Applying End-to-end Path Delay Analysis to Multi-rate Automotive Systems Developed using Legacy Tools

Saad Mubeen and Thomas Nolte
Mälardalen Real-Time Research Centre, Mälardalen University, Västerås, Sweden
{saad.mubeen, thomas.nolte}@mdh.se

***Abstract*—The end-to-end path delay analysis is used to predict timing behavior of multi-rate automotive embedded systems. Some of the assumptions used by the existing analysis may not be strictly followed by some legacy tools due to optimizations applied during the development of these systems. As a result, the existing analysis may not be applicable in some cases. In this paper we identify one such case. That is, the case in which all the tasks in a multi-rate task chain have equal priorities despite the fact that they have different periods. Furthermore, the chain contains at least one single-rate sub-chain. We also propose a preliminary solution that makes the existing analysis applicable to this case. However, the proposed solution is pessimistic. Currently, we are working on minimizing the pessimism.**

I. Introduction

Most of the automotive embedded systems are developed as *multi-rate* systems [1]. A multi-rate system contains at least one multi-rate task chain. A multi-rate task chain consists of a connected sequence of tasks such that each task in the chain is periodically activated with an independent trigger source (clock). The activating clocks along the chain often have different periods. These types of task chains can be realized in uniprocessor as well as distributed embedded systems. These task chains find their applications in many domains. However, this paper focuses only on the automotive domain.

Often, real-time requirements are specified on multi-rate automotive systems. This means, the time at which these systems respond to some stimulus is equally important as logical correctness of the response. Many of these systems are of safety-critical nature, meaning that, missing a real-time requirement may result in the system failure which can lead to catastrophic consequences. Hence, the developer of such a system has to provide guarantees that the actions by the system are taken in a timely manner when it is executed. The end-to-end path delay analysis [1] serves as one of the methods to provide such guarantees. The analysis is pre-runtime, i.e., it can validate end-to-end timing requirements specified on the system without actually running the system and performing exhaustive testing.

A. Motivation and Contribution

The motivation for this work comes from an activity of implementing the end-to-end timing analysis in an existing tool suite, Rubus-ICE [2], that is used to develop vehicular embedded systems by several international companies including Volvo. While implementing the analysis and performing the testing, we identify that the existing end-to-end path delay analysis [1] may not be applicable in the case where all tasks in a multi-rate task chain have equal priorities despite different periods; while the chain contains at least one single-rate sub-chain. The existing analysis is not flawed at all, in fact, this is due to the constraints used in the optimization tool that is used for generating some task parameters. Consequently, the tool may not strictly follow the assumptions used by the analysis in some cases. We also discuss a preliminary solution to use the existing analysis in this case. However, the solution results in pessimistic end-to-end path delays. Minimization of the pessimism from the analysis is an ongoing work.

B. Paper layout

In Section II, we discuss the background and related work. Section III describes the end-to-end path delays. Section IV presents the problem statement. Section V discusses the preliminary solution and summary of current work.

II. Background and Related Work

The timing behavior of a single-rate real-time system can be predicted by calculating response times [3] of all tasks and comparing them with corresponding deadlines. For example, consider a single-rate real-time system consisting of only one task chain shown in Fig. 1. There are three tasks in the chain represented by τ_1, τ_2 and τ_3. The tasks have equal priorities and the Worst Case Execution Time (WCET) of each task is equal to 1 time unit. There is only one activating source for the chain that triggers τ_1 periodically with a period of 8 time units. The data read by τ_1 from register[1] Reg-1 is considered as the input of the task chain. It may correspond to the data that arrives from a sensor. Whereas, the data written by τ_3 to Reg-4 is considered as the output of the task chain. It may correspond to the control data (or signals) for an actuator.

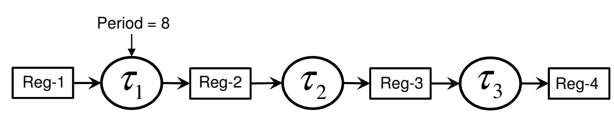

Fig. 1. Example of a single-rate task chain.

Here is the sequence of events that occur as soon as τ_1 is triggered: it reads the data from Reg-1, executes its functionality, writes the data to Reg-2 and immediately triggers τ_2. The tasks τ_2 and τ_3 follow similar execution steps with an exception of the activation by their predecessor tasks unlike τ_1. Assuming no interferences, the response time of the chain can

[1] A register may correspond to a port of a software component which is realized by the task at run-time.

be intuitively calculated by summing WCETs of all tasks in the chain, i.e., 3 time units. In this case, there is only one path through which the data can traverse through the chain from input (Reg-1) and appears at the output (Reg-3). Intuitively, the response time of the chain also represents the end-to-end path delay.

However, the timing behavior of a multi-rate real-time system cannot be completely predicted based on the response-time analysis results. For these systems, different types of end-to-end path delays should also be computed and compared to corresponding end-to-end deadlines. Nevertheless, these delays implicitly depend upon response times of the tasks. For example consider a multi-rate task chain shown in Fig. 2. The task chain consists of three tasks denoted by τ_1, τ_2 and τ_3. The tasks are triggered by independent periodic clocks with periods of 8, 8 and 4 time units respectively. The WCETs of each task is assumed to be 1 time unit. When the task τ_1 is triggered, it reads the data from Reg-1, executes some functionality and finally writes the data to Reg-1. The tasks τ_2 and τ_3 follow similar operations on the corresponding input and output registers. Since τ_3 is triggered independently, the data produced by it at Reg-4 at the time of its response corresponds to the input data in Reg-3 and it may not correspond to the fresh input data in Reg-1. Intuitively, the end-to-end delay can be significantly higher than the end-to-end response time in the multi-rate chains. These delays are discussed in detail in the next section.

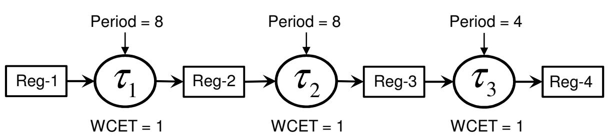

Fig. 2. Example of a multi-rate task chain.

The end-to-end timing constraints in automotive multi-rate real-time systems are formally defined by Stappert et al. [4]. Whereas, a framework for the calculations of end-to-end path delays for these systems is developed by Feiertag et al. [1]. These two works have been conducted in cooperation with an EU project, TIMMO2USE [5], in which the Timing Augmented Description Language (TADL2) [6] is developed to provide AUTOSAR [7] with a timing model. AUTOSAR is an industrial initiative to provide a standardized software architecture for the development of software for automotive embedded systems. Some timing constraints included in TADL2 correspond to semantics of the end-to-end path delays defined in [4], [1]. The corresponding end-to-end path delay analysis has also been implemented in several industrial tools including Rubus-ICE [8]. Alur et al. [9] developed a technique for the calculation of the end-to-end delays. However, this technique is based on model checking. A compositional scheduling approach based on event stream models has been introduced in [10]. However, the approach focusses on FIFO-based communication among components. On the other hand, we focus on register-based (port-based) communication which is very common in automotive systems [7], [11], [4], [1], [2].

In this paper, we consider the end-to-end path delay analysis developed in [1] due to several reasons. First, it is build upon existing response-time analysis. Second, it is flexible in a sense that it is equally applicable to the single-node as well as distributed multi-rate real-time systems. Finally, it is the most recent analysis for multi-rate real-time systems in the automotive domain. Moreover, it is acknowledged by the AUTOSAR consortium including some tool vendors.

III. End-to-end Path Delays

In order to explain the meaning of end-to-end path delays, consider again the task chain shown in Fig. 2. Since each task in the chain is activated independently with a different clock, there can be several paths through which the data can traverse from input to output. In other words, there can be multiple outputs (values in Reg-4) corresponding to one input (value in Reg-1) of the task chain as shown by the uni-directional curved arrows in Fig. 3. This results in different end-to-end path delays. These delays in the task chain shown in Fig. 2 are graphically illustrated in Fig. 3. It should be noted that the existing end-to-end path delay analysis [1] assumes fixed-priority preemptive scheduling while the tasks are assigned priorities according to rate-monotonic algorithm. The execution scenario shown in Fig. 3 is created for simplicity and may not represent the exact worst-case scenario.

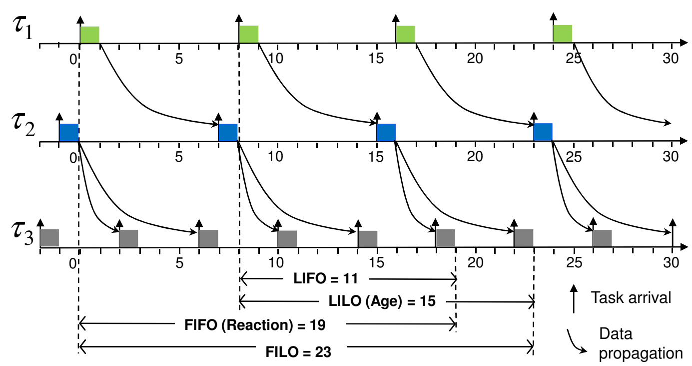

Fig. 3. End-to-end path delays in a multi-rate task chain in Fig. 2.

1) Last In First Out (LIFO) Path Delay: This delay is equal to the time elapsed between the current non-overwritten release of τ_1 (input of the chain) and corresponding first response of τ_3 (output of the chain).

2) Last In Last Out (LILO) Path Delay or Age Delay: This delay is equal to the time elapsed between the current non-overwritten release of τ_1 and corresponding last response of τ_3. This delay finds its importance in many multi-rate systems including the control systems where interest lies in the freshness or age of the produced data. For instance, in a multi-rate control system that initiates by acquiring a sensor input and terminates by producing an actuation signal, it is vital to ensure that the actuator signal does not exceed a timing constraint such as maximum age of the data. That is why this path delay is better known as the "Age" delay in the automotive domain [6], [11], [1], [8]. We overload the terms age and LILO to mean the same delay. It should be noted that the last non-overwritten input that actually propagates through the task chain towards the output is considered in both LIFO and LILO path delays.

3) First In First Out (FIFO) Path Delay or Reaction Delay: This delay specifies the longest allowed reaction time for the data produced by the initiator to be delivered to the terminator. Formally, it can be defined as the time elapsed between the previous non-overwritten release of τ_1 and the first response of τ_3 corresponding to the current non-overwritten release of τ_1. In order to understand this definition, assume that a new value of the input is available in Reg-1 "just after" the release of the first instance of τ_1 (at time unit 0 in Fig. 3). Intuitively, the first instance of τ_1 "just misses" reading the new value from Reg-1. The new data has to wait until the release of the next instance of τ_1 to propagate towards the output of the task chain. The new data is read by the second instance of τ_1. The first output corresponding to the new data (arriving just after time unit 0) appears at the output of the chain at 19 time units. This delay represents the FIFO path delay as shown in Fig. 3. This type of path delay is more obvious in distributed real-time systems where a task in the receiving node may just miss to read the fresh signals from a message arriving from the network. In the automotive domain, this delay is better know as the "Reaction" delay because it represents the first reaction corresponding to the input data [6], [11], [1], [8]. This delay is important in the button-to-reaction applications used in the body electronics domain where the first reaction to input is important. We overload the terms reaction and FIFO to mean the same delay.

4) First In Last Out (FILO) Path Delay: This delay specifies the longest time elapsed between the previous non-overwritten release of τ_1 and the last response of τ_3 corresponding to the current non-overwritten release of τ_1. The explanation about "just missing" a fresh input data discussed to explain the FIFO path delay equally applies here.

IV. Problem Statement

Before presenting the problem, first we briefly discuss a few terms. A timed path is a sequence of task instances in a multi-rate task chain whose sequential execution results in the propagation of data from input to output of the chain. There can be several valid timed paths in a multi-rate task chain. Assume $\tau_1(2)$ to denote the second instance of τ_1. An example of a valid timed path is $\tau_1(2) \rightarrow \tau_2(3) \rightarrow \tau_3(7)$, i.e., data path represented by the second, third and seventh instances of τ_1, τ_2 and τ_3 in Fig. 3. Let the end-to-end path delay for this timed path be denoted by $Delay^{E2E}_{(\tau_1(2)\rightarrow\ \tau_2(3)\rightarrow\ \tau_3(7))}$. According to the existing analysis, this delay is calculated as follows.

$$Delay^{E2E}_{(\tau_1(2)\rightarrow\ \tau_2(3)\rightarrow\ \tau_3(7))} = \alpha_3(7) + R_3(7) - \alpha_1(2) \quad (1)$$

$\alpha_3(7)$ represents the activation time of the seventh instance of τ_3. Similarly, $\alpha_1(2)$ represents the activation time of the second instance of τ_1. Whereas, $R_3(7)$ represents the response time of the seventh instance of τ_3. We refer the reader to [1] for the detailed calculations of different types of delays.

Now consider a special multi-rate task chain in which all tasks have equal priorities while the chain contains a single-rate sub-chain as shown in Fig. 4. The sub-chain, consisting of τ_1 and τ_2, is single-rate because there is only one independent activating source along the sub-chain that triggers τ_1 with a period of 8 time units. Whereas, τ_2 is activated only by its predecessor τ_1. Hence, there is a precedence relation between τ_1 and τ_2. Overall, the task chain is a multi-rate chain because τ_1 and τ_3 are triggered independently with different clocks.

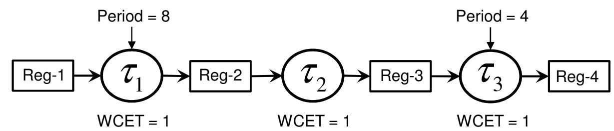

Fig. 4. Example of a multi-rate task chain with a single-rate sub-chain.

Due to the presence of a single-rate sub-chain, the timed paths for the multi-rate chain in Fig. 4 are different from the timed paths for the task chain in Fig. 2. For instance, the first instance of τ_2 cannot execute before the first instance of τ_1 due to precedence relation between the two tasks. As soon as τ_1 finishes its execution, it activates τ_2. This is in contrary to the task chain in Fig. 4 where the first instance of τ_2 can be executed before the first instance of τ_1 as shown in Fig. 3. The end-to-end path delays for the task chain in Fig. 4 are shown in Fig. 5 for the scenario where the smallest offset (1 time unit) between τ_1 and τ_2 is used in the single-rate sub-chain. Clearly, the offset between the two tasks cannot be zero because τ_2 cannot be activated before τ_1 has finished its execution.

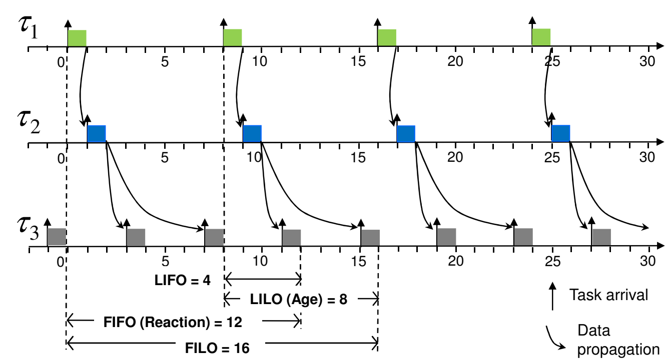

Fig. 5. End-to-end path delays of the multi-rate task chain in Fig. 4 with the smallest offset between τ_1 and τ_2 in the single-rate sub-chain.

Some of the assumptions used by the existing end-to-end path delay analysis may not hold in some legacy tools that are used for the development of multi-rate automotive systems such as Rubus-ICE [2]. The Rubus schedular applies the fixed priority preemptive scheduling [12] to transactions. Each clock trigger defines a transaction and all tasks along the trigger chain (i.e., single rate chain) are allocated to the transaction. For example, the task chain in Fig. 4 is allocated to two transactions. The first transaction with a period of 8 time units contains the two tasks τ_1 and τ_2. Whereas, the second transaction with a period of 4 time units consists of only τ_3. Within a transaction consisting of more than one task, every two neighboring tasks bear a precedence relation. The tasks within a transaction are scheduled with offsets. The offset assignment tool may apply optimizations based on response times and not on the end-to-end path delays. As a result of the optimization, a task in the single-rate sub-chain may not be scheduled immediately after its predecessor, e.g., unlike the first instance of τ_2 that is scheduled immediately after the execution of the first instance of τ_1 in Fig. 5.

However, the optimization tool guarantees to schedule the task such that it is able to complete its execution before the start of the new period of its predecessor task within the transaction. This can be observed in Fig. 6 where the first instance of τ_2 is scheduled as late as possible so that it finishes its execution before time unit 8, i.e., before the start of the next period of τ_1. The tool assigns such offsets within the transactions to accommodate event-triggered tasks and external interrupts occurring at various priority levels. It should be noted that the third instant of τ_3 is preempted by the first instance of τ_2 because the optimization tool is constrained to schedule each instance of τ_2 within the period of τ_1. In comparison, if all tasks in the chain are assigned priorities according to the rate-monotonic algorithm, the third instance of τ_3 cannot be preempted. However, the optimization tool uses the arbitrary priority assignment. This can be seen from the equal priorities assigned to all tasks in the chain despite the fact that periods of all tasks are not equal. Since, the existing end-to-end path delay analysis considers the rate monotonic scheduling, it cannot be applied to those multi-rate task chains in which all tasks have different periods but equal priorities; and there exists at least one single-rate sub-chain. This is because the legacy tool that is used for the task parameters generation may optimize the tasks with equal priorities for response times and not for the end-to-end path delays.

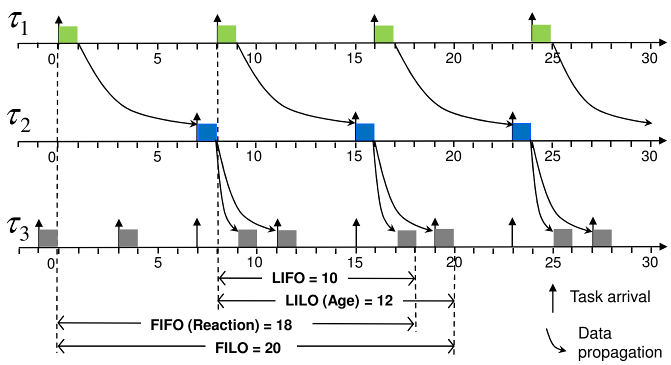

Fig. 6. End-to-end path delays of the multi-rate task chain in Fig. 4 with the longest offset between τ_1 and τ_2 in the single-rate sub-chain.

Discussion: Due to the optimizations and constraints used in some legacy tools that are used for the development of multi-rate task chains, the assumptions used by the existing analysis [1] may be violated by the tools in some cases. As a result, the existing analysis may not be applicable in such cases. For example, the existing analysis assumes rate-monotonic scheduling, whereas the tool may use arbitrary priority assignment. We identified one such case where all tasks in the chain have same priorities despite having different periods and the chain contains at least one single-rate task chain. It should be noted that this is not because of any flaw in the existing analysis. In fact, it is because of constraints specified in the optimization tools that generate task parameters and sometimes violate the assumptions used by the analysis. Apart from the case identified, the existing analysis holds good.

V. Preliminary Solution and Ongoing Work

In a bid to reuse the existing analysis, a simple yet intuitive solution to deal with the problem is to treat every multi-rate task chain having at least one single-rate sub-chain similar to an equivalent pure multi-rate task chain. That is, each task in such a task chain is assumed to be activated independently. Although some tasks in the task chain are actually activated by their predecessor tasks, the analysis engines assume them to be activated by independent clocks whose periods are equal to the triggering clocks of their predecessors. For example, the analysis engines treat the task chain in Fig. 4 exactly similar to the task chain in Fig. 2. Although τ_2 is triggered by τ_1 in Fig. 4, the analysis engines assume that τ_2 is triggered by an independent clock with a period of 8 time units. However, if we use this assumption while analyzing the special multi-rate task chain in Fig. 4, the calculated end-to-end path delays can be pessimistic. For example, after using the assumption, the end-to-end path delays (LIFO = 11, LILO = 15, FIFO = 19, FILO = 23) shown in Fig. 3 are higher than the end-to-end path delays (LIFO = 10, LILO = 12, FIFO = 18, FILO = 20) in Fig. 6 where the longest offset (7 time units) between τ_1 and τ_2 in the single-rate sub-chain is considered.

Currently, we are working on extending the existing end-to-end path delays analysis so that the pessimism in the calculated end-to-end path delays can be minimized. After that, we plan to implement the extended analysis in Rubus-ICE tool suite. Moreover, we plan to provide a proof of concept by conducting an industrial-application case study.

Acknowledgement

This work is supported by the Swedish Foundation for Strategic Research (SSF) within PRESS project. We thank the industrial partners Arcticus Systems AB and Volvo Sweden.

References

[1] N. Feiertag, K. Richter, J. Nordlander, and J. Jonsson, "A Compositional Framework for End-to-End Path Delay Calculation of Automotive Systems under Different Path Semantics," in *Workshop on Compositional Theory and Technology for Real-Time Embedded Systems*, Dec. 2008.

[2] "Rubus-ICE: Integrated component Development Environment," http://www.arcticus-systems.com.

[3] M. Joseph and P. Pandya, "Finding response times in a real-time system," *Computer Journal*, vol. 29, no. 5, pp. 390–395, 1986.

[4] F. Stappert, J. Jonsson, J. Mottok, and R. Johansson, "A Design Framework for End-To-End Timing Constrained Automotive Applications," in *Embedded Real-Time Software and Systems (ERTS), 2010.*

[5] Mastering Timing Information for Advanced Automotive Systems Engineering. In the TIMMO-2-USE Brochure, 2012. Available at: http://www.timmo-2-use.org/pdf/T2UBrochure.pdf.

[6] Timing Augmented Description Language (TADL2) syntax, semantics, metamodel Ver. 2, Deliverable 11, Aug. 2012.

[7] "AUTOSAR Techincal Overview, Release 4.1, Rev. 2, Ver. 1.1.0., The AUTOSAR Consortium, Oct., 2013," http://autosar.org.

[8] S. Mubeen, J. Mäki-Turja, and M. Sjödin, "Support for end-to-end response-time and delay analysis in the industrial tool suite: Issues, experiences and a case study," *Computer Science and Information Systems*, vol. 10, no. 1, 2013.

[9] A. C. Rajeev, S. Mohalik, M. G. Dixit, D. B. Chokshi, and S. Ramesh, "Schedulability and end-to-end latency in distributed ecu networks: formal modeling and precise estimation," in *Proceedings of the tenth ACM international conference on Embedded software*, ser. EMSOFT '10. ACM, 2010, pp. 129–138.

[10] K. Richter, D. Ziegenbein, M. Jersak, and R. Ernst, "Model composition for scheduling analysis in platform design," in *Proceedings of the 39th Annual Design Automation Conference*, 2002, pp. 287–292.

[11] "EAST-ADL Domain Model Specification, Deliverable D4.1.1," http://www.atesst.org/home/liblocal/docs/ATESST2_D4.1.1_EAST-ADL2-Specification_2010-06-02.pdf.

[12] N. Audsley, A. Burns, R. Davis, K. Tindell, and A. Wellings, "Fixed priority pre-emptive scheduling: an historic perspective," *Real-Time Systems*, vol. 8, no. 2/3, pp. 173–198, 1995.