Frame Packing under real-time constraints

Ricardo Santos Marques, Nicolas Navet, Françoise Simonot-Lion

To cite this version:

Ricardo Santos Marques, Nicolas Navet, Françoise Simonot-Lion. Frame Packing under real-time constraints. 5th IFAC International Conference on Fieldbus Systems and their Applications - FeT'2003, Jul 2003, Aveiro, Portugal. pp.185-192. inria-00107676

HAL Id: inria-00107676
https://hal.inria.fr/inria-00107676

Submitted on 28 Aug 2007

FRAME PACKING UNDER REAL-TIME CONSTRAINTS

Ricardo Santos Marques - Nicolas Navet - Françoise Simonot-Lion

LORIA - TRIO
2, Avenue de la forêt de Haye
54516 Vandoeuvre-lès-Nancy
France
{rsantos,nnavet,simonot}@ensem.inpl-nancy.fr
Tel: +33.3.83.59.55.77 - Fax: +33.3.83.59.56.62

Abstract: The set of frames of an in-vehicle application must meet two constraints: it has to be feasible from a schedulability point of view and it should minimize the network bandwidth consumption. The latter point is crucial for enabling the use of low cost electronic components and for facilitating an incremental design process. This study proposes two heuristics for the NP-complete problem of generating a set of schedulable frames that minimizes the bandwidth usage. The proposed strategies are complementary. The first one can be applied to large sized problems (in the context of in-vehicle applications) while the second one, slightly more efficient in our experiments, is limited to small size problems (less than 12 signals emitted by each stations). These proposals has proved to be effective in comparison with other possible strategies.

Keywords: Embedded Systems, Vehicles, Scheduling Algorithms, Heuristics, Bandwidth-Minimization Problems.

1. INTRODUCTION

The purpose of the study is a contribution to the configuration of a middleware in the context of in-vehicle embedded systems. More precisely, it proposes a method for building off-line the set of frames when the Medium Access Protocol (MAC) is a priority bus such as CAN which is a de-facto standard for automotive communications. Application processes in the different Electronic Control Units (ECUs) send and receive data, termed signals, over the network. The middleware layer, that masks the communication system services to producer and consumer processes, performs at run-time the packing of the frames and their emission according to the configuration.

In-vehicle applications are subject to real-time constraints and most signals have a limited life-time. The freshness constraint associated to a signal is not necessarily the same for all consumers of the signal, in the following we will consider that it is the most stringent one. Knowing the set of ECUs and for each ECU the set of signals that are to be sent over the network, their size, their deadline and their production period, one must build the set of frames, the sequence of signals composing each frame as well as decide their emission period. In the case of a priority based MAC protocol, the priority of each frame has to be chosen and those priority choices will obviously have an impact on the respect of the deadlines.

In addition to schedulability, the set of frames must be constructed with the objective of minimizing the bandwidth consumption. The first reason lies on the fact that the possibility to add new functionalities in the form of one or several ECUs must be preserved and adding more ECUs implies that more signals will be exchanged on the bus. Such an incremental design is a standard procedure in the automotive industry. The second reason to minimize the bandwidth consumption is to allow the use of low power processors which are less expensive. The problem to solve is to find a configuration of frames under schedulability and bandwidth minimization constraints. This problem is NP-complete (see Norström et al. (2000)) and, as it will be shown, it cannot be solved using an exhaustive approach even for a small number of signals and/or ECUs. The solution is thus to find efficient heuristics.

A solution to the problem of frame configuration under schedulability constraints is proposed in the same applicative context by the middleware Volcano (see Casparsson et al. (1999)) but the algorithms of this commercial product are not published. Some heuristics are presented in Norström et al. (2000) to build a set of frames over CAN that minimizes the bandwidth consumption but without explicitly searching one feasible solution. Furthermore, the proposed solutions does not decide the priority of the frames. Finally, in the context of production management, several strategies were developed to place elements of different sizes inside some boxes. This problem is usually known as the 'bin-packing' problem (see Coffman et al. (1996) for a survey).

In this paper we propose two heuristics. The first one is a greedy algorithm inspired by the 'bin-packing' approaches, its complexity allows its usage on large size problems. The second heuristic searches more extensively through the solution's space than the second. It can only be applied to limited sized problems and in this case it is slightly more efficient. In this study, the performance metrics are the network bandwidth consumption and the capability to find schedulable solutions. Our proposals will be evaluated by comparison with two effective 'bin-packing' heuristics and with a naive strategy (one signal per frame).

Section 2 is devoted to a formal description of the problem, the assumptions and notations are given and the complexity of an exhaustive approach is derived. In sections 3 and 4, the proposed heuristics are described and their performances are assessed.

2. PROBLEM FORMULATION

The problem is to constitute a set of k frames $F = \{f_1, f_2, ..., f_k\}$ from a given set of n signals $S = \{s_1, s_2, ..., s_n\}$ in such a way as to minimize the bandwidth consumption while respecting the deadline constraint. The location of application processes is fixed so each signal is associated to the station that produces it. Each signal s_i is characterized by a tuple (N_i, T_i, C_i, D_i):

- the sending station denoted N_i,
- its production period T_i on this station. In practice the clocks of the stations are not synchronized but we assume that the first production of all signals located on the same ECU takes place at the same time,
- C_i is the size of s_i in bits with the assumption that C_i is always smaller or equal than the maximal possible frame size (there is no segmentation of a signal),
- D_i is the relative deadline of s_i, it is the maximum duration between the production of the signal on the sender side and its reception by all consumers. In this paper we assume that the deadline of a signal is equal to the production period but the proposed strategies are still valid for deadlines smaller than the period.

The communication network is a priority bus that might be CAN(ISO (1993)), VAN(ISO (1994)) or the J1850(SAE (1992)). The numerical results are obtained with a CAN network at 500kbits/s. Each frame contains up to 8 data bytes and we consider an overhead of 64 bits[1] .

Each frame f_i resulting from the packing process is characterized by a tuple $(N_i^*, T_i^*, C_i^*, D_i^*, P_i^*)$:

- N_i^* is the station that sends the frame,
- T_i^* is the frame emission period that is the smallest period of all the signals composing the frame,
- C_i^* is the size of the frame in bits. It is composed of data plus the constant overhead fixed to 64 bits,
- D_i^* is the deadline of the frame which depends on the signals composing the frame,
- P_i^* is the priority for the MAC protocol.

It is usual on in-vehicle networks that a part of the traffic has no real-time constraint (for instance, diagnosis frames) and such frames are assigned a priority lower than all real-time frames. It is assumed that these frames have a size of 128 bits (8 bytes of data plus the overhead) which will be the blocking factor for all real-time frames (ie. the

[1] The exact size of the CAN frame overhead is dependent of the data because of the 'bit-stuffing' mechanism. An overhead of 64 bits by frame is a reasonable value.

maximum time interval during which one frame can be delayed by a lower priority frame).

2.1 Problem complexity

The problem of building frames from signals is similar to the 'bin-packing' problem and it was proven to be NP-complete in Norström et al. (2000). Nevertheless on small size problems an exhaustive approach might be used. One thus has to determine the exact complexity of our specific problem.

To group a set of n signals in k non-empty frames comes to create all the partitions of size k from the set of signals. The complexity of this problem is known, it is the Stirling number of the second kind (see Abramowitz and Stegun (1970) page 824):

$$\frac{1}{k!}\sum_{i=0}^{k}(-1)^{(k-i)}\binom{k}{i}i^n$$

The number k of frames per station can vary from 1 to n where n is the number of signals produced by the station. The number of frames to envisage on a station i that produces n signals is thus:

$$\mathcal{S}_i=\sum_{k=1}^{n}\frac{1}{k!}\sum_{i=0}^{k}(-1)^{(k-i)}\binom{k}{i}i^n$$

For example, three signals a, b, and c on a same station leads to five different partitions : $[(a,b,c)]$, $[(a),(b,c)]$, $[(a,b),(c)]$, $[(a,c),(b)]$ and $[(a),(b),(c)]$.

If we consider a set of m stations, the solutions space becomes $\prod_{i=1..m} S_i$ where S_i is the number of possible frames for station i. The solutions space grows very quickly as soon as n and m increase. For instance, with 10 signals per station ($S_i = 115975$) and 5 stations, an exhaustive search would need to consider about $2\cdot10^{25}$ cases. Moreover, this evaluation did not take account of the determination of frame priority. One can see that for real industrial cases an exhaustive approach cannot be applied. This justifies the design of specific heuristics.

2.2 Optimal priorities allocation

Under the Fixed Priority Preemptive (FPP) policy, the response time of a task only depends on the set of higher priority tasks and not on the relative priorities between these higher priority tasks. Starting from this observation, Audsley proposed in Audsley (1991) a priority allocation algorithm that runs in $O((n^2+n)/2)$ where n is the number of tasks. This algorithm is optimal: if a solution exists then it will necessarily be found by the algorithm. The strategy is to start from the lowest priority level (m) and to search a task that is feasible at this priority level. In case of failure, one can conclude that the set of tasks is not schedulable. The first feasible task at level m is assigned to that priority. Once the priority level m is given to a task, the algorithm tries to find a feasible task at level $m-1$ and so on until priority 1 which is the highest priority of the system.

In the general case this algorithm cannot be applied to message scheduling under the Non-Preemptive Fixed Priority (NPFP) policy which is the medium access strategy for priority buses. Indeed, on a network the response time of a frame not only depends on the higher priority frames but also on the set of lower priority frames because of the blocking factor (maximal time interval during which one frame can be delayed by a lower priority frame). In our particular context the blocking factor is equal for all frames since we assume the existence of a non real-time traffic (e.g. diagnostic frames). Without any other assumptions on this traffic, one must consider the blocking factor as being the size of the largest frame compliant with the communication protocol. In this case, the conditions are fulfilled to apply the Audsley algorithm in order to evaluate the schedulability of a set of frames.

3. THE "BANDWIDTH-BEST-FIT DECREASING" HEURISTIC

The name of this heuristic has been given by analogy with the terminology used in 'bin-packing' problems with off-line resolution (*Best-Fit decreasing - First-Fit decreasing*, see Coffman et al. (1996)). In 'bin-packing' problems, the objective is to minimize the number of boxes (of frames here). In our context the goal is to minimize the network bandwidth consumption without taking account of the number of frames. The underlying idea of the heuristic is to place a signal in the frame that minimizes the bandwidth consumption induced by the signal. The heuristic builds only one solution which feasibility is assessed with the Audsley algorithm. If the solution is feasible then a local optimization procedure is applied. If it fails to find a feasible priority assignment, the heuristic applies some transformations on the frames in order to shift some load to lower priority levels.

3.1 Description

The algorithmic description of the 'Bandwidth-Best-Fit decreasing' heuristic is given below:

(1) On each station, sort all signals in decreasing order of bandwidth consumption.
(2) Insert the signal s_i in a frame:
    (a) there is at least one frame in the station that can accept s_i, it means a frame with a size including s_i smaller or equal to the protocol maximum and whose deadline is strictly positive (see Appendix A). Find which is the frame that minimizes the most the bandwidth consumption with s_i as part of the data field. Compare this bandwidth consumption augmentation with the one caused by placing s_i alone in one frame. Insert the signal s_i in the frame f_k corresponding to the best solution. If f_k corresponds to a frame that was already created, then one updates the frame timing characteristics: $T^*_k = \min(T^*_k, T_i)$ and D^*_k is set according to the algorithm described in Appendix A. If f_k corresponds to a frame whose data field is only composed by s_i then one sets the frame timing characteristics to $T^*_k = T_i$ and $D^*_k = D_i$.
    (b) there is not a single frame that can accept s_i: create a new frame with timing characteristics $T^*_k = T_i$ and $D^*_k = D_i$.
(3) As long as there is a signal not inserted in a frame, return to step 2.
(4) Feasibility test of the set of frames with the Audsley algorithm:
    (a) the configuration is not feasible:
        (i) one constructs $\hat{F}$, the set of frames for which no priority has been found.
            (A) there is one frame in $\hat{F}$ that contains at least two signals. Find the frame in $\hat{F}$ with at least two signals and for which the difference between the worst-case response time and deadline is the smallest when the frame is given the lowest priority that has not been successfully assigned. This frame is denoted $\hat{f}$.
            (B) all the frames were fully decomposed: FAILURE.
        (ii) one removes from the signals composing the frame $\hat{f}$, the one with the smallest deadline and one places it in a new frame. Update $\hat{f}$'s timing characteristics (period, deadline and size).
        (iii) return to step 4 (the algorithm will stop when all non-feasible frames have been completely decomposed).
    (b) the configuration is feasible: SUCCESS

This heuristic is composed of distinct parts. The first part (steps 1 to 3) aims at constructing a solution that minimizes the bandwidth consumption. These steps are inspired from the very efficient 'Best-Fit decreasing' heuristic from the 'bin-packing' domain. However, the choice criterion is not the size of the objects but the bandwidth consumption. The second part (step 4) deals with the feasibility of the proposed solution. Should the Audsley test fail, then the heuristic tries to reduce the deadline constraints of the frames by isolating the most demanding signals. The frame initially chosen for the decomposition is the one that exceeds the least its deadline at the lowest priority level that has not been successfully assigned. It is thus the frame that is the more likely to respect its deadline if it should contain less data. To determine which frame to decompose, it is necessary to compute a response time for each frame for which a priority has not been found (one gives the considered frame the first priority not assigned by the Audsley algorithm, the higher priority frames stay unchanged and the lower priority frames have no influence).

The step 1 of the heuristic is a sorting process, its complexity can thus be reduced to $n\log(n)$ (where n is the number of signals in the station). Steps 2 and 3 require in the worst case $O((n^2 + n)/2)$ steps. During step 4, there is at most n response time computations (no priority level was already assigned) and n calls to the Audsley algorithm. In our experiments made on industrial-scale problems (one hundred signals distributed over ten ECUs), this complexity did not raise any computation time problem.

3.2 Performances evaluation

The performances of the BBFd heuristic are evaluated with regard to two metrics that are crucial in our context: the feasibility of the system and the network bandwidth consumption. The competing strategies are:

- 'One Signal per Frame' (1SpF): each frame contains only one signal,
- 'First-Fit decreasing' (FFd): the signals are sorted decreasingly according to their size and they are inserted in the first frame that can contain them,
- 'Best-Fit decreasing' (BFd): the signals are sorted decreasingly according to their size and they are inserted in the frame that will have the smallest space left after the insertion.

The heuristics were implemented in C++ and the feasibility of each solution is tested by the *rts* software (written by Jörn Migge, see Migge (2002)) that implements the Audsley algorithm (see paragraph 2.2) and response time computation on CAN.

3.2.1. Bandwidth consumption Only the results induced by configurations that are feasible with all strategies are taken into account since the use of non-feasible set of frames is ruled out in the context of in-vehicle applications. Figure 1 shows the average network load on 100 feasible configurations for an useful load varying from 15 to 35% (the useful or nominal load is the load brought by the data alone). Signals are randomly generated with a period ranges from 5 to 100 ms (uniform distribution - step of 5ms), a deadline equals to the period, a size between 1 and 8 bytes (uniform distribution - step of 1 byte). For each test, the number of generated signals is the average number of signals necessary to reach the desired nominal load level. The signals are randomly located on a set of 10 stations. The network transmission rate is 500 Kbits/s and the overhead is 8 bytes for one data frame that can contain up to 8 data bytes (CAN network).

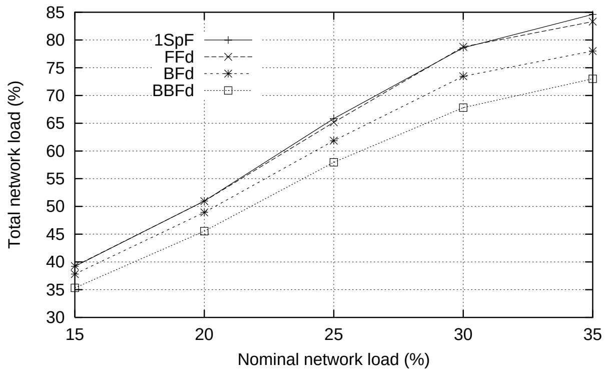


Fig. 1. Total network load for a nominal load ranging from 15 to 35%. Average results on 100 feasible configurations for the 4 competing strategies.

As it can be seen on figure 1, the relative performance of the heuristics remains identical whatever the network load. BBFd always produces the best results with a gain varying from 9.9 to 13.7% over the 1SpF approach and from 6.3 to 7.6% over BFd which is the closest heuristic. The gains in terms of bandwidth consumption do exist for all network load without being very important.

3.2.2. Configurations feasibility The conditions of the experiments are those described in paragraph 3.2.1. The results are based on 100 randomly generated tests. Table 1 shows the results obtained for the 4 competing strategies.

Up to 25% of nominal load, all the heuristics find a feasible solution. Beyond that load level, the performances strongly differ but the relative performances remains identical as for bandwidth consumption. BBFd is always the best. In particular it enables us to obtain 13 supplementary feasible configurations at the highest load level.

nominal load	15%	20%	25%	30%	35%
1SpF	100	100	100	97	36
FFd	100	100	100	97	54
BFd	100	100	100	97	76
BBFd	100	100	100	100	89

Table 1. Number of feasible configurations over 100 tests for a nominal load varying from 15 to 35%.

This suggests to us that the constraint relaxation technique used in case of non-feasibility is effective (see step 4 of the algorithm in paragraph 3.1).

3.3 Local optimization algorithm

When the BBFd heuristic returns a feasible solution, one can additionally execute a local optimization (LO) algorithm as it is classically done in optimization problems (see for instance Osogami and Okano (1999)). We propose a LO strategy that basically tries to permutes pairs of signals or to move some signals. This procedure is executed a large number of times (n) in each station ($n = 10000$ in our experiments) and the solution of the LO becomes the current best solution if it improves the bandwidth consumption and if it is feasible. The following algorithm is executed n times in each station:

(1) $\{f_a,\ f_b\}$ are two frames randomly chosen from the set of frames with more than one signal, bc_{before} is the bandwidth consumption of $\{f_a,\ f_b\}$. Choose randomly one signal from each frame: s_a from f_a and s_b from f_b and execute the following operations:
  (a) remove s_a from f_a and insert it in f_b. If the size of f_b is smaller or equal than the protocol maximum and if the deadline of both frames is strictly positive then bc_{after1} = bandwidth consumption of $\{f_a,\ f_b\}$ otherwise $bc_{after1} = \infty$.
  (b) same procedure as step (1a) with s_b instead of s_a. The corresponding bandwidth consumption is bc_{after2}.
  (c) remove s_a from f_a and insert it in f_b, remove s_b from f_b and insert it in f_a; if the size of both frames is smaller or equal than the protocol maximum and if their deadline is strictly positive then bc_{after3} = bandwidth consumption of $\{f_a,\ f_b\}$ otherwise $bc_{after3} = \infty$.
(2) if one of the bandwidth consumptions $\{bc_{after1}$, bc_{after2}, $bc_{after3}\}$ is lower than bc_{before} then update the set of frames of the station with the best solution otherwise go to step (1).
(3) Feasibility test of the new set of frames with the Audsley algorithm:
  (a) the new set of frames *is not* feasible, undo the changes operated made at step (2) and go to step (1).

(b) the new set of frames is feasible, go to step (1).

The possibility of improvement comes from the fact that, by swapping or moving signals, one can test a configuration with a signal located in a frame that did not exist, or not with the same content, at the time where the signal's insertion was decided (see step (2a) of the algorithm described in Section 3.1).

3.3.1. Performance evaluation The performance of the LO procedure is evaluated by the gain of network bandwidth consumption and by the percentage of cases where it improves the solution of BBFd.

The conditions of the experiments are those described in subsection 3.2.1. LO, with parameter $n = 10000$, is applied on 100 feasible solutions computed with BBFd. Table 2 shows the resulting network bandwidth usage.

nominal load	15%	20%	25%	30%	35%
BBFd	35.32	45.54	57.96	67.81	73.01
BBFd - with LO	35.21	45.36	57.63	67.47	72.73

Table 2. Average network bandwidth consumption over 100 feasible solutions for a nominal load varying from 15 to 35%.

Although the improvements are modest, BBFd with LO is clearly always better than BBFd with a gain varying between 0.3% and 0.5%. Table 3 shows the percentage of cases where LO brought a gain against BBFd alone.

nominal load	15%	20%	25%	30%	35%
%	56	64	87	94	95

Table 3. Percentage of cases where BBFd with LO decreased the bandwidth consumption. Experiments made over 100 feasible solutions with a nominal load varying from 15 to 35%.

The improvement with LO tends logically to be more important (Table 2) and more frequent (Table 3) when the load increases since BBFd is likely to be more distant from the optimal solution.

4. THE 'SEMI-EXHAUSTIVE' HEURISTIC

The term 'semi-exhaustive' is chosen because this strategy is an exhaustive search through the solution's space where one would cut some branches that are judged not promising. This algorithm starts by the construction of the set of all possible frames. In practice, the complexity of this first step, determined in paragraph 2.1, does not allow us to deal with cases where there are more than 12 signals in one station ($S_i = 4213597$ for $n = 12$, see paragraph 2.1). With less than 12 signals per station, the generation of the set of possible frames can be done using the technique detailed in Orlov (2002).

4.1 Description

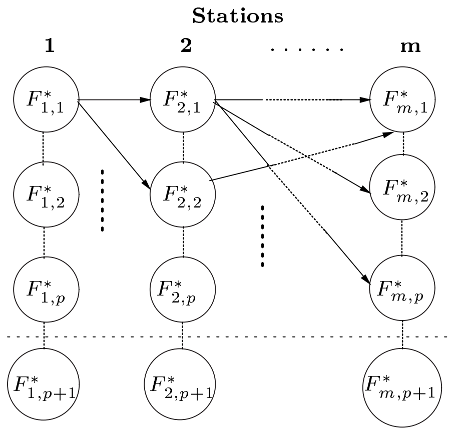
The algorithmic description of the 'semi-exhaustive' heuristic is given below:

(1) On each station i, construction of F_i^* which is the set of all possible partitions of the set of signals. A partition is 'possible' if it does not include a frame that violates the maximum data size imposed by the communication protocol or a frame having a negative deadline. The transmission period of a frame is the smallest production period of the signals it contains while the deadline of the frame is set according to Appendix A.
(2) On each station i, the set F_i^* is sorted in increasing order according to the partition bandwidth consumption (i.e. the bandwidth consumed by all the frames of a partition).
(3) $F_{i,1}^*$ is the partition that minimizes the bandwidth usage on station i. The stations are sorted in increasing order according to $F_{i,1}^*$ (e_k is the index of the k-th station in this order). For each F_i^*, one defines a depth p_i from which no further search will be performed: the partitions $F_{i,j}^*$ with $j > p_i$ won't be considered since they are the least satisfactory in terms of bandwidth consumption. The values chosen for p_i enable us to control the complexity of the heuristic. Let a_i be a temporary index variable initialized to 1 on each station i.
(4) The current set of frames is composed of the frames contained in the partitions $F_{e_1,a_{e_1}}^* \bigcup F_{e_2,a_{e_2}}^* \bigcup ... \bigcup F_{e_m,a_{e_m}}^*$ where m is the number of stations in the network.
(5) Feasibility test of the set of messages with the Audsley algorithm:
    (a) the configuration is not feasible: the next solution to test is built according to the algorithm

```
for u:=$a_{e_m}$ downto $a_{e_1}$ do
    if (u<$p_u$)
    then $a_u$:=$a_u$+1; break;
    else $a_u$:=1; fi
```

    If all the configurations were tested: FAILURE otherwise return to step 4. Figure 2 illustrates this search technique.
    (b) the configuration is feasible: SUCCESS.

As soon as the first feasible solution is found, the algorithm stops. Nevertheless it is possible to continue the search for instance until a fixed number



Feasibility Tests

$$F^*_{1,1} \bigcup F^*_{2,1} \bigcup \cdots \bigcup F^*_{m,1}$$
$$F^*_{1,1} \bigcup F^*_{2,1} \bigcup \cdots \bigcup F^*_{m,2}$$
$$\cdots$$
$$F^*_{1,1} \bigcup F^*_{2,1} \bigcup \cdots \bigcup F^*_{m,p}$$
$$F^*_{1,1} \bigcup F^*_{2,2} \bigcup \cdots \bigcup F^*_{m,1}$$
$$F^*_{1,1} \bigcup F^*_{2,2} \bigcup \cdots \bigcup F^*_{m,2}$$
$$\cdots$$
$$F^*_{1,1} \bigcup F^*_{2,2} \bigcup \cdots \bigcup F^*_{m,p}$$
$$\cdots$$
$$\cdots$$
$$F^*_{1,p} \bigcup F^*_{2,p} \bigcup \cdots \bigcup F^*_{m,p}$$

Fig. 2. Search trough the solution's space in the 'semi-exhaustive' heuristic.

of feasible solutions is found. This algorithm does at most $\prod_{i=1..m} p_i$ calls to the Audsley algorithm and the choice of the values of p_i should be done according to the available computation time. In our experiments, the values of p_i were all chosen equal to 25 but we think that it is possible to obtain better results by individualizing the values of p_i on each station according to some criterion that is to be defined. It is noteworthy that the algorithm finds the optimal solution in terms of bandwidth consumption in the particular case where the set of partitions $F^*_{e_1,1} \bigcup F^*_{e_2,1} \bigcup \ldots \bigcup F^*_{e_m,1}$ is feasible.

4.2 Performances evaluation

The experiments were conducted with the parameters described in subsection 3.2.1 except for the number of signals on each station and the number of stations which are no longer randomly chosen. Indeed, to ensure that the 'semi-exhaustive' (SE) heuristic may be applied (no more than 11 signals on each station), the number of signals is fixed to 10 per station while the number of stations varies according to the desired nominal load.

On figure 3, one observes that SE and BBFd-wlo are always significantly better than 1SpF (up to 13.46% for SE and 13.22% for BBFd with LO) which shows the effectiveness of the proposed heuristics. The performance of SE and BBFd with LO are very close one to each other (the curves are almost superposed on Figure 3) but SE remains better whatever the load. However the gain is limited since it ranges from 0.3 to 0.56%. From a schedulability point of view, SE, as well as BBFd with LO, always found a feasible solution in our experiments (made with deadline equal to period) for nominal loads lower than 35%. Beyond this load level, the computation time for SE may become problematic due to the rarefaction of feasible solutions.

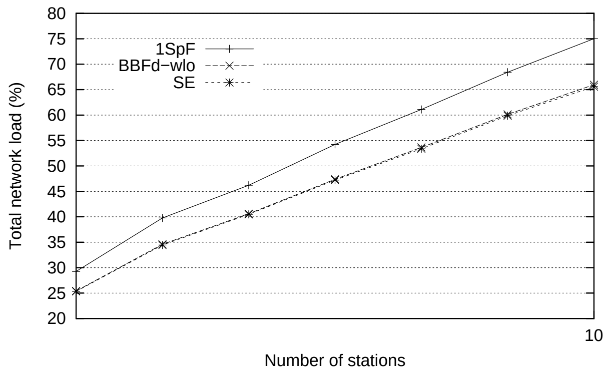


Fig. 3. Total network load for a number of stations varying from 4 to 10 with 10 signals by station. The corresponding nominal load is 11% with 4 stations, 13.75% with 5 stations, 16.5% with 6 stations, 19.25% with 7 stations, 21.5% with 8 stations, 24% with 9 stations and 27.5% with 10 stations. Average results over 100 feasible configurations with "one signal per frame" (1SpF), BBFd with LO (BBFd-wlo) and 'semi-exhaustive'(SE) heuristics.

CONCLUSION

In this study, two heuristics for building feasible sets of frames that minimize the bandwidth consumption have been proposed. The strategies are complementary, the first one (BBFd with LO) may be applied on large size problems (in the context of in-vehicle applications) while the second one (SE), slightly more efficient in the experiments, is usable only on limited size problems (less than 12 signals by ECU). These proposals have proven to behave much better than other a priori possible approaches that are 'Best-Fit decreasing', 'First-Fit decreasing' and 'One Signal per Frame'.

These heuristics can be used as a starting point for other optimization algorithms to direct the search towards promising parts of the solution's space. In particular, they might be included in the initial population of a genetic algorithm, the initial population having a strong impact on the performance of the algorithm (see for instance Westerberg and Levine (2001)).

REFERENCES

M. Abramowitz and I.A. Stegun. *Handbook of Mathematical Functions.* Dover Publications (ISBN 0-486-61272-4), 1970.

N.C. Audsley. Optimal priority assignment and feasibility of static priority tasks with arbitrary start times. Technical Report YCS164, University of York, 1991.

L. Casparsson, A. Rajnák, K. Tindell, and P. Malmberg. Volcano - a revolution in on-board communications. Technical report, Volvo Technology Report, 1999.

E.G. Coffman, M.R. Garey, and D.S. Johnson. *Approximation Algorithms for NP-Hard Problems*, chapter Approximation Algorithms for Bin Packing: a Survey. PWS Publishing Company, 1996.

ISO. ISO International Standard 11898 - Road vehicles - Interchange of digital information - Controller Area Network (CAN) for high-speed communication, 1993.

ISO. ISO International Standard 11519-3 - Road vehicles - Low-speed serial data communication - Vehicle Area Network (VAN), 1994.

J. Migge. RTS, 2002. program and manual available at `http://www.loria.fr/~nnavet`.

C. Norström, K. Sandström, and M. Ahlmark. Frame packing in real-time communication. Technical report, Mälardalen Real-Time Research Center, 2000.

M. Orlov. Efficient generation of set partitions, 2002. available at `http://www.cs.bgu.ac.il/~orlovm/papers/partitions.pdf`.

T. Osogami and H. Okano. Local search algorithms for the bin packing problem and their relationships to various construction heuristics. In *Proc. IPSJ SIGAL*, 1999. also available as IPSJ Technical Report number 99-AL-69-5.

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

SAE. SAE J1850 Class B data communication network interface, 1992.

C. H. Westerberg and J. Levine. Investigation of different seeding strategies in a genetic planner. In Springer-Verlag, editor, *Proc. EvoWorkshops2001: EvoCOP, EvoFlight, EvoIASP, EvoLearn, and EvoSTIM*, LNCS 2037, 2001.

Appendix A. DEADLINE OF A FRAME AFTER THE ADDITION OF A SIGNAL

The transmission period of a frame containing several signals is the smallest production period of the signals and the transmission of the frame will be synchronized with the production of the signal having the smallest period. On the other hand, the deadline of the frame is not the smallest deadline due to possible offsets between the production dates of the signals and the actual transmission dates of the frame. Let us for instance consider the example shown on figure A.1 with two signals s_1 and s_2 having respectively a period $T_1 = 10$ and $T_2 = 14$ and a deadline (relatively to the production date) $D_1 = 10$ and $D_2 = 14$. The signal s_2 produced at time 14 is actually sent at time 20 and the deadline of the frame must thus be equal to 8 in order to respect the timing constraint of s_2. One can also note that the transmissions made at times 10 and 40 do not include any new value for s_2. In practice, the value of s_2 might be re-transmitted or not but there will not be any timing constraint on the frame induced by s_2.

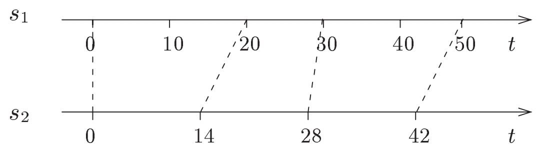


Fig. A.1. Two signals with production periods equal to 10 and 14. The signals are transmitted in a frame synchronized with the signal having the smallest period. The dotted line indicates when the signal with period 14 is actually transmitted.

To determine the deadline of the frame, one must find the largest offset between a production date and the transmission of the next frame. One wants to include signal s_i in frame f_k already containing the signals $s_1^k, s_2^k, ..s_n^k$. One denotes s_{min} the signal with the smallest period of the set $\{s_1^k, s_2^k, ..s_n^k\} \cup s_i$, the period of f_k becomes $T_k^* = T_{min}$. The relative deadline of f_k is $D_k^* = \min\{D_j - \text{Offset}(T_{min}, T_j) \,|\, s_j \in \{s_1^k, s_2^k, ..s_n^k\} \cup s_i\}$ where $\text{Offset}(a, b)$ returns the largest possible duration between the production date of a signal with period $b \geq a$ and the transmission of the frame of period a that contains the signal. It has been shown in Quan and Hu (2000) that $\forall k_1, k_2 \in \mathbb{N}\; k_1 \cdot a - k_2 \cdot b = q \cdot \gcd(a, b)$ with $q \in \mathbb{Z}$. In our context, one imposes $a > k_1 \cdot a - k_2 \cdot b \geq 0$ and thus $\text{Offset}(a, b) = (\frac{a}{gcd(a,b)} - 1) \cdot gcd(a, b) = a - gcd(a, b)$. In our example, the deadline must be set to 6.