Configuring the communication on FlexRay
- the case of the static segment[*]

Mathieu Grenier[1], Lionel Havet[2], Nicolas Navet[2]

1: LORIA - Nancy Université, BP 239, 54506 Vandoeuvre, France
2: INRIA - RealTime-at-Work, 615 Rue du Jardin Botanique, 54506 Vandoeuvre-lès-Nancy, France
Contact author: Nicolas Navet (Nicolas.Navet@realtimeatwork.com)

Abstract: This paper deals with the configuration of the static segment of a FlexRay network, in the case where the tasks producing the signals are not synchronized with the FlexRay communication cycle, as it can be the case, for instance, if legacy software is to be re-used. First, we provide solutions to verify the freshness constraints of the signals exchanged in the static segment, under the form of both simple non-schedulability tests and an exact analysis. Then we propose a heuristic to construct the communication schedule, which proved to be efficient in our experiments. Finally, we highlight some future work that should help us further optimize the configuration of FlexRay networks, be it in regard to hardware resource usage or dependability objectives.
Keywords: FlexRay, message scheduling, static segment, freshness constraint.

1   Introduction

Context of the study.   FlexRay [3, 14] is an industry initiative developed to fulfill the needs of upcoming automotive real-time control applications, that require high-speed transmission and dependability-oriented services. FlexRay supports both time-triggered and event-triggered communications through, respectively, the *static segment* and the *dynamic segment*, which once concatenated form the *communication cycle*. The dynamic segment obeys a priority based protocol, similar in the spirit to CAN, which is well-suited to event-triggered traffic and which facilitates the re-use of legacy software. This paper focuses on the configuration of the static segment where the transmissions are organized according to a TDMA-based protocol (Time Division Multiplexed Access): each ECU (Electronic Control Unit) possesses a certain number of slots with a system-wide length in each communication cycle, each slot allowing the transmission of a single FlexRay frame at most. Configuring the communication involves defining 64 distinct communication cycles, where each slot always belongs to the same station but the frames that are transmitted can differ.

Contribution of the paper.   We are interested here in the configuration of the static segment and do not address the case of the dynamic segment - the reader interested in the dynamic segment can refer to [10, 9]. We consider the situation where the tasks are executed asynchronously with regard to the FlexRay Communication cycle, they transmit and receive signals through a middleware layer

---

[*]This document is a working paper on the basis of a publication at the 4th European Congress on Embedded Real Time Software (ERTS 2008), Toulouse, France, January 29 - February 1, 2008. This version dated February 13, 2008.

which, when assumptions are needed, is assumed to be AUTOSAR given its widespread adoption. In particular, the specific requirements of the AUTOSAR FlexRay Interface [7] will be taken into account. The configuration problem addressed is to set the characteristics of the communication in the static segment in such a way as to 1) meet the freshness constraints of the signals and 2) minimize the bandwidth utilization (*i.e.*, use the lowest data rate and the least number of slots). The latter point is important for enabling the use of low cost electronic components and for facilitating an incremental design process.

Organization of the paper. The assumptions made on the use of FlexRay, as well as the notations used throughout this study, are introduced in Section 2. In Section 3, it is shown how to check that the signals timing constraints are met. Section 4 presents an algorithm to build an optimized communication schedule and some experiments to assess its performance. Finally, Section 5 highlights some insights gained from this study.

2   System model

2.1   Data production

We consider a set of applicative-level tasks that are running *asynchronously* with regard to the FlexRay communication cycle. Asynchronous means here that the instances of the tasks are not driven by the FlexRay network (*i.e.*, they are not triggered by communication related events such as the arrival of messages - their periods are not related the length of the communication cycle). This is usually the case if the application has not been conceived and configured specifically for a given FlexRay communication cycle, as with legacy software that is re-used. However, we assume that the scheduling of tasks is brought up by the network, for instance, the scheduling on each node starts right after the network has been synchronized, at the beginning of the first communication cycle[1].

Task instances are assumed to produce signals at the end of their execution, we assume that all signal instances must be transmitted at least once (*i.e.*, no signal is permitted to be overwritten in a transmission buffer if it has not been transmitted). The model is illustrated in Figure 1. Each signal s_i is characterized by a tuple $(N_i, T_i, O_i, C_i, D_i)$ where:

- N_i is the identifier of the ECU which produces the signal,

- T_i is the *period of production* - generally corresponding to the period of the producing task,

- O_i is the *offset of the signal*, that is the latest time after which the first instance of the signal is produced (and subsequent values of the same signal will be released at times $O_i + k \cdot T_i$ at the latest). The offset is expressed with regard to the start of the first FlexRay communication cycle,

- C_i its *size* in bits,

---

[1]The tasks are thus not fully asynchronous with regard to the bus, but, by synchronous, we mean something stronger where the task periods are multiple of the communication cycle's length and each task instance is triggered by some communication-related events. If the startup of the scheduling is not triggered by the network, without further assumptions, it implies that each signal can be produced at each and every point in time during the communication cycle, and little can be done in terms of optimization. In particular, it is always possible that the signal value is produced after the slot where its transmission was planed, which implies a delay, before reception by the consumer of the data, equal to at most the transmission period (see the beginning of section 3 for an example). In that case, oversampling - at the cost of an increased bandwidth usage - is probably the only way to ensure better temporal performances.

- The *age of a signal* is the duration between the production of the signal on the sender side and the end of transmission of the first frame carrying the signal. The same signal value can be transmitted in several FlexRay cycles until it is updated by the producing task but only the first frame is significant since the subsequent ones do not provide any new information. D_i is the *freshness constraint* associated to signal s_i, that is the maximum age acceptable at the consumer end. The maximum age that is actually experienced by signal s_i in a given configuration is denoted by $\mathcal{A}_i$, and one needs $\mathcal{A}_i \leq D_i$ for the system to be schedulable[2].
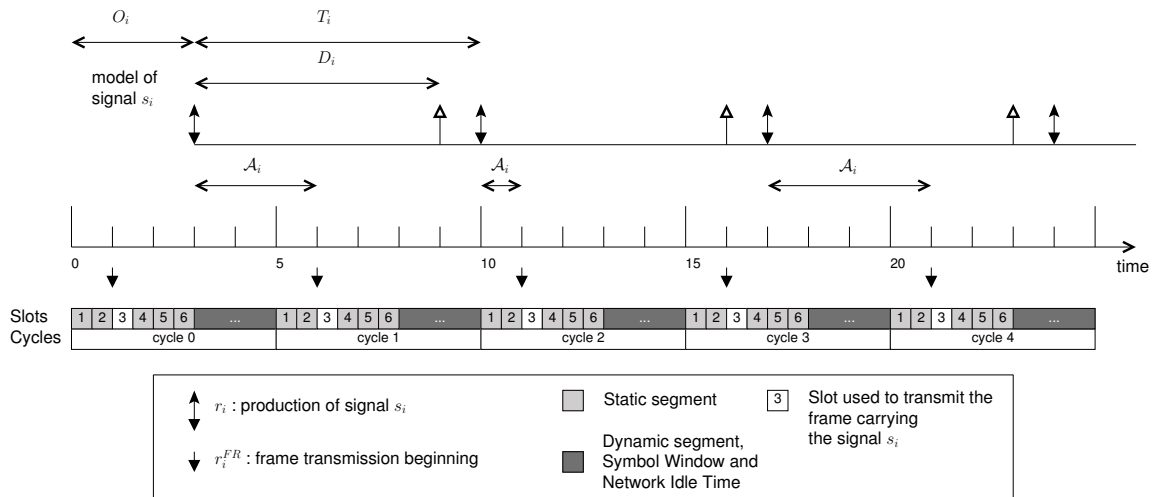

Figure 1: Illustration of the system model. Successive signal instances are produced and transmitted in the third slot of every communication cycle where r_i is the release time of an instance of the signal s_i and r_i^{FR} is the beginning of transmission of an AUTOSAR frame sending s_i.

2.2 Assumptions

Throughout this study, we place a set of assumptions on the use of FlexRay. These assumptions are in line with the design choices made by BMW (see [13] and [8]) and, more generally, with what is foreseen at the time of writing regarding the use of FlexRay in the automotive context:

1. The size of the communication cycle and the static segment are fixed respectively to 5 and 3ms (*i.e.*, the dynamic segment plus the Symbol Window and Network Idle Time last 2ms). These are parameters that cannot be optimized. With this setup, it is possible to accommodate 2.5ms freshness constraints that are thought to be possible in the next few years, by allocating 2 slots in the static segment. In this study, we limit ourselves to the case where each signal is sent at most once in a communication cycle because sub-5ms signals will most often be produced by safety-critical tasks running synchronously with the communication cycles, and thus are outside the scope the paper[3]. However, the part of the static segment left for the transmissions of the asynchronous tasks should be used at the fullest of its potential, and this is what is dealt with in this study,

---

[2]In the following, schedulable and feasible are synonyms.

[3]It would be possible as well to meet sub-5ms using the dynamic segment but strong timing and dependability constraints are more easily and efficiently ensured with time-triggered communications, thus the likely choice of the static segment for such kind of signals probably related to the control of the vehicle. The reader is for instance referred to the extensive work on the Time-Triggered Architecture (TTA) by the team of Prof. Kopetz at Vienna University for further information.

2. In the static segment, the size of the frames, and thus the duration of the slots, is fixed to a "reasonable value" chosen by the car manufacturer. For instance, BMW in its $X5$ uses 16 bytes frames (see [13]), but 24 bytes may be a viable option as well. Since, in addition, the length of the static segment is fixed, the maximum number of static slots available, *gNumberOfStaticSlots,* is known (see paragraph 3.1.1 for more details),

3. Only one communication channel is used since for the time being redundant transmission supports seem not to be considered necessary by carmakers, be it for dependability reasons or in order to increase the bandwidth,

4. FlexRay is used within an AUTOSAR communication stack (see, in particular, AUTOSAR FlexRay Interface [7]). In this context, configuring a frame, called an *AUTOSAR frame* in the remainder of the paper, involves deciding :

   (a) the *slot* used to transmit the frame (parameter *FRIF_SLOT_ID*),

   (b) the *base cycle* (*FRIF_BASE_CYCLE*), which is the first cycle out of the 64 (see [3]) in which the frame is scheduled for transmission,

   (c) the *repetition* of the frame (parameter *FRIF_CYCLE_REPETITION*), which is the period of transmission expressed as a multiple of the communication cycle with the AUTOSAR constraints that the repetition takes its value in $\{2^n \mid n \in \mathbb{N},\ n \leq 6\}$.

5. Each signal fits within one frame (*i.e.*, no segmentation as provided by the FlexRay Transport Layer [6]), and is sent in a separate frame. If some frame packing strategy is implemented (see [12] and the discussion in the conclusion of this document), it is done at the applicative level,

6. We consider the typical case where $D_i \leq T_i$, which implies that, if timing constraints are met, no buffer overwrite takes place,

7. We assume that the clock used for the applicative tasks does not drift apart from the FlexRay clock. In practice, the possible drift can be controlled by resynchronizing the OS clock on the FlexRay clock, as it is possible in the AUTOSAR context.

In the following, the term *configuration* denotes the allocation of the slots of the static segment to the ECUs and, for each slot, the allocation of the signals to the frames sent in the 64 communication cycles. Once the configuration is set, the communication schedule in the static segment is fully defined.

3   Verifying timing constraints on FlexRay

It is widely said that configuring communications with guaranteed response times on time-triggered buses is much easier than on event triggered buses. This is certainly partly true in the sense that no really complex schedulability analysis is needed to assess the feasibility of a given configuration. However if one aims to optimize the utilization of the hardware resources - in order to make use of a lower data rate or simply to connect more ECUs on the network - then the problem becomes much more involved, and communication cycles should be explicitly conceived with the objective in mind to minimize the bandwidth usage while meeting timing constraints.

There are two main reasons why optimizing the design of the communication is important. First, as discussed in paragraph 4.4 (considering the static segment alone), the bandwidth that is actually available is much smaller than the nominal workload of the network, in particular because oversampling can not be avoided in the general case. In addition, on the contrary to CAN, the communications in the static segment are driven by FlexRay's own internal clock. If - as in this paper - we consider applicative tasks that are running asynchronously with regard to the communication then it happens that a signal has to wait several communication cycles before being transmitted. As an example, let us consider a signal sent every 100ms, with a 5ms communication cycle, if the possibility of transmission errors is overlooked, it will be transmitted every 16 cycles (by assumption 4 in paragraph 2.2 and because no signal overwrites are permitted - see equation 2). Shall a new value of the signal be released immediately after the beginning of transmission of the frame that carries the signal, then the transmission will take place 90ms later. In that case, the response time, and thus the signal age at the consumer end, is greater than that what would have been obtained on most CAN networks. Indeed, in our experience, a worst-case response time equal to 90ms can only be observed in fairly loaded 125Kbit/s CAN buses (see [5]).

3.1 Basic non-schedulability condition

We propose a simple test to detect that a given set of signals is for sure not schedulable on a network with certain characteristics (i.e., data rate, number of slots in the static segment, length of the communication cycles, number of stations). For a configuration, not being ruled out by the test is a necessary but non-sufficient condition for feasibility, which means that some set of signals might not be discarded by the test while there is no configuration leading to a feasible schedule. However, the test is useful to rule out the use of some data rate, and to provide an global overview of the system's load and possibility of evolution. The first step is to determine the number of slots available in the static segment.

3.1.1 Number of slots available in the static segment

FlexRay Specification, see Annexe B.4 in [3], provides the necessary information about configuration constraints and configuration parameters determination. Various error terms influence the precision that can be achieved by the FlexRay clock synchronization algorithm. In order to choose the proper configuration parameters, it is first necessary to know the attainable precision of the clock synchronization, which is mainly influenced by the network topology : a large and complex network with several stars will result in a low synchronization accuracy.

The *action point offset*, denoted $gdActionPointOffset$, is the time instant in a slot after which the frame starts being transmitted. In Annexe B.4.6, it is precised that $gdActionPointOffset$ must be greater than the attainable precision. Hence, the larger the $gdActionPointOffset$, the larger the slot size, and thus fewer slots in the communication cycles. In this study, we assume a simple network topology (*i.e.*, no star, bus length $\leq$ 20 meters) which leads to a good attainable precision and allows for a small action point offset. We use the following FlexRay configuration settings, which are in line with the settings disclosed by BMW ([13]):

- The data payload of a static frame, $gPayloadLengthStatic$, is set to 8 2-bytes words, which is 16 bytes,

- The communication cycle is fixed to 5ms, denoted here by $gdCycleLength$, and the length of the static segment is 3ms (see paragraph 2.2),

- The length of the macrotick, $gdMacrotick$, is set to $2\mu s$.

Based on these settings, and following the equations in Annexe B, one can derive the maximum number of slots in the static segment ($gNumberOfStaticSlots$) at different bit rates. The results are shown in Table 1.

Bit Rate (Mbit/s)	2.5	5	10
$gNumberOfStaticSlots$	**27**	**51**	**93**

Table 1: Number of slots in the static segment for different data transmission rates.

Logically, the configuration at 10Mbit/s is close to what BMW is using in the latest X5 model, where $gNumberOfStaticSlots$ is equal to 91.

3.1.2 Minimum number of slots required

The non-schedulability condition is based on the minimum number of slots required by the set of signals of a given application. The required number of slots must be lower than $gNumberOfStaticSlots$, otherwise we know for sure that the set of signals is not schedulable. We stress that this is a necessary but non-sufficient condition: there are configurations that will pass the test without being actually schedulable. Also, the condition holds under the assumption that there is no frame-packing: one signal uses a full slot.

For a node k, sending a set of signals S, the minimum number of slots needed is :

$$n_k = \sum_{\{i \mid N_i = k\}} \frac{1}{\overline{T_i}} \tag{1}$$

where $\overline{T_i}$ is the *repetition* of the frame (*FRIF_CYCLE_REPETITION* in AUTOSAR terminology), which is the period of transmission expressed as a multiple of the communication cycle, with the constraints[4] that

$$\overline{T_i} = max\{2^n \mid n \in \mathbb{N},\ n \le 6 \text{ and } 2^n \cdot gdCycleLength \le T_i\}. \tag{2}$$

The minimum number of slots required by all stations is thus $\sum_k \lceil n_k \rceil$. For instance, let us consider a network with 4 stations sending each 10 signals with a period of 10ms and 10 signals with a period of 20ms. This configuration requires at least $4 \cdot \lceil 10 \cdot 0.5 + 10 \cdot 0.25 \rceil = 32$. This is larger than the number of static slots available at 2.5Mbit/s (see Table 1), and thus we know that the configuration is not schedulable at 2.5Mbit/s.

This non-schedulability condition offers a simple coarse-grained test, that is independent of the slot allocation and the deadlines of the signals. In our experience, the test is relatively accurate as long as the freshness constraints of the signals are close to their periods. This test will be refer to as "non-schedulability test 1", or "test 1" for short, in the experiments of paragraph 4.5.

The next paragraph presents a necessary and sufficient test for the case where a configuration is defined. In paragraph 3.3, this latter test is used in turn to come up with a more precise non-schedulability test that takes deadlines into account without requiring that the configuration of the communication is defined.

---

[4]Indeed, if $\overline{T_i} \cdot gdCycleLength > T_i$ (*i.e.*, the transmission period is greater than the signal production period), some signals will be lost.

3.2 Schedulability of a configuration

Here we explain how to decide on the schedulability of a given configuration. It consists in checking that the freshness constraint associated to each signal is verified. If at least one signal is sent in an AUTOSAR frame that does not ensure the freshness constraint, then the configuration is non-schedulable. In the remainder of the paragraph, we show how to estimate the maximal age $\mathcal{A}_i$ experienced by a signal in a given configuration.

3.2.1 General case

Let us consider a signal s_i with production period T_i, offsets O_i and freshness constraint D_i (see paragraph 2.1). This signal is assigned to an AUTOSAR frame with given *FRIF_SLOT_ID* and *FRIF_BASE_CYCLE* characteristics. From the two latter information, we deduce the beginning of the first slot after the startup of the communication in which the frame is scheduled for transmission, this time point is called the *offset of the frame* and is denoted by O_i^{FR}. The next characteristic of interest here is *FRIF_CYCLE_REPETITION*: T_i^{FR} denotes the time between two subsequent transmissions of the frame ($T_i^{FR} =$ *FRIF_CYCLE_REPETITION* $\cdot gdCycleLength$).
Let r_i be the release time of any instance of the signal s_i, let r_i^{FR} be the beginning of transmission of any AUTOSAR frame sending s_i. We know from Lemma 2 in [11] that

$$r_i^{FR} - r_i = p \cdot g + (O_i^{FR} - O_i) \bmod g \tag{3}$$

where $g = \gcd(T_i^{FR}, T_i)$, $p \in \mathbb{Z}$, with gcd being the greatest common divisor. The quantity $r_i^{FR} - r_i$ is equal to $\mathcal{A}_i$ in Figure 1.
Since the construction of an AUTOSAR frame requires some time, we place the assumption that signal s_i must be released at least *PackingTime* before r_i^{FR}:

$$r_i^{FR} - r_i \geq PackingTime. \tag{4}$$

As explained in paragraph 2.1, we only consider the first frame carrying the signal to evaluate the age. The release r_i^{FR} of the AUTOSAR frame which contains the value of the signal for the first time is such that:
$$r_i^{FR} - r_i < PackingTime + T_i^{FR} . \tag{5}$$

Indeed, otherwise r_i^{FR} is not the release of the first frame carrying the signal.
We now aim to identify the maximum value of the quantity $r_i^{FR} - r_i$, which comes to find the largest value that can be taken by p in equation 3. From equations 3 and 5, one has:

$$p < \frac{PackingTime + T_i^{FR} - \left((O_i^{FR} - O_i) \bmod g\right)}{g} .$$

Thus, the maximal duration between the production of the signal and the beginning of transmission of the first frame carrying that signal is $p_{\max} \cdot g + (O_i^{FR} - O_i) \bmod g$, where:

$$p_{\max} = \left\lceil \frac{PackingTime + T_i^{FR} - \left((O_i^{FR} - O_i) \bmod g\right)}{g} \right\rceil - 1 . \tag{6}$$

The maximum age of a signal is thus equal to

$$\mathcal{A}_i = p_{\max} \cdot g + (O_i^{FR} - O_i) \bmod g + slotSize \tag{7}$$

where $slotSize$ is the length of each slot in the static segment.

It is worth pointing out that in equations 6 and 7, the only two terms influencing the age that can be configured are O_i^{FR} and T_i^{FR}. In particular, reducing the transmission period below what we call the "natural" period (*i.e.*, the largest possible period $\overline{T}_i \cdot gdCycleLength$, see equation 2), and thus oversampling, will reduce the age. This strategy will be used when looking for feasible configurations.

3.2.2 Specific properties in the case where deadlines equal periods

In the case where the freshness constraint of each signal is equal to its period ($D_i = T_i$) - if we neglect the packing time and the frame transmission time - any possible choice for the parameters of the AUTOSAR frame will guarantee the freshness constraint since $T_i^{FR} \leq \overline{T}_i \cdot gdCycleLength \leq T_i = D_i$. Let us describe the following naive strategy executed on each station considered in an arbitrary order:

- Construct the set of AUTOSAR frames with their transmission period T_i^{FR} equal to $\overline{T}_i \cdot gdCycleLength$ (where i is the index of the signal the frame carries), rank the frames by increasing value of their period,

- While there are unassigned AUTOSAR frames on the station, consider the first available slot and fill it up by placing iteratively the frames in the first available cycle (i.e., set the frame base cycle to the first available cycle).

As shown in Appendix A, this naive strategy comes up with a configuration that uses exactly the minimum number of slots required, as computed in paragraph 3.1.2. With the aforementioned assumptions, schedulability test 1 becomes a necessary and sufficient condition.

However, if one really needs to be accurate, one can not neglect the packing time and the frame transmission time, and all choices are not equivalent in this context. Let us for instance consider the example shown in Figure 2 where a signal A is produced with a period and deadline equal to $gdCycleLength$ (length of the communication cycle) and the frame transmitting that signal is assigned to the first slot of the static segment (and transmitted in every cycle given its period). Now, imagine that signal A is produced after the beginning of the first slot of the FlexRay cycle, it will therefore have to wait for the first slot of the next communication cycle to be transmitted. The age of the signal at reception is thus $gdCycleLength + slotSize$, and thus the freshness constraint is not met.
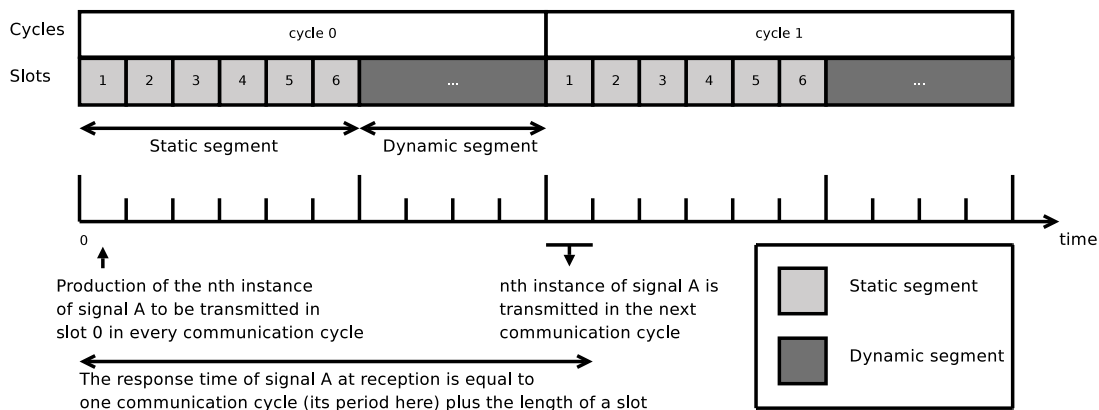

Figure 2: Example showing that the slot location matters even in the case where the signal deadline is equal to its period.

In practice, for an AUTOSAR frame, only one, or at most a few choices for the transmission slot and base cycle (typically out of several hundreds) are not compatible with the freshness constraint in the case where the deadline is equal to the period. This explains why, in the experiments of paragraph 4.5.1, all the signals sets that are not ruled out as unschedulable by test 1 are indeed schedulable with our algorithm.

In many practical cases the freshness constraints will be much smaller that the signals periods, especially for the signals with the largest periods, and, in that case, fewer configurations lead to a feasible schedule.

3.3 Improved non-schedulability condition

The frame repetition that is considered in equation 1 is the largest one that meets the AUTOSAR constraint expressed in assumption 4c (see equation 2 also). However it can happen that - with that repetition value - none of the slots leads to a feasible schedule. This occurs when, whatever the value of O_i^{FR} in equation 6, the age of the signal is greater than the freshness constraint. In the latter case, the only solution is to increase the transmission frequency. Precisely, the AUTOSAR constraint forces us to double the frequency each time. This can be accounted for in the non-schedulability condition, and equation 2 becomes:

$$\overline{T_i} = max\{2^n \mid n \in \mathbb{N},\, n \leq 6 \text{ and } \exists O_i^{FR} \text{ such that } \mathcal{A}_i \leq D_i\}. \tag{8}$$

where $\mathcal{A}_i$ is computed using equation 7. Of course, if for at least one signal no suitable value has been found for $\overline{T_i}$ then the set of signals as a whole is non-schedulable. This test will be refer to as "non-schedulability test 2" in the experiments of paragraph 4.5. The complexity of this second test does not raise any practical problems given the limited number of slots in the static segment (*i.e.*, a few hundreds at most).

It is worth pointing out that the difference between the number of slots required that is returned by test 2 and the number of slots returned by test 1, enables to estimate the cost of oversampling. This difference can be also computed on a signal per signal basis, which can serve to identify the signals that induce the most overheads in the static segment, and thus ones possibly best suited to the dynamic segment.

4 Optimizing cycle configuration

4.1 Performance metrics

The primary objective is to come up with configurations that are feasible in regard to the respect of the freshness constraints. The secondary objective is to minimize the bandwidth usage, which translates into minimizing the number of slots used. This latter objective is intended to preserve the possibility of evolution of the embedded system, which is crucial given the incremental design process in use in the automotive industry.

Another meaningful secondary objective would be to minimize the age of the signals at the consumer end (or, said differently, maximize the freshness of the data). The corresponding cost function could be for instance $\sum_{s_i} (D_i - \mathcal{A}_i) / D_i$, but other choices would be meaningful as well. However, this latter criterion is not compatible with minimizing the bandwidth usage since improving the freshness generally comes at the price of some oversampling. In this study, we focus on feasibility and bandwidth usage.

4.2 Size of the search space

Allocating slots belongs to the family of NP-hard problems (see, for instance, [10] for further information on the complexity issue of this problem), which means that this problem has no solution in polynomial time. The worst case scenario in terms of size of the search space occurs when each signal is sent in one slot and one cycle out of the 64 ($FRIF\_CYCLE\_REPETITION = 64$). In this setup, there are $n = 64 \cdot gNumberOfStaticSlots$ possible choices for where to send a signal. Assigning $\#signals$ consists in a permutation $A_n^{\#signals} = n!/(n - \#signals)!$, which, given the number of signals in practical cases, is much too big for an exhaustive search. To cope with the complexity, we propose a heuristic algorithm and assess it performances with a random search strategy as benchmark.

4.3 Configuration algorithm

The proposal consists in a greedy algorithm (*i.e.*, locally optimal decisions are made at each step) that maximizes, at each step, the number of signals scheduled for transmission in a single slot. The heuristic is called *Best Slot First* (BSF) because the slot, whose schedule is decided a given step, is the one that can accommodate the greatest number of signals. Given the greedy nature of the algorithm, the global optimum will not be obtained in general, however, as shown in paragraph 4.5, the algorithm performed well in our experiments.

Best Slot First Heuristic (BSF):

1. Construct the set of signals that each station is able to transmit during each unassigned slot. This is done by filling up the slots with as many AUTOSAR frames as possible according to the following procedure that is executed for each station and each slot:

   (a) Construct the list L of AUTOSAR frames that respect the timing constraints of the signals they carry. This list is obtained by enumerating and testing all base cycles and repetitions that are possible for each signal,

   (b) The AUTOSAR frames in L are ranked by decreasing values of the couples $(\overline{T}_i \cdot gdCycleLength/T_i^{FR}, \overline{T}_i)$: the smaller the numerical value of $\overline{T} \cdot gdCycleLength_i/T_i^{FR}$, the higher the rank in the list - the second component is used to distinguish frames having the same first component. The underlying idea is to favor the AUTOSAR frames having a repetition value close to the natural period of the signal the frame is carrying because it allows limiting the oversampling,

   (c) Assign the first frame in L to the slot, remove from L all AUTOSAR frames that are in conflict with this scheduling choice (*i.e.*, frames scheduled for transmission during the same cycle). If L is not empty, go to step 1b, otherwise return the set of signals transmitted with this schedule.

2. Set the schedule of the slot that can transmit the greatest number of signals - the schedule is the one computed at step 1c. If there is at least one unassigned slot, go to step 1. Otherwise, if there is no more slot available, and at least one signal still to place, no schedulable solution has been found.

For a given network, the algorithm runs in $O(\#signals^2 \cdot \log(\#signals))$, which does not raise practical problems.

Below is the description of the benchmark algorithm, called Randomized Slot Selection (RSS), to which BSF is compared. It should be pointed out that, mainly, what is random at each step of RSS is the choice of the scheduling solution that is kept among all feasible solutions, but not the construction of the solutions themselves.

Randomized Slot Selection (RSS):

1. For each signal, construct the list of all AUTOSAR frames that respect the freshness constraint,

2. One signal S is chosen randomly,

3. In the list of the frames sending S, if not empty, one frame F is chosen at random - the frame is scheduled for transmission according to its characteristics (see 4 in paragraph 2.2). If the list is empty, there is no more possible frame for signal S: the random allocation has failed and "non feasible" is returned,

4. Frame F that has just been placed is removed from the list, as well as all other frames that have become incompatible with the schedule of F. Precisely, 1) all frames scheduled for transmission in the same slot as F but not sent by the same ECU, 2) frames from the same ECU, sent in the same slot, that have a conflict with the choice made for F (*i.e.*, one instance of the frames collides with one instance of F),

5. Go to step 2 as long as the list of signals is not empty.

4.4 Experimental setup

In the experiments, we choose to consider the traffic exchanged on powertrain or chassis networks because that is where FlexRay is the most likely to be used in a first step. Sets of signals were generated using NETCARBENCH [1], a GPL-licensed software that creates random sets of signals, or random set of frames, written in a simple XML format, according to user-defined parameters (*e.g.*, network load, number of ECUs, distribution of the periods of the frames, etc.).

We would like to stress that, if 10Mbit/s is the data rate, the bandwidth that can actually be used in our experiments is much less important. Indeed, the static segment is $3/5$ of the whole communication cycle, the size of the slot is $slotSize = 3.10^{-3}/93 = 3.2 \cdot 10^{-5}$ second while each slot is only used to transmit a signal of at most 8 bytes (or $6.4 \cdot 10^{-6}$ second). Thus the useful bandwidth is: $10^7 \cdot 93 \cdot 6.4 \cdot 10^{-6}/5 \cdot 10^{-3} \approx 1.2$Mbit/s, and the corresponding overall efficiency of the protocol - without the dynamic segment here - is only 12%. In addition, this figure is obtained without taking into account the AUTOSAR constraint which says that the repetition factor must be a power of two in a certain range (see equation 2). For instance, a frame having a period equal to one second will be transmitted at least once every 64 cycles, which is every 320ms in our setup. The constraint on the repetition factor reduces the available bandwidth, which explains why 1Mbit/s, instead of 1.2Mbit/s, is the rate of applicative-level data used in the NETCARBENCH configuration file shown in Figure 3.

```
<?xml version="1.0" encoding="UTF-8"?>
<config>
<ecu Min="5" Max ="15" />
<load Min="30" Max="40" />
<bandwidth Value="1000" />
[...]
<p1 Value="10" Weight="5" [...]/>
<p2 Value="20" Weight="5" [...]/>
<p3 Value="50" Weight="5" [...]/>
<p4 Value="100" Weight="5" [...]/>
<p5 Value="200" Weight="5" [...]/>
<p6 Value="1000" Weight="5" [...]/>
<p7 Value="2000" Weight="2" [...]/>
<messages_sizes>
<m8 Length="8" [...] />
</messages_sizes>
</config>
```

Figure 3: Excerpt of a NETCARBENCH configuration file used to generate the sets of message, [...] means that non relevant information has been omitted. The periods are expressed in milliseconds, and randomly chosen according the distribution indicated by the parameters. For instance, a period equals to 2 seconds will be allocated with a probability $2/\sum \text{Weight}$ as specified by `<p7 Value="2000" Weight="2" [...]/>`.

The number of ECUs is randomly chosen between 5 and 15 (`<ecu Min="5" Max ="15"/>`). The load range (*i.e.*, only applicative level data, protocol overheads are not taken into account since here sets of signals are generated) for signals set drawn at random from this configuration file is specified by the line `<load Min="5" Max="10"/>` in Figure 3. Experiments were done at various levels of load: precisely, one hundred sets of messages were generated for each load level in the set $\{(Min = 0.3Mbit/s, Max = 0.4Mbit/s), (0.4, 0.5), (0.5, 0.6), (0.6, 0.7), (0.7, 0.8), (0.8, 0.9), (0.9, 1)\}$. Experiments were performed with and without offsets for the signals (parameter O_i, see paragraph 2.1), only the results without offsets are shown since the conclusions that can be drawn from both sets of experiments do not vary substantially.

4.5 Performance evaluation

Performances are evaluated with regard to the percentage of feasible configurations and the bandwidth usage. Two distinct cases are considered: deadlines equal to periods, and deadlines smaller than periods.

4.5.1 Deadlines equal to periods

Table 2 summarizes the results regarding feasibility. As can be seen in that Table, BSF leads to the best possible result in terms of feasibility when $D_i = T_i$. In that specific context, BSF makes, except in some particular cases, exactly the same configuration choices as the naive strategy that is shown to be near-optimal (see the explanations in paragraph 3.2.2 and the proof in Appendix A), which explains the results. On the contrary, the performance of RSS becomes quickly unacceptably low, one reason being that RSS does not strive to put additional signals in a slot already allocated to a station. In regard to the bandwidth usage (see Table 3), BSF is also clearly more efficient than RSS.

Load	$(0.3, 0.4)$	$(0.4, 0.5)$	$0.5, 0.6)$	$(0.6, 0.7)$	$(0.7, 0.8)$	$(0.8, 0.9)$	$(0.9, 1)$
Test 1&2	100	100	100	100	100	100	97
BSF	100	100	100	100	100	100	97
RSS	71	23	3	0	0	0	0

Table 2: Percentage of signal sets that are schedulable with Best Slot First (BSF) and Randomized Slot Selection (RSS) Heuristic. Schedulability test 1 and test 2 (row "Test 1&2") indicate the number of signal sets that might be feasible (*i.e.*, not unfeasible for sure), they both have the same outcome since $T = D$ here. The load is the signal bit rate in Mbit/s (*i.e.*, useful data only - no protocol overhead). The statistics are made on samples of 100 sets of signals. When the network load exceeds 1Mbit/s, the minimum number of slots required (see paragraph 1) is most often larger than the number of available slots.

Load	$(0.3, 0.4)$	$(0.4, 0.5)$	$0.5, 0.6)$	$(0.6, 0.7)$	$(0.7, 0.8)$	$(0.8, 0.9)$	$(0.9, 1)$
Test 1&2	34.2	44.1	51.8	61	68.9	77.4	86.3
BSF	34.2	44.1	51.8	61	68.9	77.4	86.3
RSS	93	93	93	NA	NA	NA	NA

Table 3: The row "Test 1&2" shows the minimum number of slots that is required at each load level as computed by test 1 and test 2 (both tests are equivalent here since $T = D$). Average number of slots used with Best Slot First (BSF) and Randomized Slot Selection (RSS) Heuristic. The maximum number of slots available is 93. The load is the signal bit rate in Mbit/s (*i.e.*, useful data only - no protocol overhead). Statistics made on the signal sets that are feasible in Table 2. NA means that no statistics can be made since RSS is unable to come up with feasible solutions in the most loaded configurations.

4.5.2 Deadlines smaller than periods

The deadlines are now more stringent since they are set to 30ms for all signals having a period greater than 30ms (deadlines are unchanged for the other signals). The first observation that can be drawn from Table 4 is that test 2 is much more accurate than test 1 when deadlines are smaller than periods. In this setup, test 1 is too imprecise to be really useful. What is more informative is the difference between the number of slots required by test 2 and test 1 (see Table 5), which indicates the cost of oversampling required to meet the freshness constraints. Indeed, for a given deadline - here 30ms - there is a threshold for the transmission period above which oversampling is necessary. At least part of the oversampling overheads could be saved by placing some or all of the signals with a period above that threshold in the dynamic segment of FlexRay, especially by taking advantage of the nice "slot multiplexing" feature that allows several stations to share the same dynamic slot.
From Table 4 and Table 5 it is also clear that BSF greatly outperforms the random strategy RSS both in feasibility and number of slots used, as it was the case as well when deadlines were equal to periods. At low and moderate bus load, BSF is efficient and produces feasible configurations for most signal sets, which, in addition, do not require many more slots than the lower bound given by test 2. At higher bus loads, above 600Kbit/s, the difference between BSF and test 2 becomes more pronounced. There are two possible reasons: test 2 is probably fairly optimistic in the sense that unfeasible signal sets are not detected as such (*i.e.*, there are conflicting requirements between signals that are not captured by the test), and BSF might not be able to come up with solutions in some cases. The extent to which the root cause is the first or second reason should be determined in the light of additional experiments, especially experiments involving more CPU-intensive search heuristics, such as genetic algorithms or local search techniques as hill-climbing.

Load	$(0.3, 0.4)$	$(0.4, 0.5)$	$0.5, 0.6)$	$(0.6, 0.7)$	$(0.7, 0.8)$	$(0.8, 0.9)$
Test1	100	100	100	100	100	100
Test2	100	100	91	45	11	0
BSF	100	89	59	7	0	0
RSS	48	5	0	0	0	0

Table 4: Percentage of signal sets that are schedulable with Best Slot First (BSF) and Randomized Slot Selection (RSS) Heuristic. The figures shown for non-schedulability test 1 (see paragraph 1) and non-schedulability test 2 (see paragraph 3.3) denote the number of signal sets that might be feasible using respectively test 1 and test 2. The load is the signal bit rate in Mbit/s (*i.e.*, useful data only - no protocol overhead). Statistics made on samples of 100 sets of signals.

Load	$(0.3, 0.4)$	$(0.4, 0.5)$	$0.5, 0.6)$	$(0.6, 0.7)$	$(0.7, 0.8)$	$(0.8, 0.9)$
Test1	34.2	44.1	51.8	61.1	69.1	77.2
Test2	51.8	68.2	78.8	87.6	90.8	NA
BSF	55.5	75.5	90	90.9	NA	NA
RSS	93	93	NA	NA	NA	NA

Table 5: Average number of slots used with Best Slot First (BSF) and Randomized Slot Selection (RSS) Heuristic, as well as the average number of slots required as computed by non-schedulability test 1 and non-schedulability test 2. The maximum number of slots available is 93. The load is the signal bit rate in Mbit/s (*i.e.*, useful data only - no protocol overhead). Statistics made on the signal sets that are feasible in Table 4.

5   Concluding remarks

The configuration of the dynamic segment of FlexRay has already received quite a lot of attention in the literature. This paper provides a first study focused on the static segment in the case where tasks are executed asynchronously with regard to the communication cycle. The contribution of the paper is threefold. First, we tried to highlight a set of assumptions that correspond to what can be foreseen at the time of writing of the use of FlexRay in the automotive context. Configuring a FlexRay cluster is complex problem, more than 70 independent parameters have to be set [2], and thus deciding beforehand what is outside the scope of optimization is crucial to reduce the search space and come up with efficient solutions. Second, we propose ways to verify the freshness constraints of the signals exchanged on the bus, under the form of both simple non-schedulability tests and an exact analysis. Beside the answer on the feasibility of the scheduling, these tests might help to identify signals that would be best transmitted in the dynamic segment. Finally, we propose a heuristic to construct the communication schedule in the static segment, which proved to be efficient in the first sets of experiments run for this study. The heuristic can be used as a starting point to direct the search towards promising parts of the solution space in more CPU-intensive optimization algorithms. In particular, the configuration produced by the heuristic can be included in the initial population of a genetic algorithm, knowing that the initial population has often a strong impact on the final performance of the algorithm.

FlexRay is still lacking higher-level protocol layers providing dependability-oriented services such as group membership, clique avoidance, or even frame acknowledgment. When safety-critical functions will be implemented on top of FlexRay soon, dependability cannot be overlooked anymore. Fortunately, the experience gained over the years with other TDMA networks, TTP/C in particular,

can be re-used to some extent. This is true in particular for the configuration of the communication cycle, and the work presented here can be extended to take into account dependability objectives, for instance, by integrating the results presented in [4] where the possibility of transmission errors is considered.

The results and the analysis presented here are preliminary and further investigation is certainly required. What is needed also is some more feedback about the use of FlexRay, be it on production or prototype cars. From our perspective, the following insight into how to use FlexRay can be stated:

- If the applicative tasks are executed asynchronously with regard to the FlexRay communication cycle, then the response times of the frames sent in the static segment can be equal to the transmission period (without oversampling). This can be larger than what would have been observed on a CAN bus, especially when offset strategies are implemented (see [5]).

- The amount of applicative-level data that can be transmitted in practice during the static segment of FlexRay - even on a 10Mbit/s network - is rather limited. The first consequence is that best configuring the static segment is crucial. The second consequence is that the static segment should probably be used preferentially for communication involving tasks running synchronously with the communication cycles.

- Given the discrepancy between the typical size of a signal (a few bytes at most) and the typical size of a FlexRay frame (at least 16 bytes in practice), frame packing strategies, as there are for CAN (see [12]), are needed. The problem is more complex in the context of an AUTOSAR stack because the packing can be done at two levels:

  1. packing signals into AUTOSAR bus independent PDUs, whose size are ≤ 8 bytes, this is done at the AUTOSAR COM Layer,
  2. packing PDUs into FlexRay frames, according to the *Frame Construction Plan*, at the FlexRay Interface level.

Configuring a FlexRay network is very complex given the number of parameters involved and their interactions, and also given the number of, sometimes contradictory, design objectives. Software tools are needed to help, and when possible to guide the FlexRay system designer. This is our ongoing work to extend our software tool NETCAR-Analyzer in that direction, see reference [5] and `http://www.realtimeatwork.com` for further information.

References

[1] C. Braun, L. Havet, and N. Navet. NETCARBENCH: a benchmark for techniques and tools used in the design of automotive communication systems. In *Proc. of the 7th IFAC International Conference on Fieldbuses and Networks in Industrial and Embedded Systems (FeT'2007)*, November 2007. Software and manual available at *http://www.loria.fr/~nnavet/netcarbench/*.

[2] *J. Broy and K. Muller-Glaser. The impact of time-triggered communication in automotive embedded systems. In* IEEE Second International Symposium on Industrial Embedded Systems (SIES'07)*, pages 353–356, 4-6 July 2007.*

[3] *FlexRay Consortium. FlexRay communications system - protocol specification - version 2.1. Available at $http:// www. flexray. com$, December 2005.*

[4] B. Gaujal and N. Navet. *Maximizing the robustness of TDMA networks with applications to TTP/C.* Real-Time Systems*, 31(1-3):5–31, December 2005.*

[5] M. Grenier, L. Havet, and N. Navet. *Pushing the limits of CAN - scheduling frames with offsets provides a major performance boost. In* ERTS Embedded Real Time Software 2008*, 2008.*

[6] AUTOSAR Development Partnership. *Specification of FlexRay transport layer. Available at* $http://www.autosar.org$*, June 2006. Version 2.0.1.*

[7] AUTOSAR Development Partnership. *Specification of module FlexRay interface. Available at* $http://www.autosar.org$*, June 2006. Version 2.0.0.*

[8] M. Peteratzinger, F. Steiner, and R. Schuermans. *Use of XCP on FlexRay at BMW. Translated reprint from HANSER Automotive 9/2006, available at url* $https://www.vector-worldwide.com/vi\_downloadcenter\_en,,223.html?product=xcp$*, 2006.*

[9] T. Pop, P. Pop, P. Eles, and Z. Peng. *Bus access optimisation for flexray-based distributed embedded systems. In* DATE '07: Proceedings of the conference on Design, automation and test in Europe*, pages 51–56, San Jose, CA, USA, 2007. EDA Consortium.*

[10] T. Pop, P. Pop, P. Eles, Z. Peng, and A. Andrei. *Timing analysis of the FlexRay communication protocol. In* ECRTS '06: Proceedings of the 18th Euromicro Conference on Real-Time Systems*, pages 203–216, Washington, DC, USA, 2006. IEEE Computer Society.*

[11] G. Quan and X. Hu. *Enhanced fixed-priority scheduling with (m,k)-firm guarantee. In* Proceedings IEEE Real-Time System Symposium (RTSS'2000)*, 2000.*

[12] R. Saket and N. Navet. *Frame packing algorithms for automotive applications.* Journal of Embedded Computing*, 2:93–102, 2006.*

[13] A. Schedl. *Goals and architecture of FlexRay at BMW. Slides presented at the Vector FlexRay Symposium, March 2007.*

[14] B. Schätz, C. Kühnel, and Gonschorek. The FlexRay protocol. In N. Navet and F. Simonot-Lion, editors, Automotive Embedded Systems Handbook. *CRC Press/Taylor and Francis, to appear in 2008.*

A Cycle configuration when deadlines are equal to periods

Here we show that, under some hypotheses, the naive algorithm described in paragraph 3.2.2 leads to an optimal schedule in the sense that it uses the minimum number of slot required, as computed by the schedulability condition of paragraph 3.1. Precisely the results is shown under the hypotheses that the transmission time of the frame and the frame packing time are neglected in the evaluation of the age of the signal. The reader is referred to paragraph 3.2 for the motivation and the notations. Some additional notations are needed here. In paragraph 3.2, T_i^{FR} was expressed in time unit, we now need to quantify the repetition factor as a multiple of the communication cycle denoted by $\bar{T}_i^{FR}$. Precisely, one has $\bar{T}_i^{FR} = T_i^{FR}/gdCycleLength$. The slot h during the gth communication cycle is identified by $S_{h,g}$ and called a "*slot-cycle*". With this definition, there are 64 available slot-cycles for each slot of the static segment. One says that the slot-cycles $S_{h,g}$ and $S_{h,g+1}$ are *consecutive*. At this point, some observations can be done:

1. As shown in paragraph 3.2.2, in this context, any possible choice for the parameters of the AUTOSAR frame guarantees the freshness constraint,

2. In a given slot, the base cycle for a frame having a repetition equal to $\bar{T}_i^{FR}$ must be chosen in the $\bar{T}_i^{FR}$ first slot-cycles (AUTOSAR constraint, see item 4 in paragraph 2.2),

3. Let $\bar{T}_i^{FR}$ and $\bar{T}_k^{FR}$ be the repetitions of two AUTOSAR frames f_i and f_k with $\bar{T}_k^{FR} \geq \bar{T}_i^{FR}$. With the "power of two constraint", one has $\bar{T}_k^{FR} = 2^n \cdot \bar{T}_i^{FR}$ with $n \in \mathbb{N}$. AUTOSAR frame f_i uses $\frac{\bar{T}_k^{FR}}{\bar{T}_i^{FR}}$ slot-cycles over $\bar{T}_k^{FR}$ consecutive slot-cycles. For instance, an AUTOSAR frame f_1 with $\bar{T}_1^{FR} = 4$ uses 2 slot-cycles out of $\bar{T}_2^{FR} = 8$ consecutive slot-cycles.

4. Let $\mathcal{S}_h$ be the set of AUTOSAR frames assigned to a slot h. If $\sum_{f_i \in \mathcal{S}_h} \frac{1}{\bar{T}_i^{FR}} = 1$ then the 64 slot-cycles of slot h are used.

For each slot, the naive strategy places iteratively the frames in the first available slot-cycle: i.e., it sets the frame base cycle to the first available slot-cycle. At each step, the algorithm takes the first AUTOSAR frame (in the order of the transmission period), denoted by f_k in the following, among the frames that have not been assigned a slot-cycle yet. Let S_h be the set of AUTOSAR frames already allocated to the current slot h. We show that if there is at least one free slot-cycle in slot h, then the base cycle of f_k can be assigned to this slot-cycle.

Proof:

The proof is done in two steps:

1. show that if there is at least one free slot-cycle in slot h, there is at least one free slot-cycle in the $\bar{T}_k^{FR}$ first slot-cycles,

2. if the base cycle of f_k is set to the first free slot-cycle, then all instances of f_k (all slot-cycles used by f_k in the 64 communication cycles) will be transmitted during slot-cycles that are still free (*i.e.*, no conflicts with other AUTOSAR frames).

Step 1:

From observation 4, if there is at least one available slot-cycle in slot h then $\sum_{f_i \in \mathcal{S}_h} \frac{1}{\bar{T}_i^{FR}} < 1$ (otherwise all slot-cycles are used). This induces that:

$$\bar{T}_k^{FR} \cdot \sum_{f_i \in \mathcal{S}_h} \frac{1}{\bar{T}_i^{FR}} < \bar{T}_k^{FR}$$

$$\underbrace{\bar{T}_k^{FR} - \sum_{f_i \in \mathcal{S}_h} \frac{\bar{T}_k^{FR}}{\bar{T}_i^{FR}}}_{\substack{\text{number of slot-cycles} \\ \text{used by the frames in } \mathcal{S}_i \\ \text{over } \bar{T}_k^{FR} \text{ consecutive slot-cycles}}} > 0 \tag{9}$$

Since the algorithm considers the frames by increasing periods, one has $\forall f_i \in \mathcal{S}_h$, $\bar{T}_i^{FR} \leq \bar{T}_k^{FR}$, thus $\frac{\bar{T}_k^{FR}}{\bar{T}_i^{FR}} \geq 1$ and we know by the "power of two constraint" that $\frac{\bar{T}_k^{FR}}{\bar{T}_i^{FR}} \in \mathbb{N}$. Equation 9 becomes $\bar{T}_k^{FR} - \sum_{f_i \in \mathcal{S}_h} \frac{\bar{T}_k^{FR}}{\bar{T}_i^{FR}} \geq 1$. There is thus at least one free slot-cycle in the $\bar{T}_k^{FR}$ first slot-cycles in the slot h.

Step 2:

Let O_k^{FR} be the base cycle of frame f_k, we know that:

1. O_k^{FR} is set to the first available slot-cycle. O_k^{FR} is thus greater than the base cycles of all the other frames already assigned:

$$\forall f_i \in \mathcal{S}_h \text{ and } \forall j \in \mathbb{N}, \, O_k^{FR} \neq O_i^{FR} + j \cdot \bar{T}_i^{FR}. \tag{10}$$

2. Since $\bar{T}_k^{FR} \geq \bar{T}_i^{FR}$ and the "power of two constraint" then $\gcd(\bar{T}_i^{FR}, \bar{T}_k^{FR}) = \bar{T}_i^{FR}$.

From equation 3, the distance between the release r_k^{FR} of the beginning of transmission of any AUTOSAR frame f_k and the release r_i^{FR} of the beginning of transmission of any AUTOSAR frame f_i in $\mathcal{S}_h$ is:

$$r_k^{FR} - r_i^{FR} = p \cdot g + (O_k^{FR} - O_i^{FR}) \bmod g \tag{11}$$

where $g = \gcd(\bar{T}_i^{FR}, \bar{T}_k^{FR})$ and $p \in \mathbb{Z}$. From item 2, equation 11 becomes:

$$r_k^{FR} - r_i^{FR} = p \cdot \bar{T}_i^{FR} + (O_k^{FR} - O_i^{FR}) \bmod \bar{T}_i^{FR}.$$

and from equation 10, $O_k^{FR} - O_i^{FR} \neq j \cdot \bar{T}_i^{FR}$, thus:

$$r_k^{FR} - r_i^{FR} \neq 0$$

There is thus no other AUTOSAR frame released in the same slot-cycle, and the schedule is collision-free.

■

Now, it remains to be shown that the naive strategy comes up with a schedule that uses exactly the minimum number of slots required as computed by test 1 in paragraph 3.1. The first observation is that no oversampling is required (*i.e.*, all frames are transmitted at their natural period) since whatever the slot, and whatever the base cycle, the signal will respect its freshness constraint. The second observation is that each slot is fully used (if there are signals left) and thus no place is wasted. From observation 1 and 2, one concludes that the naive strategy is optimal under the aforementioned assumptions in the case where $D = T$.