Frame Packing in Real-Time Communication

Kristian Sandström, Christer Norström, and Magnus Ahlmark
Department of Computer Engineering
Mälardalen Real-Time Research Centre
www.mrtc.mdh.se

Abstract

A common computational model in distributed embedded systems is that the nodes exchange signals via a network. Most often a signal represents the state of some physical device and has a signal size ranging from a single bit up to a few bytes. Furthermore, each signal typically has a deadline requirement. The communication networks used are often based on a broadcast bus where fixed or variable sized frames are transmitted. The amount of data that can be transmitted in each frame is almost always bigger than the size of a signal. Thus, from a resource perspective it would be desirable if each frame could transport several signals.

In this paper we investigate how to assign signals to periodic frames with the objective function to minimise the network bandwidth requirement while not violating specified deadlines. This problem is NP-hard, but can for most typical applications be solved efficiently by using simple heuristics. The effectiveness of our algorithm is demonstrated by applying it to signal sets derived from automotive applications for a CAN based system and for the newly developed, low cost and low speed, Local Interconnect Network (LIN). The results can be of great use in cost sensitive embedded systems such as car control systems, where the used hardware, communication networks and nodes (typically micro-controllers), have to be highly utilised to keep the production cost at a minimum level.

1. Introduction

Today most modern cars are computer controlled in order to decrease the production cost (especially to reduce the amount of installed cables) and to facilitate the implementation of new functionality such as anti skid, which is very hard, or even impossible, to implement in purely mechanical systems [4].

When replacing classical solutions, such as connecting a switch directly to a device, e.g., a motor or a lamp, with a computer network based solution; the status of the sensor has to be sampled, transmitted over the network, received by the consuming node, and finally actuated to the device within an appropriate time interval. Each sensor entity sent over the network is called a signal. A common computational model in distributed embedded systems is that the nodes exchange signals via a network [5].

The timing requirements for each signal sent over the network have to be derived from the controlled process. Thus, each signal has a size and a timing requirement specification. The timing terminology used in this paper: The End To End Deadline (ETED) is the maximum delay from a stimuli until a response is given to the environment for a specific function. An ETED timing requirement for a function has to be broken down to individual timing requirements for the components that constitute the function. This is the application engineer’s task. We will use *deadline* in this context to denote the timing requirement for a signal sent on the network. More specifically the deadline specifies the maximum delay between when a signal is available at the sending node's communication subsystem until it is available for the application(s) on the receiving nodes.

Since each node most likely will send several signals, the signals should be packed in frames so that the communication bandwidth usage is minimised. Consider the case when only one signal is included in each frame (the size of a signal is considered to be less than the size of a frame) and the frame is transmitted periodically with a minimum frequency that fulfils the deadline. Then the communication cost would be high because each signal would get the burden of all overheads in a frame, such as control information and checksum. Consider the opposite situation, the signals are packed in as few frames as possible and there exists two different sizes of frames (i.e. frames that carry different amounts of data and have different transmission times). Furthermore, assume that the signals fit into one large frame. If the signals have about the same deadline, it would be beneficial to send them in one large frame. On the other hand if we have for example two groups of signals that have quite a large difference in deadlines then it would be beneficial to divide the data into two frames. Because the smallest deadline in each frame determines the period time of the frame, and thus the bandwidth utilisation would become less than packing all signals into a large frame.

To assign signals to frames is difficult since (1) the signals are asynchronous (i.e., the different signals are available for the communication subsystem at non synchronised times) and (2) many protocols for embedded systems allow different frame sizes with different transmission times, e.g., in a CAN-based system, the data is transmitted in frames containing between 0 and 8 bytes of data.

Thus, the problem investigated in this paper is: Given a finite set of signals for each node, where each signal is characterised by a deadline and a size. Further a finite number of different sized types of frames with different transmission times are given. Find a mapping of signals to periodic frames, which will minimise the bandwidth utilisation of the communication network such that all of the signals are uniquely assigned to frames and that the frames are globally schedulable.

When comparing different packing alternatives we have chosen to define a utilisation measure for a frame as the transmission time divided by the deadline of the frame, and consequently the utilisation measure for the network as the sum of utilisation measures for all frames. Note that we use deadline instead of period time in the definition of the utilisation measure, because we want to separate the frame packing from scheduling. In the scheduling phase, periods are determined based on deadlines, frame transmission time, and the scheduling method used. A straightforward solution is to transmit each frame with a period time that is equal to half of the deadline then the deadline requirement for each frame will be fulfilled, but possibilities exist [13].

Thus we have a set of frames where each frame has a period time and a transmission time, which we have to perform schedulability analysis on. Several mature techniques for schedulability analysis of periodic frames for different protocols exist, including the technique developed for the CAN-bus by Tindell et al. [[1][2][3]] and techniques for off-line generation of timetables [9].

The problem we address is similar to the task allocation and scheduling problem that has been studied by many researchers, e.g. [9][11]. The main difference is that most often in task allocation, a system with a finite set of nodes is given while in our case we have non-finite set of frames. Furthermore, the task allocation and scheduling problem is harder since tasks often have relations between each other, including mutual exclusion and precedence. Our work also relates to the work done in multimedia applications where multiple streams are to be guaranteed as in [12], where they model the problem as a multidimensional bin-packing problem. Their problem is slightly more complex since they handle different kinds of resources like, disk, CPU and network resources. However, we have not found any work that has attacked the frame-packing problem.

The contributions of this paper are that we:

- Formulate the packing problem.
- Show that the packing problem is NP-hard
- Present a simple heuristic for frame packing that we show is very effective.
- Demonstrate the effectiveness of the algorithm on realistic sized problems derived from the automotive industry.

The rest of this paper is organised as follows. Section 2 presents our system model and the formalisation of the problem. Section 3 presents the proposed algorithms, whereas in Section 4 the corresponding analysis results are presented. Finally in Section 5 we will draw some conclusion.

2. Problem statement

System model

We assume a distributed system consisting of a set of nodes interconnected via a communication network. The communication protocol is assumed to be a packet transmission protocol with a limited set of frame sizes. A frame contains one or more signals and the size of a signal is assumed to be less or equal to the size of the largest frame. Each node transmits and receives signals, where a signal has one producer and one or more consumers. Each signal has a specified size and deadline. We assume that the period time of generation of new signal values is greater than the deadline of the signal. The nodes may or may not have a global synchronised time base.

Problem formulation

For each node the following problem has to be solved. Given a finite set $S = \{s_1, s_2, ..., s_n\}$ of signals with size $sz(s_i) \in N^+$ and a deadline $d(s_i) \in N^+$. We define a frame *f* as a collection of signals from *S*. Each frame has an associated transmission time $c(f_j) \in N^+$ and a size $sz(f_j) \in N^+$, defined by the used communication protocol.

The problem is now to find a mapping of S into a set of frames $F = \{f_1, f_2, ..., f_l\}$, such that each $s_i \in S$ is included in a unique f_j, with $\sum_{\forall s_i \in f_j} sz(s_i) \leq sz(f_j)$, and which minimises the bandwidth utilisation measure $U = \sum_{\forall f_k \in F} \frac{c(f_k)}{\min_{\forall s_i \in f_k} (d(s_i))}$.

Each frame has to be transmitted with a rate that fulfils the deadline requirement on the signal with the shortest deadline in the frame f_i. The objective is to map the signals into frames such that the bandwidth requirement U is minimised, while making sure that frames are schedulable.

This problem is *NP*-hard in the strong sense since it easily can be shown that it is a special case of the well known "bin packing" problem, which is a *NP*-hard combinatorial optimisation problem [7]. The "bin packing" problem is obtained when all signals have the same deadline and when there is only one size of frames. Then our optimisation problem becomes to pack the signals in as few frames as possible, which is exactly the "bin packing" problem. So if our problem is proven to belong to class *P* then should also the "bin packing" problem belongs to that class, which is a contradiction, unless *P* = *NP*.

3. An engineering approach: mapping signals to frames

The frame-packing problem is a NP-hard problem and hence we need to solve the problem by using heuristic techniques. To get a measure of the effectiveness of our algorithms, a theoretical lower bound for the utilisation is derived for the signals. This theoretical lower bound is never higher than the real lower bound.

The lower bound is calculated by assuming that each signal is transmitted in a frame with the lowest cost per bit and the deadline of the frame is the same as the deadline of the signal. A frame has a transmission time and a data size.

We define the lowest theoretical overhead per bit by $MINOH = \min_{\forall f} (c(f)/sz(f))$. The minimal theoretical signal utilisation, *SU*, for a signal *s* is calculated by $SU(s) = \frac{sz(s)}{d(s)} \times MINOH$. The theoretical lower bound of the utilisation for all signals is calculated by $LB(S) = \sum_{\forall s \in S} SU(s)$. Intuitively, this corresponds to packing each signal in a minimum overhead frame, together with other frames with the same deadline that completely fills up the frame.

Our heuristic approach is to first sort the signals in increasing deadline order and then pack the signal into frames by a heuristic algorithm. We will consider two type cases of packing, the first packing algorithm (fixed frame size) considers only one size of the frames and exploits the first fit algorithm and the second algorithm (linear frame selection) uses heuristics for deciding which frame size to be used. A more detailed description of the algorithms can be found in [6].

Fixed frame size

The algorithm for fixed size frames assigns signals to a frame until a signal does not fit into the frame, then a new frame is created and the signal is assigned to that frame.

Linear frame selection

The algorithm starts off with a frame of the smallest frame size and assigns signals to that frame. When a signal *s* does not fit into the frame a selection is made; the cost (in bandwidth usage) for using a larger frame that fits all signals including *s* is compared with the cost of keeping the original frame and assigning *s* to a new frame with the smallest possible size. The alternative with the lowest cost is preferred. Moreover, when several frames have been created the algorithm first traverses the frames in order, trying to fit the signal into some unused space. If that is not successful the procedure described earlier is started.

A nice property of both algorithms presented is that they are polynomial time algorithms. Which in practice mean that they are very fast to run even for large signal sets.

4. Simulation

To evaluate the quality of our algorithms we will perform analysis for type-cases of signal sizes and deadline distributions, both for a Controller Area Network (CAN) [10] based system and the slow and low cost Local Interconnection Network (LIN)[8]. CAN is a broadcast bus designed to operate at speeds up to 1 Mbps. Data is transmitted in frames containing between 0 and 8 bytes of data. A LIN installation usually runs at the speed of 5-20 Kbps/s and is intended to be used for control of internal lights, window drivers, selection switches, etc. in automotive systems. Data is transmitted in frames containing 2, 4 or 8 bytes of data.

We have chosen to study these two buses because they operate on different speeds and have different sets up of possible frame sizes. Further, a CAN based system is more likely to be used for sending larger signals in terms of number of bits since it is mostly used for sending control data, while the LIN based system is mostly used for replacing simple on/off logic. The sizes and deadline distributions for each bus have been derived from discussions with our industrial partners [14].

To generate signal sets we have developed a test case generator that takes the following as input:

- The theoretical lower bound bandwidth, which is used for regulating the amount of signals to be generated.

- The distribution of signal sizes (e.g., 70% 1 bit signals 20% 2 bit signals and 10% 4 bit signals)
- The distribution of deadlines (e.g., 20% of the signals has a deadline of 10, 25% of the signals have a deadline of 25 etc.)

CAN simulation

Signals were created with a distribution of the signal size according to Table 1 and each signal was given one out o nine different deadlines. Table 1 gives also the probability for assigning a specific deadline to a signal.

Size distribution		Deadline distribution	
Size	**Probability**	**Deadline**	**Probability**
1	0.20	20	0.07
2	0.20	40	0.20
3	0.10	50	0.25
5	0.10	75	0.05
8	0.20	100	0.10
10	0.05	150	0.10
16	0.15	200	0.10
		250	0.10
		400	0.03

Table 1. Distribution of signal sizes (a) and deadlines (b) for the CAN simulation.

The graph presented in Figure 1 shows the bandwidth utilisation of the frames as a function of generated signal sets with different loads. The graph was obtained by running 10000 generated signal sets for each load level. The graphs include the result from the lfs algorithm and the fixed frame size algorithm. The fixed frame size algorithm was executed for 8 different frame sizes, however smaller CAN frames have been omitted as they result in much higher bandwidth utilisation. The network was assumed to operate at 500 Kbps.

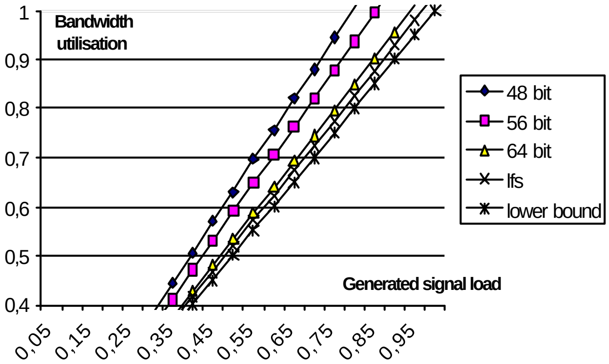

Fig. 1. The performance at different load levels.

As can be seen from the graph we are close to optimal in fact we are just some percents above the optimal and thus it seems that we have a rather good heuristic.

LIN-simulation

Signals were created with a distribution of the signal size according to Table 2 and each signal was given one out of seven different deadlines. Table 2 gives also the probability for assigning a specific deadline to a signal. The cost for the three different frame sizes was assumed to be 15, 20 and 25 respectively.

Size distribution		Deadline distribution	
Size	**Probability**	**Deadline**	**Probability**
1	0.50	50	0.05
2	0.20	75	0.10
3	0.20	100	0.20
10	0.05	150	0.20
16	0.05	200	0.20
		400	0.20
		1000	0.05

Table 2. Distribution of signal sizes (a) and deadlines (b) for the LIN simulation.

The graphs presented in Figure 2 shows the bandwidth utilisation of the frames as a function of generated signal sets with different loads. The graphs were obtained by running 10000 generated signal sets for each load level.

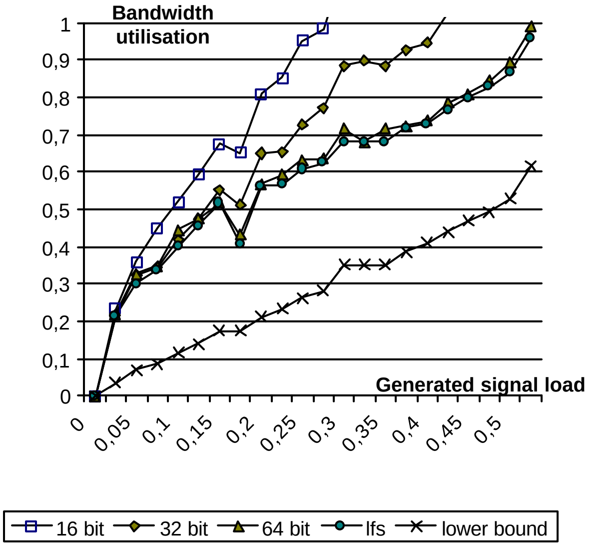

Fig. 2. The performance of the algorithms at different load levels generated

For small signal sets the 16 and 32 bit frames in the LIN simulation gives better performance than the 64 bit frame because the effect of not filling up the "last" frame is less significant. Further, compared to the CAN simulation, the LIN simulation also has a larger gap between the lower bound and the lfs algorithm since the price of not filling up the last frame is much higher (because the CAN-bus runs on a much higher speed), the CAN simulation includes significantly more signals and frames, and that only 3 frame sizes can be used in LIN.

Discussion

- The CAN simulation includes many more signals, and hence more frames, than the LIN tests and thus the price of not filling up the "last" frame is less significant.
- For small signal sets the 16 and 32 bit frames in the LIN simulation gives better performance than the 64 bit frame, because the effect of un-used space in the last frame is in average much higher.
- It is quite easy to construct "pathological" cases where for example the 64 bit fixed frame behave much worse than the lfs algorithm.
- Since all algorithms are so cheap to run one can always select the best result provided by any of the algorithms.

5. Conclusion

In this paper we have presented the frame-packing problem, made a formalisation of the problem, showed that the problem is NP-hard, presented a heuristic solution, and demonstrated the heuristics effectiveness on signal sets that have been derived from real automotive applications.

The results from this paper can be used for many different communication networks where several small signals have to share the space available in one frame.

Further research includes looking into the issue of adjusting the period times of the frames in an efficient way.

An interesting theoretical problem is to find out if it is possible to find an approximation algorithm, which can give a worst case upper bound on the waste of bandwidth for the algorithms presented in this paper.

Acknowledgements: We would like to thank Hans Hansson, Sasikumar Punnekat, Jukka Mäki-Turja, Ralf Elvsén, and Henrik Thane for valuable discussions and for reviewing earlier versions of this paper.

Mälardalen Real-Time research Centre (MRTC; www.mrtc.mdh.se) is a research centre in Västerås, Sweden, supported by Swedish industry, the Swedish Foundation for Knowledge and Competence Development (KK-stiftelsen) and Mälardalen University.

6. References

[1] Tindell K., J. Clark, Holistic Schedulability Analysis for Distributed Hard Real-time Systems. Technical Report YCS197, Real-Time Systems Research Group, Univ. of York, 1993.

[2] K. W. Tindell and A. Burns. Guaranteed message latencies for distributed safety-critical hard real-time control networks. Technical Report YCS229, Dept. of Computer Science, University ofYork, June 1994.

[3] K. W. Tindell, A. Burns and A. J. Wellings. *Calculating Controller Area Network (CAN) message response times*, Control Engineering Practice 3(8):1163-1169, 1995.

[4] K. Melin. Volvo S80: Electrical system of the future Volvo Technology Report. 98-12-11.

[5] L. Casparsson, A. Rajnak, K. Tindell, and P. Malmberg Volcano a revolution in on-board communications. Volvo Technology Report. 98-12-10.

[6] C. Norström, K. Sandström, Magnus ahlmark. Frame Packing in Real-Time Communication, MRTC Technical report, July 2000, www.mrtc.mdh.se.

[7] M. R. Garey and D. S. Johnson. Computers and Intractability: A Guide to the Theory of NP-Completeness. Freeman San Francisco, CA, ISBN 0-7167-1045-5, 1979.

[8] LIN Protocol Specification, http://www.lin-subbus.org/

[9] K. Ramamritham. Allocation and Scheduling of Complex Periodic Tasks. In 10th Int. Conf. on Distributed Computing Systems, pages 108-115, 1990.

[10] Road Vehicles - Interchange of Digital Information - Controller Area Network (CAN) for High Speed Communication, ISO DIS 11898, February 1992.

[11] J Jonsson and J Vasell. Evaluation and comparison of task allocation and scheduling methods for distributed real-time systems. In proceeding of Second IEEE International Conference on Engineering of Complex Computer Systems, 21-25 Oct. 1996.

[12] H. Jiandong and D. Ding-Zhu. Resource management for continuous multimedia database applications. In proceedings of RTSS'94. 1994.

[13] A. K. Mok. FUNDAMENTAL DESIGN PROBLEMS OF DISTRIBUTED SYSTEMS FOR THE HARD–REAL-TIME ENVIRONMENT. Ph.D. thesis MIT 1983.

[14] C. Norström, K. Sandström, M. Gustafsson, J. Mäki-Turja, and N.-E. Bånkestad. Findings from introducing state-of-the-art real-time techniques in vehicle industry. In industrial session of the 12th Euromicro Conference on Real-Time Systems, Stockholm, Sweden, 2000.