Reliability-Aware Frame Packing for the Static Segment of FlexRay

Bogdan Tanasa, Unmesh D. Bordoloi, Petru Eles, and Zebo Peng
Department of Computer and Information Science,
Linköping University Sweden
{bogdan.tanasa, unmesh.bordoloi, petru.eles, zebo.peng}@liu.se

ABSTRACT

FlexRay is gaining wide acceptance as the next generation bus protocol for automotive networks. This has led to tremendous research interest in techniques for scheduling signals, which are generated by real-time applications, on the FlexRay bus. Signals are first packed together into frames at the application-level and the frames are then transmitted over the bus. To ensure reliability of frames in the presence of faults, frames must be retransmitted over the bus but this comes at the cost of higher bandwidth utilization. To address this issue, in this paper, we propose a novel frame packing method for FlexRay bus. Our method computes the required number of retransmissions of frames that ensures the specified reliability goal. The proposed frame packing method also ensures that none of the signals violates its deadline and that the desired reliability goal for guaranteeing fault-tolerance is met at the minimum bandwidth cost. Extensive experiments on synthetic as well as a industrial case study demonstrate the benefits of our method.

Categories and Subject Descriptors

C.4 [**PERFORMANCE OF SYSTEMS**]: Fault tolerance

General Terms

Algorithms, Design, Reliability

Keywords

FlexRay, Frame packing, Reliability, Scheduling

1. INTRODUCTION

Modern day automotive vehicles are equipped with high end electronic functionalities. Such advanced functionalities are supported by an in-vehicle electronic network which is, in essence, a complex distributed embedded system with tens of Electronic Control Units (ECUs). The ECUs communicate with each other by exchanging signals over a field bus which is governed by an arbitration protocol. Traditionally, such protocols in the automotive domain have followed either a time-triggered or an event-triggered paradigm. Of late, however, hybrid protocols like FlexRay [5] and FTT-CAN [4] have become immensely popular as they combine the advantages of both time-triggered and event-triggered protocols. FlexRay, in particular, has garnered widespread support because it has been developed by a consortium of major industrial players like General Motors, BMW and Audi.

EMSOFT'11, October 9–14, 2011, Taipei, Taiwan.

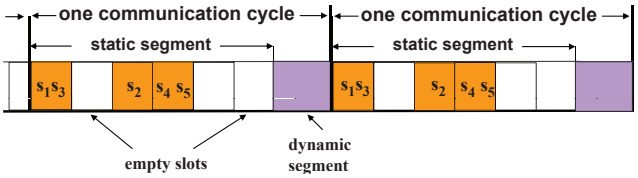


Figure 1: Illustration of the FlexRay communication cycle. Multiple signals are packed together into frames that are then transmitted on the slots of the ST segment.

The popularity of FlexRay has sparked tremendous interest in techniques for scheduling signals on the FlexRay bus [3, 11, 7]. Signals are, essentially, elementary units of communication data that need to be transmitted from one ECU to another. In practice, signals are first packed together into frames at the application-level and the frames are then transmitted over the bus. The frames must be scheduled such that the hard real-time deadlines, as demanded by automotive applications, are satisfied. However, apart from real-time issues, we note that frames on the bus may become corrupt due to transient faults, thereby posing reliability issues [15, 8]. Electronic devices, including communication buses, are becoming increasingly vulnerable to transient faults. Transient faults occur for a short duration of time and cause a bit flip, without causing any permanent damage to the logic. They are caused by factors like electromagnetic radiation and temperature variations. In contrast to permanent faults (e.g., those caused by physical damage) which cause long term malfunctioning, transient faults occur much more frequently [2].

In spite of such reliability concerns, existing frame packing techniques have assumed a fault-free transmission of frames over the bus. In this paper, we propose a technique for frame packing for the FlexRay bus that guarantees to achieve reliability against transient faults while satisfying the timing constraints. To achieve fault-tolerance our technique relies on temporal redundancy, i.e., our proposed scheme relies on frame retransmissions. However, this increases the bandwidth utilization cost. Our proposed scheme is constructed to minimize the bandwidth utilization required to guarantee the fault-tolerance of frames transmitted on the FlexRay bus. In the rest of this section, we highlight our contributions in light of (i) the problem being addressed and (ii) related research work in this domain.

1.1 Overview of the Problem

Communication on the FlexRay bus is performed over a set of periodic cycles, where each cycle is divided into the time-triggered

static (ST) segment and the event-triggered dynamic (DYN) segment. In this paper, we focus on the ST segment of the FlexRay bus. The ST segment is divided into a set of equal length slots. Each slot is assigned to a particular ECU that is allowed to transmit a frame during that slot. In practice, each ECU transmits a set of signals which are packed into a frame to be transmitted during its assigned slot. Figure 1 illustrates two communication cycles of FlexRay. In Figure 1, the FlexRay ST segment has six slots that transmit three frames. Frames $s_1 s_3$ and $s_4 s_5$ consist of two signals while s_2 is comprised of one signal.

As mentioned above, frames might become corrupt due to faults on the bus. In order to achieve reliability in presence of such faults, it is imperative to design a scheme to tolerate faults. In this work, we propose a novel scheme towards this that relies on frame retransmissions. Our method consists of the following components — (i) packing signals into frames, (ii) computing the number of times each frame must be retransmitted in order to guarantee the desired reliability level and (iii) scheduling the frames, i.e., assigning each frame to a slot in the ST segment such that the deadlines are met. Our computation for the required number of retransmissions is based on a probability analysis that connects the probability of failure of each signal to the overall reliability goal. Retransmission of frames increases the load on the bus bandwidth and, hence, our proposed method optimizes the bandwidth utilization by minimizing the number of required retransmissions.

An instinctive approach to solve the above frame packing problem is to de-couple the various components mentioned in the above paragraph, i.e., pack the signals into frames and, then, compute the required number of retransmissions for reliability. However, such a method (henceforth referred to as the *three-step* algorithm) could be out-performed (with respect to bandwidth utilization) by methods that tightly couple the three components together. This will be illustrated in Section 3 with an example. In this paper, we propose a heuristic (see Section 6) where all the components are tightly coupled together, i.e., our frame packing heuristic is aware of the reliability and bandwidth optimization goals. In the rest of the paper, we call this proposed heuristic as Reliability Aware Frame Packing (RAFP) algorithm. A wide range of experiments (see Section 7) conducted by us show that RAFP outperforms the *three-step* algorithm by a significant margin.

Note that the frame packing problem, without considering reliability issues, is already an NP-hard problem [9]. Thus, any approach to solve the problem addressed in this paper optimally, could not be expected to run in polynomial time. However, to evaluate the quality of our proposed heuristic (RAFP), we also formulate a Constraint Logic Programming (CLP) [1] problem (see Section 5). This allows us to solve the problem optimally by invoking available CLP tools. The CLP based approach exhaustively searches the complete design space using branch and bound and, hence, suffers from scalability issues. Our experiments (see Section 7) show that results returned by our heuristic (RAFP) are close to the optimal results returned by the CLP-based approach.

1.2 Related Work

The frame packing problem has been studied in the literature in various contexts. In [13], several heuristics were presented to pack frames that are exchanged over a CAN network so as to minimize the bandwidth consumption. The frame packing problem also has been addressed in [10] in the context of distributed systems consisting of both time-triggered and event-triggered networks interconnected via gateways. More recently, the problem of packing signal into frames for FlexRay networks has also been studied in [14, 9]. However, all the above lines of work were oblivious to the issue of fault-tolerance and reliability. In contrast, we focus on the problem of frame packing for the ST segment of FlexRay while providing guarantees regarding reliability against faults.

It should be mentioned here that there have been other attempts [8, 15] that have proposed fault-tolerant message passing schemes on FlexRay. However, our paper differs significantly from them. This is because the respective schemes for fault-tolerance both in [8] and in [15] assume that frames are already packed. This implies that the techniques proposed in [8, 15] cannot be directly applied to design scenarios that start from scratch, i.e., when signals are the only known units of communication. On the other hand, if one assumes packing of frames into signals has been already performed, then, as illustrated in Section 3, the approaches in [8, 15] might lead to poor bandwidth utilization. In contrast, in this paper, we assume that signals (the elementary communication units) are the known inputs and the proposed technique handles both frame packing and frame scheduling for reliable transmission over FlexRay.

2. SYSTEM MODEL

Before we delve into details of the problem in successive sections, in this section, we present the system model that we consider. Our system model comprises of (i) signals, (ii) frames, (iii) the FlexRay bus and (iv) the fault model.

Signals: Our system is a distributed automotive architecture consisting of a set of ECUs $\{E_1, E_2, \cdots, E_N\}$ where each ECU E_i generates a set of signals $S_i = \{s_1^i, s_2^i, \cdots, s_{N_i}^i\}$. Each signal is characterized by the following parameters.

- **Period (T_j^i):** denotes the rate at which signal s_j^i is produced by the node E_i.
- **Offset (O_j^i):** is the time after which the first instance of signal s_j^i is produced. The offset is expressed with regard to the start of the first FlexRay communication cycle, when $t = 0$. Thus, the subsequent instances of signal s_j^i are produced at $O_j^i + u \times T_j^i$, where u is a positive integer.
- **Deadline (D_j^i):** is the latest time instant, relative to the instant when the signal s_j^i is produced, by which the transmission of s_j^i must be completed. We assume that no signal instance can be overwritten in a transmission buffer if it has not been transmitted. This implies that each instance of a frame must be transmitted before the next instance is ready, i.e., $D_j^i \leq T_j^i$.
- **Length (W_j^i):** denotes the size of the signal s_j^i in bits.

Frames: A frame f_v^i consists of a set of signals S_v^i, where $S_v^i \subseteq S_i$, produced by the ECU E_i. Its characteristics, period T_v^i, offset O_v^i, deadline D_v^i and length W_v^i, are defined as follows:

$$\begin{cases} T_v^i = \min_{u=1}^{|S_v^i|}\{T_u^i\} \\ O_v^i = \min_{u=1}^{|S_v^i|}\{O_u^i \mid T_u^i = T_v^i\} \\ D_v^i = \min_{u=1}^{|S_v^i|}\{D_u^i - \Delta_{s_u}\} \\ W_v^i = \sum_{u=1}^{|S_v^i|} W_u^i \end{cases} \quad (1)$$

In the above, Δ_{s_u} represents the largest possible duration between the production time of signal s_u^i with period $T_u^i \geq T_v^i$ and the transmission of frame f_v^i with period T_v that contains the signal s_u^i. Δ_{s_u} is computed as follows.

$$\Delta_{s_u} = T_v^i - \gcd(T_v^i, T_u^i) \quad (2)$$

Here, gcd is the greatest common divisor of two positive integers.

In the following, we will explain the above formulas concisely. For a more detailed discussion, we refer the reader to [12]. While packing a group of signals into one frame, the period of the resulting frame will be the minimum period among the initial periods of signals. In other words, the transmission of the remaining signals has to be synchronized with the signal having the smallest period. The offset of the frame is equal to the offset of the signal which has the minimum period. For the case when several signals have the same minimum period, the offset of the frame will be the same as that of the signal having the smallest offset. The length of the frame is equal to the sum of the lengths of the individual signals.

The computation of the deadline of the resulting frame is more involved, as expressed by the formulas above. In general, if signals are packed together, the deadline of the resulting frame is smaller compared to the deadlines of the constituent signals. On the other hand, if all signals have their periods as multiples of the minimum period, then the deadline of the frame is the minimum deadline amongst all constituent signals. This is encapsulated by the above formula to compute the deadline. Note that, it is possible that the deadline of the resulting frame f_v^i has a negative value and this implies that no matter how the frame is scheduled, the constituent signals will still miss their deadlines. This situation may arise when for a signal s_u^i, we have $D_u^i < \Delta_{s_u}$. In such cases, we say that the signal s_u^i **cannot** be packed with frame f_v^i. For more details and derivation of the formulas, please refer to [12].

FlexRay bus: In this paper, we assume that the frames will be transmitted on the ST segment of FlexRay. Let us consider that the length of the FlexRay communication cycle is FC and the length of the static segment is ST. The static segment is partitioned into a fixed number of equal length slots. Let us denote the length of each slot as SD. Each node E_i is allowed to send a frame only during a slot that is allocated to that particular node and this allocation is determined statically. If a frame is not ready when the slot allocated to the respective ECU is scheduled to start, the slot will go empty, i.e., no other ECU is allowed to use it. The least common multiple of the periods of the frames and the FlexRay communication cycle is denoted as the hyperperiod H.

Fault Model: The automotive industry currently refers to the international standard (IEC61508) for functional safety of electronic safety-related systems. The standard identifies various levels of integrity or system reliability. For each level, the standard constrains the permissible probability of system level failure in a **time unit**, τ, which is typically one hour. Following this, we assume that the maximum probability of a system failure due to faults on the FlexRay communication bus in a time unit, τ, is constrained by γ. Given γ, we define $\rho = 1 - \gamma$ as the **reliability goal** which represents the quantified performance level with respect to transient faults which has to be met by the FlexRay communication sub-system. We assume that the probabilities of failure p_j^i for each frame f_j^i is known to us. For instance, if the Bit Error Rate (BER) of the FlexRay bus is known, p_j^i may be computed as, $p_j^i = 1 - (1 - BER)^{W_j^i}$, where W_j^i is the size of the frame f_j^i.

3. MOTIVATIONAL EXAMPLE

As mentioned in Section 1, one approach to construct fault-tolerant schedules for FlexRay is to decouple the frame packing and the reliability computation and the scheduling components of the overall algorithm (the *three-step* approach). In this section, we will illustrate, with the help of an example, that such techniques may lead to poor bandwidth utilization. Towards this, let us consider an example where the FlexRay communication cycle $FC = 4$ ms, the slot length $SD = 512$ bits and number of slots in one FlexRay cycle is $NS = 80$. Let the Bit Error Rate be $BER = 10^{-2}$ and the reliability goal be defined as $\rho = 0.80$ over a time unit $\tau = 32$ ms. We consider that there are 6 signals from one ECU to be packed. The characteristics of the 6 signals are defined in Table 1. We would like to mention here that all the results discussed below, for this example, were found using a CLP-based framework (see Section 5) that we have implemented which will provide the optimal solution.

	Offset (ms)	Period (ms)	Deadline (ms)	Length (bits)
s_1	1	8	8	20
s_2	1	8	8	15
s_3	2	4	4	20
s_4	1	12	12	25
s_5	2	12	12	20
s_6	1	16	16	14

Table 1: Signal Parameters

In the first step of the *three-step* approach, the signals are packed into frames with bandwidth minimization as optimization goal, without any reliability concerns. For the example under consideration, the optimal packing is to pack all the 6 signals into one frame. In the second step, we compute the number of times this frame must be retransmitted in order to meet the reliability goal ρ. The number of retransmissions for the frame turns out to be 9 and, thus, a total number of 10 slots will be used in each FlexRay cycle (the initial transmission plus the retransmissions).

Above, we computed bandwidth utilization following the *three-step* approach. However, let us now consider an optimal reliability aware packing that integrates all the three components together. This results in two frames f_1 and f_2, where f_1 consists of signals s_1, s_2 and s_3, and f_2 contains the rest of the signals. Note that, compared to the frame packing in the *three-step* approach that results in only one frame, this packing now results in two frames. However, when we compute the number of retransmissions for f_1 and f_2, we obtain 3 and 4, respectively. Thus, the total number of occupied slots while considering retransmissions is now 9 (considering initial transmissions plus the retransmissions of f_1 (1+3) and f_2 (1+4)). Thus, the reliability aware frame packing has saved one slot (9 versus 10) compared to the *three-step* method. This highlights the limitations of the *three-step* algorithm.

Computing the appropriate set of signals to be packed relies on a complex interplay of periods and lengths that influence the overall failure probability as well as deadlines that influence the schedulability. The frame packing step in the *three-step* algorithm ignores such details. Hence, all signal characteristics must be carefully encapsulated into any frame packing algorithm to be effective. These details will be formally discussed in Section 5.

4. PROBLEM STATEMENT

Our problem statement is formulated as follows. Given the system model described in Section 2, for each ECU E_i, (i) construct a set of frames $F_i = \{f_1^i, f_2^i, \cdots, f_{M_i}^i\}, M_i \leq N_i$ from a given set of signals $S_i = \{s_1^i, s_2^i, \cdots, s_{N_i}^i\}$, (ii) compute the required number of retransmissions k_j^i for each frame, and, (iii) assign slots to each frame, such that:

- the resulting frames and their retransmissions are schedulable, i.e., slots may be assigned to all frames such that the

deadline of frames and thus, the deadlines of the constituent signals are satisfied,

- the reliability goal ρ is achieved,
- the total bandwidth consumption is minimized.

In the next section, we will present a Constraint Logic Programming (CLP) formulation which allows us to solve the problem optimally using CLP solvers. However, due to the intractability of the problem, the CLP solver does not scale beyond small sized problems and hence, in Section 6 we will also present an efficient heuristic to solve the problem.

5. CLP-BASED OPTIMAL APPROACH

In this section, we shall describe the constraints of our CLP formulation for the problem that was formally stated in Section 4.

5.1 CLP Formulation

We will present the constraints related to the reliability analysis (Equations 7, 9, 10), frame packing (Equation 12), FlexRay protocol (Equation 15) and scheduling (Equations 16, 21 and 22), respectively, in the following.

Reliability constraints: Let k_j^i denote the number of times each instance of a frame f_j^i is retransmitted. For a successful transmission of each instance of frame f_j^i, at least one of the total k_j^i + 1 transmissions of the frame must be successful, i.e., fault-free.

We recall (see Section 2) that p_j^i is the probability of failure of the jth frame of ECU E_i. Given p_j^i, the probability of one instance of a frame f_j^i to encounter faults in each of its transmissions (including the initial transmission and the following k_j^i retransmissions) is:

$$PF(f_j^i, k_j^i) = {p_j^i}^{k_j^i+1} \tag{3}$$

Following Equation 3, the probability of one instance of the frame f_j^i to have at least one transmission without faults is:

$$PS(f_j^i, k_j^i) = 1 - {p_j^i}^{k_j^i+1} \tag{4}$$

The above calculation considers only one instance of the frame f_j^i. However, as discussed in Section 2, the system reliability ρ is defined for a time unit τ. During the time interval τ, the frame f_j^i occurs with a period T_j^i for $\frac{\tau}{T_j^i}$ times. Extending Equation 4 to consider all instances of the frame f_j^i, the probability to have at least one transmission without faults for each instance over a period of time τ is:

$$GPS(f_j^i, k_j^i) = \left(1 - {p_j^i}^{k_j^i+1}\right)^{\frac{\tau}{T_j^i}} \tag{5}$$

We will call $GPS(f_j^i, k_j^i)$ as the global success probability of frame f_j^i. Finally, considering all frames and all instances of them within τ, Equation 5 can be extended to obtain the global success probability GP of all frames:

$$GP = \prod_{i=1}^{N}\left[\prod_{j=1}^{M_i}\left(1 - {p_j^i}^{k_j^i+1}\right)^{\frac{\tau}{T_i^j}}\right] \tag{6}$$

In the above equation, N represents the total number of ECUs, M_i represents the total number of frames to be sent on the bus by the ECU E_i. In order to satisfy the reliability goal, we have the following constraint.

$$GP \geq \rho \tag{7}$$

In the following, we will bound the upper and lower limits on the number of required retransmissions. CLP solvers could compute solutions without such bounds as well, however, we provide these bounds to constrain the search space for the CLP. Towards this, note that, in order for Equation 7 to hold true each k_j^i individually must be greater then a value $(k_j^i)^L$, where $(k_j^i)^L$ is the minimum value that satisfies the condition:

$$\left(1 - {p_j^i}^{k_j^i+1}\right)^{\frac{\tau}{T_i^j}} > \rho \tag{8}$$

The above result follows directly from the fact the probability values are all fractional numbers. Following this, we have the constraint for lower bounds of k_j^i as below:

$$k_j^i \geq (k_j^i)^L \tag{9}$$

For the upper bound, we note that the total number of used slots (where each slot is occupied by one frame) cannot exceed the number of existing slots NS. This gives us the following constraint.

$$\sum_{i=1}^{N}\left(\sum_{j=1}^{M_i}(1 + k_j^i)\right) \leq NS \tag{10}$$

Packing: For each ECU E_i we introduce the boolean variables $x^i(u, v)$ to denote whether signal s_u^i belongs to frame f_v^i or not. Thus we have the following matrix PM_i of boolean variables.

$$PM_i = \begin{pmatrix} & f_1^i & f_2^i & \cdots & f_{P_i}^i \\ s_1^i: & x^i(1,1) & x^i(1,2) & \cdots & x^i(1,P_i) \\ s_2^i: & x^i(2,1) & x^i(2,2) & \cdots & x^i(2,P_i) \\ \cdots & \cdots & \cdots & \cdots & \cdots \\ s_{N_i}^i: & x^i(N_i,1) & x^i(N_i,2) & \cdots & x^i(N_i,P_i) \end{pmatrix} \tag{11}$$

The size of the matrix PM_i is $N_i \times P_i$ where N_i represents the total number of signals produced by ECU E_i and P_i represents the maximum number of possible frames that can result after packing. Note that P_i cannot exceed N_i. A signal can be assigned to only one frame. This constraint can be formulated as follows:

$$\sum_{u=1}^{P_i} x^i(j, u) = 1, \quad \forall j = \overline{1, N_i} \tag{12}$$

A frame f_u^i is declared to be empty if:

$$col_u^i = \sum_{j=1}^{N_i} x^i(j, u) = 0 \tag{13}$$

The total number of non-empty frames for a given ECU E_i is denoted as M_i and is computed as follows:

$$M_i = \sum_{u=1}^{P_i} c_u^i, \quad \text{where: } c_u^i = \begin{cases} 0, & col_u^i = 0 \\ 1, & col_u^i > 0 \end{cases} \tag{14}$$

For a non-empty frame ($col_u^i \neq 0$) f_u^i consisting of signals $f_u^i = \{s_{a_1}^i, s_{a_2}^i, \cdots, s_{a_q}^i\}$ the parameters period T_u^i, offset O_u^i, deadline D_u^i and length W_u^i are computed based on the equations presented in Section 2.

FlexRay and Scheduling constraints: The frame lengths must not exceed the slot capacity SD:

$$W_j^i \leq SD \tag{15}$$

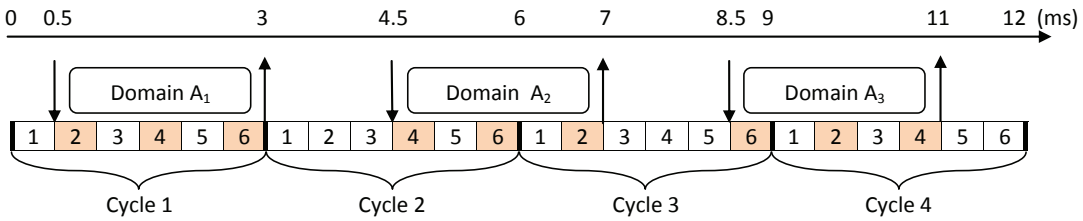


Figure 2: Feasible set of slots or *domains* for instances of frame f.

From Section 2, we know that deadlines of frames must be positive. Thus, we have the following constraint:

$$D_j^i > 0 \tag{16}$$

Apart from Equation 16, we must also ensure that all instances of a given frame f_j^i (including the k_j^i retransmissions) are schedulable. Hence, a frame f_j^i is schedulable if k_j^i+1 distinct slots can be allocated to it such that the deadline D_j^i is not violated. Towards this, we introduce the concept of "domains". Domains are the set of feasible slots for each instance of a frame. Again, we note that the CLP can find solutions without constrains on domains. However, we build the domains and provide the constraints to the CLP to limit the search space for the CLP. Before formally presenting the computation of domain, we illustrate the basic intuition behind it with an example.

Example: Let's assume that the length of the FlexRay cycle is $FC = 3$ ms and the total number of slots is $NS = 6$. Let us consider a frame f with an offset $O = 0.5$ ms, a period $T = 4$ ms and a deadline $D = 2.5$ ms. The frame f is schedulable if we can assign a slot to f before its deadline and this must be ensured for all instances of f. The total number of instances to be accounted for while finding the feasible set of slots for frame f is $NI = \frac{\text{lcm}(FC,T)}{T} = 3$. An instance $u, \forall u = \overline{1, NI}$ of frame f will be produced at moment $a(u) = O + (u-1) \times T$ and it must be transmitted on the bus before time instant $b(u) = a(u) + D$ (see Section 2). Hence, the first instance will be produced at $a(1) = 0.5$ ms and it must be transmitted to its destination before $b(1) = 3$ ms. From the characteristics of the FlexRay bus under consideration, we know that the candidate slots for sending the current instance on the bus are in the domain $A_1 = [2..6]$ (see Figure 2).

Similarly, the second instance is produced at $a(2) = 4.5$ ms and must be transmitted on the bus before $b(2) = 7$ ms. Hence, the domain of feasible slots for this instance is $A_2 = [1..2] \bigcup [4..6]$. The third instance is produced at $a(3) = 8.5$ ms and must be transmitted before $b(3) = 11$ ms and hence its domain $A_3 = [1..4] \bigcup [6]$. According to FlexRay protocol, all instances of the frame f must use the same slot and, hence, the set of feasible slots that can be allocated to f is computed from the intersection of sets A_1, A_2 and A_3. Thus, we get $A = A_1 \bigcap A_2 \bigcap A_3 = \{2, 4, 6\}$ as the feasible set of slots for frame f as shown by the shaded slots in Figure 2. □

In the following we will formally show how the domains (feasible set of slots of a given frame f_j^i) is computed in the general case. For any given instance u of frame f_j^i, the cycle in which this instance is generated - sc_u and the cycle in which the deadline of the current instance is going to expire - ec_u can be computed as shown below (FC is the length of the FlexRay cycle).

$$sc_u = \left\lfloor \frac{a(u)}{FC} \right\rfloor + 1 \text{ and } ec_u = \left\lfloor \frac{b(u)}{FC} \right\rfloor + 1 \tag{17}$$

For any given instance u of frame f_j^i the first slot - fs_u from the cycle sc_u that can be used, is computed as below.

$$fs_u = \left\lfloor \frac{a(u) - FC \times (sc_u - 1)}{SD} \right\rfloor + 1 \tag{18}$$

At the same time the last slot - ls_u from the cycle ec_u which can be used by the u^{th} instance of frame f_j^i, is computed as follows.

$$ls_u = \left\lfloor \frac{b(u) - FC \times (ec_u - 1)}{SD} \right\rfloor - 1 \tag{19}$$

Based on the calculated fs_u and ls_u, the domain $A_{j,u}^i$ of the u^{th} instance of frame f_j^i is computed as follows (NS is the number of slots in the FlexRay ST segment).

$$A_{j,u}^i = \begin{cases} [fs_u..ls_u] & \text{if } fs_u \leq ls_u \text{ and } ec_u = sc_u \\ [1..ls_u] \bigcup [fs_u..NS] & \text{if } fs_u > ls_u \text{ and } ec_u = sc_u + 1 \\ [1..NS] & \text{if } fs_u > ls_u \text{ and } ec_u \geq sc_u + 2 \end{cases} \tag{20}$$

Having the domains $A_{j,u}^i$ for each instance of frame f the domain of frame f_j^i is computed as the intersection $A_j^i = \bigcap_u A_{j,u}^i$. The first set of conditions for schedulability is that there must be enough slots for each frame f_j^i and its k_j^i retransmissions. This condition is is given as follows for each frame.

$$|A_j^i| \geq (k_j^i + 1) \tag{21}$$

The second set of scheduling constraints refers to the fact that no two frames can share the same slot. This condition can be formulated as follows by introducing the variables $s_{j,l}^i$ for slots (i denotes the ECU which produces the frame f_j^i, j represents the index of the frame and the l index identifies the retransmission $l = \overline{0, k_j^i}$ of the frame):

$$\begin{cases} s_{j,l}^i \in A_j^i, \forall l = \overline{0, k_j^i} \\ s_{b,c}^a \neq s_{e,f}^d, \forall a \neq d, \forall b \neq e, \forall c \neq f \end{cases} \tag{22}$$

Optimization objective: The optimization objective is to minimize the number of used slots in the FlexRay cycle. The number of used slots is same as the total number of required transmissions since each frame occupies one slot.

$$\text{minimize: } \sum_{i=1}^{N} \left[\sum_{j=1}^{M_i} \left(1 + k_j^i\right) \right] \tag{23}$$

6. THE PROPOSED HEURISTIC (RAFP)

The CLP formulation described in the section above will return optimal solutions but is computationally intensive and cannot scale to large designs. Hence, in this section, we propose an efficient heuristic for the optimization problem. We refer to this as the Reliability Aware Frame Packing (RAFP) algorithm in this paper. The pseudo-code of the heuristic is given in Algorithm 1. We provide

a short outline below that is followed by a detailed description of each step of the heuristic.

The heuristic starts by assigning each signal as a separate frame. Thereafter, the algorithm proceeds according to the following steps.

1 Compute the required number of retransmissions for the current set of frames based on the *reliability analysis* described in Section 6.1.

2 For each ECU E_i, choose the *best* pair of frames and pack them into one frame. The pairs are chosen based on a *packing metric* described in Section 6.2. We note that it is sufficient to evaluate only feasible pairs, i.e., pairs where the resulting frame has positive deadline, a length smaller than the slot capacity SD and a non-empty domain.

The previous two steps will be iterated until the bandwidth utilization (optimization objective) cannot be improved. Lines 2 to 17 in Algorithm 1) refer to the above iteration. Thereafter, the heuristic proceeds as follows:

3 Build a schedule based on the required number of retransmissions, scheduling constraints and FlexRay parameters.

4 In case step **3** fails, a deadline relaxing scheme is invoked. Signals will be selectively extracted from frames in order to increase the deadlines of frames and, thus, increase the chances of finding a schedulable solution.

When a signal is extracted from one frame two new frames are generated and the reliability analysis has to be rerun since the total number of frames has changed. Therefore, if step **4** is reached, the heuristic will iterate steps **1, 3** and **4** until a schedulable solution is found (step **2** will be skipped). Lines 18 to 23 refer to these two steps in Algorithm 1.

6.1 Step 1 - Reliability Analysis

In this section, we will discuss step **1** of our heuristic that computes the required number of retransmissions for a given set of frames. As discussed above, when step **1** is invoked for the first time, each signal is assumed to be a separate frame and in the following iterations, a set of packed frames will provided as an input to this step. The goal of this step is to compute the required number of retransmissions k_j^i for each frame. For clarity in elucidation, in this section, we drop the superscripts of the variables k_j^is for the frames. Instead, let us assume that all the frames generated by all ECUs are denoted as $\{k_1, k_2, \ldots, k_L\}$, where L the total number of frames considering all ECUs, i.e., $L = \sum_{i=1}^{N} M_i$, where M_i is the set of frames from ECU_i. For the purpose of our analysis we assume that the k_i retransmissions can be of non-integral value. At the end of the analysis, when the frame packing has been completed, our heuristic computes the ceilings of k_is in order to give us the practically viable values k_is.

The number of retransmissions must be such that the constraint $GP \geq \rho$ (Equation 7) is satisfied. Allowing continuous values of k_is and considering the fact that the designer's intention is to achieve the reliability goal ρ at a minimum cost, we can rewrite Equation 5 as follows.

$$\prod_{i=1}^{L} \left(1 - p_i^{k_i+1}\right)^{\frac{\tau}{T_i}} = \rho \tag{24}$$

We rewrite our objective function (Equation 23) as the following.

$$F(k_1, k_2, \cdots, k_L) = \sum_{i=1}^{L} (k_i + 1) \tag{25}$$

Algorithm 1 Reliability Aware Frame Packing Heuristic

Input: Signals of ECU_i as $\{s_1^i, s_2^i, \ldots, s_{N_i}^i\}$, with probability of failure p_i and characteristics T_j^i, O_j^i, D_j^i and W_j^i, N number of ECUs in the system, reliability goal ρ and FlexRay bus parameters FC, SD and NS

1: Initialize each signal s_j^i as a frame, enqueue frame to set F
2: compute k_i for each frame (see step **1**, Section 6.1)
3: compute B (total number of occupied slots)
4: Initialize $B' < B$
5: **while** $B' < B$ **do**
6: $B' = B$
7: **for** $i \in \{1, 2, \ldots, N\}$ **do**
8: find f_u^i and f_v^i to be packed into f_w^i (see step **2**, Section 6.2)
9: **if** $f_w^i \neq NULL$ **then**
10: $F = F \backslash \{f_u^i, f_v^i\} \bigcup f_w^i$
11: **else**
12: $i = i + 1$ (proceed to next ECU)
13: **end if**
14: **end for**
15: compute k_i for each frame (see step **1** in Section 6.1)
16: compute B (total number of occupied slots)
17: **end while**
18: compute schedules for frames in F (see step **3**, Section 6.3)
19: **while** $NOT - SCHEDULABLE$ **do**
20: relax deadline, unpack frames (see step **4**, Section 6.4)
21: compute k_i for each frame (see step **1**, Section 6.1)
22: compute schedules for frames in F (see step **3**, Section 6.3)
23: **end while**

From Equation 24 we can write the variable k_1 as a function of $k_2, k_3, \cdots, k_L$:

$$k_1 = \frac{\ln\left[1 - \frac{\rho^{\frac{T_1}{\tau}}}{\prod_{j=2}^{L}\left(1 - p_j^{k_j+1}\right)^{\frac{T_1}{T_j}}}\right]}{\ln p_1} - 1 \tag{26}$$

This allows function F (Equation 25) to be written as:

$$F = \frac{\ln\left[1 - \frac{\rho^{\frac{T_1}{\tau}}}{\prod_{j=2}^{L}\left(1 - p_j^{k_j+1}\right)^{\frac{T_1}{T_j}}}\right]}{\ln p_1} + \sum_{j=2}^{L}(k_j + 1) \tag{27}$$

In order to obtain the values of the variables k_i that minimize the function F one should solve the following set of equations (a total number of $L - 1$ equations):

$$\frac{\partial F}{\partial k_j} = 0, \quad \forall j = \overline{2, L} \tag{28}$$

The above set of $L - 1$ equations can be re-written as only one equation using a set of algebraic transformations as follows (please refer to the Appendix in Section 9 for the derivation):

$$\rho \prod_{i=1}^{L} \left(1 + \alpha_i \beta(k_L)\right)^{\frac{\tau}{T_i}} = 1 \tag{29}$$

where:

$$\alpha_i = \frac{T_i}{T_L} \frac{\ln p_L}{\ln p_i}, \quad \beta(k_L) = \frac{p_L^{k_L+1}}{1 - p_L^{k_L+1}}$$

Once $\beta(k_L)$ is computed, we can compute $k_2, k_3, \ldots, k_L$ and then, by using the Equation 26, we can compute k_1. Unfortunately, Equation (29) cannot be solved exactly using analytical methods. Therefore, we compute (please refer to the Appendix in Section 9 for the derivation) an approximate solution $\beta^*(k_L)$ as follows:

$$\beta^*(k_L) = L\frac{1-\rho^S}{\rho^S}\frac{1}{\sum_{u=1}^{L}\alpha_u}, \quad S = \frac{1}{\sum_{u=1}^{L}\frac{\tau}{T_u}} \tag{30}$$

6.2 Step 2 - Frame Packing

In this section we will explain how a pair of frames is chosen to be packed and the metric on which the decision is based. Initially, the process of packing signals into frames starts with frames containing only one signal. Once the optimization process advances, smaller frames will be packed into bigger frames until the utilization of the available slots cannot be improved. For each ECU E_i, having an existing set of frames $F^i = \{f_1^i, f_2^i, \cdots, f_{M_i}^i\}$, the heuristic finds the *best* pair of frames f_u^i and f_v^i to be packed into a frame f_{uv}^i, i.e., $f_{uv}^i = (f_u^i \circ f_v^i), u \neq v$, among all possible combinations. Note that when packing two frames into one, only $\frac{M_i \times (M_i - 1)}{2}$ combinations must be explored since the packing process is commutative. In the following we describe our packing metric based on which the best pair is chosen.

Our metric consists of **two components**. In what follows, T_{uv}^i, D_{uv}^i and W_{uv}^i corresponding to frame f_{uv}^i are computed based on the equations shown in Section 2, T_{max}^i, D_{max}^i and k_{max}^i represent the maximum values among periods, deadlines and required number of retransmissions over all the frames in the current set F^i corresponding to ECU E_i.

The **first component** of the metric represents the ratios of lengths with respect to periods.

$$\alpha_{uv}^i = \left(\frac{W_u^i}{T_u^i} + \frac{W_v^i}{T_v^i} - \frac{W_{uv}^i}{T_{uv}^i}\right) D_{max}^i T_{max}^i \tag{31}$$

In this metric, $\frac{W_{uv}^i}{T_{uv}^i}$ represents the rate at which the new frame will transmit bits and $\frac{W_u^i}{T_u^i} + \frac{W_v^i}{T_v^i}$ represents the rate at which the two frames under consideration transmit when they are two distinct frames. Ideally, we would like to see no increase in this rate even when the two frames are packed. The intuition is that frames that occur less frequently will be less often affected by transient faults and hence, they will require less number of retransmissions to achieve reliability. Hence, this metric is the difference of these two terms favoring those packings that lead to as little increase in the rate of transmission as possible. The term $D_{max}^i T_{max}^i$ is used to normalize the metric with the second metric presented below.

The **second component** of the metric represents the ratios of deadlines with respect to the required number of retransmissions as a measure of schedulability. The intuition is that frames should be packed such that the resulting deadline D_{uv}^i decreases as little as possible while the required number of retransmissions k_{uv}^i increases as little as possible. We note that the length W_{uv}^i of the resulting frame f_{uv}^i is larger then the individual lengths (W_u^i and W_v^i) of the constituent frames (f_u^i and f_v^i). Thus the required number of retransmissions k_{uv}^i will grow but our metric is constructed to minimize this increase.

$$\beta_{uv}^i = \left(\frac{D_u^i}{k_u^i} + \frac{D_v^i}{k_v^i} - \frac{D_{uv}^i}{k_{uv}^i}\right) k_{max}^i SD \tag{32}$$

The value k_{uv}^i is the required number of retransmissions for the resulting frame f_{uv}. This value cannot be computed unless we conduct the analysis described in Section 6.1 for the future set of frames containing the frame f_{uv}^i. On the other hand, we cannot invoke the reliability analysis unless we have chosen a set of frames to be packed into f_{uv}^i, which is actually the eventual outcome of the current step. Therefore, we estimate the value of k_{uv}^i, while guaranteeing that the obtained value is safe (the reliability goal ρ is achieved) as shown below.

$$\left(1 - {p_{uv}^i}^{k_{uv}^i+1}\right)^{\frac{\tau}{T_{uv}^i}} = \left(1 - {p_u^i}^{k_u^i+1}\right)^{\frac{\tau}{T_u^i}} \left(1 - {p_v^i}^{k_v^i+1}\right)^{\frac{\tau}{T_v^i}} \tag{33}$$

Based on the two components, α_{uv}^i and β_{uv}^i, we define the packing metric as:

$$M_{uv}^i = \alpha_{uv}^i - \beta_{uv}^i \tag{34}$$

The combination $f_{uv}^i = (f_u^i \circ f_v^i), u \neq v$ with the highest value of metric M_{uv}^i will be chosen as the best combination of frames for ECU E_i. After this, the process of packing frames for this ECU proceeds to the next iteration with a new set of frames $F^i \leftarrow F^i \backslash \{f_u^i, f_v^i\} \bigcup \{f_{uv}^i\}$ as input to step **1**.

6.3 Step 3 - Scheduling

The scheduling step has to assign slots to each frame f_j^j and its k_i^j retransmissions such that the deadlines of all the frames are satisfied. We will use a CLP formulation for this step based on the scheduling constraints presented in Section 5 (Equations 17 to 22). However, the CLP solver will be configured such that instead of exploring the whole solution space, it uses **limited discrepancy search - lds** heuristic [1].

6.4 Step 4 - Deadline Relaxation

In case step **3** fails to find a schedulable solution which meets the reliability goal ρ, in this step, our heuristic identifies the *critical* frames, i.e., frames that are likely to be responsible for this outcome and unpacks the most critical frame. In order to identify the critical frames we sort the frames in the increasing order of deadlines because the frames with smaller deadlines are more likely to create bottlenecks during slot assignment. Those frames having the same deadlines are sorted in the decreasing order of the required number of retransmissions k_j^i because such frames are contributing to higher bandwidth utilization. The first frame f_j^i, in the resulting sorted list, i.e., the frame having the smallest deadline D_j^i and, in case there are more than one such frame, the frame with the largest k_j^i amongst them, is declared the most critical frame. In this frame, the heuristic now identifies the critical signal. Towards this, we note that each frame consists of a set of signals, where each signal has its own initial deadline. The signal s_u^i that, if removed from the frame, increases the deadline of the frame by the maximum, is identified as the critical signal and is extracted from the frame. As a result we will have two new frames $f_{(1)}^i = f_j^i - \{s_u^i\}$ and $f_{(2)}^i = \{s_u^i\}$. In this case the reliability analysis needs to be re-run for the new set of frames $F^i = F^i \backslash \{f_j^i\} \bigcup \{f_{(1)}^i, f_{(2)}^i\}$.

7. EXPERIMENTAL RESULTS

We conducted wide range of experiments by running our proposed algorithms on synthetic test cases as well as an industrial case study. The experimental setup in described in the next section and the results are described in Section 7.2. In Section 7.3 we conduct some experiments to show the significance of the packing metrics described in Section 6.2. Finally, the industrial case study is discussed in Section 7.4.

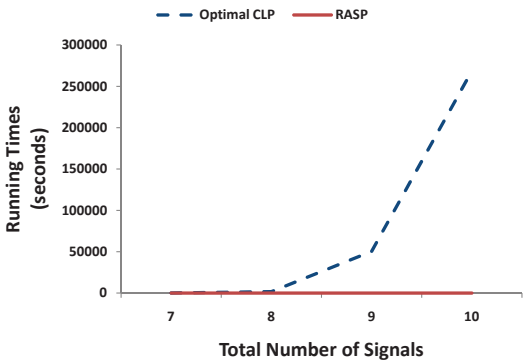


Figure 3: Running times for "small" test cases.

7.1 Experimental Setup

The framework has been implemented in Constraint Logic Programming [1] and Matlab. All the experiments were conducted on a Windows 7 machine running a 4-core Xeon(R) 2.67 GHz processor. The test cases were generated by randomly varying the signal parameters like periods and deadlines in order to cover a wide range of possible combinations. The length of the FlexRay communication cycle FC was varied between 3 ms and 10 ms following the usual design practice in the industry [9], [6]. The periods of the signals were varied between $1 \times FC$ and $10 \times FC$, the deadlines of the signals were varied between $1 \times FC$ and $7 \times FC$ while the lengths of the signals were varied between 8 and 128 bits. Note that the deadlines were generated under the assumption that $D_j^i \leq T_j^i$. We conducted two broad classes of experiments as follows.

- Small sized test cases with 7, 8, 9 and 10 signals were studied, where we considered at most 2 ECUs. For each set 20 examples were studied. In these experiments, we compared the RAFP algorithm against the CLP-based implementation.

- Two categories of large test cases were studied. First, we considered 5, 10, 15 and 20 ECUs, with each ECU producing 25 signals. Second, we considered a setup with 10 ECUs, where each ECU generated 10, 15, 20 and 25 signals. For each test case in these two categories, we experimented with 20 examples. The CLP does not scale to these large test cases. To illustrate the performance of our heuristic, we compared our RAFP algorithm with a *three-step* heuristic for these test cases. The implementation of the *three-step* heuristic will be described in Section 7.2.

For all of the above examples we assumed that the Bit Error Rate $BER = 10^{-7}$ and the reliability goal $\rho = 1 - 10^{-6}$ over a time unit $\tau = 1$ hour. We note that in practice, every frame has an overhead. For FlexRay [5], this overhead consists of a header field and a CRC field amongst others. For ease of presentation, in Section 5 and Section 6 we ignored this overhead. It is straightforward to incorporate this into our analysis and our implementation framework takes this overhead into account as well. We would like to mention that when packing two frames into one, the overhead remains constant and only the length of the payload field increases.

7.2 Results

First, we discuss the results obtained on the smaller test cases followed by the results obtained on the larger test cases.

Small Test Cases: In total, $20 \times 4 = 80$ experiments were conducted. The average running times of the conducted experiments are presented in Figure 3. As observed, the time required by the optimization problem to find the optimal solution grows exponentially with the number of signals while our heuristic ran to completion in a significantly smaller amount of time. We also compared the results of our heuristic with those of the CLP. On average, the results from the heuristic are only 15% away from the optimal solution.

Large Test Cases: Above, we compared the results of the heuristic with the optimal CLP implementation for test cases containing up to 10 signals. This is because beyond 10, the CLP did not scale at all and could not provide a solution in a reasonable amount of time (within two hours). Thus, it was not possible to compare our heuristic (RAFP) with the CLP. However, we designed a *three-step* heuristic (see Section 1.1) to compare it against RAFP for such large test cases. The *three-step* heuristic initially packs the signals into frames without reliability concerns. This frame packing problem is equivalent to the bin packing problem, where the goal is to minimize the total number of frames (used bins). Bin packing is a well-known NP-hard problem and, hence, various heuristics have been proposed for the frame packing problem [13]. In our implementation, we utilize the best fit heuristic for this step. Once the frames are packed, our *three-step* heuristic then computes the required number of retransmissions and, finally, schedules the frames by assigning a slot to each frame (see Section 6).

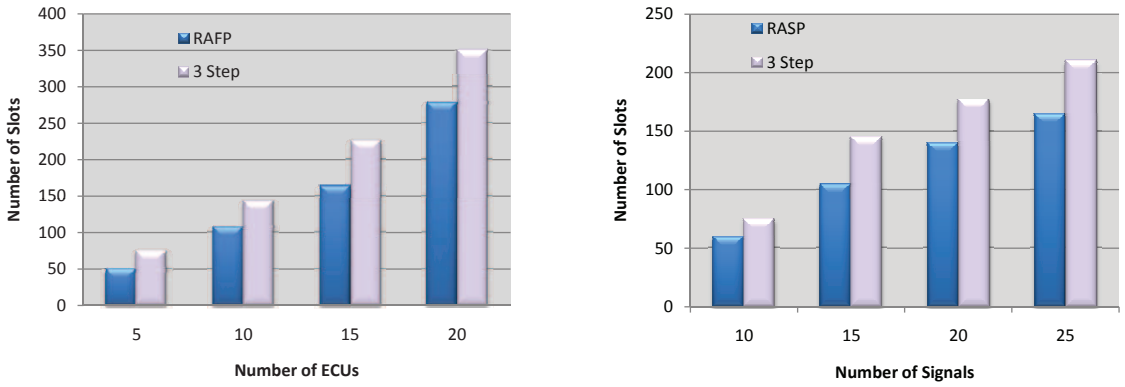
Figure 4 illustrates the number of slots required by RAFP versus the number of slots required by the *three-step* approach for these two categories of large test cases. As mentioned in the experimental setup, for each input size we considered 20 examples. Hence, each bar column in Figure 4 shows the average out of the 20 test cases. As observed in the figures, RAFP outperforms the *three-step* approach in all test cases. Moreover, as the problem size increases (i.e., with increasing number of signals or with increasing number of ECUs), the savings in slots from RAFP are even more significant. For example, if we consider Figure 4 (a) at ECU=5, RAFP outperforms *three-step* approach on the average by 25 slots but at ECU=20, RAFP is better by 75 slots. This is also reflected in Figure 4 (b) and this trend shows the significance of having a heuristic like RAFP. By considering the reliability constraints *while* packing signals into frames compared to the case when reliability is considered at the end of the packing process (as in the *three-step* approach), RAFP is able to demonstrate better performance. We would like to mention that the running times of both RAFP and the *three-step* approach were very similar.

7.3 Influence of the Metrics

As stated before, the problem of packing signals into frames must account for a complex interplay of periods, lengths and deadlines that directly influence the bandwidth utilization and schedulability of the resulting frames. We encapsulated this influence in the metrics described in Section 6.2. With the help of some experimental results, we illustrate the importance of these metrics. We considered a single ECU with 20 signals towards this and compared the quality of results (with respect to bandwidth utilization).

1. We compared $RAFP$ against $RAFP_1$ where $RAFP_1$ is same algorithm as RASP but with $M_{uv}^i = \alpha_{uv}^i$ instead of Equation 34. On average, the solutions returned by $RAFP$ were 23% better (with respect to bandwidth utilization) than those from $RAFP_1$.

2. Thereafter, we also compared $RAFP$ against $RAFP_2$ where $RAFP_2$ is same as the algorithm as $RAFP$ but with $M_{uv}^i =$



(a) (b)

Figure 4: The comparison of RAFP versus the *three-step* heuristic when (a) signals per ECU is constant at 25 but the number is ECUs is varied and in (b) number of ECUs is constant at 10 the number signals per ECU is varied.

ECU	Offset (ms)	Period (ms)	Deadline (ms)	Length (bits)	Multiplicity
ECU_1	7640	8000	8000	32	4
ECU_2	7650	8000	8000	32	4
ECU_3	105	1000	1000	32	5
ECU_3	530	1000	1000	32	1
ECU_3	530	1000	1000	16	4
ECU_3	530	1000	1000	8	2
ECU_4	120	1000	1000	32	5
ECU_4	565	1000	1000	32	1
ECU_4	565	1000	1000	16	4
ECU_4	565	1000	1000	8	2
ECU_5	160	1000	1000	32	5
ECU_5	160	1000	1000	16	2
ECU_6	200	1000	1000	32	6
ECU_6	450	1000	1000	16	2
ECU_6	450	1000	1000	8	2
ECU_7	1800	8000	8000	16	2
ECU_7	5800	8000	8000	16	10
ECU_8	1990	8000	8000	16	2
ECU_8	5990	8000	8000	16	10
ECU_9	2010	8000	8000	16	2
ECU_9	6010	8000	8000	16	10
ECU_{10}	2040	8000	8000	16	1
ECU_{10}	6040	8000	8000	16	10
ECU_{11}	4260	8000	8000	16	24
ECU_{11}	3490	8000	8000	16	7
ECU_{11}	3490	8000	8000	8	1

Table 2: Signal parameters of a x-by-wire case study.

$-\beta_{uv}^i$ instead of Equation 34. On average, the solutions returned by $RAFP$ were 30% better (with respect to bandwidth utilization) than those from $RAFP_2$.

The above results (comparing $RAFP$ with $RAFP_1$ and $RAFP_2$) illustrate the importance of our packing metrics. These metrics, combined together, contribute to the performance of RAFP. This shows that only by tightly coupling *all* parameters together with the reliability constraints one may optimize bandwidth utilization.

7.4 Case Study

We considered a X-by-wire case study with 126 signals in total and with 11 ECUs. The signal characteristics are shown in Table 2. The columns in the table, from left to right, show the ECU that produces the signal, the offset, the period, the deadline and the size of the signal. The last column shows the number of signals with those characteristics that were generated from the same ECU. The FlexRay parameters are FC = 1 ms, ST = FC, and τ = 1 hour of functionality and $\rho = 1 - 10^{-7}$. The BER value was set to 10^{-7}.

For this real-life case study, the CLP-based implementation could not find a solution even when we allowed it to run for more than two days. This highlights the scalability issue with techniques like the CLP-based implementation that rely on exhaustive search. This demonstrates the need for heuristics like RAFP proposed in this paper. For the sake of comparison with the CLP-solver, we considered a smaller case study considering only the signals from ECU_1 to ECU_4. For this case study, the CLP-solver found a solution with 22 slots but even after two days of running it could not guarantee the optimality of solution. RAFP reported a solution that occupied 28 slots in a matter of very few minutes while the *three-step* heuristic found a solution with 29 slots.

8. CONCLUSIONS

In this paper, we proposed a reliability aware frame packing heuristic (RAFP). We also presented a CLP-based framework to solve the problem optimally. We conducted experiments on synthetic test cases as well as on industrial case study. Our experimental results showed that the CLP-based approach does not scale to large test cases while our proposed scheme (RAFP) has no scalability issues. We compared the bandwidth utilization achieved by RAFP with alternative heuristics and the results demonstrate that RAFP significantly outperformed them.

9. APPENDIX

In this appendix, we provide the derivations for results that were described in Section 6.1.

Derivation of Equation 29: The partial derivatives of the cost function F are:

$$\frac{\partial F}{\partial k_j} = 1 - \frac{\ln p_j}{\ln p_1}\frac{T_1}{T_j}\frac{\rho^{\frac{T_1}{\tau}}}{\prod_{u=2}^{L}\left(1-p_u^{k_u+1}\right)^{\frac{T_1}{T_u}} - \rho^{\frac{T_1}{\tau}}}\frac{p_j^{k_j+1}}{1-p_j^{k_j+1}}, \quad j \geq 2$$

Imposing the following condition(s) in order to obtain the local extreme point(s) of function F:

$$\begin{aligned}\frac{\partial F}{\partial k_j} &= 0 \\ \Rightarrow \quad & \underbrace{\frac{\ln p_1}{T_1}\frac{\prod_{u=2}^{L}\left(1-p_u^{k_u+1}\right)^{\frac{T_1}{T_u}}-\rho^{\frac{T_1}{\tau}}}{\rho^{\frac{T_1}{\tau}}}}_{term_1} \\ = \quad & \underbrace{\frac{\ln p_j}{T_j}\frac{p_j^{k_j+1}}{1-p_j^{k_j+1}}}_{term_j}, j \geq 2\end{aligned}$$

Re-writing the previous equation(s) and identifying the common parts leads to following set of equalities:

$$\begin{aligned}\Rightarrow \quad & \underbrace{\frac{\ln p_1}{T_1}\frac{\prod_{u=2}^{L}\left(1-p_u^{k_u+1}\right)^{\frac{T_1}{T_u}}-\rho^{\frac{T_1}{\tau}}}{\rho^{\frac{T_1}{\tau}}}}_{term_1} \\ = \quad & \underbrace{\frac{\ln p_2}{T_2}\frac{p_2^{k_2+1}}{1-p_2^{k_2+1}}}_{term_2} \\ = \quad & \cdots \\ = \quad & \underbrace{\frac{\ln p_L}{T_L}\frac{p_L^{k_L+1}}{1-p_L^{k_L+1}}}_{term_L}\end{aligned}$$

Re-writing k_j as a function of k_L ($term_j = term_L, j \geq 2$):

$$\frac{\ln p_j}{T_j}\frac{p_j^{k_j+1}}{1-p_j^{k_j+1}} = \frac{\ln p_L}{T_L}\frac{p_L^{k_L+1}}{1-p_L^{k_L+1}}$$

Let us now add the following notations:

$$\alpha_j = \frac{T_j}{T_L}\frac{\ln p_L}{\ln p_j} > 0 \tag{35}$$

$$\beta(k_L) = \frac{p_L^{k_L+1}}{1-p_L^{k_L+1}} \tag{36}$$

$$1-p_j^{k_j+1} = \frac{1}{1+\alpha_j\beta(k_L)}, j \geq 2 \tag{37}$$

With the equality $term_1 = term_L$, we obtain Equation 29.

Derivation of $\beta^*(k_L)$: Assuming all parameters α are equal, Equation 29 transforms into:

$$\rho\Big(1+\alpha\beta(k_L)\Big)^{\sum_{i=1}^{L}\frac{\tau}{T_i}} = 1 \tag{38}$$

This yields $\beta(k_L)$ as follows:

$$\beta^{\alpha}(k_L) = \frac{1-\rho^S}{\rho^S}\frac{1}{\alpha} \tag{39}$$

Above, we obtain different values of $\beta(k_L)$ considering $\alpha = \alpha_1$ or $\alpha = \alpha_2$ and so on. We then compute the approximation $(\beta^*(k_L))$ as being the weighted mean of these values.

$$\beta^*(k_L) = \frac{\sum_{u=1}^{L}\alpha_u\beta^{\alpha_u}(k_L)}{\sum_{u=1}^{L}\alpha_u} \tag{40}$$

10. REFERENCES

[1] K. R. Apt and M. G. Thiran. *Constraint Logic Programming using ECL^iPS^e*. Cambridge University Press, 2007.

[2] R. C. Baumann. Radiation-induced soft errors in advanced semiconductor technologies. *Device and Materials Reliability*, 5(3):305–316, 2005.

[3] D. B. Chokshi and P. Bhaduri. Performance analysis of FlexRay-based systems using real-time calculus, revisited. In *Symposium on Applied Computing*, 2010.

[4] J. Ferreira, P. Pedreiras, L. Almeida, and J. A. Fonseca. The FTT-CAN protocol for flexibility in safety-critical systems. *Micro*, 22(4):46–55, 2002.

[5] The FlexRay Communications System Specifications, Ver. 2.1. `www.flexray.com`.

[6] M. Grenier, L. Havet, and N. Navet. Configuring the communication on FlexRay: the case of the static segment. In *European Congress Embedded Real Time Software*, 2008.

[7] A. Hagiescu, U. D. Bordoloi, S. Chakraborty, P. Sampath, P. V. V. Ganesan, and S. Ramesh. Performance analysis of FlexRay-based ECU networks. In *Design Automation Conference*, 2007.

[8] W. Li, M. D. Natale, W. Zheng, P. Giusto, A. L. Sangiovanni-Vincentelli, and S. A. Seshia. Optimizations of an application-level protocol for enhanced dependability in FlexRay. In *Design Automation and Test in Europe*, 2009.

[9] M. Lukasiewycz, M. Glaß, J. Teich, and P. Milbredt. FlexRay schedule optimization of the static segment. In *International Conference on Hardware/Software Codesign and System Synthesis*, 2009.

[10] P. Pop, P. Eles, and Z. Peng. Schedulability-driven frame packing for multicluster distributed embedded systems. *Trans. Embed. Comput. Syst.*, 4:112–140, February 2005.

[11] T. Pop, P. Pop, P. Eles, Z. Peng, and A. Andrei. Timing analysis of the FlexRay communication protocol. *Real-Time Systems*, 39(1-3):205–235, 2008.

[12] G. Quan and X. Hu. Enhanced fixed-priority scheduling with (m,k)-firm guarantee. In *Real-Time Systems Symposium*, 2000.

[13] R. Saket and N. Navet. Frame packing algorithms for automotive applications. *Journal of Embedded Computing*, 2(1):93–102, 2006.

[14] K. Schmidt and E.G. Schmidt. Message scheduling for the FlexRay protocol: The static segment. *Vehicular Technology*, 58(5):2170 –2179, June 2009.

[15] B Tanasa, U. D. Bordoloi, P. Eles, and Z. Peng. Scheduling for fault-tolerant communication on the static segment of FlexRay. In *Real-Time Systems Symposium*, 2010.