Electro-optic Switches Based on Space Switching of Multiplexed WDM Signals: Blocking vs Non-blocking Design Trade-offs

Apostolos Siokis[a,c], Konstantinos Christodoulopoulos[b,c], Nikos Pleros[d], Emmanouel Varvarigos[b,c]
[a]Computer Engineering and Informatics Department, University of Patras, Greece,
[b]National Technical University of Athens, Greece
[c]Computer Technology Institute and Press – Diophantus, Patras, Greece
[d]Aristotle University of Thessaloniki, Greece

Abstract. Short distance optical interconnects are a promising solution for tackling the bandwidth and low energy consumption requirements of next generation Data Centers (DC) and High Performance Computing (HPC) systems. The realization of optical switching should offer scalability, allowing the interconnection of multiple racks and/or servers/compute nodes, and quick reconfiguration times. To this end, fast small-radix MicroRing Resonator (MRR)- and Mach-Zehnder Interferometer (MZI)-based space switching devices, capable of supporting multiple optical signals multiplexed through Wavelength Division Multiplexing (WDM) have been reported. Using such devices as building blocks we evaluate the performance of a number of simple electro-optic switch architectures based on successive wavelength selection, WDM multiplexing and space switching, attempting to achieve scalable switching fabrics with good throughput performance on average using little additional hardware and few switching stages, thus lower total insertion losses as well as lower power consumption. The price paid for such architectural simplicity is that it introduces additional constraints on the feasible permutation matrices of such switching fabrics, affecting performance for some traffic patterns. We discuss the trade-offs between performance and hardware requirements and based, on our findings, we propose alternative architectures that overcome these limitations.

Keywords: optical interconnects; optical switches; silicon photonics; WDM; space switches; scheduling

1. Introduction

The bandwidth requirements within Data Centers (DC) are increasing rapidly due to both processor evolution and the continuous growth of Big Data analytics and cloud-based applications: the annual global cloud traffic is predicted to quadruple by 2019 (from 2.1 to 8.6 ZettaBytes [1]). Power consumption poses another important issue as global DC power consumption is projected to rise to 1012 billion kWh in 2020 [2]. A similar trend is observed in the High Performance Computing (HPC) community where off-chip bandwidth of tens of Tb/s will be demanded to meet future HPC requirements [3].

Optics is a promising energy-efficient solution providing terabit transmission through Wavelength Division Multiplexing (WDM) that could satisfy the increasing bandwidth needs of DCs and HPCs. Optics has already found its way inside DCs and HPC systems and is expected to be deployed over ever shorter distances (board-to-board, on-board, and even on-chip) [4] in the near future, for both data transmission/reception and switching. A number of hybrid and all-optical architectures for rack-to-rack communication in DC and HPC systems can be found in [5] (and references cited there) where energy consumption benefits from the application of optics in such environments are also discussed. An important part of this trend is silicon photonics (Si-Pho), emerging as a powerful technology for optical connectivity in integrated circuit environments [6]. The realization of switching in the optical domain should offer scalability in the interconnection of multiple racks and/or on-board compute nodes (in HPC systems), while achieving fast reconfiguration times. In DCs in particular, the packet sizes cluster around 200 and 1400 Bytes (see discussion in [5]). They are either small control packets or parts of large files that are fragmented to the maximum packet size of the Ethernet networks (1500 Bytes). In order for a switch to operate at packet granularity, reconfiguration times of a few nanoseconds or less are best (a 200 Byte packet needs 40 ns to be transmitted assuming 40 Gb/s channels). Optical switching can be realized by configuring various devices, such as Micro-Electro-Mechanical Systems (MEMS), MicroRing Resonators (MRR), and Mach-Zehnder Interferometers (MZI). The typical limitation of the former is their slow reconfiguration times making them suitable only for slow optical circuit switching (OCS), even though faster MEMS-based switches have been recently reported [7]. Since fast switching times are required for DC and HPC, the development of MZI- and MRR-based fast Si-Pho switches is an area of intense research focus. In particular, 4×4, 8×8, 16×16 MZI and MRR space switches have been reported [8]-[13] based on multiple 2×2 MZI and MRR switching elements, respectively delivering reconfiguration times below 5 ns.

Architectures based on such photonic switches, combined with the WDM capability of optics in order to achieve high connectivity degrees are described in [14]. [15] presents a 8x8 switch architecture where 2 groups of 4 data signals are multiplexed using WDM in a single signal that is then spatially (in the space domain) switched using a 2×2 MZI switch. In PhoxTrot project [16, 17] a similar approach is followed, where 12 optical channels each carrying a rate of 40 Gb/s are multiplexed in a single signal that is then spatially switched using a 4×4 MZI switch (and then demultiplexed), leading to a 48×48 switch with near 2 Tbp/s maximum throughput. In what follows we will refer to this approach as the PhoxTrot switch. In [12] the scalability of MRR based switch fabrics for WDM signals is examined for DC application, showing that a 128×128 switch with 6 wavelengths per port is feasible, assuming however a large power budget (35dB).

In the present work, we examine a number of straightforward switch architectures, like the ones discussed above, where multiple optical inputs are multiplexed in a WDM signal, which is then switched in the space domain using a single (MZI- or MRR-based) chip, and is demultiplexed at the output in order to reach the desired destination. In this way, high-radix switches can be built with few optical switching stages, resulting in low total insertion losses, thus addressing the main scalability limitation of silicon photonic space switching elements. We discuss the advantages in hardware requirements of these approaches compared to other wavelength-space alternatives, such as the architectures proposed in [18, 19] based on Semiconductor Optical Amplifiers (SOA). We also discuss how these architectures can be expanded in order to interconnect multiple state-of-the-art electronic switches leading to larger electro-optic switching structures. These approaches offer scalability, achieving good throughput on average with little additional hardware and smaller optical paths in terms of basic switching elements. The architectures discussed have different hardware requirements and different functionality, in terms of the input-output permutations they can switch in a single step without contention. Their blocking and non-blocking characteristics are discussed based on the scheduling complexity they require and the maximum throughput they achieve. We also discuss the performance-hardware requirements, the functionality limitations and the blocking/non-blocking characteristics of them, the trade-offs involved and, based on our findings, we propose alternative architectures to overcome these limitations.

In Section 2, we define the class of switch architectures to be studied, viewing the PhoxTrot switch as a single case of this class, and we compare them to other configurations and architectures, in terms of hardware requirements, functionality and power consumption. In Section 3, we discuss the implications on scheduling that limit throughput and lead to speedup requirements. In Section 4, we examine the performance of such architectures (in terms of throughput and required speedup) for various traffic patterns. In Section 5, we discuss a variation of these architectures interconnecting multiple state-of-the-art electronic switches that can lead to larger electro-optic switching structures. In Section 6, we present alternative architectures without the aforementioned constraints, while in Section 7 we conclude this paper.

2. Electro-Optic Switch Architectures

In this section we first give some important, well known definitions regarding the blocking/non-blocking properties of a switching fabric (Section 2.1). We also discuss the buffer organization in switching fabrics and the way they relate to speedup (Section 2.2). Then we briefly describe the PhoxTrot switch architecture. We also define a class of similar switch architectures, viewing the PhoxTrot switch as an instance of this class (Section 2.3), investigate its merits and compare its hardware requirements to those of other configurations (Section 2.4). We also discuss options for this family of architectures regarding buffer organization in the inputs and scheduling decisions (Section 2.3.1).

2.1. Blocking and Non-blocking Switching Fabrics

The switching fabric is the heart of modern routers and switches. In what follows we will assume packets of fixed and equal length, called cells. A *switching pattern* is a particular set of connections between input and output ports of the switch. The following constraints must be satisfied for any switching fabric providing point-to-point connectivity:

Constraint C1) a single input is connected to at most one output

Constraint C2) at most one input is connected to a single output

If input i wants to connect to output $\pi(i)$, i=1,2,…, N, constraints C1 and C2 basically state that $\pi(\ )$ should be a *permutation function*. A switching pattern satisfying constraints C1 and C2 (i.e., a permutation input-output pattern) is a *blocking switching pattern* if the data cells cannot be transmitted on all connections simultaneously without collisions. A switch exhibiting blocking switching patterns is a *blocking switch*, while a switch that does not exhibit such patterns is a *non-blocking switch* [20]. The number of switching patterns that satisfy constraints C1 and C2 for switches of size N×N is $N!$, equal to the input-output permutations. We define the *functionality* of a switch as the number of different input-output permutations it can handle. A non-blocking switch has a functionality of N! A blocking switch has reduced functionality, but possibly requires lower cost and fewer components for its implementation. Finally, a common distinction for non-blocking switches, is between *Strictly Non-Blocking* (SNB) and *Rearrangeably Non-Blocking* (RNB) [20]. In the former, a connection can always be set up between any idle input and any idle output without disturbing connections already set up. In the latter when establishing a connection between an idle input and an idle output, internal paths of existing connections may have to be rearranged to set up that connection. SNB is desirable for circuit switching so as not to disturb existing circuits. For packet switching, RNB switches work equally well, assuming that an appropriate algorithm is executed in every step to ensure non-blocking switching configurations.

2.2. Buffer Organization in Switches and Speedup

A traditional distinction regarding the buffer placement and organization in switches is between *Input-Queued* (IQ), *Output-Queued* (OQ) and *Combined Input and Output Queued* (CIOQ) approaches [21, 22]. An important design parameter of switching fabrics is *speedup*. Speedup *S* is defined as the ratio of the switch bandwidth provided to the minimum switch bandwidth needed to support full throughput on all inputs and outputs [23]. An N×N switch with speedup of *S* can remove up to *S* cells from each input and transfer at most *S* cells to each output in a time slot. S>1 requires faster internal line rates (compared to the input and output port line rates), and memory with shortened access time and faster scheduling decisions. Furthermore, S>1 is required to perform output buffering, given that the line rates of the output ports are the same to those of the input ports. It is well known that OQ switches require speedup S=N to achieve full throughput. IQ switches can operate with S=1 applying what is known as Virtual Output Queues (VoQ) concept in the inputs [24] to avoid performance degradation due to Head-Of-Line (HOL) blocking [21]. For $1 < S < N$ a CIOQ approach is required. CIOQ switches can exactly emulate the behavior of an OQ switch using a speedup of 2-1/N [25].

2.3. WS–mSS Switch Architectures

We first describe briefly the architecture of a 48×48 electro-optic router chip currently pursued within the PhoxTrot project and comprising a 4x4 Si-Pho switching matrix equipped with wavelength MUX/DEMUX stages at every of its I/O ports. [16, 17]. The heart of the PhoxTrot router chip is a 4×4 Benes photonic switching fabric consisting of multiple cascaded 2×2 switching elements. This photonic switch performs Space Switching (SS), since it does not consider/distinguish based on the wavelength of the switched traffic. The 2×2 switching elements are Mach Zehnder Interferometer (MZI) switches that have been so far demonstrated in several switch matrix implementations as reliable and broadband switching modules, usually exploiting electro-optic-switching mechanism [8]-[13]. MZI-based Si-Pho switches have been shown also in higher radix arrangements co-integrated even with all necessary CMOS driving circuitry on the same chip [13]. Figure 1a depicts such a 4×4 non-blocking switching matrix consisting of 6 symmetric single-arm MZI-based switching elements arranged in a Benes topology, while many 4×4 switching matrices can be combined in larger switching topologies with a higher port count, towards implementing larger photonic n×n switching fabrics. The reconfiguration time of this 4×4 switch is 1.4 ns.

The overall router chip architecture that will incorporate the 4×4 SS, under study in the framework of PhoxTrot, will route a stream of 12 multiplexed signals (using WDM) per port of the 4×4 photonic non-blocking switch, leading to 4×12 input and 4×12 output multiplexed signals, creating a 48×48 switching element. The transmitter and receiver modules of the router chip will rely on flip-chip bonded Vertical Cavity Surface-Emitting Laser (VCSELs) [26] and Photodetectors (PDs), respectively, with every VCSEL of the 12-VCSELs array emitting at different wavelength from 1520nm to 1580nm and supporting multi-level modulation formats with a bit rate up to 40 Gb/s [27]. The 12 input channels are WDM multiplexed (MUX) through a combiner or an Array Waveguide Grating (AWG) and fed into the first input port of the 4×4 SS switch. They are demultiplexed (DEMUX) at the output with an AWG, and thus each input pin is being forwarded to the corresponding output pin. The router chip follows an IQ approach with VOQ organization of the input buffers. A CIOQ approach was not preferred, as it would require opto-electronic and electro-optic conversions at both ends. An OQ approach requires speedup S=4, something not practical due to the high line rates, unless the input and output channel rates were reduced to 10Gb/s. An illustration of the router chip architecture is depicted in Fig. 1a. In order to allow Wavelength Selection (WS) in the inputs, either a 12×12 electronic switch is required in the input and/or appropriate buffer organization (see Section 2.2.1). Since a Wavelength Selection (WS) stage (implemented in the electronic processing part) is followed by a Space Switching stage for the multiplexed WDM signal (mSS), we will refer to such an architecture as a WS-mSS switch.

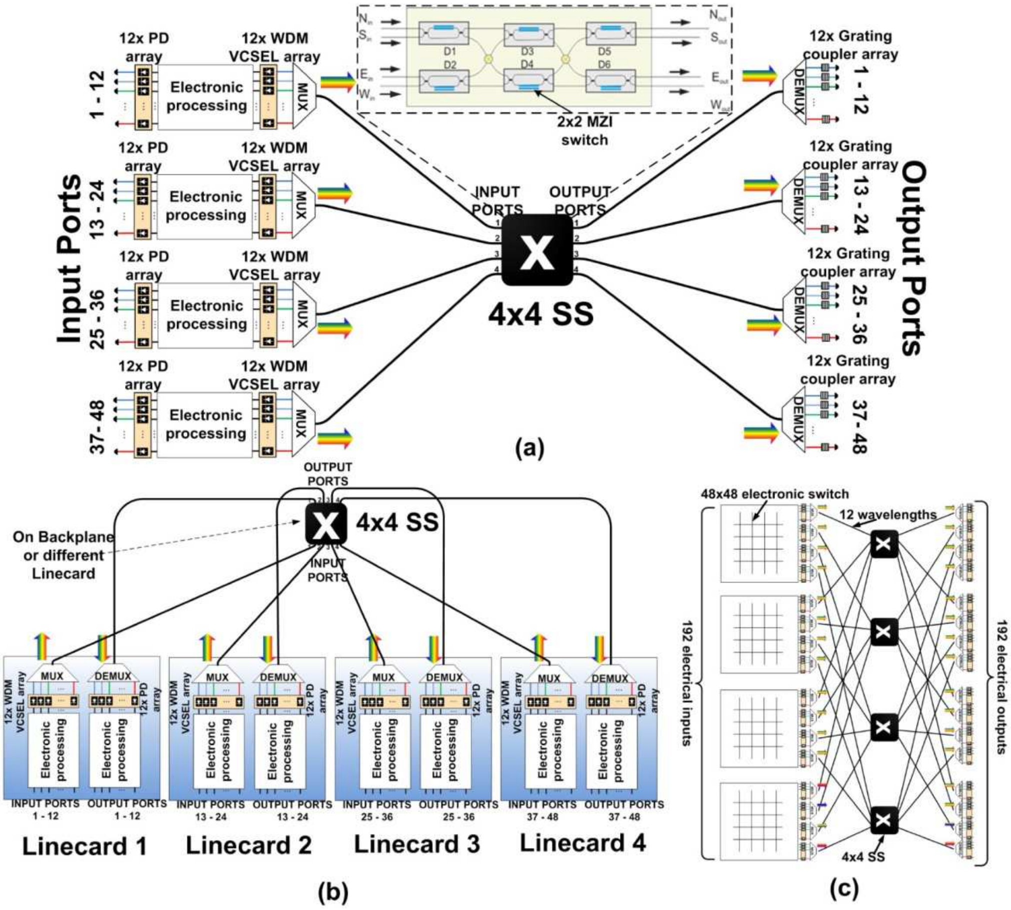

Fig. 1 a Single chip implementation of a 48×48 WS-mSS switch: 12 signals are multiplexed in each input of the 4×4 Space Switch (SS), and they are switched to the appropriate SS output port, where they are demultiplexed. **b** Alternative implementation of the 48×48 WS-mSS switch: 4 Linecards interconnected using the 4×4 optical switch on backplane or on a different card. **c** 192×192 switch constructed by using 4 48×48 electronic switches, 4 4×4 SS and 12 signals (de)multiplexed in a single (output) input.

An alternative implementation of the same WS-mSS architecture is depicted in Fig. 1b. A key difference of this version is that the inputs and outputs are electrical signals. Optics is used only internally in the switch. This allows buffering and electronic processing at the outputs as well, leading to a CIOQ approach, which can provide *S*>1. In Fig.1b this version is depicted using 4 linecards, corresponding to the 4 interconnected groups of inputs-outputs (wavelengths), and the 4×4 optical SS is located on the backplane or on a different linecard.

The aforementioned WS-mSS architectures can be generalized assuming a single n×n optical Space Switch (SS) and n m×m Wavelength Selection (WS) elements, where m is the number of wavelengths that can be multiplexed in a single optical signal, leading to N×N

swiching fabrics, with N=n·m. In this case, input (or output) port i of the N×N switching fabric belongs to group $j = \lceil i/m \rceil$ of the SS and to the input (output) pin $i-(j-1)\cdot m$ of the respective WS group. PhoxTrot case can be seen as a special case of this architecture with n=4, m=12 and N=48.

Ideally, EO and OE conversion would occur only in the inputs and outputs, respectively, of an all-optical DC or HPC network, using purely optical switching in-between. However, since large DC and HPC networks contain thousands of switches (as opposed to long-haul and metro networks where 10-100 nodes are found) and short-range optical transmitters are used, OEO conversion and/or optical signal amplification is unavoidable. The latter should be avoided as well due to power consumption: a 40 Gb/s VCSEL-based link has a power consumption of 22.3 pJ/bit leading to 0.892W [28], while an amplifier has a power consumption of 0.5-1 W depending on the type of amplifier ([2], p. 23, 24). Instead of using amplification or OEO conversion between WS-mSS chips, OEO conversion was incorporated in the chip architecture of Fig. 1a. Most "all-optical" DC and HPC architectures presented in the literature (see [5]) are based on high-radix all-optical switches used to replace the higher layers of the fat tree architectures in DC networks (aggregation, core), leading to fewer tiers and flatter architectures. Thus, up to the Top-of-Rack (ToR) layer is the electronic domain and the highest layer is the all-optical part of the fat tree, using the optical switches. A similar approach using WS-mSS is described below.

Multiple WS-mSS switching elements can be used in order to interconnect multiple state-of-the-art k×k electronic switches, with k=n·m, assuming n n×n SS and that m wavelengths can be multiplexed in a WDM signal, leading to an N×N switching element with N=k·n=m·n^2. In this way, in the first stage there are n k×k electronic switches. The output ports of the k×k switches are divided in n=k/m groups, where the m wavelengths of a single group are multiplexed in a single signal. In the second stage, n n×n SS elements are used. A single group $j \in [1,2,..,n]$ of multiplexed signals of a single kxk electronic switch $i \in [1,2,..,n]$ is connected to input port i of SS element j. In the third stage there are n groups of de-multiplexing elements. Output port $o \in [1,2,..,n]$ of SS $l \in [1,2,..,n]$ is connected to de-multiplexer l of group o. Thus, n k×k electronic switches and n n×n SS are required in total. In Figure 1c an example is presented for k=48, n=4, m=12 leading to a 192×192 swiching element requiring 4 48×48 electronic switches and 4 4×4 SS. For comparison, a 192×192 fat tree implementation based only on 48×48 electronic switches requires 12 electronic switches interconnected in a 2-layer fat tree. Using this concept electro-optic switches with extremely high radix can be realized, achieving port numbers far beyond what current exclusively electronic or exclusively photonic switches can offer.

2.3.1.Buffer organization at the inputs

In the following, we outline 2 architecture variations regarding the VOQ organization at the inputs, to avoid HOL effects, as well as switch scheduler/allocators arrangements.

The first version of the architecture requires additional n m×m electronic switches: a single m×m electronic switch (with its m VOQs per input) is present after the PDs and before the VCSELs in order to select the appropriate wavelength/VCSEL. In this version, after the PD, there are n VOQs for a single input, each corresponding to a different desired output port of the n×n SS element followed by the m VOQs for the m×m WS electronic switches (thus n+m VOQs/input port in total). This version lends itself to n+1 separate scheduling decisions: 1scheduler is used for the n×n SS and n for the respective m×m electronic switches (depicted in Fig. 2b for the PhoxTrot case). The SS scheduler will configure the n×n SS and will decide which groups of VOQs will be used to forward the cells to the m×m VOQs that follow. Note that in this case, switching using the WS-mSS consists of two separate switching stages: the SS must stay in a configuration for 2 cycles for scheduling incoming cells. In order for a cell to be able to be served in a single scheduling slot of the WS-mSS, a speedup value equal to *m* would be required for the *m*×*m* switching elements.

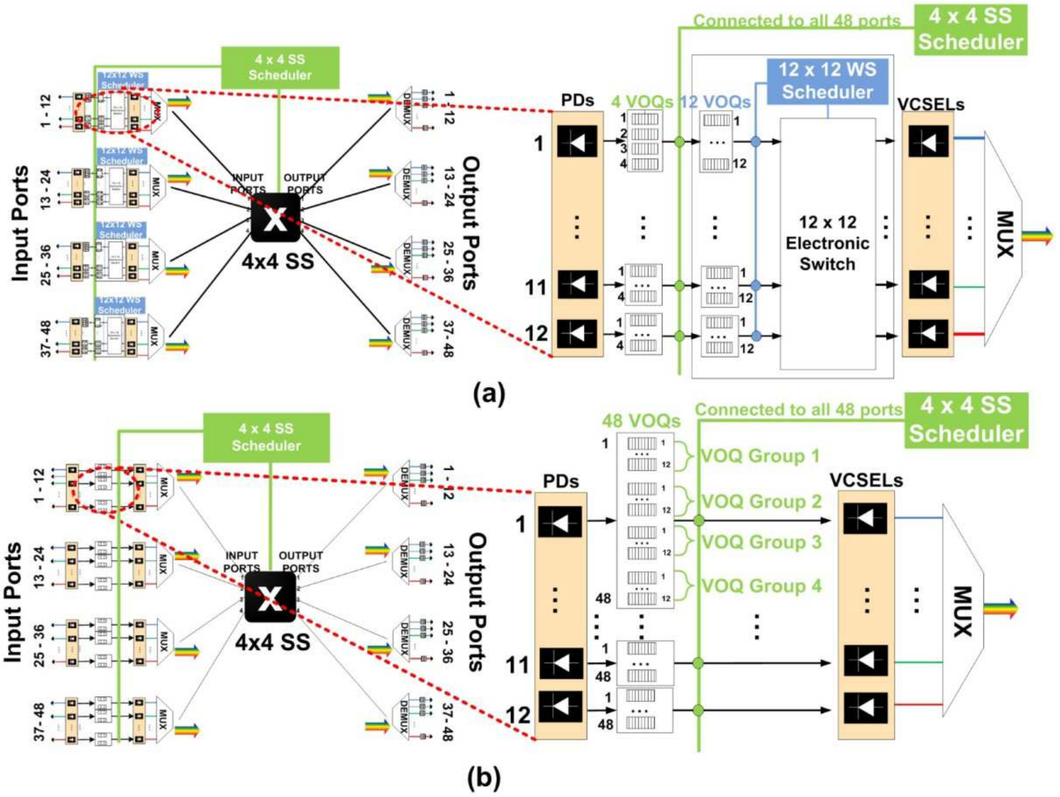

Fig. 2 a 5 schedulers: 1 for the 4×4 SS and 4 for the respective 12×12 electronic switches and the respective VOQ organization in the electronic part of the PhoxTrot chip. **b** A single scheduler for both the WS and SS tasks and the respective VOQ organization in the electronic part of the PhoxTrot chip.

The speedup requirements of the electronic switch can be eliminated if $N=n{\cdot}m$ VOQs are used for every input port. The *N* VOQs are divided in *n* groups of *m* VOQs. The *n* groups correspond to the *n* outputs of the *n*×*n* SS element. In principle the *m*×*m* switches can be avoided completely (Fig. 2b). The (single) scheduler will configure the *n*×*n* SS matrix using a specific scheduling algorithm. Based on this decision, it will use the respective groups of *m* VOQs in order to determine which cells it will finally forward to the chosen input ports of the *n*×*n* SS. For example, if the scheduler in Fig. 2b decides to connect the input port 4 of the SS to SS output port 2, it will use the 12 VOQs contained in VOQ groups 2 for all the 12 inputs of group 4. Note that the scheduling decisions could again be broken in *n*+1 schedulers: the SS scheduler decides the VOQ groups that will be used in the input ports while the remaining *n* decide which cell will be forwarded from the selected (by the first scheduler) VOQ groups.

2.4. Electro-optic Switch Architectures: Hardware Requirements, Functionality and Power Consumption Comparison

In this section we compare the Ws-mSS architectures presented above with silicon photonic switches employing only space switching, as well as other space-wavelength switching architectures, in terms of cost, functionality and energy consumption. We assume that we have packet switching and that buffering and signal regeneration are used in all cases. Thus, for all $N\times N$ packet switch architectures presented below (both SS and WS/SS), N lasers and N receivers are used. Finally, we also compare the WS-mSS approach of Fig. 1c to an all-electronic switching case.

The first two columns in table 1 present the total number (and the size) of basic switching elements as well as the number of switching elements found in the worst case optical data path for the respective architectures. The following columns present the technology of the switches, the switching type (either SS or SS combined with WS, using m wavelengths), the functionality of the switch and the minimum required speedup S to achieve 100% throughput, given that the HOL effect has been eliminated through VOQs in the inputs. For the WS-mSS case, the N VOQ's/input are treated as described in Section 2.3.1.

The first 3 architectures are common switch topologies implemented using MZI- or MMR-based silicon photonics devices [9] assuming only Space Switching (SS). A basic 2×2 MZI switching element consists of two interferometers linked by two arms with an equal length. By changing the phase difference of the two arms, the MZI is switched from "cross" to "bar" state. A basic 2×2 MRR switching element consists of two silicon MRR's and two crossing silicon waveguides. The "cross" and "bar" states are implemented by actively changing the on- and off- resonance of the MRRs to the input signal. The cross-point topology is a crossbar-like configuration requiring N^2 basic 2×2 switching elements arranged in an $N\times N$ mesh. The Switch and Select topology is a tree-like topology where the number of switching stages is $2\cdot\log_2 N$ for all paths, employing only 1×2 and 2×1 basic switching elements. The Benes topology requires the minimum (for non-blocking design) number of switches and switching stages, and this is why it is arguably the most popular choice for such switch implementations [8]-[13]. The RNB feature of this topology requires that an appropriate algorithm is executed at every step to ensure non-blocking switching configurations. The main scalability limitation of all these approaches is the insertion loss. Optimistically, each MZI has insertion losses of 1.5 dB [9]. Taking into account a typical laser output value of +3dBm for the transmitter (typical value for VCSELs) and that typical receiver sensitivities hardly go lower than -11dBm for >25Gb/s data rates [9], the available power budget for >25Gb/s optical links turns out to be lower than 14dB. Relaxing the data rate to 10Gb/s can increase the power budget to higher than 20-25dB. Assuming N=n=32, the 9 switching stages of a solely SS Benes-based approach give 13.5 dB insertion losses, leaving almost no margin for additional loss parameters originating from I/O coupling stages and for additional parameters affecting signal quality, like crosstalk, when targeting 40Gb/s data rate links. The optical data path (the modules the optical signal will meet) in a WS-mSS approach, assuming an $n\times n$ Benes SS, and m wavelengths per input/output consists of the multiplexer (m×1 MUX), followed by $2\cdot\log_2 n$-1 switching stages and the de-multiplexer (1×m DEMUX). Assuming 1 dB insertion loss for the (DE)MUX [9], an N=32 WS-mSS with n=4 and m=8 requires 3 switching stages, yielding insertion losses equal to 6.5 dB. The insertion losses for SS and WS-mSS MZI switches of various sizes are shown in Fig. 3a, using the aforementioned values for the MZI and the MUX/DEMUX, assuming Benes topologies for the space switches. Already for m=2 the WS-mSS approaches give fewer losses than the respective SS approaches. For m=16 the power budget requirements using WS-mSS switches is less by 10dB than the respective SS implementations. In Fig. 3b we present the power consumption for the same cases of Fig. 3a. Since the number of transmitters and receivers is the same for all architectures, their power consumption is not taken into account. A MZI has a power consumption of 2 mW [9] while the AWG (DE)MUX are passive. Since WS-mSS switches increase the port number without additional active elements, they are more energy efficient than SS switches. For m=16 the WS-mSS switches require less than 96% power compared to the SS implementations.

Table 1: Electro-Optic Switch architectures: Cost and Functionality Comparison.

	# of switching elements	**data path (Worst case)**	**Technology**	**Switch**	**Blocking/ Functionality**	***S***
Benes	$N\cdot\log_2N$-N/2 (2×2 SS)	2·$\log_2N$-1 (2×2 SS)	MZI or MRR	SS	RNB / N!	S=1
Cross-point	N^2 (2×2 SS)	2·N-1(2×2 SS)	MZI or MRR	SS	SNB / N!	S=1
Sw. & Select	2·N·(N-1) (1×2 & 2×1 SS)	2·$\log_2N$ (1×2 & 2×1 SS)	MZI or MRR	SS	SNB / N!	S=1
SOA WS-SS	2·N (1×n & n×1 SS)	1×n SS, n×1 SS	SOA	SS/WS (m λ's)	SNB / N!	S=1
OSMOSIS	4·N (m×1 SS & m×1 WS)	m×1 MUX ,EDFA, 1×2N splitter, m×1 SS, 1×m DEMUX, 1 m×1 WS	SOA	SS/WS (m λ's)	SNB / N!	S=1
WS-mSS	$n\cdot\log_2n$ - n/2 (2×2 SS)	m×1 MUX, 2·$\log_2n$-1 (2×2 SS), 1×m DEMUX	MZI or MRR	SS/WS (m λ's)	B / $n!\cdot(m!)^n$	$S\geq1$, $S\leq\min(n,m)$

As discussed above, WS-mSS approaches allow greater scalability, with fewer losses and lower power consumption than SS approaches. The price paid is the reduced functionality of the WS-mSS switch, which is $n!\cdot(m!)^n$ (<N!). The functionality of the WS-mSS approach and its impact on scheduling is examined in detail in Section 3. This lower (blocking) functionality can reduce throughput up to 1/min(n,m) in the worst case for certain appropriately chosen traffic patterns. Vice versa, assuming the version of the architecture depicted in Fig. 1b, where the optics reside inside the switch, speedup equal to min(n,m) is needed to ensure 100% throughput for all cases. However, on average (assuming i.i.d. and uniformly distributed number of cells for every input/output port communication), the speedup requirements are much less (closer to 1). The impact of the blocking functionality of WS-mSS on throughput and speedup requirements is discussed in more detail in Section 4. In Figure 3c the maximum throughput (on average case) is depicted for the same cases of Fig. 3a and 3b. The number of cells for every input/output port communication is uniformly distributed from 1 to 1000. After m=4, the maximum throughput increases slowly. For n=4, m=32 and 64 (giving N=128 and 256) the respective throughput values are 0.903 and 0.92 (not shown in Fig 3c). Thus, WS-mSS approaches can achieve high throughput on average while exhibiting all the advantages described above: high-radices, fewer losses, low power consumption. It follows from the discussion above that in order to design a WS-mSS switch with 100% throughput in all cases and relatively low speedup requirements, at least one dimension (n or m) should be sufficiently small. As it will be argued in Section 4.3, it is preferable to use small n and large m than the opposite in order to achieve lower speedup requirements on average, for large switch sizes. Alternative WS-mSS architectures where the internal line rates of the switch are kept equal to the input and output channel rates (no speedup requirements) are discussed in Section 6.

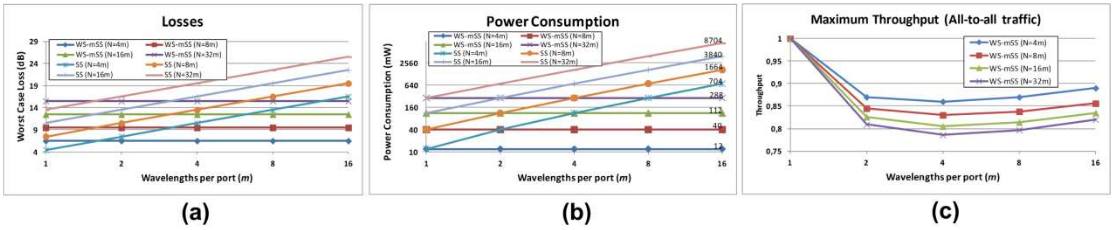

Fig. 3 a Comparison in terms of insertion losses for various SS and WS-mSS switches. **b** Comparison in terms of power consumption for various SS and WS-mSS switches. **c** Maximum throughput on average for various WS-mSS switches: all-to-all traffic with the number of cells for every input/output port communication uniformly distributed from 1 to 1000.

We also compare the aforementioned architectures with two switch architectures, presented in the literature, combining space and wavelength switching (SS and WS). Both architectures are based on Semiconductor Optical Amplifiers (SOA). The wavelength-space non-blocking architecture presented in [18] needs n m×m electronic switches for wavelength selection without using any (DE)MUX. For the space switching part, it requires N 1×n and N n×1 space switches. For every 1×n SS, n SOAs are required (assuming n≤32). For every n×1 SS, 1 SOA is required (assuming n≤32). The total number of SOAs is N·(n+1): however, only 2N SOAs will be active for a full input-output permutation (1 SOA active for every 1×n or n×1 SS). The second architecture (OSMOSIS [19]) follows a broadcast and select approach implementing a N×2N switch (each output has two receivers, Rx, for optimized performance). In the broadcast phase, n broadcast units are used, each containing an m×1 multiplexer, an EDFA amplifier and an 1×2N splitter. In the select phase, 2N select units are used, each containing two stages of m SOA optical selector gates. A first SOA gate selects the correct fiber or spatial group. A second SOA gate, after demultiplexing, then selects the correct wavelength within that fiber. The active elements for a full input-output permutation are 2N SOAs and n EDFAs. Note that OSMOSIS architecture against which MZI WS-mSS architecture is compared is the most energy efficient of the architectures examined in [5] (p.24). The disadvantage of both approaches described above is that they require large arrays of SOAs which are expensive and power hungry.

Taking into account that an active SOA requires around 0.46W [18] for N=64, with n=m=8, the SOA WS-SS and OSMOSIS switching matrices have a power consumption of 59W and 67W, respectively (assuming 1W for the EDFA amplifiers). A solely SS 64-port Benes switch requires 0.704 W (352 2×2 MZIs and 11 switching stages) and a Benes-based WS-mSS with n=m=8 requires 0.04 W (20 2×2 MZIs, 5 switching stages, passive AWG MUX/DEMUX). The WS-mSS approach can also be used in SOA-based switches in order to reduce energy consumption. For instance the 20 2×2 switches required for a 64-port SOA-based WS-mSS with n=m=8 require 36.8 W (assuming 4 SOAs for the implementation of a 2×2 switch).

Finally, we compare an architecture based on WS-mSS switches and an all-electronic architecture assuming a small DC cluster of 768 servers. For the comparison we assume that the 768 servers are interconnected using: a) a fat tree topology based on commodity 48-port electronic switches (the usual topology in modern DCs) and b) a topology based on WS-mSS switches with n=16 and m=12. In order to implement a 768-server full bisection fat tree topology using 48-port switches, 48 switches are needed, arranged in 2-layers: 32 switches are needed for the first layer (ToR) and 16 switches for the second layer. Half of the ports of the 48-port ToR switches are connected to the servers and the other half to the switches in the second layer. Assuming that the typical operating power of a commodity 48-port electronic switch is 390W [29], the total energy consumption of the all-electronic fat tree is 18.72 KW. Using WS-mSS switches, the resulting 768-server topology is similar to the one presented in Fig. 1c. In this case however we have 16 48×48 electronic switches (instead of 4 as in Fig. 1c) and 4 WS-mSS switches with n=16 and m=12 (instead of the 4 WS-mSS with n=4 and m=12 of Fig. 1c). The energy consumption of the 16 electronic switches is 6.24 KW. The optical switching matrix consists of 4 16×16 SS, each one containing 56 2×2 MZI switching elements with 2 mW power consumption. Thus, the total energy consumption of the optical switching matrix is 0.448 W. To perform a fair comparison with the all-electronic architecture, the energy consumption of the N optical links must be taken into account as well. Assuming 22.3 pj/bit energy consumption for a 40 Gb/s VCSEL-based link [28], we have 0.892W for a single link and 685,056 W for the 768 optical links required in the architecture. Therefore, the total energy consumption of the WS-mSS scenario is 6.926 KW, leading to 63% energy savings compared to the all-electronic fat tree architecture. If Valiant routing is used (see Section 4.3) then the 768-server architecture based on WS-mSS switches and S=1 will experience no throughput degradation compared to non-blocking switches.

3. WS-mSS Switch Scheduling

In this section we examine the restrictions that are imposed on scheduling a WS-mSS switch, due to its architecture peculiarities and reduced (blocking) functionality. In 3.1 we examine the constraints that a permutation matrix of WS-mSS switches must satisfy. In 3.2 we describe simple scheduling algorithms for the WS-mSS. In 3.3 we give a lower bound for the number of required scheduling steps. Based on this, we derive the speedup and throughput values for the architecture at hand.

3.1. Permutation Matrices

A permutation matrix P_s is an N×N matrix representing a feasible configuration of an N×N switch in scheduling cycle s. Entry (i, j) in P_s is either 1 (indicating that in the current time slot in-port i will send a packet to out-port j) or 0 (no communication between ports i and j). Permutation matrices for simple non-blocking switches have the following properties:

Constraint C1) there is at most one "1" along a single row, and

Constraint C2) there is at most one "1" along a single column

These are the same constraints C1 and C2 mentioned in Section 2.1, expressed in terms of permutation matrices. We will describe the permutation matrices for a WS-mSS switch by defining first two more matrices: 1) SP_s*:* an n×n permutation matrix that describes the state of the SS part of the WS-mSS switch in scheduling cycle s, and 2) $WP_{s,i,j}$ *:* an m×m permutation matrix that describes the state of the WS part of the WS-mSS switch for input i and output j of the nxn SS in scheduling cycle s. Entries in permutation matrices P_s for a WS-mSS switch are either 1 or 0 according to the following rule:

$$P_s(i,j) = \begin{cases} 1, \text{if } SP_s(i',j') = 1 \text{ and } WP_{s,i',j'}(i-(i'-1)\cdot m, j-(j'-1)\cdot m) = 1 \\ 0, & \text{otherwise} \end{cases}$$

where $P_s\,(i,j)$ denotes entry (i, j) in P_s, i, j ∈{1,2,…,N}, $i' = \lceil i/m \rceil$ and $j' = \lceil j/m \rceil$. Thus, in addition to the restrictions C1 and C2 mentioned above for non-blocking switches, two more constraints should be satisfied for permutation matrices P_s for N×N WS-mSS switches:

Constraint C3) there is at most one "1" along a single row of SP_s, and

Constraint C4) there is at most one "1" along a single column of SP_s.

An example for P_s with n=4 and m=12 (the PhoxTrot case) is depicted in Fig. 4. Since there are n! and m! SP_s and $WP_{s,i,j}$ configurations/permutations, respectively, and n input (and output) space-switching ports, a WS-mSS has a functionality of $n!\cdot(m!)^n$, as opposed to the N! functionality of an N×N non-blocking switch. The additional restrictions, coming from the smaller number of feasible input-output permutations, have implications on the switch performance that will be examined in Section 4.

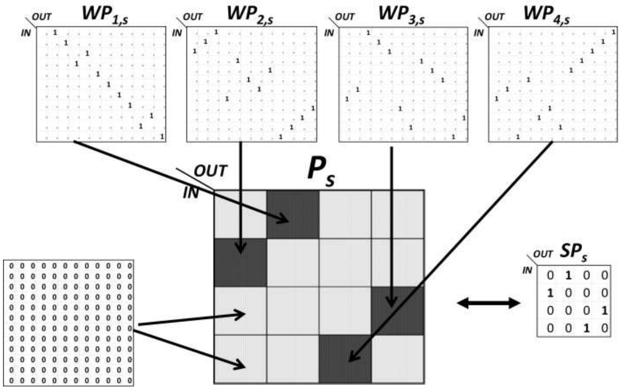

Fig. 4 Example of a specific permutation matrix for a WS-mSS switch with m=12 and n=4 (the PhoxTrot case). $P_s(48,27)=1$ since $SP_s(4,3)=1$ and $WP_{s,4,3}(12,3)=1$.

3.2. Simple Scheduling Algorithms

A (packet) switch scheduler must perform, for every scheduling cycle, a matching between the input and the output ports of the switching fabric, based on the request matrix, in order for the switch to be configured appropriately. A request matrix is an N×N matrix containing a 1 in position (i, j) if input port i wants to communicate with port j for scheduling cycle s, or 0 otherwise. The *maximum (optimal) matching* for a single scheduling step can be found in time $O(N^{5/2})$ for an N×N switch [30]. In practice *maximal matching* is usually performed, since it is simpler to implement and has a faster running time. There exist several well known maximal matching heuristics ([31]-[33]) as well as heuristics taking into account the VOQ sizes or the waiting times of the cells [23]. Below we outline simple scheduling algorithms for the (i) separate schedulers and (ii) single scheduler versions of the WS-mSS architecture outlined in Section 2 based on the scheduling algorithms mentioned above.

- Simple Scheduling algorithm for single scheduler version:

1. Execute maximum matching algorithm for N×N request matrix: generate P_s
2. View P_s as an n×n matrix whose entry (i,j) is the number of matchings achieved in the previous step for m×m block located in position (i,j) of the n×n matrix
3. Execute maximum matching algorithm on n×n matrix: determine SP_s
4. Based on SP_s update P_s (set 0 in all entries of the unused m×m $WP_{s,i,j}$ submatrices of P_s)

- Simple Scheduling algorithm for separate schedulers version:

1. SS scheduler: execute maximum matching algorithm for n×n request matrix: generate SP_s
2. WS schedulers: execute maximum matching algorithm for m×m request matrix: generate $WP_{s,i,j}$ permutation matrices
3. P_s is constructed based on SP_s and $WP_{s,i,j}$ generated in previous steps

The algorithms were described assuming that an (optimal) maximum matching algorithm, such as [30], will be used in the individual steps outlined above. Any (suboptimal) maximal matching algorithm could also be used instead.

3.3. Lower Bound for the Required Scheduling Steps using a WS-mSS switch

First, we describe known results regarding a lower bound on the number of scheduling steps, assuming N×N non-blocking switches. Based on these results we will derive similar results for N×N WS-mSS switches. We will use the notion of traffic matrices. The analysis that we follow is based on matrix decomposition [20]. A *traffic matrix* $D = [d_{ij}]$ is an N×N matrix, where entry d_{ij} is the *total* number of cells that need to be transmitted from switch port i to port j. In order to isolate the impact of the additional constraints of the WS-mSS switch architecture on scheduling performance from other factors, we will make the ideal assumption that all VOQs are infinite in all cases below (both for WS-mSS and simple non-blocking switches), thus no cells are lost due to link overloads. Let $R_i = \sum_j d_{ij}$ and $C_j = \sum_i d_{ij}$ be the i-th row sum and j-th column sum, respectively, of the traffic matrix D. Let us define as $h = \max_{i,j}(R_i, C_j)$ the *critical sum* of the traffic matrix D. A row or column of sum of all its entries equal to h is called a *critical line*. According to a well-known theorem (shown in [34], p. 57):

Theorem 1: An N×N traffic matrix D can be written as a sum of h permutation matrices.

Thus, $T_{min}=h$ where T_{min} is the minimum number of scheduling steps assuming an optimal, with respect to required steps, algorithm (such as the maximum matching algorithm in [30]). The theorem presented above holds for N×N switches with fully non-blocking functionality and full throughput, i.e., switches exhibiting only constraints C1 and C2 described in Section 3.1 and no additional constraints, due to HOL for instance.

Let T'_{min} denote the minimum number of scheduling steps assuming a WS-mSS switch (and optimal scheduling algorithms). For an N×N WS-mSS switch the N×N traffic matrix D can be viewed as an n×n matrix, where each “entry”-block of the latter is an m×m traffic matrix. The entries in an m×m matrix contain the traffic requirements between input ports located and multiplexed in the same input interface of the n×n SS, and the output ports ocated and demultiplexed from the same output interface of the SS. Let $R_{i,rc} = \sum_j d_{ij}$ and $C_{j,rc} = \sum_i d_{ij}$ be the i-th row sum and j-th column sum of the m×m block/submatrix of traffic matrix D, located in row r and column c of the n×n matrix, where $i \in [1 + (r-1) \cdot m, \dots, m + (r-1) \cdot m]$ and $j \in [1 + (c-1) \cdot m, \dots, m + (c-1) \cdot N]$. Let $h_{rc} = \max_{i,j}(R_{i,rc}, C_{j,rc})$ be the critical sum of the m×m submatrix located in row r and column c of the n×n matrix. Let us define an n×n matrix $D' = [d'_{rc}]$ where entry d'_{rc} in row r and column c is the critical sum h_{rc} of the respective m×m block in the original traffic matrix D. Let $R_r = \sum_c d'_{rc} = \sum_c h_{rc}$ and $C_c = \sum_r d'_{rc} = \sum_r h_{rc}$ be the r-th row sum and c-th column sum of this nxn matrix and $h' = \max_{r,c}(R_r, C_c)$ be its critical sum. Then:

Theorem 2: An N×N traffic matrix D can be written as a sum of h' permutation matrices assuming an N×N WS-mSS switch composed of n m×m wavelength selection (WS) elements and 1 n×n space switch (SS), with $N = n \cdot m$.

Proof. Every m×m block of traffic matrix D is itself a traffic matrix. By applying theorem 1, we obtain that scheduling the traffic in the m×m block of traffic matrix D, located in row r and column c of the n×n matrix needs h_{rc} steps. Traffic matrix D' can itself be viewed as an n×n traffic matrix where every entry d'_{rc} requires h_{rc} steps. D' will be scheduled by the n×n space switch exhibiting constraints C3 and C4. By applying theorem 1 again, this time for traffic matrix D', we obtain that the scheduling of D' requires at least h' steps. □

Note that h' is equal to or larger than h, due to constraints C3, C4 which reduce the switch functionality. The relation between h' and h for various traffic cases is examined in more detail in the following section.

In what follows we will make a distinction between S and $S(D)$. S is the speedup of a switch architecture as described in Section 2.1, while $S(D)$ is the minimum speedup a WS-mSS architecture requires in order to schedule traffic matrix D. We will also denote as $\Theta(D)$ the maximum throughput that a WS-mSS with S=1 can achieve for traffic matrix D. Note that $\Theta(D)$ is the inverse of $S(D)$. As mentioned in Section 2.1, S>1 can be provided only by CIOQ or OQ approaches. Thus, from the two WS-mSS switch architectures presented in Section 2.2 only the CIOQ WS-mSS switch implementation of Fig. 1b can provide S>1. Nevertheless, $S(D)$ is also an indicator of the performance of a WS-mSS switch compared against a full throughput switch for the same traffic matrix D.

Theorem 2 holds for WS-mSS switches with constraints C1, C2, C3 and C4 and no additional constraints, due to HOL for instance (which is eliminated by using VOQ concept appropriately adjusted, see Section 2.2.1). The only difference between h and h' is due to additional constraints C3 and C4 mentioned in Section 3.1. Since h is the minimum number of scheduling steps that can be achieved by a fully non-blocking switch with 100% throughput (without constraints C3 and C4), the minimum speedup $S(D)$ required for scheduling traffic matrix D and the maximum throughput $\Theta(D)$ can be obtained by:

$$S(D) = \frac{T'_{min}}{T_{min}} = \frac{h'}{h} \quad \text{and } \Theta(D) = \frac{T_{min}}{T'_{min}} = \frac{h}{h'} \qquad (1)$$

4. WS-mSS Switch Performance

In this section we examine the performance of a single WS-mSS switch in terms of throughput and required speedup for the best and worst cases (Section 4.1), an average case assuming i.i.d. (independent and identically distributed) and uniformly distributed number of cells for every input/output port communication (Section 4.2) and for various synthetic traffic patterns (Section 4.3).

4.1. Best and Worst Case Performance

The performance of an N×N WS-mSS switch in terms of scheduling steps for the best and worst case is given by the following theorem:

Theorem 3: The number of scheduling steps $T'_{min} = h'$ that an N×N WS-mSS switch requires to schedule any traffic matrix, is bound from below and above as: $h \leq h' \leq \min(n,m)\cdot h$, assuming optimal scheduling in terms of required steps.

Proof. The proof is based on Theorem 2 and the construction of the worst case and best case traffic matrices.

We first prove the upper bound. In order to find the worst case performance for an N×N WS-mSS switch we will construct a traffic matrix in such a way that a) it takes exactly 1 scheduling step for an N×N non-blocking switch without the constraints C3 and C4 (thus h=1), and b) it takes the maximum number of steps for the N×N WS-mSS switch due to these constraints. In order to achieve the maximum value for h' we will focus on a single line in the n×n traffic matrix D' (see Section 3.3) and we strategically place cells to get the maximum possible critical sum h'. Without loss of generality we focus on a single row r of traffic matrix

D'. In this case the worst case traffic for D' is when a single input port of n×n SS has to connect to as many as possible output ports of the SS (potentially with all of them), maintaining at the same time critical sum equal to h=1. This can be achieved by placing 1 cell in entry (i,i) of the m×m block located in position (r,i) of the n×n SS, for all 1≤ i ≤ min(n,m). Thus, the critical sums h_{ri} for the respective m×m block located in row r and columns i, 1≤ i ≤ min(n,m) will be at most 1. This can be shown by taking all cases regarding n and m sizes:

- $n \leq m$. In this case the size of the WS element is greater than the size of the SS and we have critical sum h'=n (and h=1). An example for the aforementioned traffic pattern is shown in Fig. 5a for the PhoxTrot case (n=4, m=12) creating a critical line in the first row of D' (r=1).
- n>m. In this case the size of the WS element is smaller than the size of the SS and we have critical sum h'=m. An example is shown in Fig. 5b for n=4 and m=2.

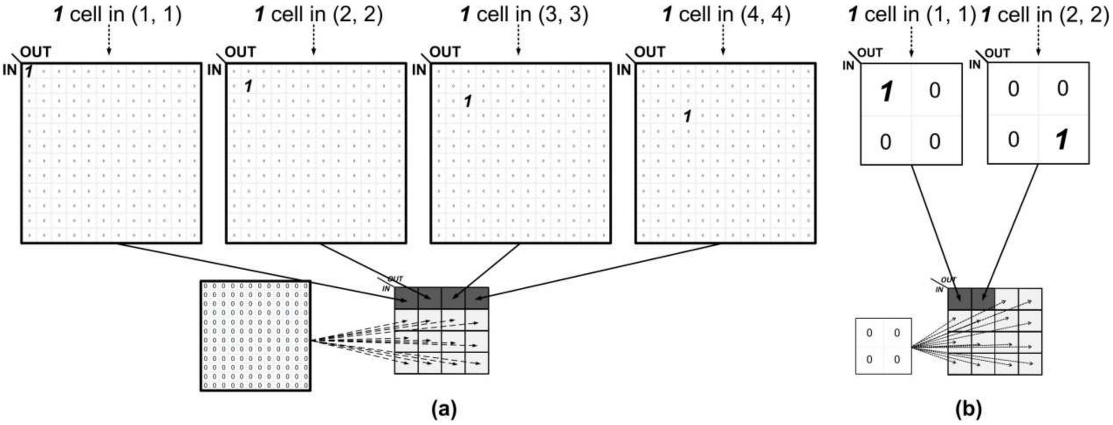

Fig. 5 a Worst case traffic matrix for n=4, m=12, and r=1. h'=n=4 (and h=1). **b** Worst case traffic matrix for n=4 and m=2 (r=1). h'=m=2 (and h=1).

From the discussion above we conclude that the worst case traffic pattern needs h'=min(n,m)=min(n,m)·h steps using an N×N switch composed of n m×m WSES and 1 n×n SS. Now we remove the assumption that a single port has to send at most 1 cell. Assuming that there are a_i cells (instead of 1) in entry (i,i) of the m×m block located in position (r,i) of the n×n SS, for all 1<i< min(n,m), then $h = \max_{1 \leq i \leq \min(n,m)} a_i$ and

$$h' = \sum_{i=1}^{\min(n,m)} a_i \leq \sum_{i=1}^{\min(n,m)} \max_{1 \leq i \leq \min(n,m)} a_i = \sum_{i=1}^{\min(n,m)} h = h \cdot \min(n,m)$$

The upper bound is reached when $a_1 = a_2 = \cdots = a$. Note that this is also the case for the initial scenario where a=1.

Now we prove the lower bound. The best performance for an N×N WS-mSS switch is achieved for a traffic pattern that simply does not require additional steps due to constraints C3 and C4. An example of best case traffic is uniform traffic with *exactly a* cells in all entries. In this case $h = N \cdot a$ for all rows and columns and $h_{rc} = m \cdot a$ for all r,c and $h' = n \cdot h_{rc} = n \cdot m \cdot a = N \cdot a = h$. □

Corollary 3.1: S(D) and Θ(D) for a WS-mSS and a traffic matrix D are bound from below and above by *1 ≤ S(D) ≤* min(n,m) and *1/* min(n,m) ≤ *Θ(D)* ≤ 1 respectively.

We can conclude that for the special case of n=4 and m=12 (the PhoxTrot case) $h \leq h' \leq 4h$ and $1 \leq S(D) \leq 4$, $0.25 \leq \Theta(D) \leq 1$. Thus, this architecture needs S=n=4 in order to ensure 100% throughput for the worst case. Based on these results we can also estimate the tradeoff between increased connectivity and worst case throughput for WS-mSS switches: the best throughput is achieved for the trivial cases when n=1, N=m or m=1, N=n, in which switching is performed entirely through wavelength selection or spatially. Increasing either n or m increases the switch connectivity, limiting however the worst case throughput to 1/min(n,m).

4.2. Performance on Average

In order to estimate the required scheduling steps *on average*, we assume that the number of cells in traffic matrix’s entries are independent and identically distributed (i.i.d.) variables obtained from the Uniform Distribution. We followed a Monte-Carlo approach, generating 1 million of random traffic matrices whose entries were randomly selected integer numbers from a to b and we estimated speedup and throughput. In the following, we present the results obtained via Monte-Carlo as well as the results obtained using theoretical approximations for speedup and throughput, which are described in detail in Appendix A. Finally, we give the asymptotic values for $S(D)$ and $\Theta(D)$ for large values of b (and a=1), derived using the aforementioned theoretical approximations (the details can be found in Appendix B).

Figure 6 depicts the obtained PDFs for T_{min} and T'_{min} for a 48×48 non-blocking crossbar switch and the PhoxTrot switch (WS-mSS with n=4 and m=12) for 3 different cases for a and b obtained via Monte-Carlo estimation, as well as the respective PDFs obtained with the theoretical approximation (denoted as i.i.d. approx.) of Eq. (2)-(10) (Appendix A). In Table 2 we present the values of $E(T'_{min})$, $E(T_{min})$, the $S(D)$ and $\Theta(D)$ for 5 various cases of a and b for both the Monte-Carlo approach and the theoretical approximation of Eq. (2)-(10). The $S(D)$ and $\Theta(D)$ columns of Table 2 for the theoretical estimation were calculated using Eq. (10). The respective columns for the Monte-Carlo approach were estimated as $E(T'_{min}/T_{min})$ and $E(T_{min}/T'_{min})$. Note that $\mathrm{E}(T'_{min}/T_{min})\approx E(T'_{min})/E(T_{min})$ and $E(T_{min}/T'_{min})\approx E(T_{min})/E(T'_{min})$, indicating that T'_{min} and T_{min} can be treated as i.i.d.

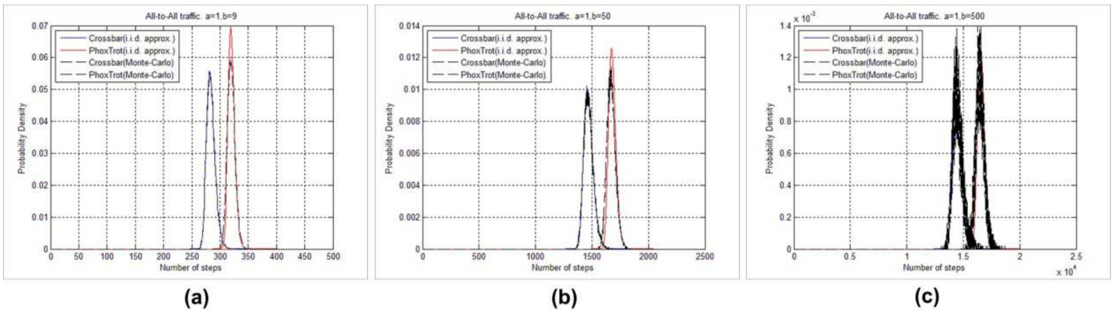

Fig. 6 PDFs for T_{min} and T'_{min} for a 48×48 non-blocking crossbar switch and a 48×48 WS-mSS switch with n=4 and m=12 (PhoxTrot case) for both theoretical approximation of Eq. (2)-(9) assuming i.i.d. variables and Monte-Carlo estimation. **a** a=1, b=9. **b** a=1, b=50. **c** a=1, b=500.

Table 2 : $E(T'_{min})$, $E(T_{min})$, $S(D)$ and $\Theta(D)$ for all-to-all traffic and a=1, estimated by Eq.(2)-(10) as well as Monte-Carlo estimation.

	b	$E(T'_{min})$	$E(T_{min})$	$S(D)$	$\Theta(D)$
	2	87.2	80.51	1.083	0.92

Monte Carlo	9	319.92	284.18	1.126	0.89
	25	847.28	747.42	1.134	0.88
	50	1670.9	1470.9	1.136	0.88
	500	16494.2	14493.7	1.138	0.88
i.i.d. approx.	2	86.07	80.16	1.074	0.93
	9	320.33	284	1.128	0.89
	25	851.41	747.65	1.139	0.88
	50	1680.76	1471.85	1.142	0.88
	500	16607.33	14506.6	1.145	0.87

The results presented above indicate that the PhoxTrot architecture achieves maximum throughput around 88% for the tested values of a and b. If a=b, then the all-to-all traffic pattern degenerates in the best case uniform traffic pattern, where both $S(D)$ and $\Theta(D)$ are equal to 1. Assuming a=1 and $b\rightarrow\infty$, we show in Appendix B that $1.79\leq S(D)\leq 2.01$ and $0.5\leq\Theta(D)\leq 0.56$. We expect however that $S(D)$ and $\Theta(D)$ converge very slowly to the asymptotic values as b increases. For instance, using the i.i.d. approximation (Appendix A), we found that for $b=10^3$, 10^4, $5{\cdot}10^4$, the respective speedup values are 1.14497, 1.14512 and 1.14513.

4.3. Monte-Carlo Estimation for Various Traffic Patterns

In this Section we present results we obtained for various traffic patterns following a Monte-Carlo approach. We generated a large number (1 million) of random traffic matrices whose (specific and according to the traffic pattern) entries were randomly selected integer numbers from a to b. In every iteration, we calculated critical sums h and h'. Thus, we were able to estimate $E(T'_{min})$ and $E(T_{min})$ for the respective traffic matrices. The results are depicted in Fig.7.

In Figure 7a we depict the results for all-to-all traffic (the same case as in Section 4.2) keeping m constant while varying the SS size n. Speaking in terms of traffic matrices, the size of traffic matrix D' increases while the size of the m×m blocks stays the same. As shown in Fig. 7a, as n increases S increases slowly. For instance, for m=4 and n=4 yields an N=16 switch with speedup requirements around 1.16. For n=16 we get a 64-port switch with speedup requirements of around 1.3 indicating good scalability of the WS-mSS architecture. In Figure 7b we present the results for all-to-all traffic, for a constant value of n, varying the number of wavelengths m. In terms of traffic matrices, the size of traffic matrix D' is kept constant while the size of m×m blocks increases. As m increases, S decreases towards the lower bound (=1). The results of Fig. 7b are similar to the results of Fig. 3c (expressed there in terms of throughput). Since large n and large m are not both feasible at the same time due to both losses and crosstalk (see design options and scalability studies in [12]), from the discussion above follows that for large N, many wavelengths multiplexed in a single input/output port and small space switches (large m and small n) are preferable to high radix space switches with few wavelengths per port (large n and small m) for lower speedup requirements on average.

In Figures 7c and d we present the respective results for various synthetic traffic patterns [23]. Several of these patterns are based on communication patterns exhibited by particular HPC applications such as fluid dynamics simulations, sorting applications, FFT. The worst case traffic pattern is constructed as described in Section 4.1. The traffic patterns used in Fig. 7d are bit permutations requiring the port number N to be a power of 2. The implied topology is a simple star network where N HPC compute nodes are interconnected using a single N×N WS-mSS. Bit reverse and transpose are pathological (worst case) traffic patterns while bit complement is a best case traffic. Bit rotation, shuffle, tornado and neighbor traffic require $S(D)$ at most 2 in all examined cases.

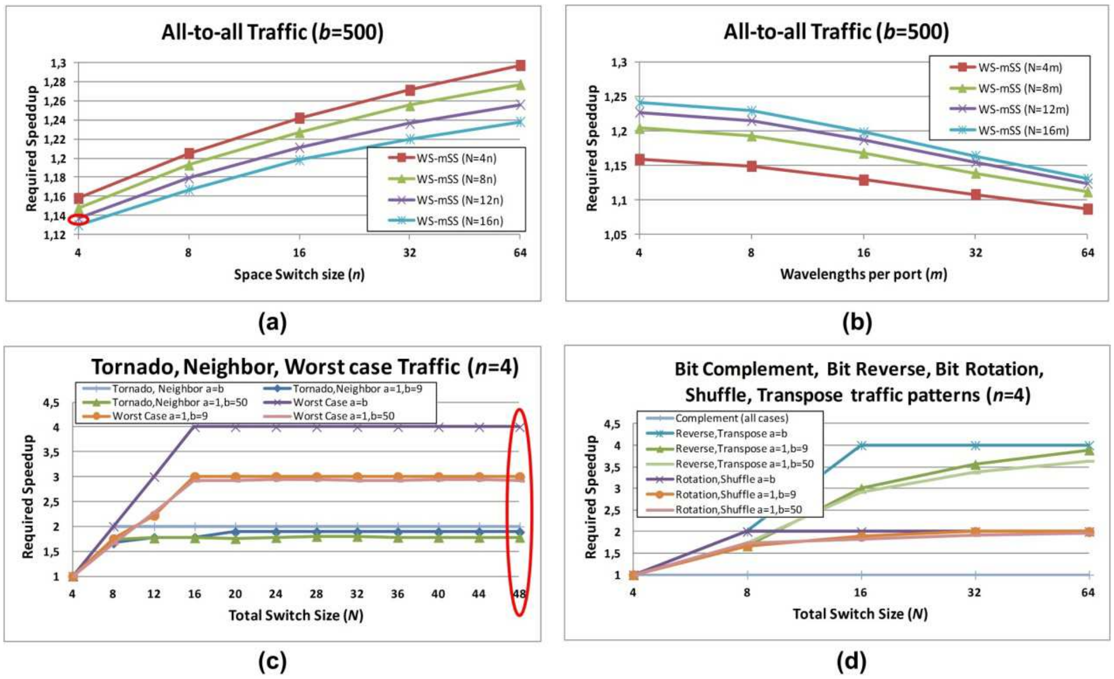

Fig. 7 Required speedup for various traffic patterns **a** All-to-all traffic (a=1, b=500) for various WS-mSS switch sizes while n increases and m is constant (PhoxTrot case marked with red). **b** All-to-all traffic (a=1, b=500), while m increases and n is constant. **c** Tornado, Neighbor, Worst Case traffic for various WS-mSS switch sizes with n=4 (PhoxTrot case marked with red). **d** Bit permutation traffic patterns for various WS-mSS switch sizes with n=4.

In order to mitigate performance degradation in such cases (or in larger networks based on WS-mSS swithes) while keeping S=1, appropriate mapping algorithms must be developed for the assignment of application tasks to nodes connected to WS-mSS ports that do not stress scheduling constraints C3 and C4. Task-to-processor assignment algorithms are already used in HPC systems based on low degree topologies, such as mesh/torus, in order to reduce the hops traveled by messages [35]. In cases like the one depicted in Fig. 1c, if Valiant routing [23] is used, no performance degradation is exhibited in the former (assuming S=1) compared to respective non-blocking architectures. In Valiant routing, for every packet (alternatively for every flow), a top-layer switch in a fat tree topology is randomly chosen as an intermediate destination. In this way every traffic pattern is transformed into uniform traffic, achieving load-balancing while avoiding bottlenecks, at the cost of increasing average distance in terms of hops in cases where communication locality exists. Therefore, the WS-mSS switches in the highest layer will have to handle uniform traffic, which can do equally well as non-blocking switches. Similar performance results to non-blocking switches are also obtained assuming Valiant routing and direct network topologies, such as mesh/torus, using WS-mSS switches (in these topologies, in order to achieve load-balancing, any other node in the network can be chosen randomly as an intermediate node).

5. Multiple WS-mSS Switches

In this section we examine the performance of the multiple WS-mSS switching elements architecture version presented in Fig. 1c. We assume that N VOQs are present in every input port (see discussion in 2.3.1). In sub-Sections 5.1, 5.2, 5.3 we examine the switch permutations, the lower bounds for scheduling and the architecture performance for various cases, respectively.

5.1. Permutation Matrices

An $N \times N$ permutation matrix P_s for the multiple WS-mSS version is divided in n^2 blocks of size $k \times k$ each, where in the case of a single WS-mSS the permutation matrices were divided in n^2 blocks of size $m \times m$. P_s is also divided in n sub-matrices P_s^i, $i \in [1,2,\dots,n]$. P_s^i corresponds to the state of WS-mSS switch i for scheduling step s. Each P_s^i is an $N \times k$ permutation matrix that has the same form as the permutation matrices of a single WS-mSS as described in Section 3.1, with the only difference being its size (the latter's size is $N \times N$). P_s^i contains columns $[1 + m \cdot (i-1) + k \cdot (j-1), \dots, i \cdot m + k \cdot (j-1)]$, $j \in [1,2,\dots,n]$ of permutation matrix P_s. A single '1' in a row (column) of SP_s^i (the permutation matrix that describes the state of the SS part of the WS-mSS switch i) prohibits another '1' only in the respective rows and columns of SP_s^i. An example for the form of a permutation matrix P_s is shown in Fig. 8 for the architecture of Fig. 1c where N=192, k=48, m=12, n=4

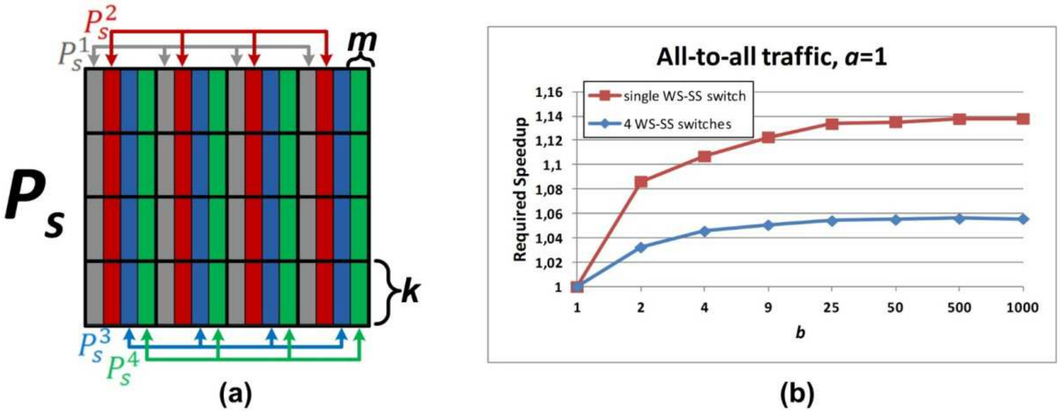

Fig. 8 **a** Permutation Matrix of an $N \times N$ switch, $N = k \cdot n$ composed of n (=4) WS-mSS switching elements interconnecting n $k \times k$ electronic switches with $k = n \cdot m$, where m wavelengths are multiplexed in a WDM signal. **b** $S(D)$ for All-to-all traffic, with traffic matrix entries uniformly distributed for a=1 to b for a single WS-mSS switch with n=4, m=12 (architectures of Fig. 1a, 1b) and for 4 WS-mSS switches with n=4, m=12 interconnecting 4 48×48 electronic switches (architecture of Fig. 1c).

5.2. Lower Bound for Scheduling Steps

In the case of multiple WS-mSS switches, traffic matrix D is divided in n sub-matrices $D_i, i \in [1,2,\dots,n]$. D is divided in sub-matrices D_i similar to the way permutation matrix P_s is divided in sub-matrices P_s^i, and as so it has the same size as P_s^i. If we view $N \times k$ matrices D_i as $n \times n$ matrices D'_i whose entries are the critical sums of the respective $k \times m$ blocks of D_i's and denote the critical sums of D'_i as h'_i, then the following theorem gives the lower bound for the required scheduling steps for the multiple WS-mSS switch architecture:

Theorem 4: An $N \times N$ traffic matrix D can be written as a sum of $T'_{min} = \max(h, h'_1, h'_2, \dots, h'_n)$ permutation matrices assuming an $N \times N$ switch with n WS-mSS switching elements interconnecting n $k \times k$ electronic switches with $k = n \cdot m$.

Proof. h found in the bound, above follows from the fact that D is a traffic matrix that should satisfy constraints C1 and C2. $h'_1, h'_2, \dots, h'_n$ arise as the n WS-mSS switches serve the portions of traffic found in matrices D_i in parallel. Thus, for serving traffic matrix D at least h steps are required, unless one or more WS-mSS swiches i with $h'_i = \max(h'_1, h'_2, \dots, h'_n)$ responsible for scheduling the portion of traffic contained in D_i requires $h'_i > h$ steps due to constraints C3 and C4 for D_i. An example where the first term dominates is for a traffic pattern where a single input port needs to send a cells in each output. In this case $h = N \cdot a$ and $h'_i = k \cdot a$. □

5.3. Performance Estimation

The performance of an $N \times N$ WS-mSS switch in terms of scheduling steps for the best and worst case is given by the following theorem:

Theorem 5: The number of steps T'_{min} required by an $N \times N$ switch with n WS-mSS switches interconnecting n $k \times k$ electronic switches is bound from below and above by: $h \leq T'_{min} \leq \min(n,m) \cdot h$ assuming optimal in terms of steps scheduling.

Proof. The proof follows the same line of argument found in the proof of theorem 3. The worst case performance is exhibited when the worst case traffic described in the proof of theorem 3 occurs for at least one sub-matrix D_i, giving $T'_{min} = \min(n,m) \cdot h$. A best case example is again uniform traffic with a cells in all D entries. □

Corollary 5.1: $S(D)$ and $\Theta(D)$ for an $N \times N$ switch composed of n WS-mSS switches interconnecting n $k \times k$ electronic are bound from below and above by $1 \leq S(D) \leq \min(n,m)$ and $1/\min(n,m) \leq \Theta(D) \leq 1$, respectively.

The best and worst case performance bounds of the multiple WS-mSS switches architecture are the same to the bounds for a single WS-mSS switch of this architecture. Figure 8b depicts the S values for all-to-all traffic with traffic matrix entries uniformly distributed from a to b (a=1 in all cases) for the 192×192 architecture of Fig. 1c (N=192, k=48, m=12, n=4) against a single WS-mSS switch with m=12 and n=4 (Fig. 1a, 1b). As expected the multiple WS-mSS architecture is closer to the lower bound for speedup than the single WS-mSS architecture (maximum throughput 95% and 88% respectively for b=1000), since as discussed in Section 5.1, in the former case a single ‘1’ in a row (column) of SP_s^i prohibits another ‘1’ only in the respective rows and columns of SP_s^i while the entries contained in $SP_s^j, \forall j \neq i$ are unaffected. For all the other cases the obtained speedup value is equal for both architectures (we also examined bit permutation traffic patterns with n=4, m=16 and k=64 giving N=256).

6. Alternative Switch Architectures

In this section we present two architecture alternatives to the single WS-mSS architecture outlined in Section 2 without the additional scheduling constraints C3, C4 and we discuss the trade-offs between performance and additional hardware requirements.

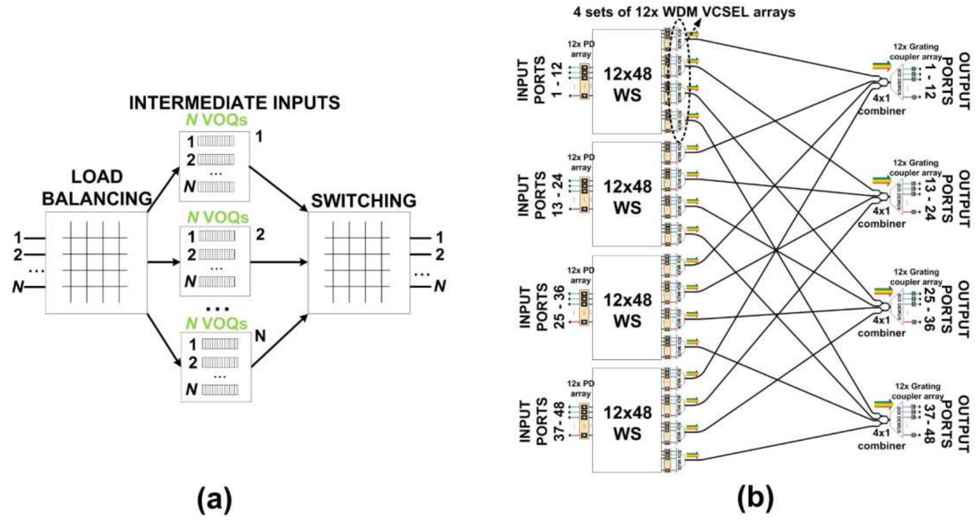

Fig. 9 a Alternative architecture for the 48×48 WS-mSS where the 4×4 SS has been replaced by point-to-point optical links. **b** Load Balanced Birkhoff–von Neumann switch architecture requiring 2 switching stages.

The first architecture variation is based on the observation that WS-mSS switches can handle uniform traffic as well as simple non-blocking crossbars without the additional scheduling constraints C3 and C4 and relies on the Load Balanced Birkhoff–von Neumann (LBBN) switch architecture [36]. The basic LBBN switch architecture is depicted in Fig. 9b. The switch consists of 2 identical switching stages and a single buffering stage between these stages where every buffer is partitioned in N Virtual Output Queues (VoQ). All the switch external lines are assumed to be synchronized [37]. Both switching stages follow a fixed sequence of periodic configurations such as simple Round-Robin scheduling (where input i connects to output port $[i + s -1] \bmod N +1$ in scheduling step s). The first switch uniformly balances the input traffic over all the VoQs of the intermediate stage, thus transforms any traffic in a pseudo-uniform traffic pattern. The second stage is an input-queued crossbar switch in which each VOQ is served at a fixed rate over the load-balanced input traffic. The main advantage of the LBBN switch architecture is that it trivializes scheduling while achieving 100% throughput for a large class of traffic patterns. The same architecture can be realized using 2 successive WS-mSS switches without throughput degradation due to the additional scheduling constraints, while using S=1. The first WS-mSS switch does not maintain VOQs (the cells that arrive in an input port are immediately forwarded to its output port that happens to be connected in this scheduling cycle), and the second switch uses VOQs as described in Section 2.2.1. The WS-mSS switches should execute a modified version of round-robin scheduling that we will refer to “2-level round-robin” since a simple round-robin cannot ensure that all N inputs are connected with N outputs (full switch configuration) in every scheduling cycle for a WS-mSS switch due to the additional scheduling constraints. In the 2-level round robin a round-robin algorithm is performed for the SS element. The SS remains in every configuration for m scheduling cycles. For every SS configuration, the “selected” m×m switching elements (represented by the m×m blocks of the permutation matrix for which SP_s=1) will concurrently perform a round-robin algorithm. Thus, the configurations of all “selected” m×m switching elements will be the same in every cycle s. The 2-level round-robin ensures full switch configurations in every cycle s for WS-mSS switches. A disadvantage of the LBBN architecture described above is that it requires more opto-electronic and electro-optic conversions due to the presence of 2 switching steps. In principle the first switching stage could be a simple *n*×*n* SS executing a simple round-robin algorithm, staying in a single switch configuration for *m* scheduling cycles, omitting the first opto-electronic and electro-optic conversion - assuming that this is feasible with respect to insertion losses for the given power budget (alternatively, amplification is needed). Thus, all the electronic components for this architecture variation are placed between the two optical *n*×*n* space switches. The load-balancing *n*×*n* SS can dissolve the worst case traffic pattern mentioned in Section 4.1, if such pattern occurs, so that the WS-mSS switch in the second stage can handle the (now) uniform traffic (as was also shown in Section 4.1). In this case using only two *n*×*n* SS elements can guarantee full throughput for a large number of traffic scenarios. It should be mentioned however that LBBN switches do not guarantee the correct cell sequencing in the output, while there are some pathological traffic patterns that reduce throughput. Dealing with these issues requires additional buffering in the input as well as the outputs stages or more complex buffer structures and policies between the 2 switches [38]-[40].

The second variation of the WS-mSS switch architecture described in Section 2 can be obtained by replacing the *n*×*n* SS with n^2 fixed optical links and the *m*×*m* electronic switches with *m*×(*n·m*) electronic switches. In every input, after the *m*×(*n·m*) switching element, *n* (identical) sets of *m* WDM VCSEL arrays are required, instead of a single set of 1 such set of *m*=12 VCSELs as in the original architecture. The *n* groups of *m* multiplexed signals are connected to all *n* outputs. In every output, an *n*×1 combiner is used to combine the signals of the *n* groups of *m* multiplexed signals in a single signal which is then demultiplexed in *m* signals (collisions should be avoided by the scheduling algorithm). This architecture provides $S=n$ ($\geq \min(m,n)$) in the space domain: more than or equal to the required speedup for the worst case. An example for this alternative with parameters *n*=4 and *m*=12 (PhoxTrot case) is depicted in Fig. 8a. Its disadvantage is that it requires *n* times more VCSEL sets ($m \cdot n^2$ VCSELs in total) and multiplexers (n^2 in total) than the original, as well as *n* *n*×1 (passive) combiners.

7. Conclusion

Some of the most prominent devices for DC and HPC application are MZI- and MRR-based, small radix Si-Pho space switches that exhibit fast reconfiguration times, and are capable of supporting multiple optical signals multiplexed through WDM. In this work we examined scalable electro-optic switch architectures which combine small port number (radix) MZI or MRR space switching of multiplexed WDM signals, to achieve large port numbers and good throughput on average using few optical switching stages and low total insertion losses, which is the main scalability limitation for silicon photonic switching elements. The price paid for multiplexing multiple signals using WDM and then switching that multiplexed signal in the space domain using SS matrices, is two additional constraints which restrict the feasible permutation matrices of the switching fabric in every scheduling cycle. Therefore more scheduling steps are required in order to schedule incoming traffic. We showed that these constraints reduce the maximum throughput to 1/min(*n*,*m*) in the worst case, or alternatively speedup requirements equal to min(*n*,*m*) (assuming the architecture version were optics reside in the internal of the switch) in order to ensure 100% throughput in all cases. Assuming *S*=1, throughput more than 80% can be achieved for all average traffic cases examined. Based on our analysis we also proposed alternative switch architectures for HPC and DC countering the performance degradation in the worst case traffic patterns in the initial approach and we discussed the trade-offs between performance and additional hardware requirements.

Appendix

A. Theoretical Approximations for Throughput and Speedup (*D* with i.i.d. entries)

Let us define N^2 i.i.d. discrete random variables: $X_{i,j} \sim U[a,\ b]$, $i, j \in [1,2,\ldots,N]$, representing the entries located in row i and column j of traffic matrix D (with mean $\mu=(a+b)/2$ and variance $\sigma^2=[(b-a+1)^2-1]/12$). Let $Y_1,Y_2,\ldots,Y_{2N}$ be random variables, with variables $Y_1,Y_{2,..,}Y_N$ representing the sums of rows and variables $Y_{N+1},Y_{N+2,..,}Y_{2N}$ representing the sum of columns of traffic matrix D. Then, the critical sum of traffic matrix D (non-blocking switch case) is

$$T_{min} = Y_{max} = \max(Y_1, Y_2 \ldots, Y_{2N}) \qquad (2)$$

If we denote the CDF (cumulative distribution function) of a specific Y_i as $F_Y(y) = P(Y_i \leq y)= P(Y \leq y)$, then the CDF of T_{min} is:

$$F_{T_{min}}(y) = P(Y_{max} \leq y) = P(Y_1 \leq y, Y_2 \leq y, \ldots, Y_{2N} \leq y) \leq$$
$$\leq P(Y_1 \leq y, Y_2 \leq y, \ldots, Y_N \leq y) \cdot P(Y_{N+1} \leq y, Y_{N+2} \leq y, \ldots, Y_{2N} \leq y) =$$
$$= P(Y_1 \leq y)P(Y_2 \leq y) \ldots P(Y_{2N} \leq y) = F_Y(y)^{2N} \qquad (3)$$

The equality would hold if $Y_1,Y_2,\ldots,Y_{2N}$ were mutually i.i.d. The row sums $Y_1,Y_{2,..,}Y_N$ are mutually i.i.d as well as the column sums $Y_{N+1},Y_{N+2,..,}Y_{2N}$, but variables belonging to one set are not independent from variables of the other set. The covariance between the sum of a single row Y_r, $r \in [1,2,\ldots,N]$ and the sum of a single column Y_c, $c \in [N+1, N+2,\ldots,2N]$ is $cov(Y_r, Y_c) = E(Y_r Y_c) - E(Y_r)E(Y_c) = E\left(X_{r,c-N}^2\right) - E(X_{r,c-N})^2 = \sigma^2 = [(b-a+1)^2 - 1]/12$. The respective correlation coefficient is $\rho_{Y_r,Y_c} = \frac{cov(Y_r,Y_c)}{\sigma_{Y_r} \cdot \sigma_{Y_c}} = \frac{\sigma^2}{\sqrt{N\cdot\sigma^2}\cdot\sqrt{N\cdot\sigma^2}} = \frac{1}{N}$. It is well known that the probability distribution of the sum of N i.i.d uniform variables can be approximated by a normal distribution $\mathcal{N}(N \cdot \mu, N \cdot \sigma^2)$ where μ and σ^2 are the mean and variance of the uniform variables. Already for $N = 4$ the difference between the normal approximation and the exact distribution is often negligible [41]. Thus, $Y_1,Y_2,\ldots,Y_{2N}$, for $N \geq 4$ can be viewed as normally distributed variables. According to an important result [41], the condition

$$\lim_{n\to\infty} \rho_n \ln n = 0 \qquad (4)$$

for stationary standard normal random variables $Y_1,Y_2,\ldots,Y_n$ with $\rho_n=\mathrm{cov}(Y_0,\ Y_n)$ and $Y_{max}=\max(Y_1,Y_2,\ldots,Y_n)$, implies that the asymptotic distribution of Y_{max} behaves as if Y_i, $i \in [1,\ldots,n]$ were i.i.d. random variables. In our case

$$\lim_{2N\to\infty} \frac{1}{N} \ln 2N = \lim_{N\to\infty} \frac{2\ln N}{N} = 0$$

Thus, in principle, for large N we can treat $Y_1,Y_2,\ldots,Y_{2N}$ as mutually i.i.d. with $F_Y(y) = P(Y_i \leq y)= P(Y \leq y)$, $i \in [1,\ldots,2N]$ where

$$Y = X_1 + X_2 + \ldots + X_N \qquad (5)$$

and $X_j \sim U[a,\ b]$, $j \in [1,\ldots,N]$ are i.i.d. and we can also take the equality in (3), hence $F_{T_{min}}(y) = F_Y(y)^{2N}$. Then, the PDF (Probability Density Function) of T_{min} is $f_{T_{min}}(y) = 2N \cdot f_Y(y) \cdot F_Y(y)^{2N-1}$. Then, by definition,

$$E(T_{min}) = 2N \int_{-\infty}^{\infty} y \cdot f_Y(y) \cdot F_Y(y)^{2N-1} dy.$$

Following the same reasoning we can estimate the critical sum T'_{min} of traffic matrix D' (WS-mSS switch), assuming i.i.d. variables in every step. Consider random variable Y' defined as:

$$Y' = X_1 + X_2 + \ldots + X_m \qquad (6)$$

where X_1, $X_{2,}$ …, X_m are i.i.d. variables obtained from the Uniform Distribution as before ($U[a,b]$). Then we define random variable X' as:

$$X' = Y'_{max} = \max(Y'_1, Y'_2 \ldots, Y'_{2m}) \qquad (7)$$

(where Y', $Y'_1,\ldots,Y'_{2m}$ are all i.i.d.) representing the critical sum of a single *m*x*m* element. Then, in a similar way we define:

$$Y'' = X'_1 + X'_2 \ldots + \ X'_n \qquad (8)$$

and finally

$$T'_{min} = Y''_{max} = \max(Y''_1, Y''_2 \ldots, Y''_{2n}) \qquad (9)$$

In order to estimate scheduling performance we would like to estimate the expected values of T'_{min} and T_{min} in order to obtain required speedup and maximum throughput:

$$S(D) = E(T'_{min})/E(T_{min}) \text{ and } \Theta(D) = E(T_{\min})/E(T'_{min}) \qquad (10)$$

The probability distribution of the sum of i.i.d. random variables [Eq. (5), (6), (8)] can be obtained by convoluting the probability distributions in pairs [43]. The probability distributions of Eq. (2), (7), (9) can be calculated using order statistics (the largest order statistic) [44]. We developed a script using Matlab performing all necessary convolutions, the largest order statistics calculations for Eq. (2), (3) and (5)-(10) and the calculation of the expected values of T'_{min} and T_{min}. Figure 6 and Table 2 (Section 4.2) present the obtained results for T_{min} and T'_{min} for a 48×48 non-blocking switch and the PhoxTrot switch for various cases for *a* and *b*, obtained via both Monte-Carlo estimation, as well as the theoretical approximation of Eq. (2)-(10). Note that the estimation of T'_{min} using Eq. (9) leads to a slightly bigger error for $E(T'_{min})$ compared to the estimation for $E(T_{min})$. This is due to the fact that $N=n{\cdot}m$ and *m* and *n* are small in the examined case. For example, since $n=4$, by using Eq. (9) where we treat $Y''_1,\ldots,Y''_{2n}$ as i.i.d. we tend to overestimate $E(T'_{min})$.

B. Closed Form Approximations for Throughput and Speedup (*D* with i.i.d. entries)

In this section we give closed form approximations for $E(T_{min})$, $E(T'_{min})$ and thus for Eq. (10), treating the sums of traffic matrices's lines as i.i.d variables in every step as in the theoretical approximation of the previous section. We also present the obtained values for the same cases examined in Section 4.2 using Eq. (10). Based on these closed form approximations we estimate the asymptotic values for $S(D)$ and $\Theta(D)$ for large values of *b* (and a=1).

The probability distribution of the sum of *N* i.i.d uniform variables can be approximated by a normal distribution as mentioned in Appendix A. A handy and exact closed form formula for the maximum of normal variables cannot be easily obtained. The largest order statistic of *N* normal random variables has a probability distribution also known as power normal distribution [45] (presented there for standard normal variables). The expected value for power normal distribution is calculated there recursively using function:

$$L_{2n+1}(\lambda) = \sum_{i=1}^{2n+1} (-1)^{i+1} \binom{2n+1}{i} \frac{1}{2^i} L_{2n+1-i}(\lambda)$$

where

$$L_n(\lambda) = \int_{-\infty}^{\infty} [\Phi(\lambda x)]^n \, \varphi(x) dx$$

Below we give two approximations for the expected value of Z_{max}=max(Z_1,Z_2,…,Z_n) where Z_i, $i \in [1,2,\ldots,n]$ are i.i.d. standard normal variables, and one approximation for the variance and then we apply them to derive closed form formulas for $E(T_{min})$ and $E(T'_{min})$. Given the closed form formulas for the expected value and variance of standard normal variables Z_i, the respective formulas for normally distributed variables $X_i \sim \mathcal{N}(\mu,\sigma^2)$, where X_i=σ·Z_i+μ can be easily obtained. Naturally, X_{max}=σ·Z_{max}+μ. From basic properties of expected value and variance we can obtain $E(X_{max})$=$E(\sigma \cdot Z_{max}+\mu)$=σ·$E(Z_{max})$+$\mu$ and Var(X_{max})=Var($\sigma \cdot Z_{max}+\mu$)= σ^2·Var(Z_{max}).

EVT approximation

An easy to use approximation for expected value and variance for the maximum of i.i.d. standard normal variables can be obtained by Extreme Value Theory (EVT). It is known that standard normal distribution is in the max-domain of attraction of the Gumbel distribution. If Z_{max}=max(Z_1,Z_2,…,Z_n) where Z_i, $i \in [1,2,\ldots,n]$ are i.i.d. standard normal variables, then an approximation (underestimation) for expected value can be obtained by [46]:

$$E(Z_{max}) = \sqrt{2 \cdot \ln n} \cdot \beta(n) \qquad (11)$$

where:

$$\beta(n) = \left[1 - \frac{\ln(4\pi \cdot \ln n)}{4 \cdot \ln n}\right]$$

An approximation for variance with accuracy within 5% for n>10 is [46]:

$$Var(Z_{max}) = \sigma^2_{Z_{max}} = \frac{\pi^2 \cdot n}{12 \cdot \ln n} \qquad (12)$$

Eq. (11) gives for $E(T_{min})$:

$$E(T_{min}) = \sigma\sqrt{2N \cdot \ln 2N} \cdot \beta(2N) + N \cdot \mu \qquad (13)$$

since variables Y_1,Y_2,…,Y_{2N} can be approximated as normally distributed with $Y_i \sim \mathcal{N}(N \cdot \mu, N \cdot \sigma^2)$, $i \in [1,2,\ldots,2N]$ where μ=(a+b)/2 and σ^2=[(b – a + 1)^2-1]/12. Similarly, $E(X')$ can be obtained by

$$E(X') = \sigma\sqrt{2m \cdot \ln 2m} \cdot \beta(2m) + m \cdot \mu \qquad (14)$$

since $Y'_i \sim \mathcal{N}(m \cdot \mu, m \cdot \sigma^2)$. Variance of X' can be approximated by Eq. (12):

$$\sigma^2_{X'} = \frac{\pi^2 \cdot m \cdot \sigma^2}{12 \cdot \ln 2m} \qquad (15)$$

As discussed above, X' [Eq. (7)], as well as T_{min}, are power normal variables. Power normal distribution is actually a skewed normal distribution [45]. We approximate X' with a normal distribution of mean $E(X')$ and variance $\sigma^2_{X'}$. Thus, Y'' which is a sum of n such random variables can be approximated by a normal distribution, $Y'' \sim \mathcal{N}(n \cdot E(X'), n \cdot \sigma^2_{X'})$. Hence, using Eq. (11) again we get:

$$E(T'_{min}) = \sigma_{X'} \cdot \sqrt{2n \cdot \ln 2n} \cdot \beta(2n) + n \cdot E\left(X'\right) \qquad (16)$$

where $E\left(X'\right)$ and $\sigma^2_{X'}$ are given by Eq. (14), (15).

Jensen's upper bound

An upper bound for $E(Z_{max})$ can be also obtained by Jensen's inequality [47] (p. 40), which states that for f convex:

$$E(f(x)) \geq f(E(x))$$

Jensen's equality for $e^{t\cdot Z_{max}}$ gives:

$$e^{t\cdot E(Z_{max})} \leq E(e^{t\cdot Z_{max}}) = E\left(\max_i e^{t\cdot Z_i}\right) \leq \sum_{i=1}^{n} E(e^{t\cdot Z_i}) = n \cdot e^{\frac{t^2}{2}}$$

where $E(e^{t\cdot Z_i})$ is the moment generating function of Z_i. By taking natural logarithms we get:

$$E(Z_{max}) \leq \frac{\ln n}{\mathrm{t}} + \frac{t}{2}$$

The minimum t for which this inequality holds is $t = \sqrt{2 \cdot \ln n}$. Subtituting above, gives:

$$E(Z_{max}) \leq \sqrt{2 \cdot \ln n} \qquad (17)$$

Using the same reasoning as before, $E(T_{min})$ and $E(T'_{min})$ can be approximated combining the upper bound from Jensen's equation [Eq. (13)] and Eq. (12). Eq. (17) gives for $E(T_{min})$:

$$E(T_{min}) \leq \sigma\sqrt{2N \cdot \ln 2N} + N \cdot \mu \qquad (18)$$

For $E(T'_{min})$ we have:

$$E(T'_{min}) \leq \sigma_{X'} \cdot \sqrt{2n \cdot \ln 2n} + n \cdot E\left(X'\right) \qquad (19)$$

where $\sigma^2_{X'}$ is given by Eq. (16) and $E(X')$ is given by:

$$E\left(X'\right) \leq \sigma\sqrt{2m \cdot \ln 2m} + m \cdot \mu \quad (20)$$

In Table 3 we present the values for $E(T'_{min})$, $E(T_{min})$, $S(D)$ and $\Theta(D)$ obtained using Eq. (13), (16) (EVT approximation) and Eq. (18), (19) (Jensen's upper bound) for the same cases of Table 2.

Table 3 : $E(T'_{min})$, $E(T_{min})$, $S(D)$ and $\Theta(D)$ for all-to-all traffic and a=1, estimated by Eq. (14), (17) (EVT) as well as Eq. (18), (19) (Jensen).

	b	$E(T'_{min})$	$E(T_{min})$	$S(D)$	$\Theta(D)$
EVT Approx.	2	86.58	80.15	1.08	0.93
	9	315.13	282.06	1.117	0.90
	25	833.81	741.47	1.125	0.89
	50	1643.9	1459.1	1.127	0.89
	500	16224	14375	1.129	0.89
Jensen	2	93	82.47	1.128	0.89
	9	348.44	294	1.185	0.84
	25	926.9	774.95	1.196	0.84
	50	1830.1	1526.1	1.199	0.83

500	18086	15045	1.202	0.83

Asymptotic values for *S*(*D*) and *Θ*(*D*)

Using the above approximations we can estimate the asymptotic values for *S*(*D*) and *Θ*(*D*) for large values of *b* (and *a*=1). Using Eq. (10), (13), (16), and substituting *a*=1 we get:

$$S(D)=\lim_{b\to\infty}\frac{E(T'_{min})}{E(T_{min})}=\frac{\sqrt{\dfrac{\pi^2\cdot N\cdot\ln2n}{6\cdot\ln2m}}\cdot\beta(2n)+n\cdot\sqrt{2m\cdot\ln2m}\cdot\beta(2m)}{\sqrt{2N\cdot\ln2N}\cdot\beta(2N)}\qquad(21)$$

For the PhoxTrot numbers (*m*=12, *n*=4 and *N*=48), Eq. (21) yields *S*(*D*)≈1.79 and *Θ*(*D*)≈0.56. Using Eq. (10), (18), (19), and substituting *a*=1 we get:

$$S(D)=\lim_{b\to\infty}\frac{E(T'_{min})}{E(T_{min})}=\frac{\sqrt{\dfrac{\pi^2\cdot N\cdot\ln2n}{6\cdot\ln2m}}+n\cdot\sqrt{2m\cdot\ln2m}}{\sqrt{2N\cdot\ln2N}}\qquad(22)$$

For *m*=12, *n*=4 and *N*=48, Eq. (22) yields *S*(*D*)≈2.01 and *Θ*(*D*)≈0.50. Therefore, 1.79≤*S*(*D*)≤2.01 and 0.5≤*Θ*(*D*)≤0.56. Note that the approximations we presented in this and the previous section are the same (if not more accurate) if variables $X_{i,j}$ were normally distributed in the first place.

Acknowledgments This work was supported by the European Commission through the FP7 ICT-PHOXTROT (ICT 318240) project.

References

[1] Cisco Global Cloud Index: Forecast and Methodology, 2014–2019 White Paper.

[2] Make IT Green: Cloud Computing and its Contribution to Climate Change. Greenpeace International, 2010.

[3] M. Haney, N. Rohit and T. Gu., "Chip-scale integrated optical interconnects: a key enabler for future high-performance computing", SPIE OPTO. International Society for Optics and Photonics, 2012.

[4] M.A. Taubenblatt, "Optical Interconnects for High-Performance Computing", JLT 30(4), 2012, pp.448,457.

[5] C. Kachris and I. Tomkos, "Power consumption evaluation of all-optical data center networks", Cluster Computing 16.3, 2013, pp. 611-623.

[6] A.Liu, L. Liao, Y. Chetrit, J. Basak, H. Nguyen, D. Rubin and M. Paniccia, "Wavelength division multiplexing based photonic integrated circuits on silicon-on-insulator platform", IEEE JSTQE 16.1, 2010, pp.23:32.

[7] S. Han, T. J. Seok, N. Quack, B. Yoo and M. C. Wu, "Monolithic 50x50 mems silicon photonic switches with microsecond response time", OFCC, Optical Society of America, 2014.

[8] L. Lu, L. Zhou, Z. Li, X. Li and J. Chen, "Broadband 4x4 Nonblocking Silicon Electrooptic Switches Based on Mach–Zehnder Interferometers", Photonics Journal, IEEE 7.1, 2015, pp. 1-8.

[9] B. G. Lee, N. Dupuis, P. Pepeljugoski, L. Schares, R. Budd, J. R. Bickford and C. L. Schow, "Silicon photonic switch fabrics in computer communications systems", Journal of Lightwave Technology 33.4, 2015, pp. 768-777.

[10] R. Stabile, A. Albores-Mejia, A. Rohit and K. A. Williams, "Integrated optical switch matrices for packet data networks", Microsystems & Nanoengineering 2, 2016, 15042.

[11] L. Qiao, W. Tang, and T. Chu, "16×16 Non-blocking Silicon Electro-optic Switch Based on Mach-Zehnder Interferometers", in Optical Fiber Communication Conference, OSA Technical Digest (online), Optical Society of America, 2016, Th1C.2.

[12] D. Nikolova, S. Rumley, D. Calhoun, Q. Li, R. Hendry, P. Samadi and K. Bergman, "Scaling silicon photonic switch fabrics for data center interconnection networks", Optics express 23.2, 2015 , pp. 1159-1175.

[13] B. G. Lee, A. V. Rylyakov, W. M. J. Green, S. Assefa, C. W. Baks, R. R.-Donadio, D.l M. Kuchta, M. H. Khater, T. Barwicz, C. Reinholm, E. Kiewra, S. M. Shank, C. L. Schow and Y. A. Vlasov, "Monolithic

Silicon Integration of Scaled Photonic Switch Fabrics, CMOS Logic, and Device Driver Circuits", Journal of Lightwave Technology, vol. 32, no. 4, Feb.15, 2014, pp. 743-751.

[14] L. Chen, E. Hall, L. Theogarajan and J. Bowers, "Photonic switching for data center applications", IEEE Photonics journal 3.5, 2011, pp.: 834-844.

[15] S. Papaioannou, K. Vyrsokinos, O. Tsilipakos, A. Pitilakis, K. Hassan, J.-C. Weeber, L. Markey, A. Dereux, S. I. Bozhevolnyi, A. Miliou, E. E. Kriezis and N. Pleros, "A 320 Gb/s-Throughput Capable 2x2 Silicon-Plasmonic Router Architecture for Optical Interconnects", in Journal of Lightwave Technology, vol. 29, no. 21, 2011, pp. 3185-3195.

[16] A. Håkansson, T. Tekin, L. Brusberg, N. Pleros, C. Vyrsokinos, D. Apostolopoulos, R.Pitwon, A. Miller, K. Wang, D. Tulli, S. Dorrestein, R. Smink, J. Tuin, M. Rijnbach and J. Duis, "PhoxTroT-a European initiative toward low cost and low power photonic interconnects for data centres", ICTON, 2015.

[17] http://www.phoxtrot.eu

[18] O. Liboiron-Ladouceur, I. Cerutti, Pier G. Raponi, N. Andriolli and P. Castoldi, “Energy-efficient design of a scalable optical multiplane interconnection architecture”, IEEE Journal of Selected Topics in Quantum Electronics, no. 99, 2010, pp. 1–7.

[19] R. Luijten, W. E. Denzel, R. R. Grzybowski and R. Hemenway, "Optical interconnection networks: The OSMOSIS project", The 17th Annual Meeting of the IEEE Lasers and Electro-Optics Society, 2004, Vol. 18.

[20] W. Kabacinski, “Nonblocking electronic and photonic switching fabrics”, Springer Science & Business Media, 2005.

[21] M. Karol, M. Hluchyj and S.P. Morgan, "Input versus output queueing on a space-division packet switch." Communications, IEEE Transactions on 35.12, 1987, pp. 1347-1356.

[22] B. Prabhakar and N. McKeown, "On the speedup required for combined input-and output-queued switching." Automatica 35.12, 1999, pp. 1909-1920.

[23] W. Dally and B. Towles, "Principles and practices of interconnection networks", Elsevier, (2004).

[24] N. McKeown, A. Mekkittikul, V. Anantharam and J. Walrand. "Achieving 100% throughput in an input-queued switch." Communications, IEEE Transactions on 47.8, 1999, pp. 1260-1267.

[25] S.-T. Chuang, A. Goel, N. McKeown and B. Prabhakar, "Matching output queueing with combined input output queued switches", IEEE JSAC, vol. 17, no. 6, 1999, pp. 1030-39.

[26] Y. Wang, S. S. Djordjecvic, J. Yao, J. E. Cunningham, X. Zheng, A. V. Krishnamoorthy, M. Muller, M.-C. Amann, R. Bojko, N. A. F. Jaeger and L. Chrostowski, "Vertical-cavity surface-emitting laser flip-chip bonding to silicon photonics chip," 2015 IEEE Optical Interconnects Conference (OI), San Diego, CA, 2015, pp. 122-123.

[27] C. Xie, S. Spiga, P. Dong, P.J. Winzer, M. Bergmann, B. Kögel, C. Neumeyr, and M. Amann, "Generation and Transmission of a 400-Gb/s PDM/WDM Signal Using a Monolithic 2x4 VCSEL Array and Coherent Detection", in Optical Fiber Communication Conference: Postdeadline Papers, (Optical Society of America, 2014), paper Th5C.9.

[28] A. V. Rylyakov, C. L. Schow, J. E. Proesel, D. M. Kuchta, C. Baks, N. Y. Li, C. Xie and K. P. Jackson, "A 40-Gb/s, 850-nm, VCSEL-based full optical link", Optical Fiber Communication Conference (OFC), Optical Society of America, 2012.

[29] Cisco Nexus 5548P Switch. Datasheet, Cisco Inc. (2016).

[30] J. Hopcroft and R.M. Karp, "An n^5/2 algorithm for maximum matchings in bipartite graphs", SIAM Journal on computing 2.4, 1973, pp. 225-231.

[31] N. McKeown, "The iSLIP scheduling algorithm for input-queued switches", Networking, IEEE/ACM Transactions on 7.2, 1999, pp. 188-201.

[32] T. E. Anderson, S. S. Owicki, J. B. Saxe and C. P. Thacker, "High-speed switch scheduling for local-area networks", ACM Transactions on Computer Systems (TOCS) 11.4, 1993, pp. 319-352.

[33] Y. Tamir and H. Chi, "Symmetric crossbar arbiters for VLSI communication switches", Parallel and Distributed Systems, IEEE Transactions on 4.1, 1993, pp. 13-27.

[34] H. J. Ryser, "Combinatorial Mathematics", Rahway, NJ: The Mathematical Association of America, 1965.

[35] A. Bhatelé, G. R. Gupta, L. V. Kale˙ and I-Hsin Chung, "Automated mapping of regular communication graphs on mesh interconnects", International Conference on High Performance Computing, IEEE, 2010.

[36] C. Chang, D. Lee, and Y. Jou, "Load balanced Birkhoff-von Neumann switches", High Performance Switching and Routing, 2001 IEEE Workshop on. IEEE, 2001.

[37] H. Chao and B. Liu, “High Performance Switches and Routers”, Hoboken,NJ, USA: John Wiley & Sons, Inc., 2007.

[38] H. Lee, “A two-stage switch with load balancing scheme maintaining packet sequence”, IEEE Commun. Lett., vol. 10, no. 4, 2006.

[39] C. Chang, D. Lee, and Y. Shih, “Mailbox switch: A scalable two-stage switch architecture for conflict resolution of ordered packets”, in IEEE INFOCOM, 2004, 2004, pp. 1995–2006.

[40] I. Keslassy and N. McKeown, “Maintaining packet order in two-stage switches”, in IEEE INFOCOM, 2002, pp. 1032–1041.
[41] F. Killmann and E. Collani, "A note on the convolution of the uniform and related distributions and their use in quality control", Economic Quality Control 16.1, 2001, pp. 17-41.
[42] S. Berman, "Limit theorems for the maximum term in stationary sequences." The Annals of Mathematical Statistics, 1964, pp. 502-516.
[43] C. M. Grinstead, and J. L. Snell, "Introduction to probability", American Mathematical Soc., 2012.
[44] G.Casella and R. L. Berger, "Statistical inference", Vol. 2, Pacific Grove, CA: Duxbury, 2002.
[45] R. D. Gupta and R. C. Gupta, "Analyzing skewed data by power normal model", Test 17.1, 2008, pp. 197-210.
[46] A. Stuart and K. Ord, “Kendall's Advanced Theory of Statistics”, Vol. I, Sixth Ed, Arnold, London, 1994.
[47] S. M. Ross, “Stochastic processes”, Vol. 2. New York: John Wiley & Sons, 1996.