Sliding Time Interval based MAC Protocol and its Temporal Validation

Karen Godary-Dejean, David Andreu, Guillaume Souquet

Sliding Time Interval based MAC Protocol and its temporal validation.

K. GODARY[1], D. ANDREU[1,2], G. SOUQUET[1]

[1] University of Montpellier 2, LIRMM, Montpellier, France
[2] INRIA, DEMAR project, Sophia Antipolis, France

Abstract: In the context of distributed systems, the communication requirements are very different depending on the supported application, the system topology and the environment. The functional electrical stimulation is a critical and real time application domain: communications have to be safe (no loss, neither long nor unexpected delay). Therefore, an important part of this system from an efficiency point of view is the medium access mechanism. To fit with the specific constraints of our context, a new MAC protocol has been designed: STIMAP (Sliding Time Interval based Medium Access Protocol). This article presents a formal validation of this new protocol, allowing the validation of its behavior in an exhaustive way.

Keywords: MAC protocol, formal modeling, validation, medical application

1. INTRODUCTION

Functional electrical stimulation (FES) is a palliative solution to improve the daily life living of para- and quadriplegic patients; it has been successfully used in numerous applications such as pacemakers, deep brain stimulation, pain control or hearing restoration. Implanted electrical stimulation is also studied for movement rehabilitation such as foot droop for hemiplegic patients or even more complex movements, as well as for restoration of bladder function. Solutions for implanted FES are actually mainly based on centralized architectures, except the Bion implant (Loeb, et al., 2001). They lead to complex surgery, high risk of failure during and after surgery, and global infection problems involving the whole explantation of the device (Guiraud, et al., 2006). Indeed, using multipolar electrodes favors the selectivity in fibers recruitment but induces the connection of a lot of wires from electrodes to the central implant (one wire ofr each pole of each electrode). We thus developed a new architecture for such implanted FES system based on a network of distributed stimulation and measurement units. The main drawbacks of the centralized architecture can thus be limited to minimally invasive surgery, lower failure probability and local infection issue. Indeed, whatever the number of poles and electrodes are only one 2-wire bus has to be implanted to connect all units (a 35N LT metal based implantable 2-wire medical cable, not presented in this paper). Connections of poles to distributed units being thus limited on short distance, as a unit is implanted near its multiplar electrode. Moreover, such a distributed FES architecture favors new possibilities such as evolving architecture, complex multi-polar multi-phasic stimulations (Stieglitz, et al., 2005) that can be performed neither by the existing centralized implant nor by the Bion implant.

In this paper, we deal with the communication part of the distributed FES architecture (SENIS) we developed for activating and monitoring the peripheral nervous system. More precisely, we focus on the medium access mechanism used in this architecture: a new MAC protocol, called STIMAP (Sliding Time Interval based Medium Access Protocol) (Guiraud, et al., 2004). This protocol has been designed to fulfill the context specific requirements, which are not entirely satisfied with existing MAC protocols, like simplicity, reliability and efficiency. The behavior of this protocol has been validated by means of a validation methodology which has ever been used for the validation of the communication level of the Time-Triggered Architecture (Kopetz and Bauer, 2003), an embedded critical architecture for automotive (see (Godary, et al., 2004) or (Godary, 2004) for more details of these validation works). This methodology allows properties verification through an exhaustive analysis of the whole possible states of a formal model of the system.

After the presentation of the context in section 2, section 3 is devoted to the study of STIMAP, the medium access strategy we propose. The three following sections are dedicated to the validation of this protocol: section 4 presents the methodology of validation, section 5 describes the model of the system and section 6 presents the results. Then the last section concludes and presents future works.

2. CONTEXT

In order to explain the expected characteristics of the protocol which will be presented (section 3), we first briefly introduce the communication aspects of the context for which it has been designed. Even if the proposed protocol is presented in the context of functional electro-stimulation, it can be exploited in other kinds of applications (Bernard, et al., 2006).

2.1. FES application domain

The SENIS architecture is constituted by a global controller and several distributed stimulation units (DSU) and measurement units (DMU) which are connected together via an intra-body communication network (a two wire bus at present as no intra-body wireless solution exists yet). The bus topology of the network is essential: indeed, for the approach to remain valid for both wire and wireless busses, we assume that the medium fundamentally corresponds to one (shared) domain of (not detectable) collision.

The communication module embedded in a DSU must manage all the exchanges between DSU and the controller, via the communication bus. These exchanges can be of different types: download of micro-programs, start request, error notification, and also different messages dedicated to DSU configuration. Of course the exploitation of the medium must be reliable and deterministic, *a fortiori* in this context of artificial devices that "control" natural organs. The communication module is based on a 3-layers protocol stack, according to the reference given by the structure of the (reduced) OSI model. These layers are: the Application-layer, the Medium Access Control-layer (MAC) and the Physical one. Doing so, the modularity of the protocol stack allows to easily modify any layer: what we have already done on the application layer by adding new types of application requests, as well as on the physical layer by using an existing (not implantable) wireless technology to develop extern wireless stimulation.

This protocol stack is presented according to a top-down approach.

2.2. Application layer

The application protocol is mainly dedicated to the management of the micro-machine embedded in the DSU, by modifying micro-programs and by piloting the activity of the micro-machine (specific very small processor that runs micro-programs written in a dedicated reduced instruction set). But it also supports error notification from DSU. Some of the possibilities given by this protocol are:
- Micro-program modification packets: they allow to download/modify/upload/erase the micro-programs to be executed by the micro-machine.
- Micro-machine piloting packets: they correspond to different requests as start/stop/rearm /commute, sent to the micro-machine.
- Error notification from DSU: those errors are mainly relative to the respect of physiological constraints whose verification is directly ensured within the DSU (by means of reference models specifying for instance the maximum stimulus duration, the maximum stimulus magnitude, the minimum discharge phase duration, etc.). Errors can also be relative to micro-programs (endless micro-program for instance) or to request inconsistent with the DSU state (e.g. request to start the stimulation on a DSU whereas no micro-program has been downloaded).

2.3. Medium Access Control layer

Only expected characteristics of the MAC method are exposed here, the proposed method being detailed next section.

On this network, we use two types of logical addresses, allowing unicast, multicast and broadcast communications: individual addresses (one for each DSU) and group addresses (for different groups of DSU, and all DSU). These addressing modes are necessary since the controller can communicate for example with a single DSU to program it, or with a group of DSU to start a stimulation, or with all DSU to stop any stimulation. The notion of group is significant in our context since at a given instant of time the movement control only concerns a subset of muscles (simultaneous stimulation of agonistic and antagonistic muscles for example) and thus a subset of DSU (those associated to these muscles). This implies that it is possible to dynamically impose to a DSU to subscribe/unsubscribe to one or several groups. The possibility to dynamically unsubscribe a DSU from a group is also important, particularly for medium sharing efficiency purposes.

Consequently, basic functionalities of the MAC layer are to filter incoming packets, since at the physical layer we systematically broadcast frames, and to manage subcription/unsubscription to groups (a DSU can presently be member of maximum 8 groups). But the most important functionality is to ensure a deterministic and efficient medium sharing. It must ensure that solely one node of the network uses the medium at any instant of time, to avoid any collision. The risk of collision must be avoided to be sure that no error notification neither any request can be lost, even if for reliability purpose acknowledgements can be used. Two types of acknowledgements are provided: frame reception (MAC layer) and request execution (Application layer) acknowledgements.

2.4. Physical layer

It permits to receive frames coming from the physical medium, to verify the frame validity (often based on a Cyclic Redundancy Check), to extract data contained in the frame and to stock them while waiting for their treatment by the upper layer. It can also emit data coming from the upper layer, ensuring the encoding of the

frame (and the calculation of the CRC). Several physical medium have been used, like Ethernet (used in conditions such that the CSMA-CD method does not impact the qualitative tests performed), implantable 2-wire bus and 2.4 GHz low consumption RF transceiver. This layer won't be presented in more details.

2.5. Characteristics summary

The topology is fixed; the system topology is not lead up to change while running (no mobility neither dynamic node insertion). It is a bus topology on which we assume that there is no collision detection. The application needs two kind of logical addresses: unicast and multicast (broadcast being a particular multicast address). The set of nodes (group of DSU for instance) that communicates at the different phases of the stimulation is dynamically set; moreover it must be possible to exclude or suspend members of the communicating group, at any time. The application obviously requires a reliable and efficient communication medium; for reactivity, and security purposes the global controller (of our application) must have the possibility to quickly react, i.e. the possibility to quickly emit its requests without waiting for a long time before acceding to the medium (cf. section 2). The context imposes the solution to be simple as it must be embedded in small implantable electronic devices. Moreover this solution must be based on independent communication devices, as Ethernet couplers are; this is important for the system evolutivity.

3. MEDIUM ACCESS STRATEGY

3.1. Existing approaches

There is a plethora of existing MAC methods, used for LAN, WLAN, WPAN, field buses, sensors networks and body sensors networks. We only position the proposed MAC method according to main MAC strategies (principles). An important constraint being that we must rely on small size, and low power consumption, technological solutions as it has to be embedded within implantable devices.

First, the critical context and the impossibility to detect collision on the medium prevent to use medium access strategies based on competition like CSMA_CD in Ethernet (IEEE 802.3 norm). Indeed determinism is necessary to avoid (prevent) collisions and guaranty reliability of the communications. Solutions exist to minimize the collision risks, like for instance CSMA/CA (IEEE 802.11 norm), but this approach is too complex to be embedded in a small implantable device. Contention-based solutions are not adequate to our context.

Second, strategies based on static sharing of the medium imply either (static) frequency division (FDMA), or (static) code division (CDMA), or (static) time division (TDMA). Even if research activities on implanted antennas for medical wireless communication systems are important (Rahmat-Samii & al., 2006) (Falcon, 2003), there is not yet available solution for intra-body wireless network; FDMA based solution are barely conceivable at present. CDMA based solutions have too high computational requirements to be used on implanted devices. TDMA-based solutions, for wired-buses like in TTP[1] or wireless ones like in sensors networks, could be used in our context. In TTP for instance, rigidity of static sharing should be improved in order to be able to support dynamical management of group communication. Moreover, this solution is based on a global system view which is assured by too complex mechanisms. In sensors networks, solutions are logically motivated by power constraints, i.e. maximizing node/network lifetime due to low power capacities of these wireless sensors; thus, the other performance criteria are often secondary objectives. S-MAC for example (Ye & al., 2004), is based on scheduled synchronizations and periodic sleeps; the lack of efficiency of constant predefined periods and the potential high latency are important drawbacks (for our context).

Solutions based on a combination of previously mentioned principles (TDMA, CSMA, etc), like in ZigBee (IEEE 802.15.4 norm) for example, are often complex ones as they also address other issues like multi-hop exchanges, contention and contention-free periods, etc. Even if they sometimes guarantee time-slots within free contention periods, those solutions are not fully deterministic, and time performances are often not adequate for stimulation purposes.

At last, in the election-based MAC strategy, the solution for deterministic access is the "token" assignment: the medium access "right" is represented by a token moving around the network, like Token Bus (IEEE 802.4 norm). The token is transmitted to the nodes one after the other, which is neither efficient nor adapted to the context of broadcast groups. Another solution for the election-based strategy is the master/slave approach, which is an interesting concept for us because the topology imposed by our application context is clearly closed to the centralized control. The master is the central point of the system, initiating all the communications. One slave transmits a frame only if the master has demanded or authorized it. The master/slave approach seems to be the most appropriate, like that of the fieldbus WorldFIP (European Standard EN50170). But it has two drawbacks: it requires a "scheduler of traffic" and slaves must be individually polled. Thereby, this approach can be used but the limited efficiency due to the protocol overhead must be improved, taking into account the broadcast group characteristic.

---

[1] TTP/C specifications, www.tttech.com

So, we propose a new MAC protocol: STIMAP.

3.2. STIMAP

A (simple) deterministic medium access is ensured by the master-slave model of cooperation, the controller (of our application) being thus the master. Doing so, we can be sure that no DSU will emit any frame if not authorized to do so. However this model of cooperation is not really efficient when dealing with a group of slaves (DSU) as it requires to poll slaves, i.e. to individually communicate with each one.

We defined (and patented) a method based on the master-slave model which has been modified for efficiency considerations. It obviously allows basic individual master-slave (and slave-master) communications, but offers a way to manage the communications with a group of slaves without polling all the members of the group. The controller manages the access of the slaves to the medium by means of a 'Speaking Right', similar to a token, it allocates to the members of the network; we distinguish Individual Speaking Right (ISR, individual token) and Group Speaking Right (GSR, group token).

The DSU which receives an ISR has the authorization to emit one or several packets without time limit. Theoretically, only the DSU knows when it has finished using the ISR. But practically, knowing all the possible exchanges of our application, the DSU normally emits only one packet (every response to the master requiring only one packet) apart from the case of a micro-program upload (i.e. the master wants to get the micro-program). Even in that case, at most two packets will be emitted (the longest possible micro-program requiring only two frames to be transmitted). The following figure 1 shows these two cases: first the controller sends an ISR to DSU_a which emits one frame, and seconds the controller sends an ISR to DSU_b which emits two frames.
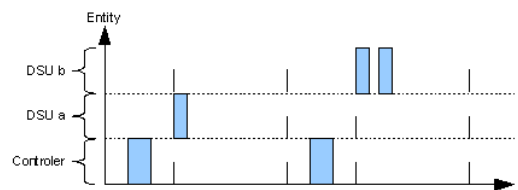

Fig 1. ISR communication

The ISR based mode is trivial, contrary to the GSR based mode that we expose now in more details.

3.3. Sliding Time Interval mechanism for group medium access sharing

When dealing with a group of slaves (DSU), each group member (each DSU) knows the size of the group and its position in that group as this position is defined in term of 'priority' (a DSU can be member of different groups and can have different positions in these groups). When the controller allocates a GSR to a given group, it sends only one frame (this frame is "physically" broadcasted, but only dedicated to the members of the indicated group). In this frame, it also indicates which member must begin the communication (i.e. it is not necessarily the member of highest priority that begins to send its packet). Knowing its position in the group, each member determines when it will have the right to emit its packet. Each member can speak during a given time interval that has been automatically computed by the controller (but that could also be specified and not computed), and indicated to all the slaves in an initial phase taking into account the network characteristics (e.g. the maximum of the Round Trip Times between the master and each slave). Figure 2 represents a GSR communication: 5 DSU are members of the group. The DSU with the priority 0 (DSU_a) begins to emit after the reception of the GSR frame sent by the controller; the others DSU emits one after the other on their turn, depending on their priority.
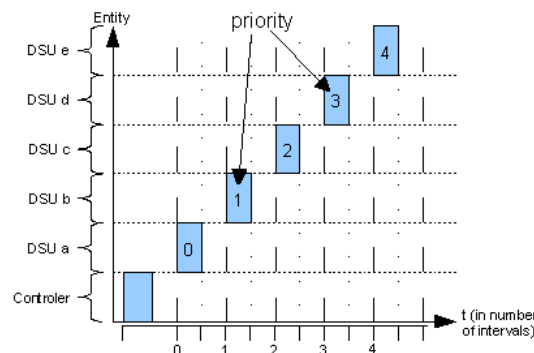

Fig 2. GSR based communication without sliding

The time interval attributed to a slave is constituted by two half-intervals: the first half-interval is dedicated to the slave communication, and the second one is reserved for a potential reaction of the master. For instance, if the slave notifies a significant error the master could have to immediately react (to stop the stimulation for example). So, to be sure that it will have access to the medium without any collision risk, the second half-interval is reserved.

This contributes to the reactivity of the distributed stimulation architecture. However, if a slave has nothing to transmit, the half-interval reserved for the controller is lost. To avoid that, i.e. to reach better performances, the MAC method integrates a time interval sliding mechanism. When waiting for its turn (i.e. its time interval), every slave listens to the medium: if the preceding member of the group did not emit a packet then it brings backward its time interval by a half time interval. In other words it recovers the half time interval that was reserved for the controller but that will never be used.

In fact, the sliding rule can be dynamically configured by the controller. The three possibilities (the sliding rules), which must be exploited in a coherent way by the controller, are:
- Sliding in case of "unused time interval": it corresponds to the case previously exposed. For example, this rule is selected when the controller asks a DSU group to notify their potential error detection. So if a DSU does not have any error to notify, it does not emit a frame and the next group member can recover the second half-interval because the extern controller will not have to react.
  An example is given on figure 3 showing that the DSU_d begins to emit its frame after 2.5 time intervals instead of 3 time intervals (as it is the case in figure 2).
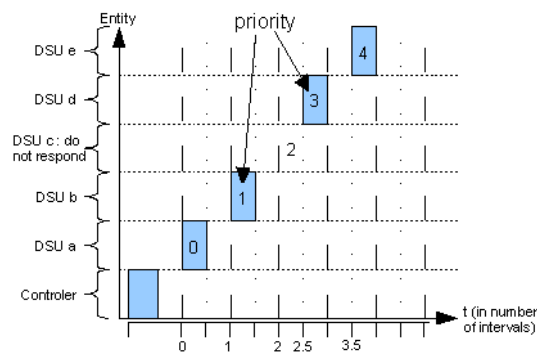

Fig 3. GSR based communication, with "unused time interval" sliding rule.

- Sliding in case of "used time interval": it corresponds to a situation in which the controller must not respond if the DSU emits a frame. For example, this rule is selected when the controller performs a test of presence on a DSU group. If a DSU responds, the second half-interval can be recovered since the controller would not have to react.
  An example is given on figure 4 showing that the DSU_b begins to emit its frame after 0.5 time interval instead of 1 time interval (as it is the case in figure 2).
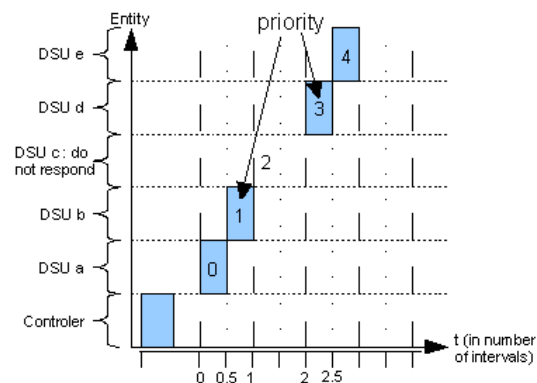- No sliding: the controller wants to always be able to react. It inhibits the sliding mechanism.

Fig 4. GSR based communication, with "used time interval" sliding rule.

This GSR based method allows a more efficient medium access sharing than the classical master/slave one: only one frame is sent by the master for all the slaves belonging to the given group and time division access between concerned slaves is optimized relatively to a compromise between determinism, reactivity and efficiency of the bandwidth exploitation.

4. METHODOLOGY OF VALIDATION

The validation of a system can be performed with different methods. The test, for example on a prototype of the system to be validated, is a well-known and effective method. But in some critical contexts, more formal methods are required (Rushby, J., 2007): simulation and verification of properties on a formal model of the system. Analysis of a model of the system allows detecting errors at an early step of the design process, before the implementation step. Furthermore, the formal approach leads to a more accurate validation of the system reliability. Simulation offers the possibility of step by step, or random, execution of the model: it can be very helpful for the design and the debugging of the model of the system, and for detection of simple errors of the system behavior. However, the more reliable results are given by an exhaustive analysis of all the possible behaviors of the system. This exhaustive validation is done by model checking (Berard, *et al.*, 2001) on the state space of the formal model i.e. all the possible behaviors. This validation method is the most appropriate one in our critical context.

The methodology we used can be resumed in four main parts. The first step of the validation process is the modeling of the system. It is necessary to choose a formal language which fits with the system to model and the properties to verify. The second step consists in abstracting the model to allow the analysis process. Indeed, combinatory explosion is a well-known problem of the analysis methods. It is then necessary to reduce the system's model, keeping only the relevant information. Next, the desired properties are modeled. And, at last, the properties must be verified analyzing the whole model.

5.   MODEL OF THE SYSTEM

5.1.   *Choice of the modeling language*

The protocol STIMAP has been modeled with Time Petri Nets (TPN) and implemented on a FPGA based prototype (Petri Nets (PN) automatically translated into VHDL components). Indeed, Petri Nets are a language allowing the expression of parallelism, synchronization, resource sharing and concurrency in a simple and natural way. They are then well-adapted for the modeling of distributed and communicating systems (Petri, 1962). At last, PN are associated to a mathematical formalism from which structural and behavioral analysis can be performed, including the model-checking analysis. Since we deal with non-autonomous systems, i.e. systems that interact with their environment via inputs (signals, sensors, etc.) and outputs (signals, actuators, etc.), we use extensions of PN that permit the description not only of the evolution of the model state but also when this occurs. A temporal extension is then necessary: the TPN which allows to model dense time as intervals associated to transitions.

5.2.   *Global model structure*

The global model represents five DSU and the medium of communication. The behavior of the controller is not modeled; it is simply represented by one initially marked place (place Appli_Master on figures 5 and 6) from which starts the communication (firing of transition t_Send). We use a set of places to indicate to all DSU the information contained into the frame sent by the controller: individual (ISR) or group (GSR) communication, and the sliding rule (places Master_ISR, Master_GSR, Master_Sliding and Master_SlidingRule at the top of figure 6).

The five independent DSU models are connected by means of the bus model.

5.3.   *Bus model*

The model of the communication medium is shown figure 5. It represents the emission of a frame by the master: the frame is emitted on the bus when the transition t_Send is fired. Then, the transition t_Propagation represents the propagation time on the medium (here, this duration is fixed to 1). When the token is in the place Bus_Reception, the frame is available for reception by the slaves (the DSU). The same kind of model is used to represents the emission of DSU frames (cf figure 6).
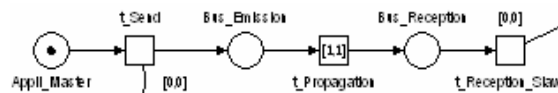

Fig 5. Bus model

5.4.   *Abstracted model of a DSU*

The part of the model which represents the DSU behavior is the same for all the DSU, except for the priority. The initial model (not given) represents all the functions of the DSU, even the application level. But these details are not all necessary in our context. The aim is the validation of the medium access mechanism, thus information about the other functions can be abstracted. Figure 6 shows the abstracted model of the DSU behavior. /The right part of the model is the abstraction of all the functionalities of the DSU except the medium access, which is modeled in the left part. When a frame is received from the bus, a token is generated in the

place Interface_Coupler (right of the model). Then the frame starts to be decoded by the MAC layer when firing the transition t_Read_Coupler. It first recovers the information relative to the speaking right. A token is generated in the place ISR or GSR depending on this information.

Then that starts the medium access mechanism at the left part of the model: transitions t_Access_ISR or t_Access_GSR can now be fired.

In the GSR case, t_Access_GSR is fired and the place Waiting is marked, representing that the DSU waits for its turn to access to the medium. Depending on the selected sliding rule (indicated by the marking of places Master_Sliding and Master_SlidingRule) and its priority, the DSU can immediately access to the medium (transition t_MyTurn) or has still to wait:

- If the sliding is not activated (the place Master_Sliding does not has a token), the transition t_NotMyTurn_NoSliding is fired and the DSU waits a duration D (place Wait_D and timed transition t_Wait_D). The DSU then returns in the place Waiting to verify again if it is its turn to access the medium or not.
- If the sliding mechanism is activated in case of "unused time interval" (the place Master_Sliding has a token but not the place Master_SlidingRule), the transition t_NotMyTurn_Sliding is fired. The DSU waits half an interval i.e. D/2. Then it tests (place Test_Reception) if the preceding DSU has emitted a frame. If so, the frame transmitted by the preceding DSU has generated a token in the place Received_Frame. Therefore, if that place is marked, the DSU has to wait: the transition t_Wait is fired and the DSU waits half an interval. On the contrary, if the place Received_Frame is not marked, the DSU does not wait: a token is directly generated in the place Waiting (firing of t_NoWait).
- In both cases (sliding activated or not), the priority of the DSU is taking into account. This priority is represented by the two places Priority and Priority_Comp, which contain different number of tokens. In the given example, the DSU has the priority 2: that means that the DSU has to fire twice the transitions t_NotMyTurn_NoSliding or t_NotMyTurn_Sliding before it can fire the transition t_MyTurn (and then access to the medium).
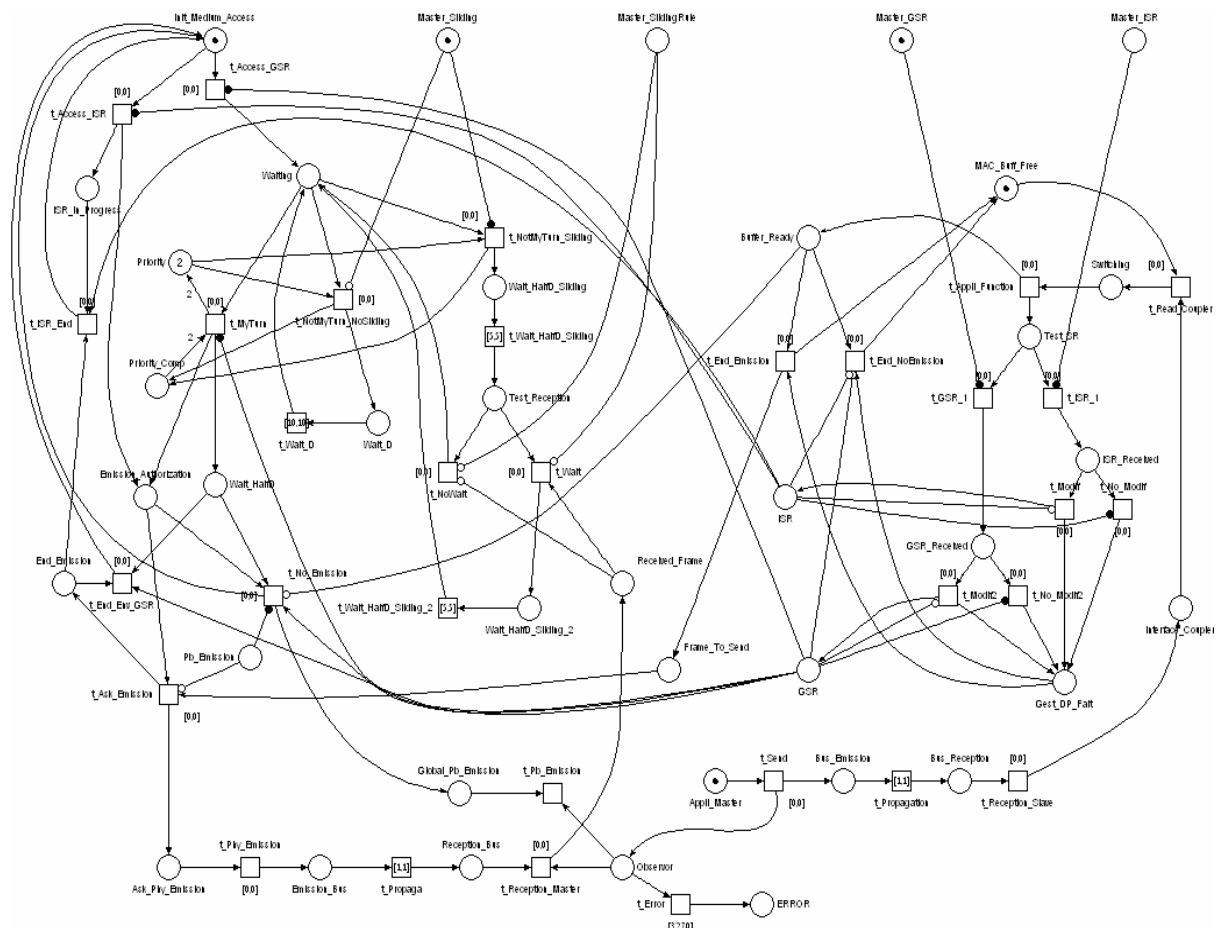

Fig. 6. DSU abstracted model, bus model and observer

6. RESULTS AND ANALYSIS

6.1. Properties to validate

To validate the access medium mechanism, it is necessary to analyze the access of the medium of each DSU for all possible cases of operation. The medium access must be verified both from a logical point of view (in which order the DSUs emit), and from the time one. Moreover, the sliding time interval mechanism must also be

validated considering its three possible rules. This article only presents the validation results of one example scenario (given in section 2.4): 5 DSU which are members of the same multicast group, with and without sliding in case of "unused time interval", and with and without a passive node. The theoretical behavior of the medium access has been given on fig.2. The aim of the validation process is to ensure that the behavior of the model of the system is identical to the expected one, i.e. the theoretical one.

6.2. Analysis methods

The validation has been done with the tool TINA (Berthomieu, *et al.*, 2004). This tool allows to model in TPN in a textual and graphical way. It allows analyzing the model with different methods: structural analysis, simulation, and reachability analysis. The last one analyzes all the possible states of the system in an exhaustive way. It is then possible to verify properties using a specific observer. This technique is often used to transform complex property verification (as liveness property) into a reachability problem (accessibility of an error state). Figure 7 shows the observer used to verify the worst time between the emission of a frame by the master (firing of the transition t_Send) and the reception by the master of the response frame of the DSU (transition t_Reception_Master). If this duration is longer than the t_Error interval, this last transition can be fired and the ERROR place can be reached.
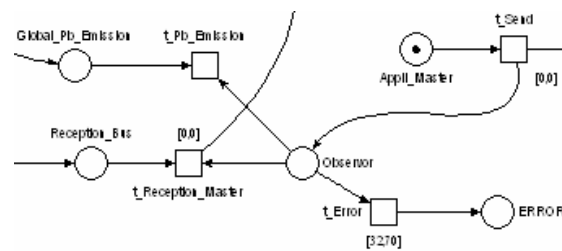

Fig 7. Property model: observer

6.3. Results

The validation has been done for the different cases of the STIMAP protocol. We give here the results of the validation of the sliding mechanism in three situations: if it is inactivated, which is the referent behavior (Table 1), if it is activated but useless when there are no passive nodes (Table 2) and useful in case of passive nodes (Table 3). Here, a node is called "passive" when it does not emit a frame in its reserved interval. In our model, the propagation time is 1 unit for each frame and the duration of the interval D is 10 units. This time is count since an initial instant which correspond to the emission of the master's frame.

Table 1 :   Worst response time for all DSU in case of GSR, without sliding

DSU	Priority	Worst response time
DSU a	0	2
DSU b	1	12
DSU c	2	22
DSU d	3	32
DSU e	4	42

Table 2 :   Worst response time for all DSU in case of GSR, with sliding in case of "unused time interval", without passive nodes

DSU	Priority	Worst response time
DSU a	0	2
DSU b	1	12
DSU c	2	22
DSU d	3	32
DSU e	4	42

Table 3 : Worst response time for all DSU in case of GSR, with sliding in case of "unused time interval", with DSUc being a passive node

DSU	Priority	Worst response time
DSU a	0	2
DSU b	1	12
DSU c	2	-
DSU d	3	27
DSU e	4	37

6.4. *Analysis*

- **Table 1:**

These results show the behavior of the modeled system in the normal case. Each DSU waits its turns and sends its frame at the desired instant. This is the same behavior than the theoretical one (figure 2). A DSU can access to the medium depending on its priority. As an example, let us considered the fourth DSU (DSU_d): before sending its frame, it has to wait the emission from the three DSU of higher priority (DSU_a, DSU_b and DSU_c), which means three D-intervals. Its worst response time is equal to:

$$2*PropTime + Prio*D = 2*1+3*10 = 32 \text{ units}$$

with:
PropTime being the propagation Time,
Prio being the priority (the higher priority being equal to 0).

- **Table 2:**

The results of table 2 correspond to the behavior of the modeled system when the sliding mechanism is activated, but in the case there is neither passive node nor message loss. Every frame is then sent, and no sliding is done (nor awaited): the results are logically the same as those of the referent scenario.

- **Table 3:**

In this scenario, the sliding mechanism is activated and the node DSU_c is considered as a passive node. The behavior should be like that exposed on figure 3: the sliding mechanism should allow saving a half-interval. This gain is observed in the worst response times of the DSU. After the no-emission of the DSU_c, DSU_d waits only a half interval. It emits its frame after 2.5 intervals instead of 3 intervals (without sliding). Its response time is equal to:

$$2*PropTime + \sum_{i=0}^{prio-1} \frac{D}{2} + (\sum_{i=0}^{prio-1} \frac{D}{2} / \text{DSUi active})$$

ResponseTime= 2*1 + 3*5 + 2*5 = 27 units.

The comparison of the three scenarios shows that the behavior of the modeled system is the same one as the theoretical one. The results given in this article are extracted from a full analysis process of the medium access mechanism, testing all the options: with or without sliding, with different configuration of sliding, with or without passive nodes. The medium access protocol STIMAP has been validated in an exhaustive way. It guaranties a reliable and deterministic access to the medium depending in the priority of the DSU.

7. CONCLUSION AND FUTURE WORKS

In the specific context of functional electrical stimulation, a new medium access protocol, called STIMAP, has been presented. In such critical context, a validation process has been performed to ensure that the behavior of the system is the desired one. We focused on the validation of the medium access mechanism proposed by STIMAP. The methodology used for this validation allows a formal and exhaustive validation of all the possible behaviors of the system. The advantage is that it is a sure and complete analysis. The inconvenience is the complexity of the analysis process which leads to combinatory explosion.

We presently perform a more precise characterization of both (FPGA-based implementation) the protocol-stack crossing performances, and the performances of different medium technologies (Ethernet, 2.4 GHz RF and a specific 2-wire implantable bus); the aim is to refine the model we analyze and to go deeper in validation. Doing that, another problem should be considered: the combinatory explosion. Exhaustive analyses lead to a too high complexity in case of big models. Solutions like systematic abstraction, hierarchical or multi-model approach have to be studied.

At last, we will soon experiments in STIMAP several methods to improve the time synchronization between DSU for a more precise automatic time-interval positioning.

REFERENCES

Berard, B., M. Bidoit, A. Finkel, F. Laroussinie, A Petit, L. Petrucci, P. Schnoebelen, 2001. Systems and Software Verification: model-checking techniques and tools. ISBN: 978-3-540-41523-7, Springer.

Bernard, S., D. Andreu, ML. Flotte, P. Cauvet, H. Fleury and F. Verju (2006). Testing System-In-Package Wirelessly. In *IEEE Int. Conf. On Design and Test of Integrated Systems (Dtis'06)*, Tunis, Tunisia, Sept. 5-7.

Berthomieu, B., P-O. Ribet and F. Vernadat (2004). The tool TINA - Construction of Abstract State Spaces for Petri Nets and Time Petri Nets, *International Journal of Production Research*, **Vol. 42**, **n° 14**, pp 2741-2756, July 2004.

Falcon, Carl. (2003). Adding a Low Data Rate Radio ASSP to an ISM Application. *Microwave Journal*, **Vol. 46**, **n°10**, pp 22 October 2003.

Godary, K. (2004). Validation temporelle de réseaux embarqués critiques et fiables pour l'automobile, PhD thesis, INSA Lyon, France.

Godary, K., I. Augé-Blum and A. Mignotte (2004). Temporal Bounds for TTA: Validation. In *Proc. of the IFIP Conference on Distributed and Parallel Embedded Systems (DIPES'04)*, Toulouse, France.

Guiraud, D., T. Stieglitz, K.P. Koch, J.L. Divoux and P. Rabischong, (2006). An implantable neuroprosthesis for standing and walking in paraplegia: 5-year patient follow-up, In *Journal of Neural Engineering*, **Vol. 3**, pp 268-275.

Guiraud, G., D. Andreu, G. Cathebras, Y. Bertrand, S. Bernard, J. Galy and J.D. Techer (2004). Dispositif de répartition de courant entre des cathodes d'une électrode multipolaire, notamment d'un implant, Patent Number
FR0409351 2004-09-0.

Kopetz, H., G, Bauer. (2003). The time-triggered architecture. *Proceedings of the IEEE.* **Vol. 91**, **n°1**, pp 112-126.

Loeb, G. E., R. A., Peck, W. H. Moore and K. Hood (2001). BION^tm system for distributed neural prosthetic interfaces, In *Medical Engineering and Physics*, **Vol. 23**, pp 9-18.

Petri, C. A. (1962). Kommunikation mit Automaten. PhD thesis, Bonn university, Germany

Rahmat-Samii Y., Kim J. (2006). *Implanted Antennas in Medical Wireless Communications*. Morgan & Claypool Publishers, August 2006.

Rushby, J., (2007). Automated Formal Methods Enter the Mainstream. In *Communications of the Computer Society of India, (Special Theme Issue on Formal Methods).* **Vol. 31, n°2**, pp 28-32.

Stieglitz, T., K.P. Koch and M. Schuettler, (2005). Flexible, Polyimide-Based Modular Implantable Biomedical Microsystems for Neural Prostheses, In *IEEE Engineering in Medicine and Biology Magazine*, **Vol. 24**, **n°5**, pp 58-65.

Ye, W. Heidemann, J. Estrin, D. "Medium Access Control With Coordinated Adaptive Sleeping for Wireless Sensor Networks", IEEE/ACM Transactions on Networking, **Vol. 12, n° 3**, pp. 493-506, June 2004.