Temporal bounds verification of the STIMAP protocol

Karen Godary-Dejean, David Andreu, Richard Romain

To cite this version:

Karen Godary-Dejean, David Andreu, Richard Romain. Temporal bounds verification of the STIMAP protocol. RTNS: Real-Time and Network Systems, Sep 2011, Nantes, France. hal-00789796

HAL Id: hal-00789796

https://hal.archives-ouvertes.fr/hal-00789796

Submitted on 19 Feb 2013

Temporal bounds verification of the STIMAP protocol

Karen Godary-Dejean, David Andreu and Romain Richard
LIRMM - Robotic Department
University of Montpellier 2
MONTPELLIER, FRANCE
{godary, andreu, richard}@lirmm.fr

Abstract

This article deals with the temporal validation of STIMAP, a medium access protocol. This protocol has been designed to meet the specific requirements of an implantable network-based neuroprosthese. This article presents the modeling of STIMAP with Time Petri Nets (TPN), and the verification of the deterministic medium access it provides, using timed model checking. The specific case of the synchronization reference time mechanism is detailed, explaining the problem it poses for the verification process and the solution we use to provide the whole protocol validation. This interesting and complex case study shows that existing formal methods and tools are not perfectly suitable for the validation of real systems, especially when some dynamic duration or hardware parameters have to be considered.

1 Introduction

1.1 Application context and communication needs

In order to improve the daily life living of para- and quadriplegic patients, Functional Electrical Stimulation (FES) is a palliative solution. FES has been successfully used to face functional deficiencies, in well-known applications such as: pacemaker, deep brain stimulation, pain control or hearing restoration. Implanted FES 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.

Effective existing solutions for implanted FES are mainly based on centralized architectures. However, this prevents the architectures to be extensible. For example, it is not possible with existing implantable neuroprostheses to add new sites of stimulation in a dynamic way. Moreover, centralized implants lead to complex surgery, high risk of failure during and after surgery, and global infection problems, which can be limited with a distributed architecture. So, we designed and developed a new architecture for such implantable FES system based on technological advances in networks and micro-electronics domains [2]. This new architecture relies on an asynchronous network of small distributed units in charge of activating and monitoring the peripheral nervous system. Thus, the asynchronous network constitutes the backbone of our neuroprosthese and it must be managed in a reliable and deterministic way in such critical embedded systems.

1.2 Why a new MAC protocol ?

The medium access mechanism used in this architecture must be studied in depth since it plays a major role regarding the communication between artificial devices that control natural organs. The characteristics of our application context lead to some conclusions referring to existing MAC methods, summarized here and detailed in [9].

First, this is necessary to remind the wanted characteristics: the communication medium must be reliable, efficient, embedded in a small implantable device, must integrate management group facilities and the signal could be transmitted in a wireless way into the human body. For all these reasons, we conclude in [9] that there is no suitable existing MAC protocol. We have eliminated the too complex or non deterministic solutions, and the ones impossible to apply at the physical level. Finally, two protocols seems to be appropriated: the master/slave approach and the TDMA one. Indeed, the topology imposed by our application context is clearly closed to a centralized management: one node is the master, others are slaves. The master is the central point of the system, initiating all the communications. Nevertheless the limited efficiency due to the master-slave protocol overhead must be improved, according to the context. The time division solution could also be used in our context, nevertheless time sharing should be improved in order to favor the reactivity of the implanted FES control architecture and to offer efficient group-based time sharing. So, to fulfill the context specific requirements and constraints, we designed a new MAC protocol, based on an association between the master/slave approach and the TDMA one, with some complementary mechanisms to enhance reliability and efficiency. This new protocol, called STIMAP (Sliding Time Interval based Medium Access Protocol), is presented section 2.

Taking early into account validation preoccupations, the behavior of this MAC protocol, has been modeled using Time Petri nets (TPN). Both TPN-based simulations and prototype experimentations contribute to the validation of STIMAP. But these two validation methods do not provide guarantees of determinism and correct behavior. With regard to the critical application context, STIMAP have to be validated applying a methodology based on formal methods. As a consequence, we went further in the validation process, applying a model checking [20, 5] based validation methodology. It allows verifying properties through an exhaustive analysis of the whole states space of a formal model of the system. The STIMAP validation process has been separated into two steps: first the validation of the basic mechanisms (detailed in [9] and resumed in section 4), and second the validation of the whole protocol considering specific hardware architecture and clock constraints. Indeed, the first validation phase is based on the hypothesis that the hardware architecture is a homogeneous one. But this hypothesis is not a realistic one, and this article focuses in section 5 on the problematic of modeling and validating STIMAP considering heterogeneous architectures. This new hypothesis significantly increase the complexity of the validation process. Furthermore, this imply considering new mechanisms as for example the *Reference time* used for synchronization.

1.3 Problematic for STIMAP validation

Model checking is now currently used for protocols validation. The problematic in our context is the medium access mechanism, and we specifically focus on the synchronization problem taking into account the hardware architecture characteristics. On the one hand, the validation of the master-slave approach has already been done for example in [7] or [4]. But in these cases of classical polling, there is no collision nor clock synchronization problem. On the other hand, the TDMA approach has also ever been validated, for example in [16], [14] or [15]. But the validation works are generally based on the hypothesis that the local clocks are synchronized, or with a guaranteed desynchronization interval. Besides, the validation works of TDMA-based protocols considered that the nodes have the same reference time to begin their TDMA sequence. This hypothesis could be contested in some cases, as the one we proposed to study in this article. Finally, the last point to consider is the used of parameters values in the modeling process. The parameters are necessary to represent the hardware and/or software architecture. Especially in the embedded context, a large number of parameters are used in the model, referring for example to hardware proportion of system components, or to configuration values of software algorithms. Nevertheless, the parametric model checking is known to be undecidable [1], and no effective solution are currently available for concrete systems validation. The only possibility is to explicitly replace parameters by numerical values, which could be obtained during an a priori experimental or theoretical step. We will see that this parameters management is not as trivial as it seems to be.

Concluding, we can remark that there are no validation results with regard to all the STIMAP aspects. The existing works often make simplifying hypotheses which could not be accepted in our case. Indeed, we consider in this article a specific application context, which implies new constraints and then new difficulties in the modeling and validation points of view. The following section presents the STIMAP protocol. Section 4 reminds the previous validation works on STIMAP [9], summarizing the used method and the validation hypotheses and results. Section 5 tackles more complex hypotheses dealing with the validation of STIMAP with for heterogeneous asymmetric architectures. It shows difficulties we encountered in the identification of time parameters that guarantee the respect of temporal constraints, and in the integration of such parameters in the validation process. This article is concluded by a discussion on limitations and improvements identified during this validation work.

2 Basic STIMAP mechanisms

2.1 Medium access principle

In our architecture, the topology is based on a centralized controller and distributed stimulation units (DSU) connected by a network. The controller, being the master, is the central point of the system, initiating all the communications with the distributed units, i.e. the slaves. The method is simple, as one slave transmits a frame only if the master has demanded or authorized it. The master/slave approach is appropriated, nevertheless the limited efficiency due to the protocol overhead must be improved, taking into account the multicast possibility. Indeed, this model of cooperation is not really efficient when dealing with a group of slaves as it requires to poll slaves, i.e. to individually communicate with each one. Time sharing (TDMA) can be a good solution to avoid polling, if clock drift problem and time-slots losses can be faced. However, a “static” TDMA is not adequate since the controller needs to communicate with a subset of slaves only when stimulation is performed, and moreover this subset of slaves is dynamically selected depending on the FES function to be performed (e.g. different set of muscles implied in a given movement).

So, the combination of time sharing and master/slave relation are the basis of the STIMAP protocol we propose. In short, we defined a method, adequate to our decentralized architecture, which is simple to implement (since it must be embedded in small implantable DSU), which allows TDMA to be contextually used, providing dynamic time-slots assignment and limiting time-slots losses.

2.2 Group medium access

STIMAP obviously allows basic individual master-slave and slave-master communications, but also offers a way to manage the communications with a group of slaves without polling all the members of the group. The master 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), which is a trivial master-slave exchange, and TDMA-based Group Speaking Right (GSR, group token). When the master allocates a GSR to a given group, it sends a broadcast frame dedicated to the concerned group members. In this frame, it also indicates which member must begin the communication (not necessarily the highest priority). 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. Figure 1 represents a basic GSR communication: 5 DSUs are members of the group. The DSU with the priority 0 (DSUa) begins to emit after the reception of the GSR frame sent by the master; the others DSUs emit one after the other on their turn, according to their priority that they use to determine their time-slot position.

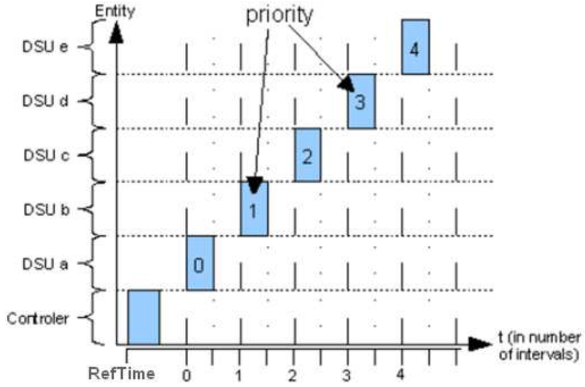

Figure 1. GSR based communication

2.3 Time slot positioning

Knowing its position in the group, the group size and the member who must begin the communication, each member determines when it will have the right to emit its packet, i.e. the position of its time-slot. Each member can speak during the time-slot duration, a given time interval D that has been automatically computed by the master or explicitly specified in an initial phase ; this parameter allowing to fit time-slot duration to the different control contexts. The positioning, in terms of member's position in the communication round, is given by the variable Pos_P, which is the position of the member of priority P:

$$Pos_P = P - BP + \alpha_P \times GS \qquad (1)$$

where: P is the priority number of the slave in the given group, BP is the priority of the first emitting DSU (which is not necessarily the lower priority), GS is the group size and $\alpha_p = 1$ if $P < BP$ else $\alpha_p = 0$.
Then, from a time point of view, the time-slot position is:

$$TSPosi = RefT + Pos_P \times D \qquad (2)$$

where: D is the time-slot duration and $RefT$ corresponds to the reference instant for every node for this communication phase (i.e. $RefT$ is not an absolute time reference). The reference time mechanism is described section 3.

2.4 Sliding time interval

The time-slot attributed to a slave is in fact constituted by two half time-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 the master will have access to the medium without any collision risk, this second half-interval is reserved. This contributes to the reactivity of the distributed stimulation architecture and is very important in our context.

However, if a slave has nothing to transmit, its half-interval is free and the half-interval reserved for the master is then unused and wasted. To avoid that, i.e. to reach better efficiency, the MAC method integrates a sliding time interval mechanism. When waiting for its time-slot, every slave listens to the medium: if the previous member of the group did not emit a packet then it brings backward its own time-slot by a half time-interval. In other words, it recovers the half time-interval that was reserved for the master but that will never be used.

In fact, the master can dynamically configure the sliding strategy according to the FES control context. Three possibilities, named sliding rules, which must be exploited in a coherent way by the master, are proposed:

- No sliding: the master wants to always be able to react. It inhibits the sliding mechanism.
- Sliding in case of "unused time interval": it corresponds to the case previously exposed. For example, this rule is selected when the master 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 time-interval because the master will not has to react. An example is given on figure 2 showing that DSUd begins to emit its frame after 2.5 time-intervals (after the reference time) instead of 3 time-intervals in the normal case (figure 1).

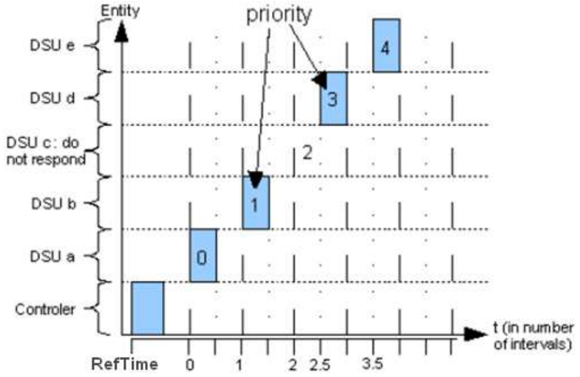

Figure 2. GSR based communication, with "unused time interval" sliding rule

- Sliding in case of "used time interval": it corresponds to a situation in which the master must not respond if the DSU emits a frame. For example, this rule is selected when the master performs a test of presence on a DSU group. If a DSU responds, the second half time-interval can be recovered since the master will not has to react. An example is given on figure 3 showing that DSUb and DSUe begin to emit their frames after $1/2$ time-interval instead of 1 time-interval (figure 1).

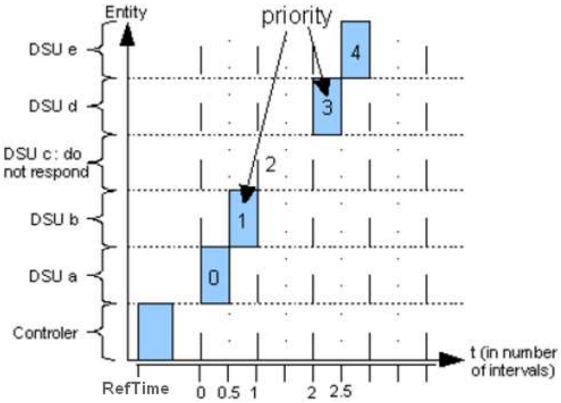

Figure 3. GSR based communication, with "used time interval" sliding rule

With this sliding mechanism, the time-slot positioning becomes more complex. It can be represented by the following equation, $\forall Pos_P > 0$:

$$TSPosi = RefT + Pos_P \times \frac{D}{2} + \sum_{i=1}^{Pos_P} \delta_i \times \frac{D}{2} \quad (3)$$

where Pos_P is computed according to equation 1, and δ_i depends on the selected sliding rule:

- in case of "no sliding" rule, $\delta_i = 1 \;\; \forall i$,
- in case of "unused time interval" rule, $\delta_i = 1$ if the previous member sent a frame else $\delta_i = 0$,
- in case of "used time interval" rule, $\delta_i = 0$ if the previous member sent a frame else $\delta_i = 1$.

3 Reference time synchronization

The preceding mechanisms ensures a deterministic access to the medium provided that the DSUs are synchronized on the master frame, for this communication phase (i.e. they not have to be continuously synchronized). This is the case in an ideal world where we do not consider hardware effects of the physical architecture: all the DSUs therefore receive the master frame at the same time and are synchronized. However in our context, we can not consider the propagation and reception times as negligible. If implanted, a wireless network is indeed a small range one but with potentially important propagation time variations, due to absorption coefficient differences within the body. In the case of an implanted wired network, even if the nodes do not move (i.e. no mobility, fixed topology, known propagation time) they can have different communication performances from a technological point of view (time to emit and receive frames).

Time synchronization has been studied in depth in the network community, notably in sensor networks [12]. More or less complex algorithms have been proposed to ensure permanent time synchronisation [19] [17] [13], since for some classes of applications it really impacts the correctness of the results (e.g. data merging). Moreover, most approaches rely on time synchronization dedicated traffic which impacts the load of the network. We do not require such a precise time synchronization but rather a contextual time synchronization, meaning that we just want a subset of nodes to be synchronized at the beginning of a contextual communication phase to limit the collision risk, knowing that the time-slot duration D can also include a "guard time" if necessary [11]. We also need a simple mechanism, as it must be embedded in small distributed units (implants). Moreover, as a communication phase has a short duration in our application context (much less than one second when communicating with group of nodes), the clock drift problem can be considered as insignificant considering for example the typical drift of 10^{-6} of the 2 MHz quartz we use in a DSU.

3.1 Symmetric vs asymmetric architectures

Our reference time mechanism must theoretically ensure a contextual reference time for all DSUs even if the propagation times are not the same for all the network members. However this supposes that transmission times between the controller and slaves are equal in both ways (controller to slave and slave to controller, for reception and emission sides), which could be an unrealistic hypothesis. Indeed, an experimental series of measurements with a platform based on Ethernet and RF technologies [6], confirmed that some hardware architectures can imply important variations between the different RTT durations. Most of all these experimental measurements showed that it is possible to have significant differences between the transmission duration from the master to a DSU and the one from this DSU to the master. This means that the system is not a perfect one and that the hypothesis of a one way communication equal to $\frac{RTT_{DSUi}}{2}$ is not verified. Thus, this hypothesis must be reconsidered: a system is neither homogeneous nor symmetric.

3.2 Reference time for symmetric architecture

Thus, we have implemented in STIMAP a synchronization mechanism: the reference time mechanism, to synchronize the DSUs at the beginning of the TDMA sequence providing a common contextual reference time. As soon as a DSU receives the GSR frame, each node determines its reference time, imposing that it must start a constant duration, chosen as a half-interval $\frac{D}{2}$, after the GSR frame has been sent. The idea of the reference time mechanism is based on subtracting the transmission time from controller to slave, estimated to $\frac{RTT_{DSUi}}{2}$ with RTT_{DSUi} the Round Trip Time between the DSUi and the master, to the previously mentioned common constant. The difference between the instant of time at which the DSUs receive the master frame will then be balanced by the subtraction of the different propagation times (that include reception and transmission performances).

The reference time mechanism is then defined by:

$$RefTWait_{DSUi} = \frac{D}{2} - \frac{RTT_{DSUi}}{2} \tag{4}$$

with $RefTWait_{DSUi}$ the duration the DSUi has to wait after the GSR reception. As it is a waiting duration, it of course can not have a negative value. Then the D parameter has to respect the following reference time constraint, with RTT^{max}_{DSU} the worst RTT of all slaves:

$$D \geq RTT^{max}_{DSU} \tag{5}$$

Figure 4 shows how this mechanism works illustrating it with two DSUs: (1) DSU2 receives the master frame first, and begins to wait $RefTWait_{DSU2} = \frac{D}{2} - \frac{RTT_{DSU2}}{2}$; (2) then DSU1 receives the master frame, and waits $\frac{D}{2} - \frac{RTT_{DSU1}}{2}$. Supposing that the transmission time of the frame from controller to slaves is equal to $\frac{RTT_{DSUi}}{2}$, the contextual reference time is then the same for both DSUs: $RefT_{DSU1} = \frac{RTT_{DSU1}}{2} + \frac{D}{2} - \frac{RTT_{DSU1}}{2} = \frac{D}{2} = \frac{RTT_{DSU2}}{2} + \frac{D}{2} - \frac{RTT_{DSU2}}{2} = RefT_{DSU2}$.

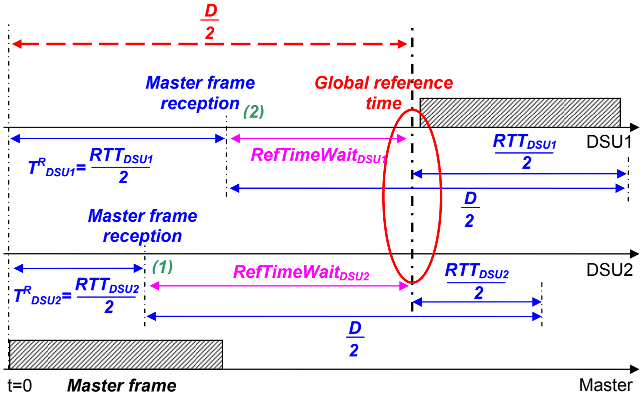

Figure 4. Basic principle of the reference time mechanism

3.3 Reference time for asymmetric architecture

The problem of the reference time mechanism should be based on the real transmission time of the GSR beacon from the controller to each slave. Let us introduce its corresponding variable T^R_{DSUi} (see figure 4). At the opposite, we will name T^E_{DSUi} the transmission duration from DSUi to the master. These variables (parameters of the model) must be known to configure the protocol before the running phase. In the previously given reference time definition, the T^R_{DSUi} parameter was estimated to $\frac{RTT_{DSUi}}{2}$. Indeed, the RTT parameter can easily be measured on the architecture, counting on the master side the time elapsed between the frame emission to the slave and the reception of the associated answer. In STIMAP, the RTT measurement is performed during the initialization phase, using an ISR mode: the duration of an individual communication is measured for each DSUi and considered as its RTT parameter value (RTT_{DSUi}).

The basic reference time mechanism resolves the synchronization problem even in case of heterogeneous system (i.e. RTT_{DSUi} not the same for all DSUs), provided that $T^R_{DSUi} = \frac{RTT_{DSUi}}{2}$. But as previously noticed, this last hypothesis implies a strong constraint on the hardware: it supposes that the network cards have the same performances in reception than in emission. Supposing on the contrary that $T^R_{DSUi} \neq \frac{RTT_{DSUi}}{2}$, the reference time instant could differ depending on the DSUs and then the theoretical equation becomes:

$$\begin{aligned} RefT_{DSUi} &= T^R_{DSUi} + RefTWait_{DSUi} && (6) \\ &= T^R_{DSUi} + \frac{D}{2} - \frac{RTT_{DSUi}}{2} && (7) \end{aligned}$$

Establishing that the RTT values can potentially be different for the all the DSUs, we can consider two hardware architecture types: the symmetric and the asymmetric ones. In a symmetric hardware architecture, the hardware couplers take the same time to emit and to receive a frame. Then the RTT durations are not necessarily the same for the different DSUs but for each DSUi the transmission time of a frame is the same whatever the direction of the transmission: $T^R_{DSUi} = T^E_{DSUi} = \frac{RTT_{DSUi}}{2}$. In asymmetric architectures, $T^R_{DSUi} \neq \frac{RTT_{DSUi}}{2}$, so the reference time instants are not the same for each DSUi. Therefore the asymmetric case has to be taken into account in the validation process since this synchronization gap impacts the protocol determinism.

However it is difficult to know the real duration of the one-way transmission of a frame in an asynchronous network; this requires specific material and experimentations, and it cannot be performed on our implanted network. Nevertheless, the aim of this study is to take into account the asymmetric case in a formal validation process to identify the limitations of the initial hypothesis.

4 First validation

A first validation step has been done to verify the basic principles of the STIMAP mechanism, considering that all the DSUs are synchronized in a common reference time. This work is presented in [9]. In this section, we globally present the validation methodology, as it is also used to validate the reference time mechanism in section 5. We then precise the hypotheses used for this first validation step, and finally remind the obtained validation results.

4.1 Validation methodology

The used methodology is a classic one from now for formal validation of communicating systems. This methodology can be resumed in four main parts. The first step of the validation process is the modeling of the system. It is then 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 exhaustive analysis methods. Thus the system model must be reduced, keeping on only the relevant information. The desired properties must also be modeled, since they can be too complex to be expressed directly in temporal logic. Next, the properties can be verified, and, at last, this validation results must be analyzed to conclude on the system reliability.

In our context, we used model checking in a Timed Petri Nets (TPN [18]) system model. Petri Nets (PN) are a formalism 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. 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 (signals, sensors, actuators, ..), 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 modeling dense time as intervals associated to transitions. In this article, the validation results have been obtained with the TINA toolbox[1] [3], which allows TPN modeling and properties verification with model checking.

Validating some specific properties, as collision absence, we can be sure that it could never append in the system provided that the system implementation is faithful to the system model. This problem is resolved in our context as the STIMAP model has been implemented on a FPGA based prototype using an automatic VHDL code generator [6] ensuring the equivalence code vs. model. HILECOP allows the automatic translation of Petri Nets into VHDL components. This ensures that the verified properties on the system model remain true on the implemented system (on FPGA paralellism is effective contrary to processor-based implementation).

4.2 First validation hypotheses and results

As said in the introduction section, the first validation step of STIMAP [9] is based on simple hypotheses: the MAC mechanism principle has been verified supposing that all the nodes are synchronized on the master frame, at the beginning of the TDMA sequence. This could be possible if the propagation time and the reception time are the same between the master and all the nodes. STIMAP has then been modeled and validated in [9] without its reference time mechanism.

This first validation step allows verifying the STIMAP bases: the group medium access, the time slot positioning and the sliding time interval mechanisms. Several properties have been validated to verified the STIMAP behavior, as the verification of the worst response time for a DSU after a master frame reception, or the verification of collision absence. All the properties have been verified even in case of emission faults: frame losses or no DSU emission.

But these validation results do not consider the heterogeneous and asymmetric architectures. The next section then studied in details the reference time mechanism and its desynchronization consequences on the STIMAP medium access mechanisms.

5 Reference time validation

The whole system comprises several DSUs communicating with one master. For complexity reasons, we validate the STIMAP mechanisms concepts on a limited number of DSUs. Moreover, models of all these components are abstracted ones: parts of the models that support the mechanisms to be studied have been kept in details whereas parts that do not have any influence have been aggregated. This section describes the DSU and the medium models as they are the necessary to understand the reference time validation. However, to simplify the model description, we focus on the GSR mode and the time slot positioning mechanism, without considering the sliding time interval mechanism.

5.1 Basic models

The DSU model of figure 5 does not represent the reference time mechanism: its modeling will be discussed section 5.3. All the DSUs start the time slot positioning immediately after the GSR beacon reception (modeled here by the marking of the `BeginMedAcc_DSUi` place). This model of the MAC behavior is a classical one, already used in [9]. It is the same for all the DSU, except for the priority parameter. The priority mechanism, which is the basis of the group-based TDMA principle, is represented by means of two places: `priority_DSUi` and `priorityComp_DSUi`. The first place is the priority of the DSU, which practically represents the number of slots the DSU has to wait before emitting. The second place is the complementary one: it represents the number of slots already waited. Depending on its priority and on the number of already waited slots, a DSU knows if it has to wait one more slot (transition `t_NotMyTurn_DSUi`, then wait for a D interval equal to 500 time units) or if it could access the medium (transition `t_MyTurn_DSUi`). The marking of the `EmissionPermission_DSUi` place represents the permission to emit, and the `FrameDuration_DSUi` frame and its corresponding transition represent the emission duration of the frame.

[1] www.laas.fr/tina

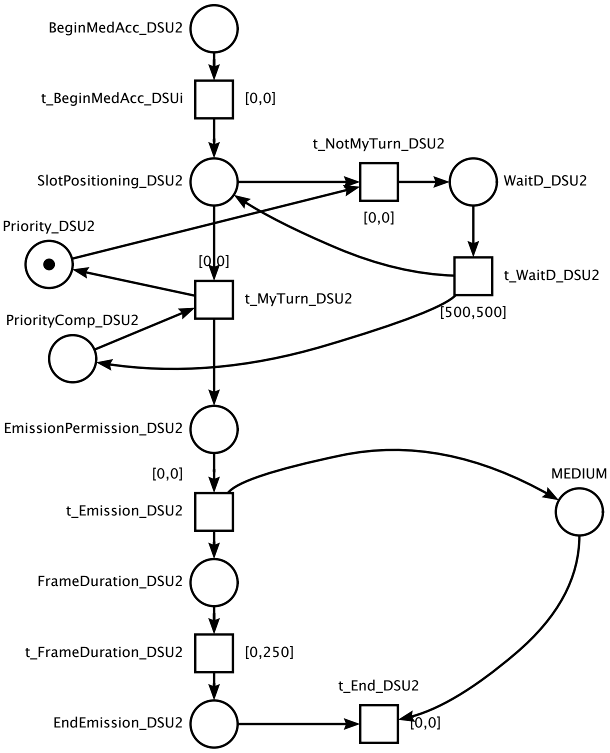

Figure 5. Basic DSU and medium model

The medium is modeled in a simple way, using only one place to represent its occupation. This representation is well-adapted to represent a shared medium where all the frames are sent in broadcast (like on wired bus as non-switched Ethernet). For wireless medium, it depends on the broadcast range: we suppose here that all nodes are reachable. Modeling the medium in a global way is a convenient over-approximation: if there is no collision detected on the global medium model, we can be sure that collision will not actually occur.

In figure 5 we see the medium model and its relation with DSU2: when DSU2 emits on the medium (transition `t_Emission_DSU2`), the place `MEDIUM` becomes marked representing that the medium is occupied. At the end of the frame emission (transition `t_End_DSU2`), this token is consumed then the medium is released.

5.2 Reference time in the symmetric case

In the symmetric architecture case, the reference time is supposed to provide a global synchronization time to all DSUs (see section 3). All the DSUs then are supposed to start the TDMA sequence $\frac{D}{2}$ time units after the beginning of the master emission. To model that, we just have to simulate a master frame transmission duration equal to $\frac{D}{2}$. Then all the DSUs will received the GSR beacon frame at the same time. The validation results with this hypothesis are logically the same as the ones obtained in the validation step without considering the reference time mechanism.

5.3 Reference time modeling in the asymmetric case

In the asymmetric architecture case, the reference time mechanism has to be entirely modeled. Figure 6 is a faithful representation of the reference time behavior. first the master frame is received by DSUi after T^R_{DSUi}, modeled by transition `t_MasterFrameReception_DSUi`. Then DSUi waits the reference time wait duration (equal to $\frac{D}{2} - \frac{RTT_{DSUi}}{2}$), staying in the `RefTimeWait_DSUi` place. Then it could begin the medium access phase: the transition `t_BeginMedAcc_DSUi` corresponds to the one of the DSU model of figure 5.

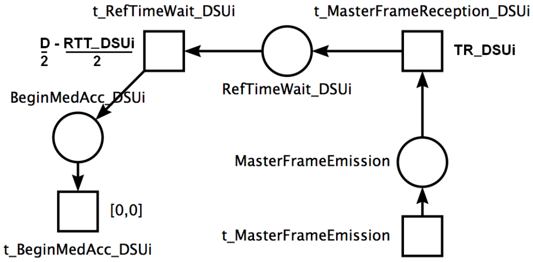

Figure 6. Theoretical reference time model

Nevertheless, this modeling is quite difficult using Petri nets. Indeed, Timed Petri Nets formalism and their associated analysis tools do not allow calculating the firing interval of transitions on a dynamic way. Then the calculation of the reference time duration must be done a priori. Furthermore, two durations of the system model are dependent on the values of the T^E_{DSUi} and T^R_{DSUi} parameters. The problem is that these parameters are not the same for all DSUs, and must not be constant: since we want to study all the possible cases, we want to represent them as random values taken in a given interval of their limit values. The first of these parameters (T^E_{DSUi}) is taken into account in the frame emission duration (transition `t_FrameDuration_DSUi` of figure 5). Indeed, we suppose that $FrameDuration_{DSUi} = T^E_{DSUi}$. In fact, the $FrameDuration_{DSUi}$ parameter represents the medium occupation duration when DSUi emits a frame, whereas T^E_{DSUi} also includes the reception duration at the master side. Therefore in our model the medium occupation duration is an over-approximation of the reality. Supposing that, our validation results will be good whatever the master reception duration is. The second of these parameters (T^R_{DSUi}) is taken into account in the master frame reception duration (transition `t_MasterFrameReception` of figure 6).

A solution of this modeling problem could be to represent the reference time instant itself as a random duration in a time interval defined by the limit values of the reference time: $[RefT^{min}, RefT^{max}]$. This interval will be associated to only one transition `t_RefTime_DSUi`, which directly represents the reference time instant of the DSUs (see figure 7). This interval, calculated in an off-line step, represents all the possible reference time values for all the T^R_{DSUi} possible values, based on the theoretical reference time equation 7. In the same idea, we could also represent the $FrameDuration_{DSUi}$ duration as a min/max interval.

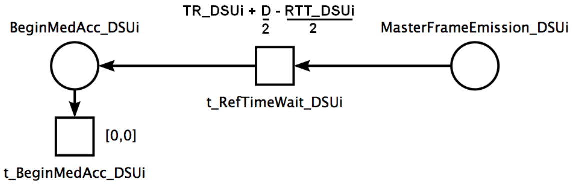

Figure 7. Simplified reference time model

However, the problem is not so easy to resolve. Indeed, these two parameters could not be independently represented since for a given DSUi, we must have $T^E_{DSUi} + T^R_{DSUi} = RTT_{DSUi}$. And of course, we could not have for a single DSUi both the maximal values of T^E_{DSUi} and T^R_{DSUi}. Ideally, we wish to determine the T^E_{DSUi} parameter value from the exact value of T^R_{DSUi}, and then to dynamically determined T^E_{DSUi} taking into account the exact firing date of the transition to which T^R_{DSUi} has been associated. But the Timed Petri Net formalism does not allow to represent the dependence between the firing instant of transitions. This is thus not possible to accurately represent the temporal behavior of the DSUs in an heterogeneous and asymmetric architecture.

The only solution for now is to represent an overset of the real states space, representing independent T^R_{DSUi} and T^E_{DSUi} values. Considering an overset of reality is safe from a validation point of view: if a property is not verified in the overset, it will never be verified in the real states. But if the property is not verified, the validation results must be analyzed in depth to know if the identified faulty scenario corresponds to a realistic situation.

5.4 Parameters values

It is now necessary to fix the interval values of the model parameters. For that, we consider the relationship between the three parameters T^E_{DSUi}, T^R_{DSUi} and RTT_{DSUi}. Experimental results [6] shows that the asymmetry is always in the same way for a given technology. For example, Ethernet hardware couplers are always faster to emit than to receive, whereas the situation is inverse for a RF technology. Thus we can consider two types of systems: with the *fast emission hypothesis*, i.e. when the DSU is faster in emission than in reception ($T^E_{DSUi} < T^R_{DSUi} \;\; \forall i$), and with the opposite hypothesis (*slow emission*). The rest of this paper focuses on the fast emission hypothesis.

To consider the worst execution case, it is necessary to consider all the possible values of the parameters. The worst asymmetric solution is when T^E_{DSUi} and T^R_{DSUi} are the most different ones. The hypothesis that $T^E_{DSUi} < T^R_{DSUi} \; \forall i$ can thus be traduced into two intervals:

$$\begin{aligned} T^E_{DSUi} &\in [0, \frac{RTT_{DSUi}}{2}] \\ T^R_{DSUi} &\in [\frac{RTT_{DSUi}}{2}, RTT_{DSUi}] \end{aligned}$$

Indeed, if T^E_{DSUi} is neglictible (considered equal to 0), and as $T^E_{DSUi} + T^E_{DSUi} = RTT_{DSUi}$, we have $T^R_{DSUi} = RTT_{DSUi}$. On the contrary, T^E_{DSUi} can not be superior to $\frac{RTT_{DSUi}}{2}$ respecting $T^E_{DSUi} < T^R_{DSUi}$.

The D parameter must respect the equation 5. In a first step (we discuss in conclusion on the possibilities of the D value dimensioning) we fix it to:

$$D = RTT^{max}_{DSUi} \tag{8}$$

Then, considering the reference time theoretical equation 7, we have:

$$\begin{aligned} RefT_{DSUi} &= T^R_{DSUi} + \frac{D}{2} - \frac{RTT_{DSUi}}{2} \\ &= T^R_{DSUi} + \frac{D}{2} - \frac{T^E_{DSUi} + T^R_{DSUi}}{2} \\ RefT_{DSUi} &= \frac{D}{2} + \frac{T^R_{DSUi} - T^E_{DSUi}}{2} \end{aligned}$$

From this equation, we can calculate the maximal and minimal values of the $RefT_{DSUi}$ parameter:

$$\begin{aligned} RefT^{min}_{DSUi} &= \frac{D}{2} + \min\left(\frac{T^R_{DSUi} - T^E_{DSUi}}{2}\right) \\ &= \frac{D}{2} + \frac{\min(T^R_{DSUi}) - \max(T^E_{DSUi})}{2} \\ &= \frac{D}{2} + \frac{\frac{RTT_{DSUi}}{2} - \frac{RTT_{DSUi}}{2}}{2} \\ RefT^{min}_{DSUi} &= \frac{D}{2} = \frac{RTT^{max}_{DSUi}}{2} \end{aligned}$$

$$\begin{aligned} RefT^{max}_{DSUi} &= \frac{D}{2} + \max\left(\frac{T^R_{DSUi} - T^E_{DSUi}}{2}\right) \\ &= \frac{D}{2} + \frac{\max(T^R_{DSUi}) - \min(T^E_{DSUi})}{2} \\ &= \frac{D}{2} + \frac{RTT^{max}_{DSUi} - 0}{2} \\ RefT^{max}_{DSUi} &= RTT^{max}_{DSUi} \end{aligned}$$

Besides, we have seen that the $FrameDuration_{DSUi}$ is supposed to be equal to T^E_{DSUi}, and then $FrameDuration_{DSUi} \in [0, \frac{RTT_{DSUi}}{2}]$. All these possible parameters values for an asymmetric hardware architecture are resumed in table 1.

Fast emission hypothesis
○ $D = RTT^{max}_{DSUi}$ ○ $FrameDuration_{DSUi} = T^E_{DSUi}$ ○ $T^E_{DSUi} \in [0, \frac{RTT_{DSUi}}{2}]$ ○ $T^R_{DSUi} \in [\frac{RTT_{DSUi}}{2}, RTT_{DSUi}]$ ○ $RefT_{DSUi} \in [\frac{RTT^{max}_{DSUi}}{2}, RTT^{max}_{DSUi}]$

Table 1. Parameters values intervals

5.5 Analysis of a collision scenario

The preceding parameters values of Table 1 have been integrated in the system model. Then, verifying the no-collision property, the result is that a collision is possible. Analyzing in depth the collision scenario, we remark that the collision is possible only if the parameters values are that of boundaries. Indeed,

The execution of this scenario is shown figure 5.5, **with** $D = 500tu$ (tu: time unit), and its detailed behavior is:

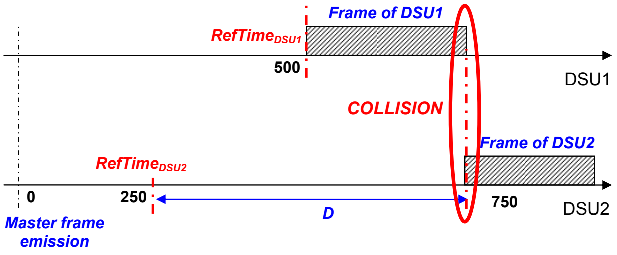

Figure 8. Collision example in an asymmetric architecture case

- $t = 250tu$: the DSU2's reference time is $250tu$; as DSU2 is not the first DSU of the GSR sequence, it waits a D time-interval before emitting.
- $t = 500tu$: the DSU1's reference time is $500tu$; as DSU1 is the first DSU to emit, it begins to emit its frame.
- $t = 750tu$: DSU2 has waited its first D interval, it's its turn now: it begins to emit its frame.
- if the frame of DSU1 is longer or equal to $= 250tu$, this frame emission is not finished and the DSU2 emission can provoke a COLLISION.

Then, a collision occurs if $RefT_{DSU1} = 500tu = D$, $RefT_{DSU2} = 250tu = \frac{D}{2}$ **and** $FrameDuration_{DSU1} = 250tu = \frac{D}{2}$**.** Even if this scenario concerns only extreme values, it seems realistic as these parameters values are included in the possible intervals. Besides we do not present in this paper the sliding time interval mechanism, which leads to the same results (collision occurs) with even more realistic parameters values.

But we can not conclude at this step that collision are possible in the real world. First of all, we have to study the specific values of T^E_{DSUi} and T^R_{DSUi} parameters to verify if they respect the constraints as discussed section **??**. First, we analyze the DSU1 parameter values:

$$\begin{aligned}
RefT_{DSU1} &= \frac{D}{2} + \frac{T^R_{DSU1} - T^E_{DSU1}}{2} \\
D &= \frac{D}{2} + \frac{T^R_{DSU1} - T^E_{DSU1}}{2} \\
\frac{D}{2} &= \frac{T^R_{DSU1} - T^E_{DSU1}}{2} \\
D &= T^R_{DSU1} - T^E_{DSU1}
\end{aligned}$$

Considering the interval values of T^R_{DSU1} and T^E_{DSU1}, the only solution to satisfy this equation, and then to fulfill the collision condition, is $T^R_{DSU1} = D$ and $T^E_{DSU1} = 0$. But the collision occurs only if $T^E_{DSU1} \geq 250tu = \frac{D}{2}$, which does not belong to the preceding values.

This analysis allows concluding that ***this collision situation is not a real one, it corresponds to one of the over-estimated states of the system model***. Moreover, all the detected collision scenarios correspond to the over-estimated states space.

5.6 Final validation results

Considering the whole heterogeneous and asymmetric hypothesis, the analysis of all the validation results leads to the same conclusion for all the protocol mechanisms: no collision is possible. The time slot positioning mechanism has been validated for all its possible sliding rules, and this validation step includes fast and low emission hypothesis, for all the possible parameters values of the reference time mechanism. A complete description of our validation process is given in [10].

Besides, another step of the validation process has considered the presence of faults in the system. Two types of faults have been modeled: frame losses, and silent nodes. They resume the most of all the possible faults at the MAC level which could affect the medium access mechanism in a temporal point of view. On the contrary, a bad frame reception is considered as a normal frame, as the medium access mechanism is only based on a signal presence in the medium. The faults consideration do not change the final conclusion of the validation.

6 Conclusion

This article deals with the validation of STIMAP, a deterministic medium access protocol. This protocol has been designed to meet the specific requirements of an implantable FES application. Thus the protocol must fit with real-time and reliability constraints, as well as with embedded ones. It must be reactive, deterministic, reliable, with a simple and light implementation. Thus, STIMAP is based on a TDMA group communication with a sliding mechanism to improve the efficiency of the classic TDMA approach. This approach is mixed with a master/slave approach to initiate a TDMA communication for one group of nodes. STIMAP also includes a reference time mechanism, which provides a common contextual reference time for the synchronization of all nodes of the network. In such a critical context, we propose to complete classical validation methods as simulation and prototype experimentation with a formal validation process, which provides more confident validation results.

This article resumes in a first part the validation of the STIMAP basic mechanisms. It presents the Time Petri Nets (TPN) protocol model and its validation, focusing on the no-collision property verification. The STIMAP validation is proved with an analysis of model checking results. The TPN formalism has been chosen to guarantee the implementation process: the validated model (in fact, an abstracted one) is directly implemented, automatically generating VHDL code for a FPGA target.

Then this article deals with the validation of the reference time mechanism. It shows the problems encountered to model this mechanism, has it needs specific dynamic considerations which can not be represented in TPN nor considered in the model checking phase. The solution is to consider an over-approximation of the real possible states and to finely analyze the validation results to conclude for the real states. The no-collision property verification result is explained in detail. This new validation phase allows the whole protocol validation, included the specific synchronization mechanism.

However, this validation work shows that the timed model checking for TPN is not as developed as we need. First, dynamic setting of time interval is not possible in the model and the validation phase. Second, the verification process often implies the use of system parameters, either in the model (for example hardware specific durations), or in the property itself (as for example the maximal execution time between two events). It could be useful to integrate the parameter concept in the validation phase. But the parameterized model checking is still an open problem, has it has been proved undecidable in [1]. Solutions to avoid the complexity are not yet mature, especially for the TPN formalism, so they can not be used for now. Therefore, an interesting continuation of this work should be the study and the improvement of the interface between the theoretical formal methods, and their associated tools, with the actual needs of the systems validation. Especially in specific contexts as embedded ones, where real-time constraints and hardware parameters have to be considered during the validation process. This work has already begun with the development of a validation tool: LPT (Little Parametric Tool) [8], which allows the parameterized verification of the maximal execution time between two transitions using automatic dichotomy. This work should be deepened to provide a useful and suitable formal validation tool. In the same idea, we work on the interfacing between the VHDL code generator and this new validation tool, in order to closely link the system design process and the formal validation one.

References

[1] R. Alur and D. L. Dill. A theory of timed automata. *Theor. Comput. Sci.*, 126:183–235, 1994.

[2] D. Andreu, D. Guiraud, and G. Souquet. A distributed implantable architecture for activating the peripheral nervous system. *Journal of Neural Engineering*, 16(6):227–258, 2009.

[3] B. Berthomieu, P.-O. Ribet, and F. Vernadat. The tool tina – construction of abstract state spaces for petri nets and time petri nets. *International Journal of Production Research*, 42(14), July 2004.

[4] B. Bordbar, R. Anane, Okano, and Kozo. An evaluation mechanism for qos management in wireless systems. In *Proceedings of the 11th International Conference on Parallel and Distributed Systems*, ICPADS'05, Washington, DC, USA, 2005.

[5] E. M. Clarke and E. A. Emerson. Design and synthesis of synchronization skeletons using branching-time temporal logic. In *Logic of Programs, Workshop*, pages 52–71, London, UK, 1982. Springer-Verlag.

[6] G. S. D. Andreu and T. Gil. Petri net based rapid prototyping of digital complex system. In *IEEE Computer Society Annual Symposium on VLSI (ISVLSI08)*, Montpellier, France, April 2008.

[7] A. David and W. Yi. Modelling and analysis of a commercial field bus protocol. In *Proceedings of the 12th Euromicro conference on Real-time systems (ECRTS'00)*, June 2000.

[8] K. Godary. LPT : Little parametric tool, outil pour la validation d'une borne temporelle paramtre. In *6ime Confrence Internationale Francophone d'Automatique (CIFA'08)*, Bucarest, Roumanie, 2008.

[9] K. Godary, D. Andreu, and G. Souquet. Sliding time interval based MAC protocol and its temporal validation. In *Proceedings of the 7th IFAC International Conference On FieldBuses & Networks in Industrial & Embedded Systems*, pages 119–126, Nov. 2007.

[10] K. Godary-Dejean and D. Andreu. Formal validation of a deterministic MAC protocol. Technical report, LIRMM, 2010. Restricted access.

[11] L. Huaming and T. Jindong. Heartbeat driven medium access control for body sensor networks. In *Proc. of the 1st International Workshop on Systems and Networking Support for Healthcare and Assisted Living Environments (HealthNet'07)*, Puerto Rico, USA, June 2007.

[12] C. E. I. Demirkol and F. Alagz. MAC protocols for wireless sensor networks: a survey. *IEEE Communications Magazine*, 44(4):115–121, 2006.

[13] D. E. J. Elson. Fine-grained network time synchronization using reference broadcast. In *Proc. of the Fifth Symposium on Operating Systems Design and Implementation (OSDI'02)*, Boston, MA, USA, December 2002.

[14] I. A.-B. K. Godary and A. Mignotte. Temporal bounds for TTA: Validation. In *IFIP Working Conference on Distributed and Parallel Embedded Systems (DIPES'04)*, Toulouse, France, August 2004.

[15] G. Leen and D. Heffernan. Formal verification of the TTCAN protocol. www.ttcan.com.

[16] H. Lnn and P. Pettersson. Formal verification of a tdma protocol start-up mechanism. In *In Pacific Rim International Symposium on Fault-Tolerant Systems (PRFTS'97)*, pages 235–242. IEEE Computer Society, 1997.

[17] G. S. M. Maroti, B. Kusy and A. Ledeczi. The flooding synchronization protocol. In *Proc. of the Second ACM Conference on Embedded Networked Sensor Systems - SenSys'04*, Baltimore, Maryland, USA, November 2004.

[18] P. Merlin. *A study of the recoverability of computing systems*. PhD thesis, Department of Information and Computer Science, University of California, Irvine, CA, 1974.

[19] R. K. S. Ganeriwal and M. Srivastava. Timing-sync protocol for sensor networks. In *Proc. of the First ACM Conference on Embedded Networked Sensor Systems (SenSys'03)*, Los Angeles, CA, USA, November 2003.

[20] J. Sifakis. A unified approach for studying the properties of transition systems. *Theoretical Computer Science*, 18(3):227–258, June 1992.