Network-on-Chip Architectures for Neural Networks

Dmitri Vainbrand and Ran Ginosar

Technion—Israel Institute of Technology, Haifa, Israel

Abstract

Providing highly flexible connectivity is a major architectural challenge for hardware implementation of reconfigurable neural networks. We perform an analytical evaluation and comparison of different configurable interconnect architectures (mesh NoC, tree, shared bus and point-to-point) emulating variants of two neural network topologies (having full and random exponential configurable connectivity). We derive analytical expressions and asymptotic limits for performance (in terms of bandwidth) and cost (in terms of area and power) of the interconnect architectures considering three communication methods (unicast, multicast and broadcast). It is shown that multicast mesh NoC provides the highest performance/cost ratio and consequently it is the most suitable interconnect architecture for configurable neural network implementation. Simulation results successfully validate the analytical models and the asymptotic behavior of the network as a function of its size.

I. INTRODUCTION

The inherent parallelism of multi-processor VLSI systems on chip (SoC) enables the efficient emulation of biological neural networks and the construction of artificial neural networks for complex tasks such as pattern recognition. When the structure and connectivity are implemented rigidly in hardware, the emulated neural networks suffer of limited flexibility and functionality [6], [7], requiring redesign if any connectivity or function needs to be changed. To overcome these limitations, we seek a SoC architecture that enables programmable and reconfigurable neural networks. Such architecture could serve as a generic medium for neuroscience and machine learning research, enabling emulation of arbitrary neural network topology and support dynamic connectivity changes as a result of training.

A key issue in emulating reconfigurable neural network is the complexity of dynamic and flexible neural connectivity. We investigate and compare several interconnect architectures (mesh NoC, tree, bus, point-to-point) and three packet-based communication methods (unicast, multicast and broadcast) and study how they support configurable communications for spiking neural networks. We show that mesh NoC using multicast is the most suitable architecture for a wide range of neural network topologies.

Background and related work are presented in Sect. II. Theoretical cost and performance analysis is provided in Sect. III, followed by simulations in Sect. IV

II. BACKGROUND AND RELATED WORK

A. Spiking Neural Networks

Artificial neural networks [5] have evolved from McCulloch-Pitts threshold on/off neurons, through the more biologically realistic continuous activation networks representing firing rates, to more accurate spiking neural networks, which also reflect inter-spike spatio-temporal relations, and which are the subject of this paper.

B. Neural Network Implementation

Parallel processors and special purpose hardware are most suitable for fast emulation of computationally intensive neural networks, for both real time applications and for very large tasks. Previous highly flexible and configurable implementations were limited to emulating a small number of neurons [8], while systems of many neurons offered limited flexibility [6], [7], and flexible systems of many neurons required very large implementations [9], due to the quadratic complexity of a fully connected design.

Hierarchical and simplified communication methods have been used to mitigate the complexity of full connectivity. Shared bus neural network implementations typically provide efficient event driven communications, whereby only the address of the spiking neuron is broadcast to all bus elements. Non-arbitrated shared bus, which detects and ignores collisions, was described in [10]. Other works employed arbitrated buses using the Address Event Representation (AER) asynchronous protocol [11]--[13]. That representation is useful for either point-to-point connections [11], [12] or broadcasting over a shared bus [13]. The former case requires expensive communication network while parallelism is quite limited in the latter case.

SoC implementations of neural networks using NoC have been proposed by [14]–[16]. NoC are particularly attractive for spiking neural networks, as they facilitate parallelism, reconfigurability, independence of the network topology, and network expandability. A small 2D torus network with four processing elements (neurons) per routing node is described in [17]. The architecture supports deadlock free X-Y routing, and uses wormhole packet switched communication. The packet combines the outputs of four logical neurons and is sent to the router of the next layer of the layered neural network router, thus

enabling only layered structure and not allowing arbitrary communications. That NoC architecture is less applicable to spiking neural networks. Scaling is also limited.

An FPGA-based mesh NoC using XY unicast routing for clustered neural network has been proposed in [19]. As shown in this paper, unicast communications may be a limiting factor in neural network implementations.

A large scale multi-chip spiking neural-network system was reported in [18]. A hierarchy of seven chips, 20 processors per chip and 1000 spiking neurons per processor is contemplated. Small AER packets are exchanged using multicast NoC. As the ability to emulate arbitrary neural network connectivity depends strongly on the size of routing tables, the paper addresses optimization techniques to reduce table sizes.

While [18] employs a mesh, [17] uses 2D torus and [19] describes a more complex hierarchy. While [17] and [19] rely on unicast, [18] chooses multicast. While [17] uses wormhole routing, [18] and [19] employ short AER packets. Packets in [18] consist of source address and [19] uses destination address. This variety of approaches raises the question of which architectural choices are preferred over others, and whether indeed NoC is the appropriate solution for the emulation of large scale neural networks. This paper attempts to address these questions, by an analytical evaluation and comparison of different interconnect architectures emulating different neural network topologies. Previous cost/performance characterizations employed either analytical [22], [23] or empirical [24], [25] approaches. We adopt the method of [20], for its generality and applicability to NoC and non-NoC architectures, as explained in the following section.

III. NETWORK ARCHITECTURE

This work investigates the emulation of neural networks on many processors inside a single SoC. Several neurons can be allocated to each processor. Spikes transmitted between neurons that are assigned to the same processor are transferred internally within the processor. Spikes between neurons assigned to different processors must be transferred over the interconnect. In the following, Section III.A defines cost and performance metrics. In Sections II.B–E we consider four alternative implementations for how the processors are interconnected, and evaluate how the interconnect architecture impacts the cost and performance of configurable neural networks. In Section II.F we summarize the results. We compare the following interconnect architectures:

NoC_Mesh: a mesh network-on-chip.
NoC_Tree: a tree network-on-chip.

AER_BUS: a broadcasting AER protocol on a shared bus.

P2p: a point to point reconfigurable connection matrix.

We consider three different communication methods ("casts"): unicast (UC), multicast (MC) and broadcast (BC). Two types of neural network models are investigated: Hopfield network [4], which requires full connectivity of the neurons (each neuron sends each spike to all other neurons), and Randomly Connected NN (RNDC), defined by random exponential local connectivity (each neurons sends its spikes to a small set of neighbors). Both NN models incur uniform traffic. The former represents maximal connectivity bounds, whereas the latter, adopted from [1], represents more realistic scenarios.

A. Cost and Performance Metrics

We use a metric similar to [20]. Cost is associated with interconnection area and power dissipation (another cost item, not discussed in this paper, relates to the size of memory needed for routing tables). Performance is evaluated by effective bandwidth, throughput and maximal spiking frequency of the neurons. The maximal theoretical bandwidth is:

$$BW(nn, arch, cast) = \frac{\sum_{i \in \{\text{links}\}} w(i) f(i)}{TotalDist(nn, arch, cast)} \quad (1)$$

were $w(i)$ is the width of link i, $f(i)$ is its switching frequency, and $TotalDist(nn, arch, cast)$ is the average total distance traversed by each neural spike (going to all its destinations), measured in the number of hops; $arch \in \{$ NoC_Mesh, NoC_Tree, AER_BUS, p2p$\}$, $cast \in \{$UC, MC, BC$\}$, and $nn \in \{$Hopfield, RNDC$\}$. While the maximum BW indicates the possible level of parallelism for a given architecture, communication method and neural network, it does not take into account inter-packet interactions and variable latency. If the network is allowed to operate at maximum BW, it stalls to a halt due to congestion. It is shown empirically in Sect IV that if the network operates below the congestion threshold, there are no congestion effects at all, and we can assume fixed small router delays. This is modeled by an empirical architecture-specific utilization factor U_{arch}, defining the effective bandwidth:

$$BW_{eff}(nn, arch, cast) = BW \times U_{arch} \quad (2)$$

For topologies with a constant number of wires per link $\bar{w}$ and constant frequency, (2) becomes:

$$BW_{eff} = \frac{\bar{w} \cdot TL_{arch}}{TotalDist} f_{arch} U_{arch} \quad (3)$$

where TL_{arch} is the total number of links in the architecture. The area cost of the architecture is:

$$A_{arch} = W_p \sum_{i \in \{\text{Arch links}\}} w(i) l(i) \quad (4)$$

where *l(i)* is the length of link *i* and W_p is the wire pitch for a given technology. We disregard router delays, since they do not scale with network size and thus the link delay is also the link cycle time:

$$T_{cycle} = R_0 C_0 \overline{l}^{\,2} \qquad (5)$$

where R_0 and C_0 are the wire resistance and capacitance per unit length, respectively, and $\overline{l}$ is the average link length. Thus the maximum link frequency is:

$$f_{arch} = \frac{1}{T_{cycle}} = \frac{1}{R_0 C_0 \overline{l}^{\,2}} \qquad (6)$$

Power dissipation is estimated as dynamic power dissipated on the link and gate capacitances:

$$P_{arch} = \sum_{i \in \{links\}} C(i) f_{arch}(i) V_{DD}^2 U_{arch} \qquad (7)$$

The maximum spiking frequency is determined by the biologically inspired *neuron refractory period* $T_{refractory}$, a "blanking" time following a spike during which the neuron cannot fire again:

$$f_{spike,\max} = \frac{1}{T_{refractory}} \qquad (8)$$

In biological cortical neural networks, the refractory period and the average synaptic (axonal) delay are typically in the same range (2-10ms), which in our case equals the packet end-to-end delay:

$$T_{refractory} \cong T_{Ax} \cong T_{cycle} \cdot \overline{Dist} \qquad (9)$$

where $\overline{Dist}$ is the average distance (in number of hops) between two connected neurons.

In the special case of unicast NoC implementations, each spike results in a succession of packets, one per destination. The delay between issuing the first and the last packets of the same spike is $N_{avg} T_{cycle}$ where N_{avg} is the average number of spike destinations for a given neural network. We arbitrarily choose a minimal time between successive spikes fired by the same neuron of $10 N_{avg} T_{cycle}$ (the inter-spike interval should be larger than the time of sending all packets of one spike) and the maximal firing frequency for unicast NoCs combines both issue and refractory delays:

$$\begin{aligned} f_{spike,\max}^{UC} &= \frac{1}{\min\left(T_{cycle} \cdot 10 N_{avg}, T_{cycle} \overline{Dist}\right)} \\ &\approx \frac{1}{10 N_{avg} T_{cycle}} \end{aligned} \qquad (10)$$

For multicast and broadcast, only one packet is issued per each spike regardless of its number of destinations. The maximal firing frequency for multicast and broadcast NoC is thus implied by (9):

$$f_{spike,\max}^{MC|BC} = \frac{1}{T_{cycle} \overline{Dist}} \qquad (11)$$

The above definition of maximal firing frequency reflects the basic property of refractory time in neural networks in the presence of the geometrical and electrical delays of the implementation. Note that the maximal firing frequency does not necessarily match the maximal NoC bandwidth. Moreover, as shown below, some of the studied implementations (especially broadcasting NoC and BUS) are jammed when all neurons fire constantly at their maximal frequency.

BW_{eff} of (3) is the average rate at which an entire network absorbs new massages. The average rate at which a single processor can feed spikes into the network is obtained by dividing into the number of processors n_p:

$$f_{p,out} = \frac{BW_{eff}}{n_p} \qquad (12)$$

The fact that the maximal firing rate $f_{spike,\max}$ may be different than $f_{p,out}$ is expressed by a factor *K*, indicating the degree to which a given implementation enables neurons to operate at their maximal rate and providing a figure of merit for comparing implementations:

$$K = \frac{f_{p,out}}{f_{spike,\max}} \qquad (13)$$

If K>1, multiple logical neurons can fit efficiently into a single physical processor. If K<1, $f_{spike,\max}$ is unachievable. Finally, the cost-performance ratio *R* is:

$$R = \frac{BW_{eff,arch}}{A_{arch} \cdot P_{arch}} \qquad (14)$$

B. Mesh NoC

A mesh NoC comprises *n* processors and *n* routers arranged in a $\sqrt{n} \times \sqrt{n}$ mesh (Figure 1).
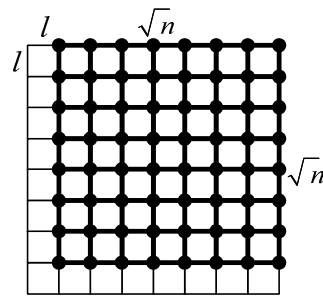

Figure 1: Mesh NoC

The total number of links in the mesh is:

$$TL_{Mesh} = 2\overline{w}\sqrt{n}\left(\sqrt{n} - 1\right) \qquad (15)$$

The average distance between two nodes in the mesh is $\overline{Dist}_{Mesh} = \frac{2}{3}\sqrt{n}$ [22].

1) Emulating Hopfield NN on a Mesh NoC

Consider a Hopfield (fully connected) NN emulated on a unicast mesh NoC. For simplicity of the analysis, assume that each processor emulates a single neuron. For each spike, a neuron sends a packet to all the other neurons (n-1 packets). The total number of hops traversed by one spike is the sum of distances between this neuron and all other neurons:

$$TotalDist_{UC,Mesh}^{Hopfield} = (n-1) \cdot \overline{Dist}_{Mesh} = \frac{2}{3}(n-1) \cdot \sqrt{n} \quad (16)$$

Substituting (15) and (16) into (3) yields:

$$BW_{eff,Mesh,UC}^{Hopfield} = \frac{3\overline{w}}{\sqrt{n}+1} f_{NoC} U_{NoC} \quad (17)$$

and the average frequency of feeding new spikes from a single processor is:

$$f_{p,out,Mesh,UC}^{Hopfield} = \frac{BW_{eff,Mesh,UC}^{Hopfield}}{n}$$

$$= \frac{3\overline{w}}{n(\sqrt{n}+1)} f_{NoC} U_{NoC} \quad (18)$$

Following (10) we conclude:

$$f_{spike,max}^{UC} \cong \frac{f_{NoC}}{10n} \quad (19)$$

Comparing the results of (17) and (19) we can derive:

$$K = \frac{f_{p,out,UC}^{Hopfield}}{f_{spike,max}^{UC}} \cong \frac{30\overline{w}U_{NoC}}{\sqrt{n}} = O\left(1/\sqrt{n}\right) \quad (20)$$

This result implies that the UC mesh NoC does not offer sufficient bandwidth to emulate a Hopfield NN at the maximal firing frequency. Only about $nK = O\left(\sqrt{n}\right)$ neurons may fire close to their maximal rate and the remaining other neurons will fire at a negligibly low rate. Alternatively, all neurons could fire at a $1/\sqrt{n}$ fraction of their maximal rate.

Turning now to multicast and broadcast NoCs, they are essentially the same for Hopfield NN, as each spike is transmitted to n-1 other neurons. The number of hops traversed per spike are the number of edges in the mesh spanning tree,

$$TotalDist_{MC/BC}^{Hopfield} \cong n \quad (21)$$

Thus:

$$BW_{eff,MC\backslash BC}^{Hopfield} = 2\overline{w}f_{NoC}U_{MC\backslash BC}\left(1 - \frac{1}{\sqrt{n}}\right) = O(1) \quad (22)$$

Intuitively, one packet, sent to all neurons, utilizes the entire network. Thus, the network can handle only one spike at a time. The average NoC input frequency is:

$$f_{p,out,MC/BC}^{Hopfield} = \frac{BW_{eff,MC/BC}^{Hopfield}}{n} \cong \frac{2\overline{w}f_{NoC}U_{NoC}}{n} \quad (23)$$

The fact that the source neuron issues one packet per enables a much tighter bound for spiking frequency, following (11):

$$f_{spike,max} = \frac{1}{T_{refractory}} \cong \frac{3f_{NoC}}{2\sqrt{n}} \quad (24)$$

Comparing (24) and (23) yields:

$$K_{Mesh,MC/BC}^{Hopfeild} \cong \frac{4\overline{w}U_{NoC}}{3\sqrt{n}} = O\left(1/\sqrt{n}\right) \quad (25)$$

The remaining parameters depend only on the interconnect topology, and are the same for all communication methods:

$$A_{mesh} = 2W_p l\overline{w}\sqrt{n}\left(\sqrt{n}-1\right)$$

$$C_{mesh} = C_0 2l\overline{w}\sqrt{n}\left(\sqrt{n}-1\right)$$

$$f_{mesh} = \frac{1}{R_0 C_0 l^2} \quad (26)$$

$$P_{mesh} = P_0 2\overline{w}U_{NoC}\sqrt{n}\left(\sqrt{n}-1\right)$$

$$P_0 \triangleq V_{dd}^{2}/R_0 l$$

The asymptotic metrics for Hopfield NN emulated on a mesh NoC are summarized in Table 1

Table 1: Hopfield network emulated on a mesh NoC

Metric	UC	MC	BC
BW	$O\left(1/\sqrt{n}\right)$	$O(1)$	$O(1)$
Area	$O(n)$		
Power	$O(n)$		
Spiking Frequency	$O(1/n)$	$O\left(1/\sqrt{n}\right)$	$O\left(1/\sqrt{n}\right)$
K	$O\left(1/\sqrt{n}\right)$		

2) Emulating RNDC NN on a Mesh NoC

For RNDC model, the probability of having a connection from neuron a to neuron b is defined similarly to [1]:

$$p(a,b) = \frac{C}{2\pi\lambda^2} e^{-D(a,b)/\lambda} \quad (27)$$

where D(a, b) is the Euclidean distance between a and b, λ is a spatial connectivity constant, and C=N$_{links}$= ‖p(·)‖ is the average number of connections per neuron. The mean distance between two connected neurons is:

$$\overline{Dist} = \frac{1}{2\pi\lambda^2} \iint_{x,y} \sqrt{x^2 + y^2}\, e^{-\sqrt{x^2+y^2}/\lambda} dxdy = 2\lambda \quad (28)$$

When emulating RNDC neural network on a unicast mesh NoC, a neuron sends individual packets to all of its C destinations. The average total number of hops for each spike packet is:

$$TotalDist = \overline{Dist} \cdot Nlinks = 2\lambda C \quad (29)$$

Thus, the bandwidth (3) of RNDC emulated on unicast mesh NoC is:

$$BW_{eff,UC}^{RNDC} = \frac{\overline{w}\sqrt{n}\left(\sqrt{n}-1\right)}{\lambda C} f_{NoC} U_{NoC} \qquad (30)$$

Simulations in [1] provide an example of small neural microcircuits (~1000 neurons) that reach optimal performance with:

$$\lambda \cong \sqrt[3]{n} \qquad C \cong \sqrt[2]{n} \qquad (31)$$

Substituting these values for unicast mesh:

$$BW_{eff,UC}^{RNDC} \cong \overline{w} \cdot \sqrt[6]{n} \cdot f_{NoC} U_{NoC} \qquad (32)$$

Using (28) and (31) in (10) we obtain:

$$f_{spike,\max}^{UC} = \frac{1}{10C \cdot T_{cycle}} = \frac{f_{NoC}}{10\sqrt{n}} \qquad (33)$$

Also,

$$f_{p,out}^{UC} = \frac{BW_{eff,UC}^{RNDC}}{n} = \left(1-\frac{1}{\sqrt{n}}\right)\frac{\overline{w}}{\lambda C} f_{NoC} U_{NoC}$$
$$= O\left(n^{-\frac{5}{6}}\right) \qquad (34)$$

Similarly, the K ratio is:

$$K_{UC} = \frac{f_{p,out}^{UC}}{f_{spike,\max}^{UC}} = \overline{w} U_{NoC}\left(1-\frac{1}{\sqrt{n}}\right)\frac{10C}{\lambda C}$$
$$= O\left(\frac{1}{\lambda}\right) = O\left(\frac{1}{\sqrt[3]{n}}\right) \qquad (35)$$

For RNDC NN emulated on a multicast NoC, the total number of hops is approximated by the length of the linear path, traversing a distance of 2λ to the first destination and then adding one hop per destination:

$$TotalDist_{Mesh,MC}^{RNN} = C + 2\lambda \qquad (36)$$

Thus, the bandwidth (3) of RNDC implemented on a multicast mesh NoC is:

$$BW_{eff,mesh,MC}^{RNDC} \cong \frac{2\overline{w}\sqrt{n}\left(\sqrt{n}-1\right)}{\left(C+2\lambda\right)} f_{NoC} U_{NoC}$$
$$= \frac{2\overline{w}\sqrt{n}\left(\sqrt{n}-1\right)}{\left(\sqrt{n}+2\sqrt[3]{n}\right)} f_{NoC} U_{NoC} = O\left(\sqrt{n}\right) \qquad (37)$$

Using (28) and (31) in (11) we obtain:

$$f_{spike,\min}^{MC} = \frac{f_{NoC}}{2\lambda} = \frac{f_{NoC}}{2\sqrt[3]{n}} \qquad (38)$$

Following (37):

$$f_{p,out}^{MC} = \frac{2\overline{w}\left(\sqrt{n}-1\right)}{\sqrt{n}\left(C+2\lambda\right)} f_{NoC} U_{NoC} = O\left(\frac{1}{\sqrt{n}}\right) \qquad (39)$$

Finally the K ratio is:

$$K_{MC} = \frac{f_{p,out}^{MC}}{f_{spike,\max}^{MC}} = \frac{2\overline{w}\left(\sqrt{n}-1\right)2\lambda}{\sqrt{n}\left(C+2\lambda\right)} U_{NoC}$$
$$= O\left(\frac{\lambda}{C+\lambda}\right) = O\left(\frac{1}{\sqrt[6]{n}}\right) \qquad (40)$$

Another practical case of random connectivity is the Locally Connected NN (LCNN) where:

$$LCNN = \left\{RNDC \mid \lambda \cong C \ll n\right\} \qquad (41)$$

Thus, since the connectivity is practically bounded,

$$f_{p,out}^{MC} = \frac{2\overline{w}\left(\sqrt{n}-1\right)}{\sqrt{n}\left(C+2\lambda\right)} f_{NoC} U_{NoC} \xrightarrow{C,\lambda \ll n} O\left(1\right) \qquad (42)$$

$$K_{MC} = \frac{\overline{w}\left(\sqrt{n}-1\right)4\lambda}{\sqrt{n}\left(C+2\lambda\right)} U_{NoC} \cong \frac{4}{3}\overline{w} U_{NoC} = O\left(1\right) \qquad (43)$$

Thus, the multicast mesh NoC offers sufficient bandwidth for emulating any size of locally connected NN. Both maximal firing frequency and the average NoC input frequency do not decrease when the network size grows. For instance, a NoC with one logical neuron in each processor is only 75% utilized even if all neurons fire constantly at their highest frequency.

With broadcasting NoC the spike is sent to all destinations regardless of connectivity pattern and the performance is independent of the neural network topology. Thus, all BC performance parameters are similar to those calculated for the Hopfield NN model (equations (22)–(25)). The remaining cost factors depend on neither the NN topology nor the communication method. They are the same as in (26).

Setting U_{NoC} constant for all communication methods (UC, MC, BC) we achieve same operating frequency and power consumption but different levels of throughput: MC provides the highest throughput using the same power. Table 2 summarizes asymptotic results for the RNDC model implemented on a mesh NoC using different communication methods (UC,MC,BC).

Table 2: RNDC with $\lambda \cong \sqrt[3]{n}$, $C \cong \sqrt[2]{n}$ on a mesh NoC

Parameter	UC	MC	BC
BW	$O\left(n^{\frac{1}{6}}\right)$	$O\left(\sqrt{n}\right)$	O(1)
Area	O(n)		
Power	O(n)		
Spiking Frequency	$O\left(1/\sqrt{n}\right)$	$O\left(1/\sqrt[3]{n}\right)$	$O\left(1/\sqrt{n}\right)$
K	$O\left(1/\sqrt[3]{n}\right)$	$O\left(n^{-\frac{1}{6}}\right)$	$O\left(1/\sqrt{n}\right)$

C. Tree NoC

Consider NoC with a binary tree topology, having n physical neurons at the leaves and n-1 routers as in Figure 2. The results can also be generalized to trees of higher degrees. The diameter of the binary tree is $2\log_2 n$. The total number of links is:

$$TL_{tree,NOC} = 2\left(n-1\right)\overline{w} \qquad (44)$$

In a Hopfield (full connectivity) neural network with unicast communications, half of all traffic passes through the root, resulting in a serious congestion. This issue has been addressed by fat-trees (FT) [2], in which link bandwidth increases when going upward to the root, maintaining more uniform traffic. We employ the approximation of FT of [3] (Figure 3), enabling nodes with small constant degree at the cost of more routers ($n\log(n)/2$) and more complex connectivity ($n\log(n)$ links). Such a FT introduces multiple alternative paths, providing additional bandwidth for short massages.
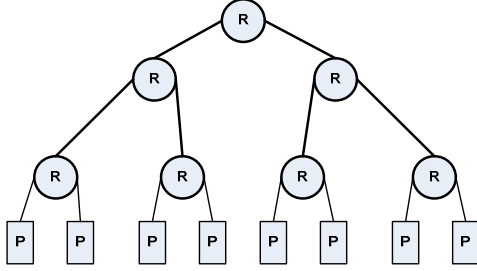

Figure 2: Regular binary tree with 8 leaves

Based on the mesh NoC analysis of Sect. III(B), we conclude that Hopfield NN is better emulated using either broadcast or multicast rather than unicast communications. Broadcast and multicast perform identically emulating Hopfield NN, and result in uniform traffic on the regular tree. For RNDC NN topology, recalling the exponential connection probability of (27), we investigate the question whether a FT is needed. The spikes sent from a neuron which is connected to other neurons at distance D traverse only a sub-tree of $k=\log(D)$ levels, or $k+1$ levels when connected to other neurons at distance $2D$ (a less likely case). The ratio of bandwidth required in levels $k+1$ and k is reflected by the ratio of the probabilities:

$$\frac{BW(k+1)}{BW(k)} = \frac{p(2D)}{p(D)} = \frac{e^{-2D/\lambda}}{e^{-D/\lambda}} = e^{-D/\lambda} = e^{-2^K/\lambda} \quad (45)$$

In a regular binary tree, the link capacity reduces by half at every level, $e^{-2^k/\lambda} = 0.5$, leading to:

$$k = \log_2(\lambda) - 0.52 \quad (46)$$

Thus, for connectivity with average distance 2λ, a sub-fat tree of height $k \sim \log_2(\lambda)$ provides the required communications, and the bandwidth of a normal binary tree suffices at the higher levels of the tree. In conclusion, the desired architecture combines a fat-tree at the bottom with a normal binary tree at the higher levels, depending on λ. To satisfy any value of λ, a full fat tree is required. Next we compare FT NoC performance to Mesh NoC. The number of links in the FT:
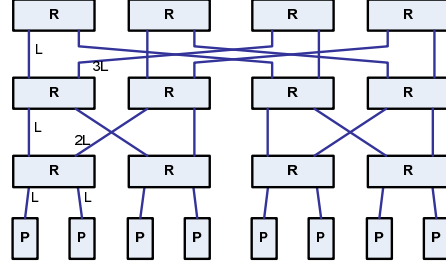

Figure 3: Fat tree topology example

$$TL_{FT,NOC} = n\log_2(n) \quad (47)$$

The average link length of the FT is computed as follows. Assuming layout as in Figure 3 and counting link length as the sum of horizontal and vertical distances, the total distances in the FT is:

$$\sum_1^{n\log n} l_i = l\left(n\left(\log(n) - \frac{1}{2}\right) + \frac{n^2}{4}\right) = O(n^2) \quad (48)$$

And the average length $\overline{l}_{FT\_NoC}$ is:

$$\overline{l}_{FT\_NoC} = \frac{1}{n\log n}\sum_1^{n\log n} l_i \cong l\left(\frac{n}{4\log(n)} + 1\right) \quad (49)$$

Recall that the average switching frequency is $f_{arch} \cong \dfrac{1}{R_0 C_0 \overline{l}^2}$. For FT NoC this yields

$$f_{FT,NoC} \cong \frac{16(\log n)^2}{R_0 C_0 l^2 n^2} \quad (50)$$

Fat Tree maximal switching frequency decreases quadratically with network size and it is $\sim n^2$ times smaller than the mesh switching frequency

$$f_{FT,NoC} \approx \frac{16(\log n)^2}{n^2} f_{Mesh,NoC} \quad (51)$$

The ratio of FT to mesh total distances (in hops) traversed by one spike is expressed by the ratio of diameters:

$$TotalDist_{FT,NoC} \cong \frac{\log(n)}{\sqrt{n}} TotalDist_{Mesh,NoC} \quad (52)$$

The total length (in number of hops) of the FT is:

$$TL_{FT,NoC} = n\log_2(n) = \log_2(n) \cdot TL_{Mesh,NoC} \quad (53)$$

Thus, the FT NoC bandwidth is:

$$BW_{eff,FT,NoC} = \frac{TL_{FT,NOC} f_{FT,NoC}}{TotalDist_{FT,NoC}^{NN}} U_{NoC}$$

$$\cong \frac{16(\log_2 n)^3 \sqrt{n} \cdot TL_{Mesh,NOC} f_{Mesh,NoC}}{n^2 \log_2(n) TotalDist_{Mesh,NoC}^{NN}} U_{NoC}$$

$$= \frac{16(\log n)^2}{n\sqrt{n}} BW_{eff,Mesh\_NoC}$$

$$\frac{BW_{eff,FT,NoC}}{BW_{eff,Mesh,NoC}} = O\left(\frac{(\log n)^2}{n\sqrt{n}}\right)$$

(54)

The area of FT is given by

$$A_{FT\_NoC} = TL_{FT,NoC} \cdot \overline{l}_{FT,NoC} \cdot \overline{w} \cdot W_p$$

$$= \overline{w} \cdot W_p \cdot l \left[\frac{n^2}{4} + n \log n \right] \quad (55)$$

And the power dissipation (using P_0 as in (26)):

$$P_{FT,NoC} = C_T f_{FT,NoC} V_{dd}^2 U_{NoC}$$

$$\cong 4 (\log n)^2 \cdot P_0 U_{NoC} \quad (56)$$

In summary, a fat tree is required for implementing RNDC on tree NoC, resulting in longer communication paths and consequently in lower frequency compared to mesh NoC.

D. AER Shared Bus

The AER shared bus naturally employs only broadcast communications. A neuron transmits address events (namely, a packet containing only its address) on the bus once it gains bus control. Each receiving neuron compares the source address with the addresses of the neurons to which it is connected. Following [20] the total length of a bus is $L_{BUS} = \dfrac{l(n-4)}{2}$ (Figure 4).

Each spike occupies the entire bus regardless of the topology of the emulated NN. Following (3), the bus effective bandwidth is:

$$BW_{AER\_BUS}^{NN} = \overline{w} \cdot f_{BUS} \cdot U_{BUS} \quad (57)$$

The bus operating frequency can be related to the mesh NoC frequency following (6) and (26):

$$f_{BUS} = \frac{1}{R_0 C_0 L_{BUS}{}^2} = \frac{4}{(n-4)^2} f_{NoC} \quad (58)$$

Likewise, the bus BW can also be expressed in terms of the BW of the MC mesh NoC (the preferred mesh NoC communication method):

$$BW_{AER\_BUS}^{Hopfield} \cong BW_{Mesh,MC}^{Hopfield} \cdot \frac{4}{(n-4)^2} \frac{U_{BUS}}{U_{NoC}} \quad (59)$$

Evidently, AER bus utilization is lower than NoC utilization, and is decreasing as the network size grows. Even when disregarding this and assuming $U_{BUS} \approx U_{NoC}$, it is evident that NoC effective bandwidth is about n^2 times higher than the AER bus bandwidth. Moreover, the latter is $n^{2.5}$ lower than the bandwidth of mesh NoC used for RNDC emulation in (37):

$$BW_{AER\_BUS}^{RNDC} \cong BW_{Mesh,MC}^{RNDC} \cdot \frac{4}{\sqrt{n}(n-4)^2} \frac{U_{BUS}}{U_{NoC}} \quad (60)$$

On the other hand, the area consumed by the AER bus is about twice smaller than the area for for NoC Mesh.
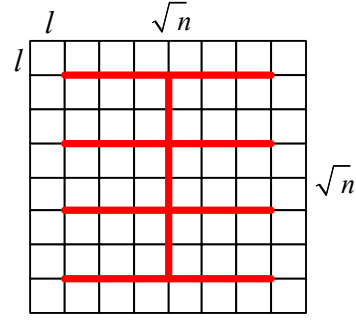

Figure 4: AER shared bus topology

The power dissipated by AER bus follows (7):

$$P_{BUS} = C_T f_{BUS} V_{DD}^2 U_{BUS} = P_0 \cdot \frac{2}{(n-4)} U_{BUS} \quad (61)$$

where $P_0 \triangleq V_{dd}{}^2 / R_0 d$. Comparing with (26), observe that both the AER bus and the mesh NoC dissipate roughly similar power, *O(1/n)*. In summary, the mesh NoC offers higher performance than the AER bus at the same cost.

E. Point to Point Architecture:

We consider *n* neurons arranged in a regular mesh and fully connected with point-to-point (PTP) unidirectional XY routed links. The total length of all PTP connections can be calculated by multiplying the total number of links $TL_{p2p} = n(n-1)/2$ by the average length of the link, $\overline{l}_{p2p} = 2l\sqrt{n}/3$

$$L_{P2P} = \frac{1}{3} l\sqrt{n} \cdot n(n-1) \quad (62)$$

The average frequency for P2P is:

$$f_{P2P} = \frac{1}{R_0 C_0 \overline{l}_{p2p}^2} = \frac{9}{4 R_0 C_0 l^2 n} = \frac{9}{4n} f_{NoC} \quad (63)$$

With Hopfield NN (full connectivity) every spike traverses *n*-1 links. The effective bandwidth is:

$$BW_{eff,P2P}^{Hopfield} = \frac{\overline{w} \cdot TL_{P2P} \cdot f_{P2P} \cdot U_{P2P}}{n-1} =$$

$$= \frac{\overline{w} \cdot n \cdot f_{P2P} \cdot U_{P2P}}{2} \quad (64)$$

It is intuitively evident that the P2P architecture can carry out *n/2* simultaneous transactions with average frequency of f_{P2P}. Similarly to the AER BUS architecture, P2P bandwidth is compared with the multicast mesh NoC implementation (22):

$$BW_{eff,P2P}^{Hopfield} = \frac{\overline{w} \cdot n \cdot 9 f_{NoC} \cdot U_{P2P}}{4n} =$$

$$= \frac{9}{4} \frac{U_{P2P}}{U_{NoC}} BW_{Mesh,NoC}^{Hopfield} \quad (65)$$

Observe that the P2P advantage of higher parallelism is balanced by the higher frequency of the NoC, so that they yield similar bandwidth.

Considering RNDC (random local connectivity) implementation on P2P, every spike traverses on average C links. Applying average connection length of $2\lambda l$ to (5) yields the average frequency of

$$f_{P2P}^{RNDC} = \frac{1}{4R_0 C_0 l^2 \lambda^2} = \frac{1}{4\lambda^2} f_{NoC} \qquad (66)$$

The effective bandwidth for RNDC implementation on P2P is

$$
\begin{aligned}
BW_{eff,P2P}^{RNDC} &= \frac{\overline{w} \cdot TL_{P2P} \cdot f_{P2P}^{RNN} \cdot U_{P2P}}{C} = \\
&= \frac{\overline{w} \cdot (n-1)n \cdot f_{P2P}^{RNN} \cdot U_{P2P}}{2C} = \\
&= \frac{\overline{w} \cdot (n-1)n \cdot f_{NoC} \cdot U_{P2P}}{8C\lambda^2}
\end{aligned}
\qquad (67)
$$

Comparing this to MC Mesh NoC architecture (37):

$$
\begin{aligned}
BW_{eff,P2P}^{RNDC} &= \\
&= \frac{\sqrt{n}\left(\sqrt{n}+1\right)\left(C+2\lambda\right)}{16C\lambda^2} \frac{U_{P2P}}{U_{NoC}} BW_{eff,mesh,MC}^{RNN}
\end{aligned}
\qquad (68)
$$

Using the assumption in (31),

$$
\begin{aligned}
BW_{eff,P2P}^{RNDC} &= \\
&= \frac{\left(\sqrt{n}+1\right)\left(\sqrt{n}+2\sqrt[3]{n}\right)}{16n^{\frac{2}{3}}} \cdot \frac{U_{P2P}}{U_{NoC}} BW_{eff,mesh,MC}^{RNN}
\end{aligned}
\qquad (69)
$$

$$
\xrightarrow{n \gg 1} \frac{\sqrt[3]{n}}{16} \cdot \frac{U_{P2P}}{U_{NoC}} BW_{eff,mesh,MC}^{RNN}
$$

Thus, while the effective bandwidth of Hopfield implementation is about the same for P2P and mesh NoC, RNDC bandwidth of P2P can be higher than the mesh NoC. The PTP area is:

$$A_{P2P} = L_{P2P}\overline{w} = \frac{l\overline{w}}{3}\sqrt{n} \cdot n\left(n-1\right) \qquad (70)$$

And P2P power dissipation is:

$$P_{P2P} \cong f_{P2P} \cdot C_T \cdot V_{dd}^2 U_{P2P} = \frac{3}{4}P_0\sqrt{n}\left(n-1\right)U_{P2P} \quad (71)$$

Comparison P2P and mesh NoC costs:

$$\frac{P_{P2P}}{P_{NoC}} \cong O\left(\sqrt{n}\right) \quad \text{and} \quad \frac{A_{P2P}}{A_{NoC}} \cong O\left(n\sqrt{n}\right) \qquad (72)$$

These results imply that although point-to-point architecture can provide better performance for certain Neural Network topologies than mesh NoC., it comes at a price of higher power dissipation and significantly higher area. The combined cost AP of P2P is n^2 times higher than the mesh NoC.

F. Summary of Cost and Performance

In the previous sections we have analyzed cost and performance of different interconnect architectures implementing neural networks. In Section B it is shown that multicast is preferred for NN emulation on mesh NoC, as summarized in Table 1 and Table 2. In Sections C—E we have analyzed the cost and performance of NoC Tree, AER shared bus and Point-to-Point connectivity and compared performance and cost to MC Mesh NoC, as summarized in Table 3.

It is evident from Table 3 that mesh NoC with MC communications is preferred for large-scale configurable VLSI implementation of neural networks. It offers the highest performance/cost ratio, provides a high bandwidth which, thanks to high level of parallelism, grows with the size of the network. Only the maximally parallel P2P provides a higher bandwidth than the mesh NoC, but at an extremely high cost. The shared bus and fat tree implementations are less favorable.

Another relevant cost item relates to the size of memory required for the routing tables. While detailed analysis is outside the scope of this paper, we have found that memory requirements are roughly similar in all architectures.

IV. PERFORMANCE SIMULATIONS

The mesh NoC architecture was simulated using UC, MC and BC in order to validate the analytical model and to gain insight into network behavior. We employed the NS2 Network Simulator [21] to investigate our neural networks. The link frequency f_{NOC}=1GHz. The network size n (n processors emulating one neuron each) was varied from 25 to 196 and higher in some cases. For each n, a MC and a UC mesh NoCs are simulated with both Hopfield and RNDC connectivity patterns, using Poisson firing rate. The realistic connectivity parameters of RNDC (31) were employed. For each simulated network, the firing rate was varied in search of a "knee point" (Figure 5), in which contentions became significant and the average network delay started to grow exponentially. For lack of analytical identification of that knee point, we place it at the point where the average delay doubles relative to its initial value. This metric is selected as the maximal firing frequency. We then examine the dependence of the maximal firing frequency on network properties, validating our analytical model.

Comparing Hopfield NN (top row of Figure 5) with RNDC (bottom row), we observe that the full connectivity of Hopfield networks comes at the price of higher average delay and lower achievable firing frequencies. Considering unicast (left hand side of Figure 5) versus multicast (right hand column), it is evident that multicast enables higher firing rates. Notice that while above the knee point the delays are intolerable, below it the delays seem constant. This validates the model assumption that the network effectively operates with no congestion. Notice further that if each processor emulates k neurons, then the maximum firing rate would decrease proportionately by k.

Table 3: Cost and performance comparison

	NoC Mesh MC	NoC Fat Tree	AER Bus	P2P
Hopfield BW	$O(1)$	$O\left(\dfrac{\log^2 n}{n\sqrt{n}}\right)$	$O\left(\dfrac{1}{n^2}\right)$	$O(1)$
RNDC BW	$O\left(\dfrac{n}{(C+\lambda)}\right)$	$O\left(\dfrac{\log^2 n}{\sqrt{n}(C+\lambda)}\right)$	$O\left(\dfrac{1}{n^2}\right)$	$O\left(\dfrac{(n-1)n}{C\lambda^2}\right)$
Practical RNDC BW	$O\left(\sqrt{n}\right)$	$O\left(\dfrac{\log^2 n}{n}\right)$	$O\left(n^{-2}\right)$	$O\left(n^{\frac{5}{6}}\right)$
Area	$O(n)$	$O\left(n^2\right)$	$O(n)$	$O\left(n^2\sqrt{n}\right)$
Power	$O(n)$	$O\left((\log n)^2\right)$	$O\left(n^{-1}\right)$	$O\left(n\sqrt{n}\right)$
perf/ cost Hopfield	$O\left(n^{-2}\right)$	$O\left(n^{-3\frac{1}{2}}\right)$	$O\left(n^{-2}\right)$	$O\left(n^{-4}\right)$
perf/ cost RNDC	$O\left(n^{-1\frac{1}{2}}\right)$	$O\left(n^{-3}\right)$	$O\left(n^{-2}\right)$	$O\left(n^{-2\frac{5}{6}}\right)$

The maximum firing rates of a Hopfield NN achieved for the simulated values on n is shown in Figure 6 (a). The firing rate achieved using UC is lower than MC and BC, and it decreases faster with n. The simulations indeed validate that the MC firing rate behaves approximately as $O(1/n)$ and UC frequency behaves as $O\left(1/n\sqrt{n}\right)$.

Similar results for RNDC NN are shown in Figure 6 (b). MC achieves higher spiking rate than both UC and BC, and the rate scales better with network size. The simulations indicate that MC spiking rate ranges between $O(n^{-0.5})$ and $O(n^{-0.8})$ depending on the specific connectivity pattern. BC spiking rate is $O(1/n)$, the same as for Hopfield NN, and UC scales slightly better than BC.

V. CONCLUSION

This paper presents theoretical analysis for determining a preferred interconnect architecture for general purpose configurable emulation of spiking neural networks. We show that a mesh NoC is preferred over other analyzed architectures (fat tree, point to point and shared bus). Further, it is shown that multicast communications outperform unicast and broadcast. Simulations successfully validate the analytical models and the asymptotic behavior of the network as a function of its size. The results may be extended to the more general case of address event transactions.

Future research may address the following open questions. Certain neural networks may be implemented efficiently on partly-reconfigurable NoCs, in which not all connectivity patterns are allowed. The architecture, mapping algorithms and memory size of such implementations should be studied. Another open issue relates to the architecture of the processors, and how it affects the NoC. Third, the architecture of multicast NoC routers may be

further optimized for emulating spiking neural networks. Multi-level hierarchical modular NoC architectures may be applied to enhance scalability of neural network emulations. Last, the NoC may be enhanced to emulate more precisely axonal delays and exact event timings.

VI. REFERENCES

[1] R. Legenstein and W. Maass, "Edge of chaos and prediction of computational performance for neural microcircuit models", *Neural Networks*, 20(3):323-334, 2007.

[2] C. Leiserson, "Fat-Trees: Universal Networks for Hardware-Efficient Supercomputing", *IEEE Trans. Comp.*, C-34(10):892-901, 1985.

[3] S. R. Ohring, M. Ibel, S. K. Das and M. J. Kumar, "On Generalized Fat-tree," *Proc. Int. Conf. Supercomputing*, pp. 276-285, 1995.

[4] R. Rojas, "Neural Networks: A Systematic Introduction," *Springer-Verlag, Berlin*, Ch. 13, pp. 338-371, 1996.

[5] W Maass, "Networks of spiking neurons: The third generation of neural network models", *Neural Networks*, 10(9), 1997.

[6] T. Morie, M. Nagata and A. Iwata, "Design of a Pixel-Parallel Feature Extraction VLSI System for Biologically-Inspired Object Recognition Method", *Proc. Int. Symp. Nonlinear Theory and its Application*, pp. 371-374, 2001.

[7] J. Schreiter et al., "Cellular pulse coupled neural network with adaptive weights for image segmentation and its VLSI implementation," *Proc. Int. Symp. Electronic Imaging: Science and Technology*, 5298, pp. 290-296, 2004.

[8] G. Indiveri, E. Chicca, and R. Douglas, "A VLSI reconfigurable network of integrate–and–fire neurons with spike–based learning synapses," *Proc. European Symp. Artificial Neural Networks*, pp. 405-410, 2004.

[9] C. Mayr et al., "Mapping Complex, Large-Scale Spiking Networks on Neural VLSI", *Int. Journal of Applied Science Eng. and Technol.*, 4(1), 2007.

[10] A. Mortara, E. Vittoz and P. Vanier, "A communication scheme for analog VLSI perceptive systems," *IEEE JSSC*, 30(6):660–669, 1995.

[11] K.A. Boahen, "Point-to-Point Connectivity between Neuromorphic Chips Using Address Events", *IEEE Trans. Circuits & Sysemst.*, 47(5), 2000.

[12] S.C. Liu et al., "Orientation-selective VLSI spiking neurons", *Neural Networks*, 14:629–643, 2001.

[13] J. Lazzaro and J. Wawrzynek, "A multi-sender asynchronous extension to the AER protocol," *Proc. Conf. Advanced Research in VLSI*, pp. 158–169, 1995.

[14] W.J. Dally and B. Towles. "Route Packets Not Wires: On-Chip Interconnection Networks". *Proc. 38th Design Automation Conf.*, pp. 684– 689, 2001.

[15] A. Jantsch, "Networks on Chip", *Workshop at the European Solid State Circuits Conference*, 2001.

[16] L. Benini and G. De Micheli, "Networks on chips: a new SoC paradigm", *IEEE Computer*, 35:70-78, 2002.

[17] T. Theocharides, G. Link, N. Vijaykrishnan and M.J. Irwin, "A generic reconfigurable neural network architecture as a network on chip", *Proc. Int. SOC Conf.*, p. 191, 2004.

[18] L.A. Plana, S.B. Furber et al., "A GALS Infrastructure for a Massively Parallel Multiprocessor". *IEEE Design & Test of Computers*, 24(5):454–463, 2007.

[19] R Emery, A Yakovlev and G Chester, "Connection-Centric Network for Spiking Neural Networks," *Proc. NOCS*, pp. 144-152, 2009.

[20] E.Bolotin, I.Cidon, R. Ginosar and A. Kolodny, "Cost Considerations in Network on Chip", Integration, 38(1):19–42, 2004.

[21] S. McCanne and S. Floyd, ns Network Simulator, http://www.isi.edu/nsnam/ns/

[22] N. Eisley and L.S. Peh. "High-level power analysis for on-chip networks," *Proc. Int. Conf. on Compilers, architecture, and synthesis for embedded systems*, pp. 104–115, 2004.

[23] U.Y. Ogras and R. Marculescu, "Analytical router modeling for networks-on-chip performance analysis", *Proc. DATE,* pp. 1096 – 1101, 2007.

[24] F. Gilabert , S. Medardoni , D. Bertozzi , L. Benini et al, "Exploring High-Dimensional Topologies for NoC Design Through an Integrated Analysis and Synthesis Framework", *Proc. NOCS,* pp. 107-116, 2008.

[25] A. Banerjee, R. Mullins and S. Moore, "A Power and Energy Exploration of Network-on-Chip Architectures", *Proc. NOCS,* pp. 163-172, 2007
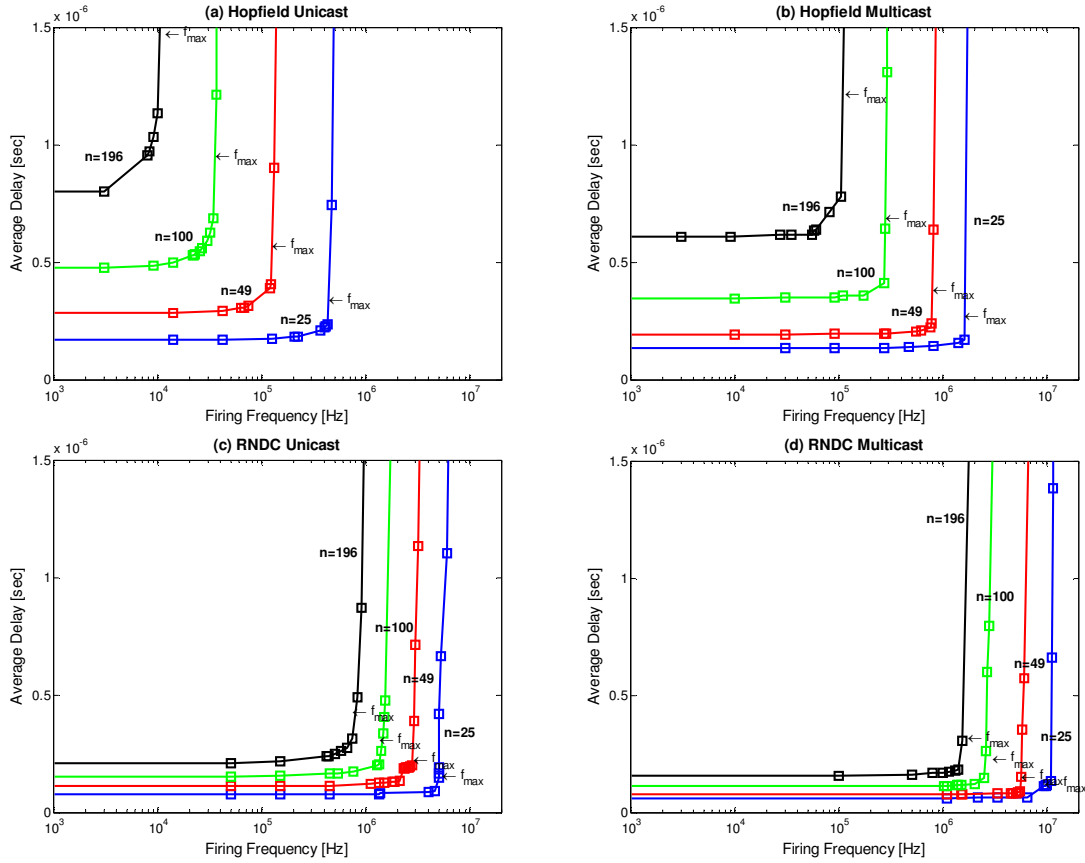
Figure 5: Average end-to-end delay vs. firing frequency on a mesh NoC: (a) Hopfield NN, unicast. (b) Hopfield NN, multicast / broadcast. (c) RNDC NN, unicast. (d) RNDC NN, multicast / broadcast.
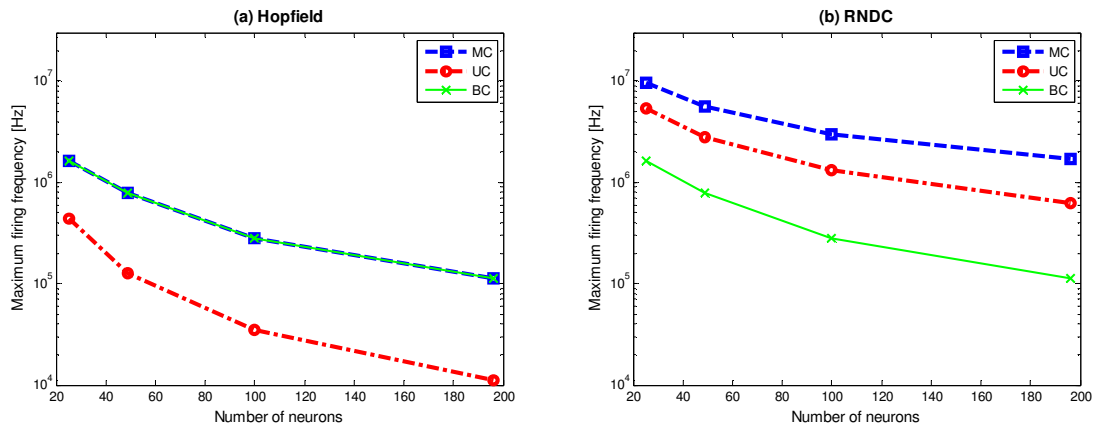

Figure 6: Maximal firing frequency vs. network size on a mesh NoC, comparing UC, MC and BC. (a) Hopfield NN. (b) RNDC