

Token-passing communication protocol in hardware based real-time spiking neural networks

B. Belhadj, J. Tomas, O. Malot, Y. Bomat, G. N'Kaoua and S. Renaud
IMS Labs, 351 cours de la libération, Bordeaux, France
Dept. of Microelectronics, Bordeaux 1 University

ABSTRACT

Biological neural networks are based upon axonal point-to-point connections which inspire connectionist architecture. As we attempt to engineer ever larger analogues of these neural networks we are forced to multiplex neural signals over time shared paths. This can alter timing of neural information, which is critical in real-time oscillatory networks. Because shared paths induce extra delay due to multiplexing signals, traveling on the channel and passing through routing devices, guaranteeing event arrival deadlines across the communication process becomes crucial. This paper addresses issues related to the guarantee of event timings with arbitrary deadline constraints in real-time distributed spiking neural network systems based on token-ring architecture. To achieve this objective, we propose an integrated method in selecting key system parameters. We show that several parameters must be set carefully if event deadlines are to be satisfied. The token holding time (THT) parameter controls the bandwidth allocation for each node in the token-ring network, and must be set properly to avoid deadline misses. The target token rotation time (TTRT) determines both the speed of token circulation and the network utilization available to nodes. TTRT should also be chosen carefully to ensure that the token circulates fast enough while maintaining a high available utilization. As proof of concept, the proposed method is applied to a multi-board spiking neural network system hosting up to 140 analog neurons spread across 7 circuit-boards. Experimental analysis shows that deadline constraints are guaranteed along with bandwidth allocation fairness when applying the proposed method.

Key words: Bio-inspired systems, spiking neural network, token-ring architecture, token-passing communication, real-time distributed systems

1. INTRODUCTION

There has been rapid progress in the development of techniques for modeling biological neural functions in hardware architectures. Simultaneous with these developments, the field of computational neuroscience has begun to yield new insights into the neurobiological substrates of pattern recognition, perception, cognition, memory and consciousness [1]. Neuromorphic systems engineering [2,3] emulates both structure and function of biological neural systems in silicon. To date, facsimiles of the initial stages of visual and auditory information processing have been implemented on single microchips [4-8]. However the complexity of neural computation beyond sensory perception requires a multichip approach and a proper communication protocol between chips to implement higher levels of processing and cognition [9].

Neuromorphic engineers have adopted time-division multiplexing to achieve massive connectivity in multichip systems, inspired by its success in telecommunications [10] and computer works [11]. Multiplexing leverages the 5-decade difference in bandwidth between a neuron (hundreds of Hz) and a digital bus (tens of MHz), enabling us to replace thousands of dedicated point-to-point connections with a handful of high-speed metal wires and thousands of transistors.

In adapting existing networking solutions, neuromorphic architects are challenged by huge differences between the requirements of computer networks and those of neuromorphic systems. Whereas computer networks connect thousands of computers at the building- or at campus-level, neuromorphic systems need to connect millions of neurons at a chip- or circuit-board level [13]. Hence, they must improve the efficiency of traditional computer communication architectures, and protocols by several orders of magnitude.

In [12], Culumciello and Androu propose a comparative study of access topologies for chip-level communication channels. Classical access techniques such as arbitration, scanning, ALOHA, and priority encoding are compared by

assessing throughput, latency and power consumption. They provide guidance in choosing the access algorithm for the appropriate bio-inspired application with efficient transmission of information and reduced latency. In [13], the author quantifies tradeoffs faced in allocating bandwidth, granting access, and queuing, as well as throughput requirements for arbitrated and non-arbitrated channels. However, to our knowledge, there is no study treating the token-ring access topology in neuromorphic systems, neither as an attractor topology to consider in the future communication systems nor as a discarded solution for weak performance. This work proposes to use token-ring based architecture to process neural data at chip- or circuit-board level. Inspired by traditional token-ring networks [18], we have adapted an existing token-ring access technique [19,20] to the context of neuromorphic systems.

The suitability of token-ring based architecture for embedded distributed real-time applications derives not only from its flexibility and simplicity to implement, but also from its property of a bounded access time. The bounded access time provides a necessary condition to guarantee real-time deadlines. The flexibility and the simplicity allow ease change in network parameters which can improve the responsiveness and the latency of the network. The network parameters must be set carefully to adapt the network responsiveness to the context of spiking neural network systems [14].

At network initialization time, a parameter called Target Token Rotation Time (TTRT) is determined which indicates the expected token rotation time. Each node is assigned a fraction of the TTRT. During this period, the node is permitted to transmit events every time it receives the token. The network parameter that represents the transmission time is called the Token Holding Time (THT). A proper selection of these parameters ensures that the token circulates fast enough while maintaining a high available utilization of the channel bandwidth.

Taking into account the increasing level of realism in neural simulation, this study quantifies the needs of real-time distributed spiking neural networks in terms of time precision, and proposes an integrated method in selecting appropriate network parameters. Section 2 outlines the framework of the study. Section 3 theoretically approaches a solution for selecting network parameters. It provides a complete bandwidth allocation scheme that guards against timing failure. A token-ring access algorithm is then proposed to convey neural information throughout the system. The last section describes our spiking neural network system emulating up to 140 spiking neurons spread across 7 circuit-boards. Experimental quantification of the proposed access technique shows that the system guards against deadline misses. No matter what happens (unless there is a network fault), events will be transmitted before their deadlines.

2. FRAMEWORK

This section describes the framework of our study. First, both network and message models are defined as well as their related constraints. Second, some of time properties of token-ring topology are addressed. These properties will be abundantly used in the subsequent section to define a method for network parameter selection. Third, a performance metric that has commonly been used for real-time processing and real-time communication is defined to determine the worst case achievable utilization of the network.

2.1 Network and message models

The network contains n nodes arranged in a ring. Each node hosts N neurons. Neurons generate events that will be encapsulated and transformed to messages in order to be conveyed into the communication channel. Outgoing messages at a node are assumed to be queued in FIFO order. It is also assumed that message transmissions are done via one-to-many broadcast to conserve bandwidth. The network is supposed to operate without any faults. The following parameters characterize the network.

- TTRT is the target token rotation time. This parameter indicates the expected token rotation time.
- τ is the token walk time. It includes the node-to-node delay and the token transmission time. τ is the proportion of TTRT that is not available for message transmission.

There are n streams of messages, S_1, \dots, S_n , with stream S_i incident on node i . S_i may contain two sort of messages: real-time-constrained messages and memory-constrained messages. The real-time-constrained messages are events (spikes with timestamps) exchanged between neurons in the network; while memory constrained messages represents the simulation data resulted from synaptic change variation. Each real-time-constrained message, or spike, must be redirected to the target neuron before the expiration of an arbitrary fixed deadline, which determines the required precision in the coincidence of spike occurrence. In contrary, memory-constrained messages do not have any real-time

constraint but instead they happen in a large number and, therefore, they need a large memory space allocation to be stored before their transmission.

Each stream S_i may be characterized as $S_i = (A_i, C_i, TH T_i)$, where

- A_i represents the activity of a node i . This parameter reflects the spike generation frequency of a given node i . It also reflects the offered load of real-time-constrained messages of the node i .
- C_i represents the afferent connectivity of the node i . This parameter serves to determine the number of synaptic changes that occur during simulation process and therefore, the number of memory-constrained messages present at the node i .
- $TH T_i$ is the token holding time of the node i . This parameter represents the amount of time for which the node i can transmit its messages.

We assume that all nodes have the same computation and communication power as well as the same time properties. Therefore, the following parameters are shared by all nodes:

- D is the relative deadline of real-time-constrained messages in the stream. The relative deadline is the maximum amount of time that may elapse between a message arrival and a completion of its transmission. Thus, the transmission of the j -th real-time message in stream S_i , which arrives at $t_{i,j}$, must be completed by $t_{i,j} + D$.
- D_M is the relative deadline of memory-constrained messages in the stream. This deadline represents the maximum amount of time allowing memory-constrained messages to stay into the buffer of outgoing messages awaiting their transmission. It is used to assess whether the buffer size of outgoing messages is enough or not.
- δ is the maximum amount of time required to transmit a message in the stream.

2.2 Constraints

The deadline constraint: In spiking neural networks, signals can be delayed getting onto shared channel, traveling on the channel, and passing through gateways or other routing devices along the channel. Each potential delay goes hand in hand with some increment in variability in arrival time at the destination. This can make subtle but real differences in event timing represented by the arrival time at the receiving neuron and, thus, interfere with the dynamics of the integration process being modeled for a certain type of cellbody. It can also interfere with the learning mechanism if it is based upon spike timing dependent plasticity rule requiring local information about the correlation of pre- and post-synaptic activity. To avoid the variability due to the communication delay, we define the deadline constraint. This constraint simply states that every message must be transmitted before its deadline. Formally, let $s_{i,j}$ be the time that the transmission of the j -th message in stream S_i is completed. The deadline constraint implies that for $i = 1, \dots, n$ and $j = 1, 2, \dots$,

$$s_{i,j} \leq t_{i,j} + D \quad (1)$$

where $t_{i,j}$ is the arrival time and D the deadline.

The bandwidth allocation constraint: This constraint states that bandwidths on all nodes must sum to less than the available network bandwidth. On the other words, the $TH T$ on all nodes must sum to less than $TTRT$ minus the token walk time,

$$\sum_{i=1}^n TH T_i \leq TTRT - \tau \quad (2)$$

2.3 Token visit number

In order to guarantee message deadlines at a node, it is necessary to have some information regarding the frequency of token visits to that node. Fortunately, extensive studies have already been carried out on the timing properties in token ring networks [15]. The generalized Johnson and Sevcik's theorem can be used to derive the following result.

In any interval of time D , the token will visit node i at least v_i times where

$$v_i = \left\lfloor \frac{D}{TTRT} - 1 \right\rfloor \quad (\text{the symbol } \lfloor x \rfloor \text{ refers to the floor of } x) \quad (3)$$

In each of these visits, node i can use its full $TH T_i$ to transmit its stream of m messages (if any).

This property will be used in the section 3 to determine the $TH T_i$ of each node as well as the optimal $TTRT$ value that optimizes the network responsiveness while maintaining high channel utilization.

2.4 Performance metric

To gauge the performance of the system, it is necessary to have an appropriate performance metric. A metric that has commonly used for real-time distributed systems is the worst case achievable utilization. Let us start by defining the effective utilization, U_i , of a message set of a node i . The effective utilization is also called the normalized offered load in the literature. In the context of spiking neural network systems, this metric may be defined as,

$$U_i = A_i \cdot (1 + C_i) \delta \quad (4)$$

where A_i is the frequency of producing real-time-constrained message and $A_i C_i$ the frequency of producing memory-constrained message in the node i . Thus, $A_i \cdot (1 + C_i)$ represents the frequency of the generation of outgoing messages in the node i . Multiplying this value by the transmission message time δ , the effective utilization U_i reflects the foreseen utilization percentage of the communication resources required for the node i . Equation (4) leads to the effective network utilization, U , by summing the utilizations at each node in the network,

$$U = \sum_{i=1}^n A_i \cdot (1 + C_i) \delta \quad (5)$$

A real-time communication protocol (with a given setting of its parameters) has an achievable utilization U' if it can meet the deadlines of any message set with utilization no more than U' . For example, if a network has an achievable utilization $U' = 0.5$, then all message sets with utilization $U \leq 0.5$ will have their message deadlines satisfied.

Consequently, to assess the performance of a scheme for choosing network parameters, we define the worst case achievable utilization U^* of a network as the least upper bound of the achievable utilizations. Hence, the network can meet the deadlines of all message sets with utilization no more than U^* . The worst case achievable utilization is usually determined when the system is submitted to the worst case situation. In section 4, we discuss the worst case situation that can happen in spiking neural network systems.

The importance of the worst case achievable utilization U^* is that it relates to the fundamental requirements of stability and predictability in hard real-time environments. If the utilization of a message set is no more than U^* , it can be predicted that all of messages will meet their deadlines. This is because the deadline of all message sets with utilization no more than U^* are guaranteed to be met. U^* also provides a measure of the stability of the system. The parameters of a message set can be freely modified while the utilization remains less than U^* . This gives a certain amount of system stability in the face of changes to message set parameters.

3. AN INTEGRATED METHOD FOR NETWORK PARAMETER SELECTION

In this section, we propose and analyze an integrated method for allocating bandwidth at each node i so that the time constraints of real-time-constrained messages are guaranteed to be met. Both $TH T$ and $TTRT$ parameters are derived from a local bandwidth allocation scheme. Based on the pre-determined network parameters, an access algorithm is then proposed to organize communication channel access.

3.1 Selecting $TH T$: a local bandwidth allocation scheme

As mentioned earlier, the selection of appropriate values of $TH T$ is a crucial step in meeting message deadlines. The node parameters (given by the A_i and C_i) and the network parameters (given by τ and $TTRT$) should be the dictating factors for the allocation of the $TH T_i$. We define an allocation scheme as an algorithm which, when given as inputs the values of all node parameters and network parameters, will produce as output the values of the $TH T_i$ to be allocated to the node i in the network. Formally, let the function f represent an allocation scheme. Then,

$$f(A_1, C_1, A_2, C_2, \dots, A_n, C_n, \tau, TTRT, \delta, D) = (TH T_1, TH T_2, \dots, TH T_n) \quad (6)$$

Allocation schemes in token ring networks may be divided into two classes: local allocation schemes and global allocation schemes. These schemes differ in the type of information they may use. A local allocation scheme uses only the information available locally to node i in allocating $TH T_i$. Locally available information at node i includes the parameters of stream S_i . τ and TTRT are also locally available at node i because these values are known to all nodes. On the other hand, a global allocation scheme can use global information related to the other nodes in its allocation of $TH T_i$. Global information includes both information locally available to nodes and external information regarding the parameters of message streams incident on other nodes. A local scheme is preferable from a network management perspective. If the parameters of the stream S_i on node i change, then only the $TH T_i$ need to be recalculated. $TH T$ s at other nodes need not change because they were calculated independently of S_i . This makes local scheme flexible and suitable for use in dynamic environments. Therefore, this paper focuses on local allocation schemes and proposes a dedicated scheme for spiking neural networks.

How should the $TH T_i$ be allocated? A message must be sent within D time units of arrival if it is to meet its deadline. Using the time property of token ring networks regarding the number of visits to node i in D time units, we have at least $v_i = \left\lfloor \frac{D}{TTRT} - 1 \right\rfloor$ visits. This suggests that for a message at node i to meet its deadline, the $TH T_i$ must be sufficient to send the message in v_i visits. On average, the number of messages arriving at a node in a given time interval must be equal to the number of messages that the node can transmit in the same interval. $A_i(1+C_i)\delta D$ can be loosely regarded as the offered load of the node i . Consider a time interval of length D , $A_i(1+C_i)\delta D$ is the traffic demand on node i during this interval. For the flow to be balanced, $A_i(1+C_i)\delta D$ must be transmitted in every interval of length D . Since the node can transmit v_i times during the time interval D , the following equation proposes an allocation scheme of $TH T_i$,

$$TH T_i = \frac{A_i \cdot (1 + C_i) \delta D}{\left\lfloor \frac{D}{TTRT} - 1 \right\rfloor} \quad (7)$$

The same equation (7) may be rewritten using the effective utilization of the node i defined in (4),

$$TH T_i = \frac{U_i D}{\left\lfloor \frac{D}{TTRT} - 1 \right\rfloor} \quad (8)$$

In the remainder of this section, we have chosen an allocation scheme that uses only local parameters to determine the amount of transmission time in each node. This scheme is adapted to the needs of spiking neural network systems. In the next section, we show how the optimal value of TTRT is derived from this allocation scheme.

3.2 Selecting TTRT : maximizing the worst case achievable utilization

The target token rotation time (TTRT) has a direct impact on the token circulation speed and network responsiveness. Therefore, it should also be chosen carefully to guarantee message deadlines. This section starts from the local allocation scheme proposed in (8) and the worst case achievable utilization defined in (4) to determine the optimal value of TTRT. The optimal value of TTRT is intended to match well the purpose of guaranteeing message deadlines.

When all nodes are fully busy (having the maximum number of messages to transmit), they use the maximum permitted time ($TH T_{max}$) to transmit their messages. According to the hypothesis stating that all nodes share the same time properties, $TH T_{max}$ is the same for all nodes. Referring to (2), $TH T_{max}$ is quantified by

$$TH T_{max} = \frac{TTRT - \tau}{n} \quad (9)$$

The case where all nodes use all of their maximum allocated bandwidth ($TH T_{max}$), must be the worst case that might happen in spiking neural network systems. Thus, the offered load on a node reaches its maximum, and the achievable utilization, U_i , becomes the worst case achievable utilization U_i^* . Hence, (8) and (9) lead to,

$$TH T_{max} = \frac{TTRT - \tau}{n} = \frac{U_i^* D}{\left\lfloor \frac{D}{TTRT} - 1 \right\rfloor} \quad (10)$$

(10), in turn, gives the worst case achievable utilization on the node i ,

$$U_i^* = \frac{TTRT - \tau}{nD} \left\lfloor \frac{D}{TTRT} - 1 \right\rfloor \quad (11)$$

Using (5), we deduce the worst case achievable utilization of the whole network,

$$U^* = \frac{TTRT - \tau}{D} \left\lfloor \frac{D}{TTRT} - 1 \right\rfloor \quad (12)$$

To determine the optimal TTRT value that goes hand in hand with the main purpose of the allocation scheme described in (7), the worst case achievable utilization of the network, U^* , described in (12) has to be maximized. The idea behind maximizing U^* comes from the fact that message deadlines are guaranteed for any achievable utilization smaller than U^* . On the other hand, maximizing U^* helps to cover the largest range of parameter settings. Thus, it is sufficient to assess system performance with the utilization U^* to draw a conclusion about constraint satisfaction.

Maximizing U^* remains to maximize the floor of $\frac{D}{TTRT} - 1$. That might be obtained only if $\frac{D}{TTRT}$ is an integer. In that case, (12) becomes

$$U^* = 1 - \frac{\tau}{TTRT} - \frac{TTRT - \tau}{D} \quad (13)$$

As we attempt to determine TTRT parameter when U^* is maximized, we derive (13) taking TTRT as the unique variable. As a result, U^* reaches its maximum at a certain value of TTRT, called $TTRT_{opt}$, determined by

$$TTRT_{opt} = \sqrt{\tau D} \quad (14)$$

Fig. 1 plots the worst case achievable utilization as a function of TTRT and that for arbitrary fixed values of the deadline D and the token walk time τ ($\tau = 0.8$). It is obvious that TTRT must be bounded by a range of values. In fact, from (13) it can be shown that when $TTRT = D$, the worst case achievable utilization is nil. Furthermore, it becomes negative for $TTRT > D$, which has no physical sense. Thus, the range of the TTRT parameter must be strictly less than the deadline D and strictly positive. For the proposed scheme, we assume that the choice of TTRT varies between 1 and $D/2$ units of time. Fig. 1 takes into account this upper bound and shows that the maximum worst case achievable utilization increases when TTRT increases. TTRT clearly has an impact on the U^* . From Fig. 1, it can be seen that when $D = 20$, $TTRT = 4$ gives a higher worst case achievable utilization than other plotted values of TTRT.

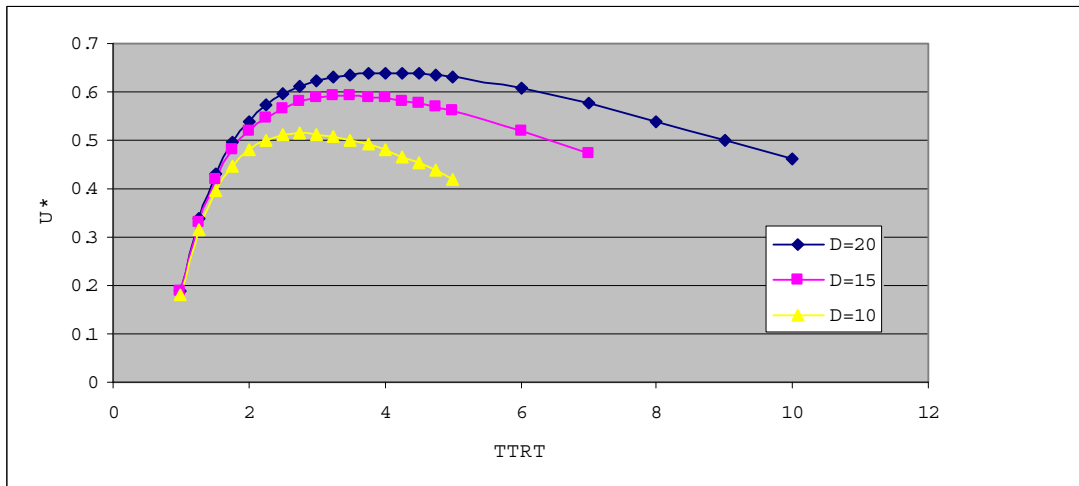


Fig. 1. The worst case achievable utilization U^* is plotted as a function of TTRT. Each curve is obtained for fixed value of the deadline D . The maximum value of the worst case achievable utilization reflects the optimal value of TTRT. It can be seen that for $D = 20$, U^* reaches its maximum for $TTRT = 4$. Thus, $TTRT_{opt} = 4$ units of time.

3.3 Satisfying the deadline constraint

This section discusses the satisfaction of both deadline and bandwidth allocation constraints in the proposed allocation scheme. Referring to (9), the bandwidth allocation constraint is satisfied if $TH T_i$ does not exceed $TH T_{max}$. This statement is always true because (8) produces values smaller than (10) in all cases. This is due to the fact that the utilization U_i is bounded by the worst case achievable utilization U_i^* . Thus, the bandwidth allocation constraint is met when using the allocation scheme defined in (7).

The deadline constraint is satisfied if the bandwidth allocation constraint is satisfied and bandwidths are allocated using the scheme in (7). Indeed, the allocation scheme in (7) guards against timing failure by allocating enough bandwidth to transmit all messages of a node within prefixed period of time. To be sure that the total bandwidth capacity is sufficient to guarantee message deadlines, the worst case achievable utilization must be less than 1. In that case, deadline constraint is satisfied.

3.4 Access algorithm

Using the bandwidth allocation scheme described in (7), the access algorithm may be merely a combination of countdown timers that represent the amount of time of network parameters.

- Token rotation timer of the node i ($TR T_i$). This counter is initialized to equal $TTR T$, and counts down until it expires ($TR T_i = 0$) or until the token is received and the time elapsed since the previous token departure is less than $TTR T$.
- Token holding timer ($TH T_i$). This counter is used to control the amount of time for which the node i can transmit messages.

Algorithm principles are simple since the allocation scheme uses only local information to determine transmission time. Whenever the node i receives the token, the countdown timer $TR T_i$ is reinitialized to equal $TTR T$. The node is allowed to transmit messages during its token possession time $TH T_i$ calculated using the scheme in (7). The priority is given to real-time messages to be sent first. If there are no real-time messages to transmit, the node starts sending memory-constrained messages until the node runs out of messages or $TH T_i$ expires ($TH T_i = 0$). An active node relinquishes the token if there is no message to transmit or the maximum allocation time ($TH T_{max}$) is reached. In the case when the $TR T_i$ expires and the node does not receive the token yet, real-time constraints remain not guaranteed anymore. This situation can cause deadline misses. The simulation is then immediately dropped and error message is raised informing the user that system misbehavior was detected. Fig. 2 illustrates an example of access algorithm operation.

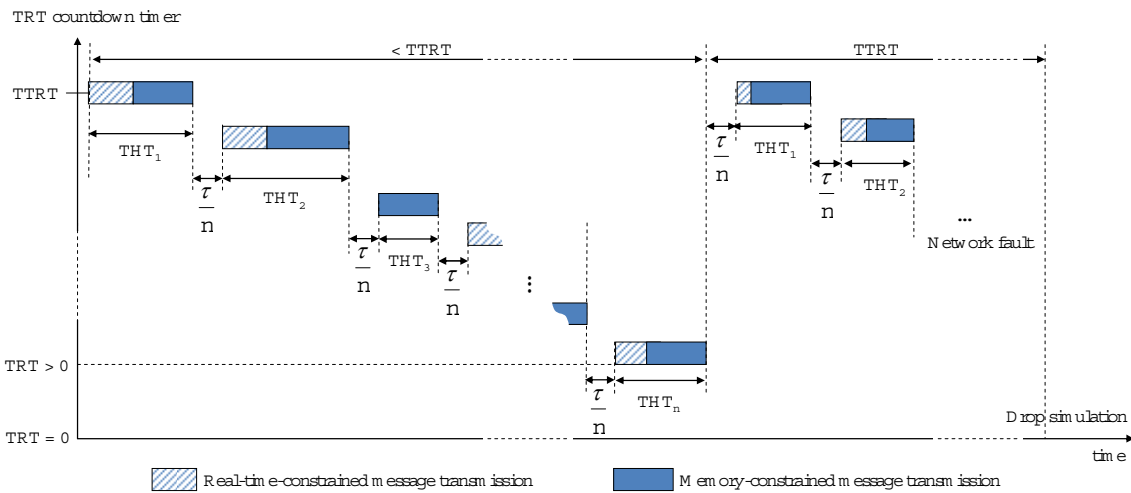


Fig. 2. Illustration of token circulation and message transmission: in the first token rotation cycle, the token goes through nodes and triggers the message transmission process. At the node level, real-time-constrained messages (if any) are transmitted before memory-constrained messages. Node 3 illustrates the case where there is no real-time message to transmit but memory-constrained messages. During the second token rotation cycle, a network fault happens (token does not reach the node 3). When $TR T_3$ expires, deadlines are not guaranteed anymore and the simulation is immediately dropped.

4. CASE STUDY : A MULTI-BOARD SPIKING NEURAL NETWORK

This section describes our multi-board system dedicated to emulate spiking neural networks. The system is designed to simulate adaptive neural networks with high degree of realism. To maintain the required accuracy during simulations, we apply the allocation scheme presented in previous sections to our system. After a brief description of the system, we describe the access topology with an illustration of its operation. Characteristics of communication media are given along with some insights of the transmission mode. Finally, a numerical analysis of the application of the local allocation scheme on our architecture is detailed, showing that message constraints are satisfied. This section does not mention the clock synchronization or detailed description of any fault situation.

4.1 System description

Fig. 3 shows a photograph of the global system. The current version of our system can host up to 140 neurons spread across 7 similar circuit-boards and can be extended up to 400 neurons over 20 boards all connected to a backplane with daisy-chain facilities. Each board is a six layers full-custom board which hosts 4 analog ASICs and one Xilinx Spartan3™ FPGA. Each ASIC incorporates 5 neurons which compute in analog mode conductance-based models following the Hodgkin-Huxley formalism [16]. Individual neurons produce in continuous time action potentials that express their intrinsic dynamic properties as well as their response to stimulations. Neurons are fully reconfigurable via parameters that select conductance-based ionic channels and characterize their response to current inputs [21]. Neuron type, firing rate and response to stimulus can be configured in each neuron. When the neuron output comparator detects an action potential, a digital 1-bit event is transmitted to the FPGA. In turn, the FPGA transmits the address of the firing neuron across the communication channel according to the token-ring access policy. In the meantime, the other FPGA of other boards scan incoming events and select addresses that have a connection with one or many local neurons. The system is designed to be flexible, providing reprogrammable connectivity. Input events are selected according to "virtual" connections stored in each FPGA. Finally, the FPGA computes synaptic changes following STDP rules [17], and generates a digital pulse whose width encodes the synaptic weight. This pulse triggers the transition to the opening state of the synaptic channels in each postsynaptic neuron.



Fig. 3. (a) The global system consists of a rack with a backplane connecting up to 20 boards. Inter-board communication is assured by FPGAs interfacing neurons of a given board to neurons located on other boards. (b) Each board contains 20 neurons spread across 4 analog ASICs, one FPGA, SDRAM memories and additional circuitry used to visualize neurons' membrane potential.

The network topology consists of 7 boards connected by point-to-point links forming a circle i.e., the token ring. A 1-bit pattern called the token circulates around the ring (from board i to boards $i+1, i+2, \dots$ until board 7, then to boards $1, 2, \dots$), granting permission to send messages (if any). Message transmission is done asynchronously over a 64-bit parallel bus. Inter-board communication operates via one-to-many broadcast to further conserve bandwidth. The motherboard gives the kickoff of the token circulation and controls simulation evolution by scanning the activity over the parallel bus. Fig. 4 illustrates system topology.

The parallel bus is composed of a handful of 64 metal wires. Each wire supports a throughput of 25M bits/s. Since the FPGA is in charge to transmit messages over the bus, it must respect the bandwidth capacity of metal wires. Each FPGA can transmit messages with a throughput of 100M bit/s which must be divided by 4 to match wire backplane capacity. Thus, the amount of time required to transmit one message is 40ns. Messages are then asynchronously transmitted over

the bus, which signify that there is no clock used to synchronize transmission between boards. This transmission regime causes problems related to sender/receiver synchronization because of clocks uncertainties (± 100 ppm in Spartan 3 FPGA oscillator). To guard against message reception failure, issuer FPGAs code neural information over a period of 40ns in such a way receiving FPGAs are sure to catch the right information. This asynchronous regime goes hand in hand with the bursty activity of spiking neural network.

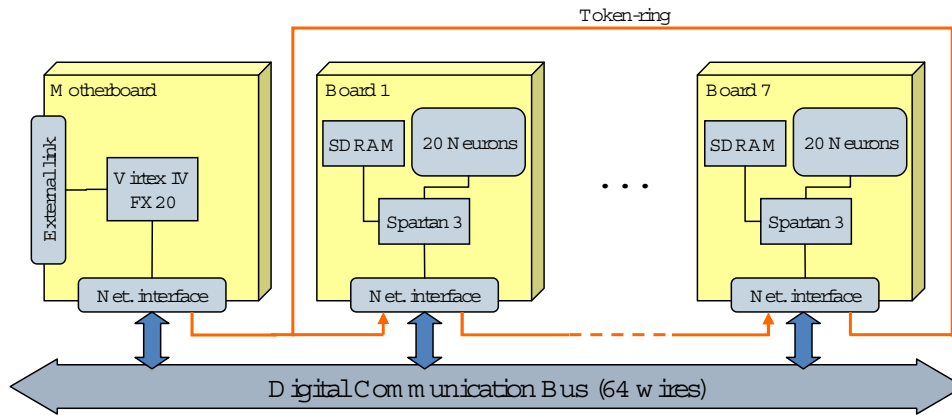


Fig. 4. A common 64-bit bus ensures data transmission, while a ring wire transfers a 1-bit token across boards. Each board communicates with the rest of the system via a network interface. An extra board, called motherboard, is in charge to control the beginning and the end of token circulation, as well as to detect misbehaviors. On the other hand, it links the whole system to an external workstation.

4.2 Application of the bandwidth allocation scheme

This section states the application of the proposed local allocation scheme in the case of our 7-board system. Network parameters as well as node parameters are expressed numerically. As a result, we show that deadline constraints are satisfied even though the worst case occurs.

The worst case situation in spiking neural networks: In spiking neural networks, when there is a priori knowledge that not all nodes are likely to produce events at the same time, simultaneous activity can happen though. Combined with "all-to-all" connectivity, simultaneous activity leads to the worst case situation. This situation floods the system with the largest number of events within a short span of time. Therefore, buffers of outgoing messages will be flooded by a large number of messages and the demand of communication resources increases instantly. In our system, neurons are configured to fire at maximum 100 spikes per second. Hence, the worst case situation can happen 100 times per second. The system is then periodically faced to the worst case situation (with a period of 10 ms). The communication policy must deal with this critical situation by guaranteeing message deadlines and providing enough buffer size to convey messages over the bus without any data loss. It is obvious that if deadline constraints are met when the worst case happens then they are met in all other possible cases. It is sufficient to assess constraints for the worst case situation to reach a conclusion about system performance.

Determining TTRT and $TH T_{max}$ parameters: Starting from the hypothesis that all boards have the same timing properties, we fixed a deadline $D = 20\mu s$ for all nodes. This value is determined so that the message arrival delay does not induce a big variability in the integration process being modeled for cellbody i.e. the impact of $20\mu s$ delay is judged tolerable. On the other hand, the token takes 40ns to move from one board to another leading to a token walk time of $\tau = 0.28\mu s$ ($40ns * 7$ boards). The TTRT parameter is then calculated according to (14) and it is equal to $2.37\mu s$. Consequently, we calculate $TH T_{max}$ using (10), and we obtain $0.31\mu s$ as a maximum time assigned to a node to transmit its messages.

Real-time-constrained message transmission during the worst case situation: Using the 64-bit bus, one message can encapsulate 3 events either time-stamped neuron addresses or synaptic changes. Knowing that $TH T_{max}$ is equal to $0.31\mu s$, the maximum amount of messages that might be transmitted during token possession time is 7 messages (40ns to transmit one message), which leads to a total of 21 events (7 messages $* 3$). As mentioned earlier, the worst case happens when neurons fire simultaneously. When that happens, 20 real-time events – corresponding to spikes of 20 neurons – are accumulated into the FIFO of each node. The node has then to transmit 7 real-time-constrained messages over the bus.

Thus, all real-time-constrained messages may be transmitted during one token rotation time. The deadline D is then met because $TTRT = 2.37\mu s < D = 20\mu s$.

Memory-constrained message transmission during the worst case situation: The worst case situation also implies that neurons are all-to-all connected. Since we calculate the plasticity of afferent events in each node, the offered load of memory-constrained messages is averaged to 54 messages (160 events) for each node. This amount of messages is released every 10ms. Having a buffer size of 1000 entries, we can store messages for $D_M = 60ms$. However, 54 messages require a little bit less than $8 * THT_{max}$ time to be transmitted, which leads to a maximum waiting time of outgoing messages less than $8 * TTRT = 18.96\mu s < D_M = 60ms$.

Satisfying deadline constraint: First of all, let us calculate the global amount of time required to transmit both real-time-constrained messages and memory-constrained messages generated in the worst case situation. We need one TTRT to transmit real-time-constrained messages and eight TTRTs to transmit memory-constrained messages. In total, we obtain a value approaching $9 * TTRT = 21.33\mu s < 10ms$. Messages are then transmitted quickly enough for the deadline constraint to be satisfied in the worst case situation. This result is also proved by means of worst case achievable utilization. The worst case achievable utilization at a node i is calculated according to (11) which leads to $U_i^* = 0.1095$. The worst case achievable utilization of the network, U^* , is obtained using (12) and worth 0.766. As a conclusion, deadline constraints are met if utilizations of all nodes are less than 0.1095, and the global network utilization does not overtake 0.766.

This numerical analysis concludes that using the bandwidth allocation scheme defined in (7) and respecting the deadline and bandwidth allocation constraints, deadline misses are avoided. This is due to the integrated method used to select network parameters THT and TTRT which offer high network responsiveness while maintaining a high available utilization.

5. CONCLUSION

This paper has proposed a token-ring access topology applied to the field of distributed real-time spiking neural network simulators. In such systems, handling event timings is crucial to avoid simulation misbehavior. We have shown that event deadlines can be guaranteed by selecting the network parameters – the bandwidth allocation time (THT) and the target token rotation time (TTRT) – in an integrate fashion. We have also proposed a local bandwidth allocation scheme that utilizes the information regarding the network activity, network connectivity and real-time message deadline in calculating THT of each node. This allocation scheme does not compromise the simplicity of the protocol implementation since it uses local node information in communication media access control. Practically, we have shown a simple access algorithm that may be easily implemented on multichip approach systems.

ACKNOWLEDGEMENT

The work described in this paper has been supported by FACETS EU project under grant FP6-IST-FETPI-2004-15879.

REFERENCES

- [1] Crick, F. and Koch, C., "Towards a Neurobiological Theory of Consciousness," *Seminars in the Neurosciences*, 263-275 (1990).
- [2] Mead, C., [Analog VLSI and Neural Systems], MA: Addison-Wesley, (1989).
- [3] Lande, T., Ed., [Neuromorphic System Engineering-Neural Networks in Silicon], Norwell, MA: Kluwer, (1998).
- [4] Koch, C. and Li, H., [Vision Chips: Implementing Vision Algorithms with Analog VLSI Circuits], Los Alamitos, CA: IEEE Computer Press, (1995).
- [5] van Schaik, A., Fragniere, E. and Vittoz, E., "Improved silicon cochlea using compatible lateral bipolar transistors," *Advances in Neural Information Processing Systems*, 671-677 (1996).
- [6] Kumar, N., Himmlbauer, W., Cauwenberghs, G. and Andreou, A., "An analog VLSI chip with asynchronous interface for auditory feature extraction," *IEEE Transactions on Circuits Syst. II*, 600-606 (1998).

- [7] Zaghboul, K. A. and Boahen, K., "Optic nerve signals in a neuromorphic chip II: Testing and results," *IEEE Transactions on Biomedical Engineering*, 667-675 (2004).
- [8] Kamada, S. and Yagi, T., "An analog silicon retina with multichip configuration," *IEEE Transactions on Neural Networks*, 197-210 (2006).
- [9] Choi, T. Y. W., Shi, B. E. and Boahen, K. A., "An ON-OFF orientation selective address event representation in a transceiver chip," *IEEE Transactions on Circuits and Systems I*, 342-353 (2004).
- [10] Schwartz, M., [Telecommunication networks: Protocols, Modeling, and Analysis], Addison-Wesley, Reading, MA, (1987).
- [11] Tanenbaum, A. S., [Computer Networks, Prentice-Hall International], Upper Saddle River, NJ, 2 edition, (1989).
- [12] Culurciello, E. and Andreou, A. G., "A comparative study of access topologies for chip-level address-event communication channels," *IEEE Transactions on Neural Networks*, 1266-1277 (2003).
- [13] Boahen, K. A., "Point-to-point connectivity between neuromorphic chips using address-events," *IEEE Transactions on Circuits & Systems*, 100-117 (1999).
- [14] Ichishita, T. and Fujii, R. H., "Performance evaluation of a temporal sequence learning spiking neural networks," *IEEE Transactions on Computer and Information Technology*, 616-620 (2007).
- [15] Sevcik, K. C. and Johnson, M. J., "Cycle time properties of the FDDI token ring protocol," *IEEE Transactions on Software Engineering*, 376-385 (1987).
- [16] Renaud, S., Tomas, J., Bomat, Y., Daouzli, A. and Saighi, S., "Neuromimetic ICs with analog cores: an alternative for simulating spiking neural networks," *ISCAS*, NO. 3, 3355-3358 (2007).
- [17] Belhadj, B., Tomas, J., Mabt, O., N'kaoua, G., Bomat, Y. and Renaud, S., "FPGA-based architecture for real-time synaptic plasticity computation," *ICECS conference*, Malta, 93-96 (2008).
- [18] Held, G., [Token-ring networks: characteristics, operation, construction and management], ISBN-13: 978-0471940418, (1993).
- [19] Malcolm, N. and Zhao, W., "Guaranteeing synchronous messages with arbitrary deadline constraints in an FDDI network," *IEEE Computer*, 186-195 (1993).
- [20] Malcolm, N. and Zhao, W., "The timed-token protocol for real-time communications," *IEEE Computer*, 35-41 (1994).
- [21] Bury, L., Saighi, S., Giremus, A., Grivel, E. and Renaud, S., "Parameter estimation of the Hodgkin-Huxley model using metaheuristics: application to neuromimetic analog integrated circuits," *IEEE Biomedical Circuits and Systems conference*, 173-176 (2008).