Resilience Analysis of Distributed Wireless Spiking Neural Networks

Tamás Borsos, Massimo Condluci, Marios Daoutis, Péter Hága and András Veres

Ericsson Research
Email: {tamas.borsos, massimo.condoluci, marios.daoutis, peter.haga, andras.veres}@ericsson.com

Abstract—Spiking neural networks (SNN) are expected to enable several use-cases in future communication networks (beyond 5G and 6G), as edge AI and battery-constrained systems can leverage the fast computation and high-power efficiency offered by SNNs. In this work we consider a Distributed Wireless SNN (DW-SNN) system and analyze its performance in terms of inference accuracy and total neural activity when radio losses are applied to spikes transferred during the inference phase. Our aim is to understand how radio losses impact performance when considering different SNN spike communication types, i.e., input, excitatory, and inhibitory spikes. Then we evaluate the impact of different traffic prioritization approaches among SNN spikes when considering a shared channel capacity being available for SNN activity. From these analyses, we derive some key insights and features that can be considered when applying a DW-SNN and handling its traffic over wireless communication systems. Finally, we report a prototype implementation of DW-SNN using custom-built IoT components, which we use to further investigate different coverage scenarios.

Index Terms—Spiking Neural Network Architecture, Distributed Wireless AI, Traffic Prioritization.

I. INTRODUCTION

Biologically inspired neural networks, and in particular spiking neural networks (SNN), are often mentioned as the next generation of AI systems [1]. SNNs typically adhere to neuromorphic computing principles, thus are thought to inherit advantages such as fast computation and power efficiency. When used in conjunction with neuromorphic hardware, SNNs become particularly well suited for edge AI systems even in battery-powered Internet of Things (IoT) devices. SNNs mimic the operation of biological neurons and their spike-based communication [2], [3]: in these neural networks, all the information is encoded and communicated among the neurons as asynchronous binary signals (spikes) by leveraging the timing of discrete events. A spike itself can be considered as binary data, where the presence of a spike implicitly carries the information. There are several examples for devices generating spike type of data. In a neuromorphic or event camera [4]–[6] each pixel directly feeds corresponding neuron(s) and these neurons emit spikes when the change in light intensity exceeds a predefined threshold [6]. Examples of popular types of sensors include artificial cochlea, skin [7], touch sensors or Neuralink’s brain-machine interface [8] directly generating spikes as output signal, or actuators, such as robotic arms, which can be controlled via spike signals. In recent years we see several neuromorphic compute hardware being introduced in research [9], notably IBM’s Truenorth [10], the Neurogrid [11] or the SpiNNaker project [12] as well as the Loihi [13] and Loihi 2 [14] chips from Intel Labs. The increasing availability of such heterogeneous devices interconnected by next generation wireless networks will enable the development of large-scale distributed neural networks [15], which we refer to as Distributed Wireless SNN (DW-SNN). DW-SNN could help in realizing the 6G vision of interactions among the human world (senses, bodies, intelligence, and values), the digital world (information, communication and computing) and the physical world (objects, organisms) [16].

SNNs exhibit several properties that make it an appealing computational framework to be considered in a wireless environment. Highly power efficient operation is a significant advantage, which translates to low computing requirements in constrained devices. Neural activations are known to be sparse in SNNs [17]. Sparseness in spike communication leads to smaller bandwidth requirements, and further energy efficiency by decreasing the time of radio transmissions. In general, the internal SNN data flow is significantly different from the data flow of a conventional Artificial Neural Network (ANN) and it is important to highlight these differences to better understand and identify the underlying communication needs imposed by the SNN: (i) in an SNN, only active neurons transmit information when their membrane potential exceeds their firing threshold, while in an ANN all the neurons communicate their outputs to their connected peers irrespective of their level of activation; (ii) information in a SNN is encoded in terms of both the active/firing neurons as well as their exact spiking times, therefore the communication medium must maintain such temporal characteristics with high fidelity; (iii) the spike rate can be highly bursty (i.e., short, high frequency train of spikes) whose distributions of size and duration are governed by power laws [18]; and (iv) depending on the used encoding, SNNs can be particularly resilient to significant amount of spike losses.

In this paper we analyze some of the feasibility aspects of a DW-SNN architecture, with the aim to explore and showcase the potential advantages of the joint design of SNN and the communication network. The inherent resilience of SNNs to spike transmissions losses opens the opportunity to relax the communication requirements, and eventually, to
design AI-native radio transmission techniques based on spike communication. The case studies we perform and present in this work (i) reveal the low impact of application level spike losses that we allow to be significantly higher than traditionally used requirements in practical wireless solutions, and (ii) help to understand the behaviour of the dynamic interaction of SNN and traffic prioritization in a limited capacity scenario. We examine these effects from the point of the application level performance metric, the inference accuracy. We also show experimental results based on a real implementation of DWSNN on custom-built embedded IoT components.

The rest of the paper is organized as follows. Sec. II describes the model and architecture of the investigated SNN system. Sec. III analyses the impact of different levels of spike losses applied during the inference. Sec. IV investigates the impact of spike prioritization in scenarios with limited channel capacity. Sec. V presents a prototype of an SNN model implemented on custom-made embedded devices. Finally, Sec. VI concludes with the findings of our work.

II. A DISTRIBUTED WIRELESS SNN MODEL

One of the main aspects of a DW-SNN relates to the different ways of distributing the neurons across wireless connected devices. One extreme considers all neuron-to-neuron spikes to be completely distributed and therefore transmitted over wireless, while another extreme considers only the input spike stream to be encoded and transmitted where all inter-neuron connections are contained and processed within a single device (hence protected from any potential losses that may be incurred by the radio link). Moreover, practical scenarios are expected to lie between the two extremes, whereby the neural network is split among a number of devices, with each device accommodating a subset of the neurons.

We investigate the various phenomena arising in DW-SNNs by selecting a spike-based implementation of the well-known MNIST hand-written digit recognition problem [19], [20]. The model implemented in [21] was modified by introducing a simulated radio link between the neurons of the SNN. In order to evaluate the overall SNN resilience, we focus on the impact of spike losses. The radio channel has a simple loss model, where the spike loss probability $P$ can be adjusted between simulation runs. The SNN is pre-trained without assuming any communication losses, so as to disallow the SNN from adapting to losses. Therefore, spike losses are only introduced during the inference phase.

The architecture of the investigated SNN MNIST system is depicted in Fig. 1. The model consists of: (i) an input layer, (ii) excitatory and inhibitory layers, (iii) an accumulator to support the inference decision. The input layer contains 784 neurons matching the number of pixels (28, 28) of the images in the MNIST Handwritten digit database [22]. The firing rates of the input neurons are proportional to the pixel intensity and transformed to spike-trains by applying Poisson-encoding schema. The input layer neurons are connected to the excitatory neurons (500 in our simulations) in a full-mesh fashion, while the excitatory neurons are connected to the same number of inhibitory neurons in a one-to-one fashion triggering an inhibitory spike for each excitatory spike. Each of the inhibitory neurons is connected to all excitatory ones, except for the one from which it receives a connection. This connectivity provides lateral inhibition and leads to an effective selection of excitatory neurons to represent image features. The excitatory neurons are also connected to the decision layer, in which the digit-assigned neurons are accumulating the excitatory spikes and are used to select the inferred digit.

The simulated radio link is applied for every connection in the neural network. All the synapses of the SNN are trained by using STDP learning rule. During the training we presented 60,000 images as input data from [23] in 10 epochs, while during the inference phase we used 1000 randomly selected images from the test database.

We experiment with different levels of distribution at various...
spike loss levels to assess the sensitivity of the entire system. The simulated wireless link is used: (i) between the input and the excitatory layer, (ii) between the excitatory layer and the decision layer and (iii) between the different agents. At different levels of distribution we split the SNN into a number of agents that use lossless internal communication (to the inhibitory and excitatory neurons located in the same agent) and wireless radio communication to other agents’ excitatory neurons (see Fig. 1). The results of the simulations are presented in Fig. 2. According to our observations, the recognition accuracy remains tolerant to high spike loss rates (even up to extreme loss rate up-to 50%). This is due to the redundant structure of the SNN where each digit is recognized by the spikes of a set of excitatory neurons, so the inference can still be performed even in the case of absence of several spikes from these those neurons. It is worth noting that the level of distribution has a significant impact on the system accuracy. When the system is distributed to extreme levels, and all transmissions are exposed to radio losses, the accuracy is impacted accordingly. On the other hand, the scale of impact converges, and there is almost no difference between distribution to 5 or 500 devices. That observation leads us to only investigate the extreme distributed case further in this paper, since that captures the fundamental behavior of DW-SNNs quite well.

III. SPIKE LOSS IMPACT ANALYSIS

The spikes generated in an SNN can be distinguished by their purpose: input numerical data are translated via a Poisson-encoding input encoding scheme into spikes on input neurons (input spikes), while spikes generated by the neurons are called excitatory and inhibitory spikes. The last two have different functions: excitatory spikes carry information about the detected features and then use them for the inference decision, whereas inhibitory spikes suppress neuron activities belonging to other feature detection. Thus, inhibitory spikes effectively regulate (i.e., significantly reduce) the neural activity of an SNN by keeping only the spikes for solid feature detection. Given the different roles of the spikes, the impact of the losses affecting them leads to significant difference in both the inference accuracy and the internal neural activity (e.g., traffic volume) of the investigated SNN. To analyze this aspect, we performed simulations in which only one kind of spike is suffering from losses (e.g., in one case only the input spikes suffer from radio losses, whereas the transmission of excitatory and inhibitory spikes remains lossless).

We analyzed three cases, where in each case only one spike type is affected by losses: (i) losses only to input spikes, (ii) losses only to excitatory spikes, and (iii) losses only to inhibitory spikes. Results in terms of inference accuracy are shown in Fig. 3. All the three curves show that the implemented SNN is highly resilient to spike losses, and differences for the investigated cases can be clearly seen for spike loss probabilities $P > 0.6$. In case of input spike losses, the accuracy remains high as long as the volume of input data is sufficient to trigger the activation of neurons ($P < 0.7$). For even higher loss probabilities of input spikes, no sufficient number of neurons is activated and so the detection of features, thus the inference accuracy drops. The impact of losing excitatory spikes is different, since some degradation in the inference accuracy starts at $P > 0.5$, meaning that the lost excitatory spikes (e.g., the detected features) are missing during the decision of the digit. Furthermore, in cases when only a few excitatory spikes are not lost (for $P > 0.8$), the inference accuracy remains high. In the case of inhibitory spike losses, the accuracy value remains high for even higher spike loss probabilities, since their role is not directly connected to the final recognition of the digit, rather than suppressing the false feature detection.

More significant differences can be seen by analysing the total neural activity of the SNN for the different spike losses. This metric has an utmost importance in real life applications.
The neural activity directly determines the power efficiency of the computation thus impacting the energy consumption of constrained devices. Neural activity is also a key aspect for the power efficiency of the radio communication. If more spikes need to be transmitted, implicitly more energy is expected to be consumed for communication purposes. Finally, spectral efficiency or the bandwidth of the radio communication is also directly affected by the number of spikes to be transmitted.

In Fig. 4, we present the (average) number of spikes emitted during an inference event for the three possible cases we used in the previous analysis. In the case of input spike losses, we see that neural activity decreases linearly as input spikes are lost. The reason for this decrease is that the lower number of input spikes triggers less neurons to fire as their firing threshold might not be reached by the input spikes. In case of excitatory spike losses, a slight decrease can be seen in the neural activity. This is due to the fact that the excitatory spikes are not fed back to other neurons in the overall SNN. Opposite to the previous cases, in the case of inhibitory spike losses, the neural activity increases dramatically due to the missing traffic regulation, e.g., the suppression of neurons detecting various false features. This phenomena can act as a positive feedback as the loss events result in significant increase in the traffic volume that should be avoided in real situations.

By comparing the numerical results of Fig. 3 and Fig. 4, one can try to find the desired balance between the required inference accuracy and the increased neural activity (i.e., the traffic volume) for the considered model. For high spike loss probabilities, one can observe that the requirements for having the best inference accuracy and the lowest neural activity contradict each other. E.g., at $P > 0.8$ the best inference accuracy can be achieved if the spike losses affect the inhibitory spikes, while the inhibitory spike losses generate huge neural activity surplus, causing large traffic impact.

IV. SPIKE DIFFERENTIATION OVER SHARED CHANNEL

In the previous sections we analyzed the impact of the communication channel on the basis of specific neural activities without considering the interaction between the traffic flows. In a realistic deployment, however, the neural agents are often distributed over devices that communicate over shared wireless channels. For example, in use-cases like collaborating mobile robots aided by distributed sensor networks on an factory floor, or vehicles and infrastructure cooperating in traffic junctions of a smart city [16], the devices can be highly localized, reusing the same spectral resources, which calls for efficient allocation of these resources. The aim of this analysis is to understand the impact of losses on different spike traffic types, and to consequently understand which types of differentiated traffic handling mechanisms can be adopted in such upcoming mobile systems, with the aim to significantly improve the application level performance metrics, such as the inference accuracy.

Here, we consider a network model of a fully shared channel with limited capacity for spike communication capable of supporting different priorities for input, excitatory and inhibitory spikes. We focus on the interoperability between a priority-based network scheduling and the inhibitory feedback mechanisms of an SNN, both of which regarded as highly dynamic systems. This interoperability is simulated in three contexts: (i) a baseline scenario where all traffic types are treated equally without prioritization, (ii) a case where input traffic is prioritized over excitatory and inhibitory spikes (which hereinafter we refer to as internal traffic) and (iii) a case where internal traffic is prioritized over the input spikes. In our analysis, we consider that lower-priority traffic is affected by losses, which could reflect the case of, e.g., a certain traffic not being transferred due to lack of radio resources after serving higher priority traffic. The channel capacity available for SNN traffic varies from 1 spike/slot (to reflect a congested channel where just a fraction of the overall capacity can be exploited by SNN) to 30 spikes/slot, which practically means no limitation.
for SNN traffic and therefore implies zero spike loss.

In Fig. 5 we present the inference accuracy for the 3 cases mentioned above. We observe that prioritization of input traffic degrades the performance in case of limited SNN channel capacity, where it affects the excitatory spikes required directly for classification. Although inhibitory spikes are also dropped at this point, which results in increased level of excitatory activations, it is seemingly not enough to compensate for the losses. Since there is no neural feedback mechanism to control the rate of input spikes, the accuracy steadily falls with the decreasing overall capacity. The reference case with equal traffic handling remains robust even at a low capacity of 5 spikes/slot, despite of high spike loss. This is in agreement with independent loss results, which show good resilience to high losses in any spike types. No prioritization means somehow equal loss ratio for all categories, regardless of spike volumes, as the arrival of spikes within a slot is uniformly distributed. The best results on accuracy are achieved by prioritizing the internal traffic, i.e., both excitatory and inhibitory spikes over the input, which can be attributed to two factors. First, internal spikes are likely to be more influential for classification accuracy than inputs, since excitatory spikes represent already processed information and features extracted in multiple iterations, which was condensed from hundreds of input spikes. On the other hand, the intensity of internal spikes is reduced by the input through a negative feedback, thus creating a self-regulatory system, where internal traffic will never suppress the input completely. As a result of this dynamic interaction, this prioritization results in maintaining high accuracy even at a capacity limit of as low as 3 spikes/slot.

Significant differences among the investigated cases can also be observed in the neural activities. This activity level is of key importance for optimal SNN operation, since it directly impacts power efficiency, communication sparsity, and, eventually bandwidth usage. Simulation results for neural activities are shown in Fig. 6. As a result of prioritizing the input spikes the activity level is surging when the channel starts to saturate. This effect is clearly due to the loss of inhibitory spikes, which allows an increasing number of neurons to fire excitatory spikes. Neural activity is increased even in the baseline case without prioritization to a small degree for similar reasons, which is an indication to expect even better performance if the inhibitory function is protected from spike losses. Prioritizing the internal traffic leads to a graceful decrease of activity without any excessive spiking at lower channel capacities.

From this analysis, we can see that there is a benefit for wireless networks supporting SNN in being able to differentiate the treatment of SNN traffic. In our considered model, we see that having higher priority treatment for internal traffic (w.r.t. input spikes) reduces the overall neural activity regardless the channel capacity allocated to SNN traffic and also gives the highest inference accuracy when few channel resources are available for SNN traffic. This implies that, to optimally handle SNN traffic, a wireless network should be aware of the different classes of the transferred spikes, so as to be able to consequently apply different traffic prioritization

---

V. PROTOTYPE OF DW-SNN IOT SYSTEM

In the previous sections we address DW-SNN aspects regarding spike losses and SNN performance over shared channel capacity in the context of transmitting spikes over radio. When considering the application of DW-SNNs in an actual use case, we encounter several challenges, especially around mapping the SNN to a set of physical devices. Here we report our experiences when materialising the proposed DW-SNN using custom-built embedded IoT components. The devices consist of an ARM M3 micro-controller (MCU), which is responsible for handling the wireless protocols while also emulating the spiking neurons. The neurons are emulated in an event-driven way, so the state of neurons (e.g., membrane potential) are updated upon the arrival or transmission of spikes. This way even a low-power, constrained MCU can emulate neurons very efficiently, allowing each device to emulate approximately 1000 LIF (Leaky Integrate and Fire) neurons in real-time.

For communication, the devices use an IEEE 802.15.4 Ultra Wide-Band radio module. The used frequency and bandwidth are 6.5 GHz and 500 MHz respectively. We modified the standard IEEE 802.15.4 frame structure to accommodate spikes better. The radio frame consists of a preamble to allow for receiver synchronization, a short type field and a so-called synapse Id to identify the synapse (as seen in Fig. 7). The total spike transmission time is approximately 70 µsec, which results in a theoretical maximum channel capacity of approximately 14000 spikes per second.

One practical challenge when mapping a DW-SNN on top of a distributed IoT system is that DW-SNN can be considered as a logical topology mapped on top of a physical topology
consisting of wireless IoT devices. We had to develop a protocol, which enables a network operator to split the DW-SNN model, and then deploy the individual configurations of spike neurons in each physical IoT device (as seen in Fig. 8). Also, because physical IoT devices are usually not very reliable, the operator needs to monitor and rearrange the physical to logical mapping in real-time whenever this is necessary. A re-deployment may be needed due to failure of IoT devices, a change of physical topology because of mobility, change of radio environment, and finally when there is a need to optimize the usage of radio resources as it was discussed in earlier chapters.

Moreover, the physical prototype allows us to investigate the impact of changes in the physical topology, for example device failures or changes in coverage due to mobility. Initially we considered a small physical configuration consisting of 4 devices, where the DW-SNN neurons are randomly assigned among them. After configuration and DW-SNN mapping onto the devices, we moved 2 devices out of coverage (depicted as 2/4 in Fig. 9), finally we only left 1 device in range (1/4). Here is worth noting that we did not let the system to reconfigure itself, nor did we do any optimization to mitigate the topology changes. In Fig. 9 we see that due to the self-regulatory behavior of inhibitory feedback, both the activity of excitatory neurons, and the overall aggregated spike rate stayed almost the same regardless of the number of connected devices. We find this behavior in agreement with our earlier observations during simulation for the impact of spike level losses.

VI. Conclusion

In this work we analyze how spike losses affect the inference accuracy and total neural activity when considering a DW-SNN implementation of the well-known MNIST handwritten digit recognition problem. Our DW-SNN model maintained high accuracy as well as low neural activity, which overall can ensure higher reliability either to excitatory and inhibitory (over input) spikes or to input and inhibitory (over excitatory) spikes. Furthermore, high inference and low neural activity was achieved by our model when excitatory and inhibitory spikes were prioritized over input spikes. These results provide key insights for the design of upcoming wireless systems aimed at better handling SNN based traffic. We expect that other DW-SNN realizations could benefit from mechanisms such as differentiation of prioritization and reliability of different types of spikes. Further studies may focus on understanding how these methods could be applied to other DW-SNN models, as well as on investigating how these methods could be dynamically adapted to the noise level and the available capacity of the channel.

ACKNOWLEDGMENT

This work has been partly funded by the European Commission through the H2020 project Hexa-X (Grant Agreement no. 101015956).

REFERENCES

[1] A. Tavanaei, et al., “Deep learning in spiking neural networks,” Neural Networks, 111, pp. 47-63, Mar. 2019. 10.1016/j.neunet.2018.12.002
[2] Gerstner, Wulfram, and Werner M. Kistler. Spiking neuron models: Single neurons, populations, plasticity. Cambridge Press, 2002.
[3] S. Ghosh-Dastidar, and H. Adeli, “Third Generation Neural Networks: Spiking Neural Networks,” in Yu W., Sanchez E.N. (eds) Advances in Computational Intelligence. Advances in Intelligent and Soft Computing, vol 116. Springer, 2009. 10.1007/978-3-642-03156-4_17.
[4] inVation AG, https://inivation.com/
[5] Prophesee Metavision for Machines https://www.prophesee.ai/, applications: https://www.prophesee.ai/event-based-vision-applications/
[6] Rebecq, Henri, et al. "Events-to-video: Bringing modern computer vision to event cameras." Proceedings of the IEEE/CVF Conference on Computer Vision and Pattern Recognition. 2019.
[7] Wang Wei Lee, et al., “A neuro-inspired artificial peripheral nervous system for scalable electronic skins,” Science Robotics, 4(32), Jul. 2019, 10.1126/scirobots.aax2198.
[8] Neuralink, https://neuralink.com/
[9] S. Furber, “Large-scale neuromorphic computing systems”, Journal of Neural Engineering, 13(5), Aug. 2016. 10.1088/1741-2560/13/5/051001.
[10] Hsu, Jeremy. "IBM's new brain [News]." IEEE spectrum 51.10 (2014): 17-19.
[11] Benjamin, Ben Varkey, et al. “Neurogrid: A mixed-analog-digital multi-chip system for large-scale neural simulations.” Proceedings of the IEEE 102.5 (2014): 699-716.
[12] Furber, Steve B., et al. “The spinnaker project.” Proceedings of the IEEE 102.5 (2014): 652-665.
[13] Davies, Mike, et al. "Loihi: A neuromorphic manycore processor with on-chip learning." IEEE Micro 38.1 (2018): 82-99.
[14] Intel Labs, Neuromorphic Computing, https://www.intel.com/content/www/us/en/newsroom/news/intel-unveils-neuromorphic-loihi-2-lava-software.html
[15] N. Skatchkovsky, H. Jang, and O. Simeone, “Spiking Neural Networks—Part III: Neuromorphic Communications,” IEEE Communications Letters, vol. 25, no. 6, pp. 1746–1750, Jun. 2021.
[16] "Hexa-X project,” 2021. [Online]. Available: https://hexa-x.eu/.
[17] Fernandez-Musoles, Carlos, Coca, Daniel, Richmond, Paul, "Communication Sparsity in Distributed Spiking Neural Network Simulations to Improve Scalability", Frontiers in Neuroinformatics. 13. 10.1088/1741-2560/13/5/051001.
[18] L. Brochini, et al., “Phase transitions and self-organized criticality in networks of stochastic spiking neurons,” Scientific Reports, 6, 35831
[19] Diehl, Peter U., and Matthew Cook. "Unsupervised learning of digit recognition using spike-timing-dependent plasticity." Frontiers in computational neuroscience 9 (2015): 99.
[20] H. Hananel, et al., “Bindsnet: A machine learning-oriented spiking neural networks library in python,” Frontiers in Neuroinformatics, Dec. 2018.
[21] LeCun, Yann, et al. "Gradient-based learning applied to document recognition." Proceedings of the IEEE 86.11 (1998): 2278-2324.
[22] LeCun, Yann, Corinna Cortes, and Chris Burges. “MNIST handwritten digit database.” (2010): 18., http://yann.lecun.com/exdb/mnist