Exploring Spiking Neural Networks for Prediction of Traffic Congestion in Networks-on-Chip

Aqib Javed*, Jim Harkin, Liam McDaid and Junxiu Liu
School of Computing, Engineering and Intelligent Systems,
Ulster University, Magee Campus, Derry, Northern Ireland, United Kingdom
*Contact: Javed-a@ulster.ac.uk

Abstract— **Networks-on-Chip (NoC) is the most modular and scalable solution for next generation hardware communication where significant data traffic loads are shared across many communication paths. One key challenge in maximising NoC performance is traffic congestion. The management of congestion at the earliest stage can significantly minimize the impact on NoC throughput. Prediction of NoC congestion offers a pre-emptive strategy in maximising NoC throughput. This paper proposes a novel spiking neural network (SNN) approach to prediction of traffic congestion. The proposed SNN exploits the temporal nature of the traffic to identify congestion patterns. The proposed SNN explores two models and both are trained and evaluated to predict local congestion 30 clock cycles in advance of occurring. Results shows that the SNN predictor utilizes 9 times less hardware area than previous approaches and can achieved up to 96.59% in accuracy.**

I. Introduction

The demand of computational intensive devices leads to the integration of more components in System-on-Chip (SoC). These many-core devices rely on shared communication paths for data transmission that can result in latency challenges [1]. Different on-chip interconnect solutions were proposed to optimize usage of shared network paths. Networks-on-Chip (NoC) is proposed as a scalable and modular communication architecture to provide multiple paths between cores and hence reduce network latency issues [2]. Depending on the application mapping and routing algorithm NoCs can support thousands of cores where it is facilitating huge communication workloads, that can ultimately cause congestion problems [3]. Quality of Service (QoS) is an important metric to validate NoC performance under different traffic status. In NoC, congestion can be produced due to non-uniform traffic routing, non-optimal flow control, inefficient traffic mapping and inappropriate network topologies. Congestion occurs at router level and can be handled locally (e.g. at router level) or globally (e.g. at network level). The management of congestion at the earliest stage can significantly minimize the impact on NoC throughput [4].

Inspired by the temporal computational capability of the human brain, neural networks are designed to perform brain related complex tasks i.e. data classification, pattern recognition [4], [5]. Traditional Artificial Neural Networks (ANNs) implicate weighted, rate-based computation to process information. However studies show that biological neurons communicate in the form of spikes, or action potential [6]. Therefore SNNs are proposed where the neurons communicate information temporally in the form of timing between spikes. SNNs encode neural information in both the spatial and temporal dimensions and require more computational power to process information as compared to non-spiking neural networks [7]. However, they perform temporal classification at a low cost in hardware given the advances in compact neural hardware implementations [8], [9]

NoCs generate temporal communication patterns while transporting packetized data traffic across the topology [10], [11]. The main advantage of the SNN over ANN is its ability to learn the temporal information with great precision [12]. Therefore, in this work an SNN based congestion prediction methodology is proposed to address NoC congestion. The scope of this work explores a cost effective SNN based prediction model with high prediction accuracy and low hardware overhead. The NoC congestion prediction is based on two levels a). Router level: Each router in the NoC has its own SNN to predict local congestion and 2). Network level: The entire NoC system has one SNN to predict local congestion for each router. The output of this work can be used in enhancing the traffic-load balancing of the NoCs, i.e. once the congestion is predicted, the SNN output can be processed by a congestion handling mechanism (e.g. adaptive routing [13]) to supress its effect before it can occur. The proposed SNN predictor can sense congestion 30 clock cycles in advance to provide enough time for a congestion handler to react.

Section II provides background on existing neural and non-neural congestion detection\prediction approaches. Section III reports on proposed SNN based NoC congestion prediction methodology and the experimental setup is outlined in section IV. Section V presents simulation results and section VI provides a conclusion and outlines of future work.

II. Background and Related Works

This section gives a brief introduction to cause and effects of congestion in NoC architectures, and presents an overview of existing congestion prediction research.

NoCs use routing algorithms to establish paths between source and destination nodes. Routing algorithms are responsible for traffic distribution of routing data. Ideally, NoC communication architecture is designed to distribute network traffic uniformly across network nodes. Because of application mapping and routing algorithm, data traffic pushes towards specific nodes. When traffic loads become significant it can lead to congestion as routing strategies have minimal time to adapt and often spare resources to compensate with path diversity [14]. Congestion is an important factor in on-chip performance degradation [15] and most NoC routers use buffer spaces to temporally store incoming data packets at input ports. NoC congestion occurs from inside router (arbitrator) to outside the router (buffer). Once input buffers are occupied, router stops receiving data to cause back-pressure towards neighbouring nodes [3](as shown in Fig. 1). Neighbouring nodes are force to keep data or find bypass path using adaptive algorithm. One promising solution is insertion of more buffering slots to compress back-pressure. These additional buffers will help to avoid congestion but cause high transmission delays. Data routed towards destination node can become part of congestion if it incur with on-path congested router. If congestion is not handled at earliest, the back-pressure will continue until the whole network get congested.

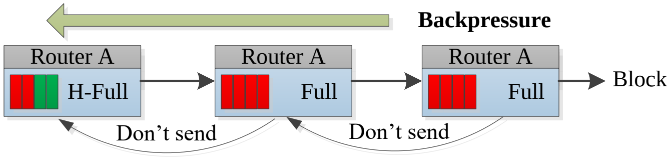


Fig.1. Effect of congestion and Backpressure.

Different techniques are developed to optimize effect of NoC congestion. These methods use buffer utilization levels, switch contention, traffic-flow patterns, traffic tables, task mapping etc. as a parameters to identify congestion[16]. Buffer utilization level is most appropriate and widely used congestion information parameter. An Output Buffer Length (OBL) selection function utilizes the next on-path router's output buffer occupancy information to decide the next hop[17]. The Neighbour-on-Path (NoP) process free slot information of neighbouring routers to identify the least congested routing path [17]. Some algorithms uses multiple information to process congestion. The Path-Congestion Aware Adaptive Routing (PCAR) and an upgraded Odd-Even adaptive routing algorithm both use switch contention along with buffer occupancy levels to route data through a least congested path [18]. Above mentioned techniques are reactive to congestion and only react when it detects on-path congestion.

Network performance (i.e. latency, throughput, QoS) can be enhanced by prior information about on-path congestion[15]. NoC congestion prediction is on-going research topic with only limited reported work to date [19].

A Traffic-Based Routing Algorithm (TBRA) is proposed to predict NoC traffic and dynamically select suitable adaptive routing algorithm to route predicted data [20]. Another motivation for traffic prediction is to optimize usage of nodes and channels thus minimizing operating power. A low-power Application Driven Traffic Pattern Table (ATPT) with small routing table in-builded inside router to record traffic flow from router [21]. Prediction using ATPT helps network to dynamically adjust voltage frequency to save up to 86% dynamic power. A predictive closed-loop flow control mechanism is proposed to predict traffic flow and to minimize NoC congestion by avoiding buffer overflow and packet drops [22]. Neural network are also involved to predict NoC traffic congestion. Artificial Neural Networking (ANN) model use buffer occupancy level to predict location of potential hotspot router with 65-92% accuracy on synthetic and real-time dataset [15]. An Evolving Fuzzy Neural Network (EFuNN), which is inspired by the combination of NNs and the fuzzy logic is proposed to predict congestion-free minimal path to improve network latency [2].

III. SNN-BASED NOC CONGESTION PREDICTION MODEL

NoC Congestion occurs with the concentration of routed data towards specific node(s). NoC performance is measured in terms of the number of flits per seconds, namely throughput. Path congestion increases transmission delays causing network latency and throughput issues. Queuing of data at router inputs is the foremost factor in performance degradation [23]. Research shows that input buffer queuing or buffer utilization data is the most effective way to identify congestion and can be used to predict local congestion [3]. Utilization data is highly dependent on the network architecture and application characteristics. This work proposes a novel SNN-based NoC congestion prediction model, at two router and network levels, using input buffer queuing information. .

A. SNN model

SNNs closely mimic biological neurons and transmit information in temporal patterns. This work considered the Leaky integrate and Fire (LIF) model with exponentially decaying (leaky) synaptic current. Spikeprop [24], a popular spiking counterpart of ANN's gradient methods is used as the learning algorithm for LIF based neural model. This paper contributes on the development, training and testing to validate the prediction coverage of the SNN models using traffic data from traced-based synthetic and real-world multimedia applications.

B. Congestion criteria

The congestion criteria is based on input buffer occupancy levels. Buffer occupancy level (buffer utilization) is a key indicator of congestion and can be viewed as temporal variations of data queuing patterns at router inputs. Fig 2 shows a 5-channel (east, west, north, south and core) NoC router with four input buffers spaces, where red depicts buffer occupancy level. For router X, (3, 2, 2, 3, 1) are the generated buffer occupancy patterns for North/West/South/East/Core ports.

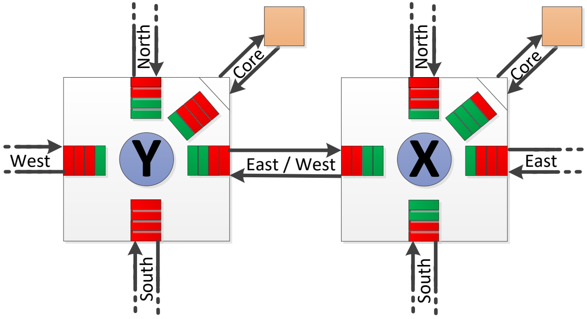
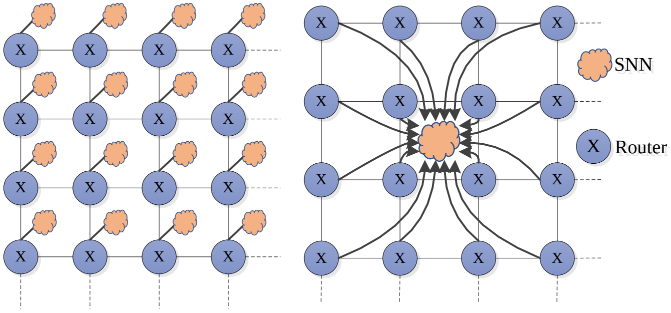


Fig. 2. Buffer utilization model with 4-buffer slots for each input (Green are free slots; Red are occupied slots)

Congestion Definition: *A router is deemed congested if the accumulated value of buffer occupancy levels is more than 60% of the total buffering slots in one router, and at least one buffer channel is fully occupied.* For example, Router Y has a total of 20 buffering slots and generates (2, 3, 4, 2, 3) patterns which occupy 14 slots in total, i.e. 14/20=70%. If the south port is also fully occupied then the router is labelled as congested and the generated pattern is classed as congested. Using the proposed congestion criteria we can generate congestion patterns that can be used for training of the SNN model to predict congestion 30 clock cycles in advance of its occurrence. This provides enough time to adapt and avoid or minimise the impact of congestion.

C. *Proposed prediction model*

The proposed SNN predication model collects the congested patterns at two different levels – one is the local *router* and other at the global *network* level. For the *router level*, the proposed model provides an individual SNNs for every router in the NoC, where buffer utilization data extracted from each router input is fed directly to SNN, and the SNN output defining the router congestion status. Fig. 3(a) shows the connections of the proposed model at *router* level. Since every router has one SNN, the SNN size depends on the NoC router location. For each SNN, the number of neurons at the input layer is same as the number of input channels of the NoC router. For example, a 4x4 2D NoC in Fig. 3(a) has 16 routers, where four corner routers have 3 inputs, eight routers in the edge are 4-inputs and four inner routers have 5-input channels.



(a) **(b)**

Fig 3. Proposed SNN prediction models (a) Router level and (b) Network level

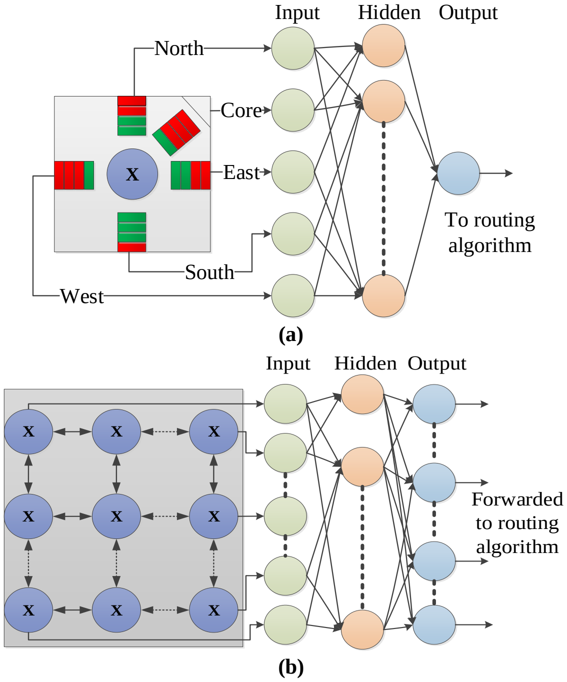
The proposed model also works at the *network level*, which uses one SNN for the whole NoC. Buffer utilization data extracted from each router is directly fed to the SNN. The buffer utilization data generated by input channels of each router are accumulated into a unified value to reduce the number of neurons at input layer of SNN. The size of the input and output SNN layers are identical to the total number of routers in the NoC. A 4x4 NoC in Fig. 3(b) depicts 16-input SNN model, where each input connects to one router. The SNNs at both levels are trained to predict congestion for each router and their performance evaluated on the basis of prediction accuracy.

IV. Experimental Setup

An experimental setup was established to verify the prediction coverage of the proposed SNN models using defined congestion criteria. This section explains the experimental processes used to perform simulations for the congestion prediction models. Simulation results of prediction performance and also expected hardware overhead are reported.

A. Simulation Environment and Setup

To evaluate the prediction model on a NoC, simulations of trace-based applications were performed using the NoC simulator Noxim [25]. Benchmarks adopted to evaluate the proposed prediction model is based on standard synthetic and real-time MPSoC applications. These benchmarks include transpose-1, transpose-2, shuffle, butterfly, Multi-Media Systems (MMS) and Moving Picture Experts Group-4 (MPEG-4). Application are mapped in Noxim on a 4x4 mesh based NoC and simulated using the standard XY-routing algorithm[26]. Each router generates and transmits data packets according to the application. Every routed data packet has eight flits, and each router channel can accumulate four data packets (32 flits) at the input buffer. Data is recorded for each router at every cycle and each simulation runs for 2,000 clocks with the first 1,000 clocks used as warm-up cycles. The generated traffic data is then used for training and testing of the proposed model.



(a)

(b)

Fig 4. SNN models at (a) Router level and (b) Network level.

To validate the prediction coverage, a 3-layer fully-connected LIF based SNN model with spikeprop as a learning algorithm is modelled and simulated in MATLAB for training and testing of proposed prediction models as shown in Fig 4. Typical router model utilizes 5x10x1 SNN (5, 10 and 1 for input, hidden and output layer neurons) for each NoC router whereas network model is connected to every NoC router through 16x30x16 SNN (16, 30 and 16 for input, hidden and output layer neurons). The output of SNN depicts the predicted congestion status of router. This can be forwarded to adaptive routing algorithms to avoid prospective network congestion. SNN is trained with 60% of simulated dataset and tested on 40% unseen dataset. Following results shows prediction accuracy and precision of proposed models on 40% unseen dataset.

B. Performance Analysis

To analyse prediction coverage of proposed congestion prediction methods, we considered two confusion matrix performance parameters: prediction accuracy (P_a) and prediction precision (P_p).

$$P_a = \frac{(\sum TP + \sum TN}{\sum (P + N)} \tag{1}$$

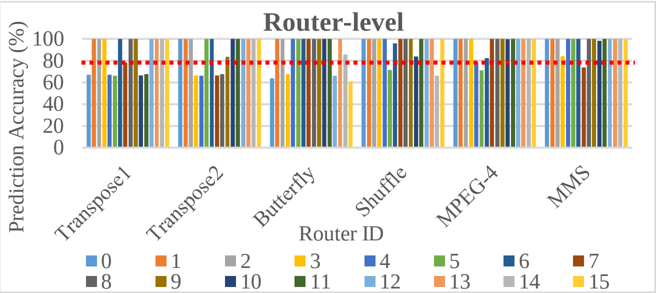


Fig 5. Prediction accuracy for router model

$$P_p = \frac{\Sigma TP}{\Sigma P} \qquad (2)$$

where congestion patterns are termed as positive (P) and non-congestion patterns are labelled as negative (N). TP and TN defines correct prediction of patterns (P) and (N) respectively. Simulation results are formulated in form of prediction accuracy and prediction precision of whole mesh network.

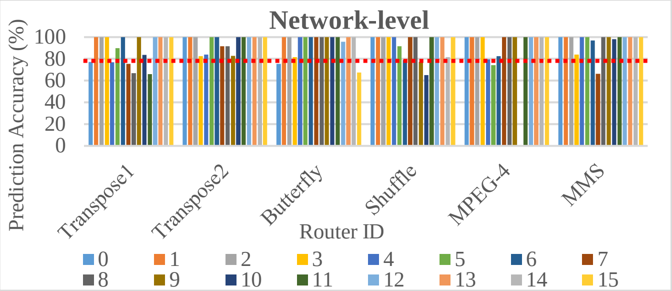
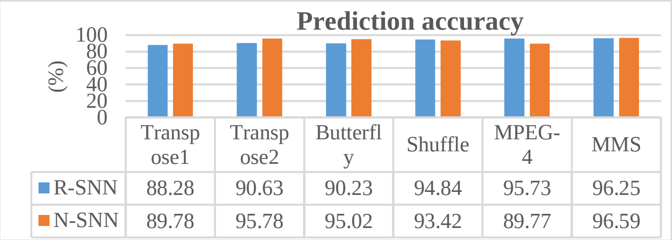


Fig 6. Prediction accuracy network model

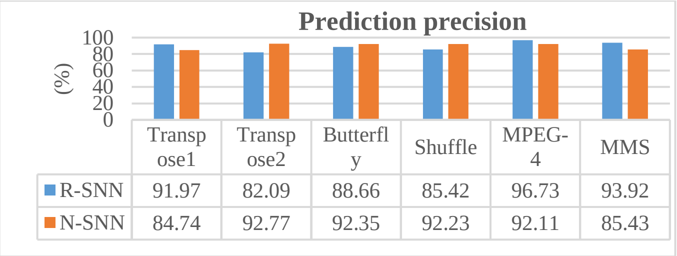
Fig. 5 and Fig. 6 presents prediction accuracy of local congestion for router and global level SNN models on synthetic and real-time applications respectively. Since, each router generate and route data packet according to mapped application and generate different patterns. Therefore, Fig. 5-6 explains the difference in prediction accuracy of proposed models for each local router. It is evident that some router shows 100% prediction accuracy whereas some routers shows ~65%. Results shows that network model have more routers with local prediction accuracy more than 80%.



	Transpose1	Transpose2	Butterfly	Shuffle	MPEG-4	MMS
R-SNN	88.28	90.63	90.23	94.84	95.73	96.25
N-SNN	89.78	95.78	95.02	93.42	89.77	96.59

Fig 7. Average prediction accuracy of proposed models

Fig. 7 shows prediction accuracy of overall network using proposed SNN models. Result shows that router-level SNN (R-SNN) with 88.28%-96.25% accurately as compared to network level SNN (N-SNN) predicted traffic patterns with 89.77%-96.59% accuracy. It is depicted that network level SNN performed exceptionally well in four applications except shuffle and MPEG-4 where router-level SNN predicted traffic patterns with better accuracy.



	Transpose1	Transpose2	Butterfly	Shuffle	MPEG-4	MMS
R-SNN	91.97	82.09	88.66	85.42	96.73	93.92
N-SNN	84.74	92.77	92.35	92.23	92.11	85.43

Fig 8. Average prediction precision of proposed models

An average network prediction precision for the router and network models are shown in Fig 8. Simulation results shows that the router model predicts congested patterns with 82.09%-96.73% precision compared with 84.74%-92.35% prediction precision for the network model. Despite delivering 96.73% accuracy by router model in the MPEG-4, the network model outperformed in four traced based applications. Therefore, on average, the network model comes with prediction precision under different traffic conditions. Simulation results demonstrate that the network model predicts local congestion with better accuracy and precision.

C. Hardware Analysis

To compute hardware area overhead, a CMOS based synaptic LIF model is used [27][8], where one synapse costs $24\times10^{-8}mm^2$ and one neuron $9\times10^{-6}mm^2$ hardware area. The hardware area of 4x4 router NoC is $8.9\times10^{-1}mm^2$ [28]. The proposed SNN models (as shown in Fig. 3-4) varies in topology, and so in hardware overhead.

TABLE 1 HARDWARE OVERHEAD

Simulator	Synaptic Area (mm^2)	Neural Area (mm^2)	Total Area (mm^2)
Router model	1.92×10^{-3}	2.16×10^{-3}	4.08×10^{-3}
Network model	2.30×10^{-3}	5.58×10^{-4}	2.86×10^{-3}

Table 1 shows hardware overhead of proposed models. Table 1 depicts that router model utilize $4.08\times10^{-3}mm^2$ hardware area as compared to network model $2.86\times10^{-3}mm^2$ area. Router model utilized almost double hardware with respect to network model. Difference in hardware is due to the utilizations of more neurons in router level. Results suggests that network model is more area efficient as compared to router model.

D. Discussion

The proposed model predicts local congestion on synthetic and real-time MPSoC application with up to 96.59% accuracy as compared to most accurate ANN based congestion prediction model which provides ~92% prediction accuracy [15]. SNNs are complex and computationally more powerful than traditional ANNs [9]. Furthermore, proposed SNNs requires 0.31%-0.45% hardware as compared to 5.8% of ANN model for base 16 router network, which makes SNNs more practical and suitable for hardware implementation.

V. Conclusion

In this work we proposed a novel SNN based congestion prediction model (both router and network models) for NoCs. The models were evaluated in term of prediction accuracy and hardware overhead. Traced-based synthetic and real-time MPSoC applications were mapped to Noxim to generate buffer utilization datasets according to the proposed congestion criteria. These patterns are used for training and testing of the SNN using MATLAB.

Results demonstrated that the network model predicts local congestion more accurately and precisely compared with the router model. Moreover, the router model exhibited ~ 40% additional area than that of the network model. Therefore, the paper concluded that the SNN at network level with a single SNN, for the whole NoC, is more scalable for congestion prediction.

The scope of this work is limited to identification of high performance congestion prediction model. Future work will explore the utilization of predicted congestion patterns to integrate with congestion aware routing algorithms [13]. Also finding low-cost interconnect solution for proposed SNN models to integration with NoC. Overall, the focus of the future research is to design an efficient SNN based prediction model that will predict congestion in advance and utilize predicted patterns to provide alternative paths to routing data.

REFERENCES

[1] M. Amin, M. Shakir, A. Javed, M. Hassan, and S. A. Raza, “Low-cost fault tolerant methodology for real time MPSoC based embedded system,” *Int. J. Reconfigurable Comput.*, vol. 2014, 2014.

[2] M. Rezaei-ravari, “Low Latency Path Prediction Mechanism in 2D - NoC,” *Electr. Eng. (ICEE), Iran. Conf.*, pp. 1565–1570, 2018.

[3] M. Tang, “Analysis on Local Congestion of Network-on-Chip,” no. Iccsee, pp. 2863–2866, 2013.

[4] S. H. Adil, M. Ebrahim, and K. Raza, “Prediction of Eye State Using KNN Algorithm,” *2018 Int. Conf. Intell. Adv. Syst.*, pp. 1–5, 2018.

[5] M. Saghir, Z. Bibi, S. Bashir, and F. H. Khan, “Churn Prediction using Neural Network based Individual and Ensemble Models,” *2019 16th Int. Bhurban Conf. Appl. Sci. Technol.*, pp. 634–639, 2019.

[6] S. Carrillo *et al.*, “Advancing interconnect density for spiking neural network hardware implementations using traffic-aware adaptive network-on-chip routers,” *Neural Networks*, vol. 33, pp. 42–57, 2012.

[7] W. Maass, “On the relevance of time in neural computation and learning,” in *Lecture Notes in Computer Science (including subseries Lecture Notes in Artificial Intelligence and Lecture Notes in Bioinformatics)*, 1997.

[8] J. Harkin, F. Morgan, L. Mcdaid, S. Hall, B. Mcginley, and S. Cawley, “A Reconfigurable and Biologically Inspired Paradigm for Computation Using Network-On-Chip and Spiking Neural Networks,” *Int. J. Reconfigurable Comput.*, vol. 2009, 2009.

[9] J. H. Lee, T. Delbruck, and M. Pfeiffer, “Training Deep Spiking Neural Networks using Backpropagation,” *CoRR*, vol. abs/1, pp. 1–10.

[10] Y. Kim, G. Kim, I. Hong, D. Kim, and H. Yoo, “A 4 . 9 mW Neural Network Task Scheduler for Congestion-minimized Network-on-Chip in Multi-core Systems,” in *IEEE Asian Solid-State Circuits Conference November 10 - 12, 2014/Kaohsiung, Taiwan*, 2014, pp. 14–17.

[11] A. Mohemmed, S. Schliebs, S. Matsuda, and N. Kasabov, “Method for training a spiking neuron to associate input-output spike trains,” in *IFIP Advances in Information and Communication Technology*, 2011.

[12] J. R. De Oliveira Neto, J. P. C. Cajueiro, and J. Ranhel, “Neural encoding and spike generation for Spiking Neural Networks implemented in FPGA,” *25th Int. Conf. Electron. Commun. Comput. CONIELECOMP 2015*, pp. 55–61, 2015.

[13] J. Liu, J. Harkin, Y. Li, and L. Maguire, “Microprocessors and Microsystems Low cost fault-tolerant routing algorithm for Networks-on-Chip,” *Microprocess. Microsyst.*, vol. 39, no. 6, pp. 358–372, 2015.

[14] Ming Li, Qing-An Zeng, and Wen-Ben Jone, “DyXY - a proximity congestion-aware deadlock-free dynamic routing method for network on chip,” *2006 43rd ACM/IEEE Des. Autom. Conf.*, pp. 849–852, 2006.

[15] E. Kakoulli, V. Soteriou, and T. Theocharides, “Intelligent hotspot prediction for network-on-chip-based multicore systems,” *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 31, no. 3, pp. 418–431, 2012.

[16] J. Latif, S. Azam, H. N. Chaudhry, and T. Muhammad, “Performance Evaluation of Modern Network-on-Chip Router Architectures,” vol. 2, no. 2, 2016.

[17] G. Ascia, V. Catania, M. Palesi, I. C. Society, D. Patti, and I. C. Society, “Implementation and Analysis of a New Selection Strategy for Adaptive Routing in Networks-on-Chip,” *IEEE Trans. Comput.*, vol. 57, no. 6, pp. 809–820, 2008.

[18] P. Huang and W. Hwang, “An Adaptive Congestion-Aware Routing Algorithm for Mesh Network- on-Chip Platform.”

[19] A. Benmessaoud Gabis and M. Koudil, “NoC routing protocols – objective-based classification,” *J. Syst. Archit.*, vol. 66–67, pp. 14–32, 2016.

[20] H. Tseng, R. Wu, W. Chang, Y. Lin, and D. Duh, “An Efficient Traffic-Based Routing Algorithm for 3D Networks-on-Chip,” in *Int’l Conf. Embedded Systems, Cyber-physical Systems, & Applications (ESCS’16)*, 2016, pp. 73–79.

[21] Y. S. C. Huang, K. C. K. Chou, and C. T. King, “Application-driven end-to-end traffic predictions for low power NoC design,” *IEEE Trans. Very Large Scale Integr. Syst.*, vol. 21, no. 2, pp. 229–238, 2013.

[22] T. Chen, W. Fu, B. Xie, and C. Wang, “Packet triggered prediction based task migration for network-on-chip,” in *20th Euromicro International Conference on Parallel, Distributed and Network-based Processing Packet*, 2012, vol. 38, no. 4, pp. 316–324.

[23] E. J. Chang, H. K. Hsin, S. Y. Lin, and A. Y. Wu, “Path-congestion-aware adaptive routing with a contention prediction scheme for network-on-chip systems,” *IEEE Trans. Comput. Des. Integr. Circuits Syst.*, vol. 33, no. 1, pp. 113–126, 2014.

[24] S. M. Bohte, J. N. Kok, and H. La Poutre, “SpikeProp : Backpropagation for Networks of Spiking Neurons Error-Backpropagation in a Network of Spik- ing Neurons,” *Esann*, no. May, pp. 419–424, 2000.

[25] V. Catania, A. Mineo, S. Monteleone, M. Palesi, and D. Patti, “Noxim : An Open , Extensible and Cycle-accurate Network on Chip Simulator,” *2015 IEEE 26th Int. Conf. Appl. Syst. Archit. Process.*, pp. 162–163, 2015.

[26] W. Zhang, L. Hou, J. Wang, S. Geng, and W. Wu, “Comparison Research between XY and Odd-Even Routing Algorithm of a 2-Dimension 3X3 Mesh Topology Network-on-Chip,” in *2009 WRI Global Congress on Intelligent Systems*, 2009, vol. 3, pp. 329–333.

[27] J. Liu, J. Harkin, M. Mcelholm, and L. Mcdaid, “Case Study : Bio-inspired Self-adaptive Strategy for Spike-based PID Controller,” *2015 IEEE Int. Symp. Circuits Syst.*, pp. 2700–2703, 2015.

[28] S. Carrillo *et al.*, “Advancing interconnect density for spiking neural network hardware implementations using traffic-aware adaptive network-on-chip routers,” *Neural Networks*, vol. 33, pp. 42–57, 2012.