A Neural Network-Based Automatized Management of $N \times N$ Integrated Optical Switches

Ihtesham Khan[1], **M Umar Masood**[1], **Lorenzo Tunesi**[1], **Paolo Bardella**[1], **Enrico Ghillino**[2], **Andrea Carena**[1], **Vittorio Curri** [1]

(1) Politecnico di Torino, Corso Duca degli Abruzzi, 24, 10129, Torino, Italy
(2) Synopsys, Inc., 400 Executive Blvd Ste 101, Ossining, NY 10562, United States
ihtesham.khan@polito.it

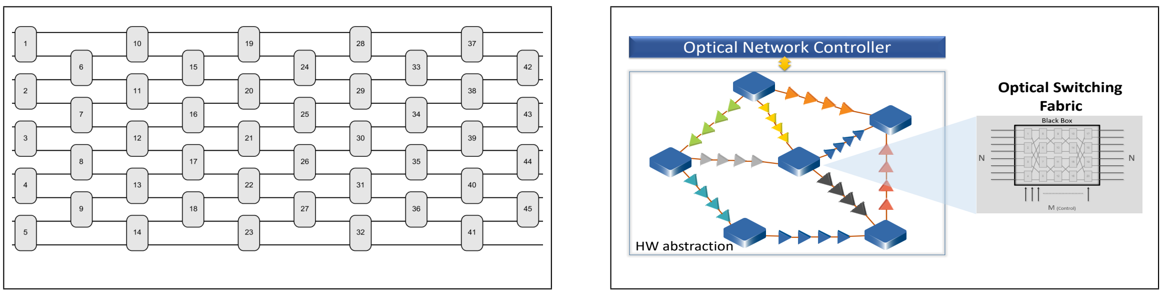
Abstract: We propose a neural network-based automatic management system for $N \times N$ optical switch in the context of a software-defined network. The proposed automatic softwarized system operates in a completely agnostic manner in real-time. © 2021 The Author(s)

1. Introduction

Recently a remarkable growth had been observed in exploiting Photonic integrated circuits (PICs) to perform complex functions at the photonic level, avoiding the bottleneck of optoelectronic conversion. Today PICs are a promising technology for next generation switching systems, due to their low energy consumption, lower latency and small footprint. These PICs attributes are highly required, particularly in core optical networks and data centres where high-speed data exchange is fundamental. This increasing application of large-scaled PIC-based switching devices demands automatic management systems. In this scenario, Software-defined networking (SDN) becomes an essential paradigm for the softwarized and automatize management of PIC-based switches.

Typically, PIC-based reconfigurable optical switches rely on the principle that the flow of light can be maneuvered by electrically controlled elements, like Mach-Zehnder (MZ) interferometers [1] or optical micro ring resonators (MRRs) [2]. A generic $N \times N$ photonic switch can be fabricated by combining multiple switching elements in different topologies, where N input signals at multiple wavelengths (λ_1, λ_2, λ_3....λ_N) are routed to any N output port with different wavelength combination (λ_2, λ_3, λ_1....λ_N) [3]. Some unique properties characterize each topological configuration: basic switching element size, non-blocking routing, minimization of optical losses, planarity, reduction of the circuit footprint and the operational power consumption [4]. Most of non-blocking optical switching networks are based on multistage crossover structures, with 2×2 crossbar switches as basic switching elements, which can be piloted through M control signals, toggling between the two switching states ($M = 1$ Cross-state ($[0,1] \to [1,0]$) and $M = 0$ Bar-state ($[0,1] \to [0,1]$)). These structures are typically based on the Clos network or the Banyan switch approach, which dictates the number of elements, stages, and interconnections.

One of the standard network topology for this class of networks is the Spanke-Beneš switch. This device is non-blocking rearrangeable, enabling the routing of all input permutations to the output ports. The main characteristic of this switch is the inter-stage planarity: while the Beneš, Clos or general Banyan networks usually relies on crossing interconnects between the stages, in the Spanke-Beneš the links between the 2×2 elements do not overlap with one another, as shown in Fig. 1a. The planarity achieved in this structure comes with a trade-off with respect to alternative implementations, as the number of switches required grows as $O(N^2)$ with respect to the Beneš structure, with a $O(N \cdot \log(N))$ dependency. This leads to a rapid increase in the number of control states available in the network $N_{st} = O(2^{N^2})$, making brute-force analysis and generalized topology-agnostic deterministic routing algorithms ineffective in tackling the problem complexity, especially concerning the best available path selection, which is the next step in the evolution of the proposed method. Due to this characteristic, the Spanke-Beneš

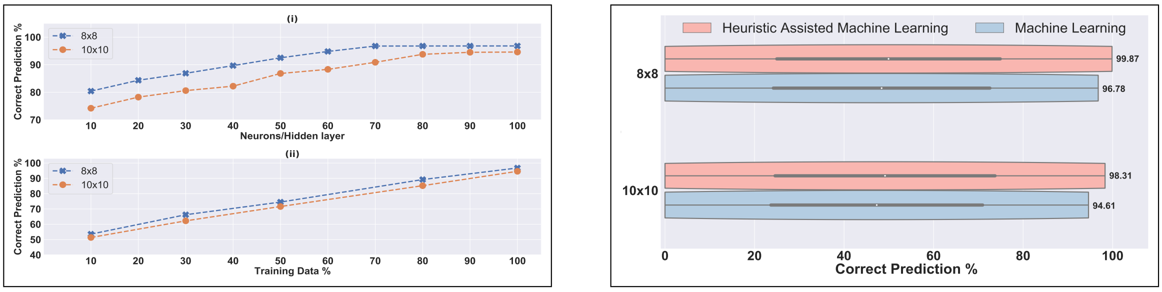


(a) (b)

Fig. 1: Examples of 10×10 Spanke-Beneš network (a), Software-defined Open optical networks (b).

Table 1: Dataset Statistics

Spanke-Benes size $N \times N$	Permutations $N!$	Number of switches M	Combinations 2^M	Dataset	Train Set	Test Set	Neurons per hidden-layer
8×8	40,320	28	268,435,456	300,000	210,000	90,000	70
10×10	3,628,800	45	35,184,372,088,832	1,000,000	700,000	300,000	90



(a) (b)

Fig. 2: Neural Network analysis (a), Neural Network prediction results (b).

network poses an effective benchmark in evaluating the effectiveness of a stochastic Machine learning (ML) based approach.

In this work, we propose to train a topology agnostic ML agent to predict control signal states of $N \times N$ optical switching system. The proposed approach is adopted in an SDN context which softwarized and automatized the configuration of any PIC-based optical switching system depicted in Fig. 1b. Given a $N \times N$ photonic switch with any arbitrary internal topology, the proposed approach undertakes it as a black-box component and tries to determine control states with a high level of accuracy.

2. Results & Conclusion

To generate the ML training and testing dataset, we considered a $N \times N$ Spanke-Beneš network. To demonstrate the proposed approach's scalability, two cases: $N = 8$ and 10; corresponding to the configurations with $M = 28$ and 45, internal switches are addressed. A subset of the total 2^M control combinations are used for the generation of the dataset, as reported in Tab. 1. The dataset is used to train a supervised neural network in the learning phase. The proposed ML model uses a *Deep neural network* (DNN), developed by using the TensorFlow© platform: it incorporates *three* hidden-layers. The proposed DNN model exploited *ReLU* as an activation function, and it is evaluated by mean square error (MSE) as a loss function. The DNN model is configured for training-steps of *4,000* and a learning rate of *0.01*. The train set for both networks consists of 70% of the dataset, while the test set uses the remaining 30%, as reported in Tab. 1. Moreover, the proposed ML agent utilizes wavelengths at the output ports as features and M control signals as labels.

The results in Fig. 2a(i) reveal the effect of increasing the number of neurons per hidden layer. An improvement in the ML model's prediction ability is observed, up to a certain extent, after which it remains constant. Along with this, in Fig. 2a(ii), the effect of the total considered training data size reported in Tab. 1 is also revealed. The trend shows that the prediction ability of the ML model improves with the increasing training data size. The rate of correct prediction is summarized for both two considered Spanke-Beneš sizes by the blue bars in Fig. 2b. We observe an excellent preliminary level of accuracy (>94%) for both the cases: $N = 8$, and 10. To further improve the ML approach's prediction capabilities, we added an auxiliary step based on a simple heuristic that we developed from observing wrong configurations. In most of the wrong prediction cases, the correct control sequence has a single switch element in an incorrect state. The proposed heuristic tries to correct a single ring error by flipping one switch simultaneously and comparing the output sequence against the desired output. For Spanke-Beneš 8×8 and 10×10, using ML assisted by heuristic, the accuracy improves to 99.87% and 98.31%.

In conclusion, we have demonstrated that a neural network can efficiently define control states for a generic $N \times N$ photonic switch without any knowledge required on the topology. The proposed approach is easily scalable to large N as a high level of accuracy is achieved with a limited size dataset.

References

1. K. Suzuki et.al, "Low-insertion-loss and power-efficient 32× 32 silicon photonics switch with extremely high-δ silica plc connector," JLT **37**, 116–122 (2019).
2. Q. Cheng et.al, "Ultralow-crosstalk, strictly non-blocking microring-based optical switch," Photonics Res. **7** (2019).
3. I. Khan et.al, "Machine learning assisted abstraction of photonic integrated circuits in fully disaggregated transparent optical networks," in *22nd ICTON,* (2020), pp. 1–4.
4. S. Arora et.al, "On-line algorithms for path selection in a nonblocking network," in *ACM symposium,* (1990).