study performed in Chapter 3, we extracted the communication and computation patterns to develop scalable kernels that mimic the real application behavior. Two main conclusions were extracted from this study. On the one hand, the confirmation that using a few channels per optical link (e.g. five or ten) provides the best performance results despite giving up network parallelism. On the other hand, the possibility of reducing the complexity of jellyfish topology, either by diminishing the number of links or by reducing the switch ports (e.g. the network cost), without a significant impact on performance.

Studies in Chapter 5 and 6 revealed that, despite the great bandwidth improvement that photonic networks present against electrical networks, photonic links are underused because classical topologies and switching techniques do not take advantage of photonic technology. In Chapter 7 we proposed Segment Switching, a switching strategy for optical circuit switching focused on reducing path length by packetizing messages and placing intermediate buffers. Packetization splits the message in packets and reduce the amount of information (i.e. the packed) that each time is sent; thus, packetization reduces the time messages block the network resources, and allows other nodes to inject earlier. The application of this mechanism improved photonic network utilization by 30% in the studied torus and fat tree topologies. This improvement results in a performance rise by 5% in the torus and 10% in the fat tree. The objective of buffering was to reduce the time that network resources are blocked by shortening

the length of the circuit established to send a message. The analysis was carried out by studying different layouts for allocating buffers to switches, varying the number of buffers and the buffer size. Experimental results showed a different buffer demand for each topology. Regarding torus, performance improves when buffering is supported by more switches, achieving a speedup of 70% with 4MB buffers in all the nodes. Regarding fat tree, the key parameter that most affects network performance was the buffer size, being enough the deployment of buffers in the upper stage of the topology. The best speedup was achieved with 16MB buffers in all stages, 90%, however, an 85% was reached with the same buffer size only in the upper stage.

8.2 Future Directions

This dissertation has explored the use of optical interconnects as an alternative and interesting approach for future exascale systems. We have made some proposals that can take advantage of the photonic technology, that feature high bandwidth capability and low power consumption. Nevertheless, we identify some issues that could be explored as future work. In particular:

- The design of network topologies from scratch that completely fit photonic technology features.
- The exploration of new routing algorithm that completely makes the most of photonic possibilities in the traditional topologies.
- Exploring the power consumption of the different proposals and analyzing the performance/power consumption trade-off.
- Design of fault tolerant photonic based networks.
- Design of photonic based redundant networks to reduce the critical path length and accelerate all-to-all communications.

8.3 Publications

The work carried out in this dissertation has resulted in publications in domestic and international conferences, as well as international journals. This section summarizes these works classifying them in three categories.

8.3.1 International Conferences

Title: Modeling a Photonic Network for Exascale Computing

Authors: José Duro, Salvador Petit, Julio Sahuquillo and María E. Gómez

Conference: International Conference on High Performance Computing & Simulation (HPCS 2017)

Location: Genoa, Italy

Year: 2017

ISBN: 10.1109/HPCS.2017.82

Title: Workload Characterization for Exascale Computing Networks

Authors: José Duro, Salvador Petit, Julio Sahuquillo and María E. Gómez

Conference: International Conference on High Performance Computing & Simulation (HPCS 2018)

Location: Orléans, France

Year: 2018

DOI: 10.1109/HPCS.2018.00069

8.3.2 National Conferences

Title: Modelado de una Red Fotónica para Computación Exascale

Authors: José Duro, Salvador Petit, Julio Sahuquillo and María E. Gómez

Conference: XXVIII Jornadas de Paralelismo Jornadas SARTECO 2017

Location: Malaga, Spain

Year: 2017

ISBN:978-84-697-4835-0

Title: Caracterización de Cargas para Redes de Computación Exascale

Authors: José Duro, Salvador Petit, Julio Sahuquillo and María E. Gómez

Conference: XXIX Jornadas de Paralelismo Jornadas SARTECO 2018

Location: Teruel, Spain

Year: 2018

ISBN:978-84-09-04334-7

8.3.3 Journals

Title: Modeling and Analysis of the Performance of Exascale Photonic Networks
Authors: José Duro, Jose A. Pascual, Salvador Petit, Julio Sahuquillo and María E. Gómez
Journal: Concurrency and Computation Practice and Experience (CCPE), volume 31, number 21, 2019.
DOI: 10.1002/cpe.4773

Title: Segment Switching: A new Switching Strategy for Optical HPC Networks
Authors: José Duro, Salvador Petit, María E. Gómez and Julio Sahuquillo
Journal: IEEE Access
Year: 2021 **Status:** Accepted

In addition to the publications mentioned above, in the realization of this dissertation together with the work done in the ExaNeSt project some technical documents and deliverables have been elaborated and favorably evaluated by a European Commission.

All publications listed in this section and technical reports mentioned in ExaNeSt project are exclusively related with this thesis. The Ph.D. candidate has contributed in the design, implementation and evaluation of the proposals, including the discussion of designs, the implementation of the experimental frameworks, the experimental execution and the analysis of the obtained results, besides the writing of the paper drafts for publications and documents and the presentation of the papers in conferences mentioned. Throughout the iterative processes, the co-authors have strongly supported the candidate, providing experienced advices and guiding him to improve the work and make it evolve into this dissertation.

Bibliography

[1] Yuichiro Ajima et al. "The tofu interconnect D". In: *2018 IEEE International Conference on Cluster Computing (CLUSTER)*. IEEE. 2018, pp. 646–654 (cit. on p. 14).

[2] Vivek Alwayn. *Optical network design and implementation*. Cisco Press, 2004 (cit. on p. 41).

[3] Shirish Bahirat and Sudeep Pasricha. "METEOR: Hybrid photonic ring-mesh network-on-chip for multicore architectures". In: *ACM Transactions on Embedded Computing Systems (TECS)* 13.3s (2014), pp. 1–33 (cit. on pp. 72, 76).

[4] Hitesh Ballani et al. "Sirius: A flat datacenter network with nanosecond optical switching". In: *Proceedings of the Annual conference of the ACM Special Interest Group on Data Communication on the applications, technologies, architectures, and protocols for computer communication*. 2020, pp. 782–797 (cit. on p. 76).

[5] Yaniv Ben-Itzhak et al. "Hnocs: Modular open-source simulator for heterogeneous nocs". In: *Embedded Computer Systems (SAMOS), 2012 International Conference on* (2012), pp. 51–57 (cit. on p. 12).

[6] Keren Bergman et al. *Photonic network-on-chip design*. Springer (cit. on p. 6).

[7] C Bintjas et al. "All-optical packet address and payload separation". In: *IEEE Photonics Technology Letters* 14.12 (2002), pp. 1728–1730 (cit. on p. 76).

[8] Francesco Bonaccorso et al. “Graphene photonics and optoelectronics”. In: *Nature photonics* 4.9 (2010), p. 611 (cit. on p. 8).

[9] Nicola Calabretta and Harm Dorren. “All-optical label processing in optical packet switched networks”. In: *2010 Conference on Optical Fiber Communication (OFC/NFOEC), collocated National Fiber Optic Engineers Conference*. IEEE. 2010, pp. 1–3 (cit. on p. 76).

[10] Charidimos Chaintoutis, Adonis Bogris, and Dimitris Syvridis. “P-Torus: Torus-based Optical Packet Switching Architecture for intra-Data Centre Networks”. In: *2018 Photonics in Switching and Computing (PSC)*. IEEE. 2018, pp. 1–3 (cit. on p. 14).

[11] J. Chan et al. “Physical-Layer Modeling and System-Level Design of Chip-Scale Photonic Interconnection Networks”. In: *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems* 30.10 (2011), pp. 1507–1520. ISSN: 0278-0070. DOI: 10.1109/TCAD.2011.2157157 (cit. on p. 13).

[12] Johnnie Chan et al. “PhoenixSim: A Simulator for Physical-layer Analysis of Chip-scale Photonic Interconnection Networks”. In: *Proceedings of the Conference on Design, Automation and Test in Europe*. DATE ’10 (2010), pp. 691–696 (cit. on p. 12).

[13] Kai Chen et al. “OSA: An optical switching architecture for data center networks with unprecedented flexibility”. In: *IEEE/ACM Transactions on Networking* 22.2 (2013), pp. 498–511 (cit. on pp. 7, 13).

[14] KK Chow et al. “Polarization-insensitive widely tunable wavelength converter based on four-wave mixing in a dispersion-flattened nonlinear photonic crystal fiber”. In: *IEEE photonics technology letters* 17.3 (2005), pp. 624–626 (cit. on p. 72).

[15] Kostas Christodoulopoulos et al. “Performance evaluation of a hybrid optical/electrical interconnect”. In: *IEEE/OSA Journal of Optical Communications and Networking* 7.3 (2015), pp. 193–204 (cit. on pp. 7, 13).

[16] Caroline Concatto et al. “A CAM-Free Exascalable HPC Router for Low-Energy Communications”. In: (2018), pp. 99–111 (cit. on p. 5).

[17] Saïd Derradji et al. “The BXI interconnect architecture”. In: *High-Performance Interconnects (HOTI), 2015 IEEE 23rd Annual Symposium on*. IEEE. 2015, pp. 18–25 (cit. on p. 28).

[18] Po Dong et al. "224-Gb/s PDM-16-QAM Modulator and Receiver based on Silicon Photonic Integrated Circuits". In: *Optical Fiber Communication Conference/National Fiber Optic Engineers Conference 2013* (2013), PDP5C.6. DOI: 10.1364/OFC.2013.PDP5C.6 (cit. on p. 7).

[19] Po Dong et al. "50-Gb/s silicon quadrature phase-shift keying modulator". In: *Opt. Express* 20.19 (2012), pp. 21181–21186. DOI: 10.1364/OE.20.021181 (cit. on p. 7).

[20] Jonathan K Doylend and Andrew P Knights. "The evolution of silicon photonics as an enabling technology for optical interconnection". In: *Laser & Photonics Reviews* 6.4 (2012), pp. 504–525 (cit. on p. 7).

[21] G. .. H. Duan et al. "10 Gb/s integrated tunable hybrid III-V/Si laser and silicon Mach-Zehnder modulator". In: *2012 38th European Conference and Exhibition on Optical Communications* (2012), pp. 1–3. ISSN: 1550-381X (cit. on p. 7).

[22] G. H. Duan et al. "Integrated hybrid III–V/Si laser and transmitter". In: *2012 International Conference on Indium Phosphide and Related Materials* (2012), pp. 16–19. ISSN: 1092-8669. DOI: 10.1109/ICIPRM.2012.6403306 (cit. on p. 7).

[23] Guang-Hua Duan et al. "Hybrid III–V on Silicon Lasers for Photonic Integrated Circuits on Silicon". In: *IEEE Journal of selected topics in quantum electronics* 20.4 (2014), pp. 158–170 (cit. on p. 7).

[24] Guang-Hua Duan et al. "New advances on heterogeneous integration of III–V on silicon". In: *Journal of Lightwave Technology* 33.5 (2015), pp. 976–983 (cit. on p. 7).

[25] Yves Durand et al. "Euroserver: Energy efficient node for european micro-servers". In: *2014 17th Euromicro Conference on Digital System Design*. IEEE. 2014, pp. 206–213 (cit. on p. 5).

[26] *ECOSCALE Website*. 2021, Jan. URL: http://www.ecoscale.eu (cit. on p. 4).

[27] René-Jean Essiambre and Robert W Tkach. "Capacity trends and limits of optical communication networks". In: *Proceedings of the IEEE* 100.5 (2012), pp. 1035–1055 (cit. on p. 41).

[28] *EuroEXA Website*. 2021, Jan. URL: http://www.euroexa.eu (cit. on p. 4).

[29] *EuroServer Website*. 2021, Jan. URL: http://www.euroserver-project.eu (cit. on p. 5).

[30] *ExaNeSt Website*. 2021, Jan. URL: http://exanest.eu (cit. on p. 4).

[31] *ExaNoDe Website*. 2021, Jan. URL: https://exanode.eu (cit. on p. 4).

[32] Nathan Farrington et al. "Helios: a hybrid electrical/optical switch architecture for modular data centers". In: *Proceedings of the ACM SIGCOMM 2010 conference*. 2010, pp. 339–350 (cit. on pp. 7, 13).

[33] Nathan Farrington et al. "Hunting mice with microsecond circuit switches". In: *Proceedings of the 11th ACM Workshop on Hot Topics in Networks*. 2012, pp. 115–120 (cit. on p. 40).

[34] Edgar Gabriel et al. "Open MPI: Goals, concept, and design of a next generation MPI implementation". In: *European Parallel Virtual Machine/Message Passing Interface Users' Group Meeting*. Springer. 2004, pp. 97–104 (cit. on p. 17).

[35] Filippo Giorgi et al. "RegCM4: model description and preliminary tests over multiple CORDEX domains". In: *Climate Research* 52 (2012), pp. 7–29 (cit. on p. 27).

[36] Crispín Gomez et al. "How to reduce packet dropping in a bufferless NoC". In: *Concurrency and Computation: Practice and Experience* 23.1 (2011), pp. 86–99 (cit. on p. 14).

[37] *Green 500 Website*. 2021, Jan. URL: https://www.top500.org/lists/green500 (cit. on p. 3).

[38] William Gropp, Ewing Lusk, and Anthony Skjellum. *Using MPI: portable parallel programming with the message-passing interface*. Vol. 1. MIT press, 1999 (cit. on p. 19).

[39] *Horizon 2020 Website*. 2021, Jan. URL: https://ec.europa.eu/programmes/horizon2020/ (cit. on p. 4).

[40] Hemayet Hossain et al. "Gpnocsim-a general purpose simulator for network-on-chip". In: *Information and Communication Technology, 2007. ICICT'07. International Conference on* (2007), pp. 254–257 (cit. on p. 12).

[41] Lavina Jain et al. "NIRGAM: a simulator for NoC interconnect routing and application modeling". In: *Design, Automation and Test in Europe Conference* (2007), pp. 16–20 (cit. on p. 12).

[42] Andrew B Kahng et al. "ORION 2.0: a fast and accurate NoC power and area model for early-stage design space exploration". In: *Proceedings of the conference on Design, Automation and Test in Europe* (2009), pp. 423–428 (cit. on p. 12).

[43] M Katevenis et al. "The ExaNeSt Project: Interconnects, Storage, and Packaging for Exascale Systems". In: *Digital System Design (DSD), 2016 Euromicro Conference on*. IEEE. 2016, pp. 60–67 (cit. on pp. 4, 38).

[44] John Kim et al. "Technology-driven, highly-scalable dragonfly topology". In: *2008 International Symposium on Computer Architecture*. IEEE. 2008, pp. 77–88 (cit. on p. 45).

[45] Sophie Lange et al. "Sub-nanosecond optical switching using chip-based soliton microcombs". In: *Optical Fiber Communication Conference*. Optical Society of America. 2020, W2A–4 (cit. on p. 76).

[46] Odile Liboiron-Ladouceur et al. "The data vortex optical packet switched interconnection network". In: *Journal of Lightwave Technology* 26.13 (2008), pp. 1777–1789 (cit. on p. 14).

[47] Ansheng Liu et al. "High-speed optical modulation based on carrier depletion in a silicon waveguide". In: *Opt. Express* 15.2 (2007), pp. 660–668. DOI: 10.1364/OE.15.000660 (cit. on p. 7).

[48] Liangjun Lu et al. "16 x 16 Non-blocking silicon optical switch based on electro-optic Mach-Zehnder interferometers". In: *Optics Express* 24 (May 2016), p. 9295 (cit. on p. 39).

[49] X. Ma et al. "LioeSim: A Network Simulator for Hybrid Opto-Electronic Networks-on-Chip Analysis". In: *Journal of Lightwave Technology* 32.22 (2014), pp. 4301–4310. ISSN: 0733-8724. DOI: 10.1109/JLT.2014.2356515 (cit. on p. 12).

[50] *Modified INSEE*. URL: https://bitbucket.org/joseduro/photonic-insee (cit. on p. 53).

[51] *Nanophotonics Technology Center Website*. 2021, Jan. URL: https://ntc.webs.upv.es (cit. on p. 6).

[52] Javier Navaridas et al. "INRFlow: An interconnection networks research flow-level simulation framework". In: *Journal of parallel and distributed computing* 130 (2019), pp. 140–152 (cit. on pp. 15, 75).

[53] Javier Navaridas et al. “Simulating and evaluating interconnection networks with {INSEE}”. In: *Simulation Modelling Practice and Theory* 19.1 (2011). Modeling and Performance Analysis of Networking and Collaborative Systems, pp. 494 –515. ISSN: 1569-190X. DOI: http://doi.org/10.1016/j.simpat.2010.08.008 (cit. on p. 38).

[54] Masaaki Ono et al. “Ultrafast and energy-efficient all-optical switching with graphene-loaded deep-subwavelength plasmonic waveguides”. In: *Nature Photonics* 14.1 (2020), pp. 37–43 (cit. on pp. 8, 76).

[55] *Original INSEE*. URL: https://gitlab.com/ExaNeSt/insee (cit. on p. 53).

[56] Pier Stanislao Paolucci et al. “Distributed simulation of polychronous and plastic spiking neural networks: strong and weak scaling of a representative mini-application benchmark executed on a small-scale commodity cluster”. In: *arXiv preprint arXiv:1310.8478* (2013) (cit. on p. 25).

[57] Steve Plimpton. “Fast parallel algorithms for short-range molecular dynamics”. In: *Journal of computational physics* 117.1 (1995), pp. 1–19 (cit. on p. 23).

[58] Steve Plimpton, Paul Crozier, and Aidan Thompson. “LAMMPS-large-scale atomic/molecular massively parallel simulator”. In: *Sandia National Laboratories* 18 (2007), p. 43 (cit. on p. 23).

[59] George Porter et al. “Integrating microsecond circuit switching into the data center”. In: *ACM SIGCOMM Computer Communication Review* 43.4 (2013), pp. 447–458 (cit. on p. 40).

[60] Rajiv Ramaswami, Kumar Sivarajan, and Galen Sasaki. *Optical networks: a practical perspective*. Morgan Kaufmann, 2009 (cit. on p. 72).

[61] Francisco Ramos et al. “IST-LASAGNE: Towards all-optical label swapping employing optical logic gates and optical flip-flops”. In: *Journal of Lightwave Technology* 23.10 (2005), p. 2993 (cit. on p. 76).

[62] Fco. Javier Ridruejo Perez and José Miguel-Alonso. “INSEE: An Interconnection Network Simulation and Evaluation Environment”. In: *Proceedings of the 11th International Euro-Par Conference on Parallel Processing*. Euro-Par’05. Lisbon, Portugal: Springer-Verlag, 2005, pp. 1014–1023. ISBN: 3-540-28700-0, 978-3-540-28700-1 (cit. on pp. 15, 37, 38).

[63] Sébastien Rumley et al. "Phoenixsim: Crosslayer design and modeling of silicon photonic interconnects". In: *Proceedings of the 1st International Workshop on Advanced Interconnect Solutions and Technologies for Emerging Computing Systems* (2016), p. 7 (cit. on p. 12).

[64] *scalasca Website*. 2021, Jan. URL: `http://www.scalasca.org` (cit. on p. 28).

[65] Ankit Singla et al. "Jellyfish: Networking data centers randomly". In: *9th {USENIX} Symposium on Networked Systems Design and Implementation ({NSDI} 12)*. 2012, pp. 225–238 (cit. on p. 62).

[66] Ankit Singla et al. "Proteus: a topology malleable data center network". In: *Proceedings of the 9th ACM SIGCOMM Workshop on Hot Topics in Networks*. 2010, pp. 1–6 (cit. on p. 13).

[67] R. Soref and B. Bennett. "Electrooptical effects in silicon". In: *IEEE Journal of Quantum Electronics* 23.1 (1987), pp. 123–129. ISSN: 0018-9197. DOI: `10.1109/JQE.1987.1073206` (cit. on p. 7).

[68] Volker Springel. "The cosmological simulation code GADGET-2". In: *Monthly notices of the royal astronomical society* 364.4 (2005), pp. 1105–1134 (cit. on p. 21).

[69] Chen Sun et al. "DSENT-a tool connecting emerging photonics with electronics for opto-electronic networks-on-chip modeling". In: *Networks on Chip (NoCS), 2012 Sixth IEEE/ACM International Symposium on* (2012), pp. 201–210 (cit. on p. 12).

[70] E Temprana et al. "Overcoming Kerr-induced capacity limit in optical fiber transmission". In: *Science* 348.6242 (2015), pp. 1445–1448 (cit. on p. 41).

[71] D. J. Thomson et al. "High contrast 40Gbit/s optical modulation in silicon". In: *Opt. Express* 19.12 (2011), pp. 11507–11516. DOI: `10.1364/OE.19.011507` (cit. on p. 7).

[72] David Thomson et al. "Roadmap on silicon photonics". In: *Journal of Optics* 18.7 (2016), p. 073003 (cit. on p. 7).

[73] *Top500 Website*. 2020, Jun. URL: `http://www.top500.org/` (cit. on pp. 2, 45).

[74] András Varga and Rudolf Hornig. "An Overview of the OMNeT++ Simulation Environment". In: *Proceedings of the 1st International Conference on Simulation Tools and Techniques for Communications, Networks and Systems & Workshops*. Simutools '08 (2008), 60:1–60:10 (cit. on p. 12).

[75] Sebastian Werner, Javier Navaridas, and Mikel Luján. "Designing low-power, low-latency networks-on-chip by optimally combining electrical and optical links". In: *2017 IEEE International Symposium on High Performance Computer Architecture (HPCA)*. IEEE. 2017, pp. 265–276 (cit. on pp. 72, 76).

[76] Kang Xi, Yu-Hsiang Kao, and H Jonathan Chao. "A petabit bufferless optical switch for data center networks". In: Springer, 2013, pp. 135–154 (cit. on p. 72).

[77] Fulong Yan, Xuwei Xue, and Nicola Calabretta. "HiFOST: A scalable and low-latency hybrid data center network architecture based on flow-controlled fast optical switches". In: *IEEE/OSA Journal of Optical Communications and Networking* 10.7 (2018), pp. 1–14 (cit. on p. 14).

[78] Fulong Yan et al. "Opsquare: A flat DCN architecture based on flow-controlled optical packet switches". In: *IEEE/OSA Journal of Optical Communications and Networking* 9.4 (2017), pp. 291–303 (cit. on p. 14).

[79] Xiaohui Ye, SJB Yoo, and Venkatesh Akella. "AWGR-based optical topologies for scalable and efficient global communications in large-scale multi-processor systems". In: *Journal of Optical Communications and Networking* 4.9 (2012), pp. 651–662 (cit. on p. 72).

[80] Xiaohui Ye et al. "DOS: A scalable optical switch for datacenters". In: *Proceedings of the 6th ACM/IEEE Symposium on Architectures for Networking and Communications Systems*. 2010, pp. 1–12 (cit. on pp. 7, 13, 71).

[81] Yawei Yin et al. "LIONS: An AWGR-based low-latency optical switch for high-performance computing and data centers". In: *IEEE Journal of Selected Topics in Quantum Electronics* 19.2 (2012), pp. 3600409–3600409 (cit. on pp. 13, 71, 72).

[82] SJ Ben Yoo et al. "Rapidly switching all-optical packet routing system with optical-label swapping incorporating tunable wavelength conversion and a uniform-loss cyclic frequency AWGR". In: *IEEE Photonics Technology Letters* 14.8 (2002), pp. 1211–1213 (cit. on p. 76).

[83] Maxwell I Zimmerman et al. “Citizen Scientists Create an Exascale Computer to Combat COVID-19”. In: *BioRxiv* (2020) (cit. on p. 2).