On the Equivalence of Maximum Reaction Time and Maximum Data Age for Cause-Effect Chains

Mario Günzel
Department of Computer Science, TU Dortmund University, Germany

Harun Teper
Department of Computer Science, TU Dortmund University, Germany

Kuan-Hsun Chen
University of Twente, The Netherlands

Georg von der Brüggen
Department of Computer Science, TU Dortmund University, Germany

Jian-Jia Chen
Lamarr Institute for Machine Learning and Artificial Intelligence, Germany
Department of Computer Science, TU Dortmund University, Germany

Abstract

Real-time systems require a formal guarantee of timing-constraints, not only for individual tasks but also for data-propagation. The timing behavior of data-propagation paths in a given system is typically described by its maximum reaction time and its maximum data age. This paper shows that they are equivalent.

To reach this conclusion, partitioned job chains are introduced, which consist of one immediate forward and one immediate backward job chain. Such partitioned job chains are proven to describe maximum reaction time and maximum data age in a universal manner. This universal description does not only show the equivalence of maximum reaction time and maximum data age, but can also be exploited to speed up the computation of such significantly. In particular, the speed-up for synthesized task sets based on automotive benchmarks can be up to 1600.

Since only very few non-restrictive assumptions are made, the equivalence of maximum data age and maximum reaction time holds for almost any scheduling mechanism and even for tasks which do not adhere to the typical periodic or sporadic task model. This observation is supported by a simulation of a ROS2 navigation system.

2012 ACM Subject Classification Computer systems organization → Embedded and cyber-physical systems; Software and its engineering → Real-time systems software

Keywords and phrases End-to-End, Timing Analysis, Maximum Data Age, Maximum Reaction Time, Cause-Effect Chain, Robot Operating Systems 2 (ROS2)

Digital Object Identifier 10.4230/LIPIcs.ECRTS.2023.9

Funding This work has been supported by Deutsche Forschungsgemeinschaft (DFG), as part of Sus-Aware (Project No. 398602212). This result is part of a project (PropRT) that has received funding from the European Research Council (ERC) under the European Union's Horizon 2020 research and innovation programme (grant agreement No. 865170). This work was partially funded by the German Federal Ministry of Education and Research (BMBF) in the course of the 6GEM research hub under grant number 16KISK038.

1 Introduction

In various embedded system applications, e.g., in automotive or avionics systems, a sequence of tasks is necessary to perform a certain functionality. The data dependency between these tasks is described by a cause-effect chain. A typical example is a cause-effect chain from a sensor to an actuator, where the first task reads the sensor value (cause), the second task processes the data, and the third task produces an output for the actuator (an effect is triggered). Timing properties of such cause-effect chains must be validated to ensure system safety.

Typical metrics to describe end-to-end timing properties are maximum reaction time (MRT) and maximum data age (MDA). The MRT is the longest time interval from an external cause until the earliest time where this external cause is fully processed (also called the maximum button to action delay). The MDA is the longest time interval starting from sampling a value and ending at an actuation that takes place based on that sampled data (also called the data freshness). Due to their importance, multiple approaches to calculate or bound MRT or MDA have been provided [2, 3, 6, 8–11, 13, 17, 24, 26].

However, the relation between the MRT and MDA values is less analyzed. This is kind of surprising, since answering the questions *if, how, and in which scenarios the MRT and MDA values are related* is interesting – both from a practical perspective (since it may be sufficient to analyze one metric instead of two) and from a research perspective (since analysis methods for one metric may also be applied when analyzing the other). Hence, in this work, we focus on such relations between MRT and MDA. The strongest analytical result known has been provided by Günzel et al. [13], showing that the MRT is an upper bound for the MDA.

Nonetheless, empirical observations suggest a stronger relation. More specifically, the AUTOSAR Timing Extensions [1] provide an important observation about the relation between MRT and MDA, namely, that "*without over- and undersampling, age and reaction are the same*" [1, Section 7.2, p. 149]. However, while this observation seems to imply that MRT and MDA can differ for systems with over- or undersampling, recent measurements in Robot Operating System 2 (ROS2) show that the observed MRT and MDA always coincide [27]. Hence, it is unclear in which scenarios MRT and MDA coincide. Even more, while both observations suggest a strong relation between MRT and MDA, no proof for such a relation is provided.

Hence, in this paper, we further investigate MRT and MDA through analytical discussion to determine *if, how, and in which scenarios the MRT and MDA values are related.* Specifically, we formally prove that they are equivalent after a warm-up period, i.e., after the data passes the complete cause-effect chain once. This insight allows the verification of timing constraints for both metrics at the same time. Moreover, analytical results in the literature for one metric can be utilized for the other one.

We build on the established result [9, 13] that for each cause-effect chain MRT and MDA can be calculated based on the length of the related job chains; that is, the time interval between the moment the first job in the chain reads data and the moment the last job in the chain writes data. On a high level, our idea is to first examine the job chains that must be considered for the MRT; that is, the job chains from the first task in the chain (i.e., the sensor) to the last (i.e., the actuator). In the next step, we consider a chain comprised of two sub-chains, both starting from the second task in the chain – one going back to the sensor and one going forward to the actuator. We call such a chain a p-partitioned job chain, where p denotes the position of the task in the chain (in this case, $p = 2$). We show that one of these 2-partitioned job chains has at least the same length as the job chain that determines the MRT. We continue by induction over the tasks in the chain, showing that the MRT is upper-bounded by the MDA. Afterwards, we similarly show that the MDA is upper-bounded by the MRT by starting from the last task in the chain.

Contributions: We show the equivalence of MRT and MDA while making only very few non-restrictive assumptions regarding tasks, communication, and scheduling model. Therefore,

Variable	Definition
$\mathbb{T}$	task set under analysis
$\tau \in \mathbb{T}$	a task
$\tau(m),\ m \in \mathbb{N}^+$	m-th job of task τ
$\preccurlyeq$	job ordering for jobs of one task
$\tau = (C_\tau, T_\tau, \phi_\tau)$	periodic task
$\tau = (C_\tau, T_\tau)$	sporadic task
$\mathrm{re}(J)$	time of read-event of a job J
$\mathrm{we}(J)$	time of write-event of a job J
$\overline{E}$	cause-effect chain under analysis
E	any cause-effect chain (can be $\overline{E}$ or a sub-chain of $\overline{E}$)
$\|E\|$	number of tasks in E
$E(i),\ i \in \{1, \dots, \|E\|\}$	i-th task of E

Table 1 Notation in this work.

our results apply for a large variety of systems. Specifically, our results can be applied for but are not limited to periodic or sporadic tasks and implicit communication or logical execution time (LET) [15]. The underlying scheduler may be (i) a job-level or task-level static priority scheduler (e.g., EDF and rate-monotic, respectively), (ii) work-conserving or non-work-conserving, and (iii) preemptive or non-preemptive. Furthermore, jobs may be executed on different processing units (e.g., on different electronic control units (ECUs)). In detail, we make the following technical contributions:

- In Section 4, we define p-partitioned job chains, where p is an integer value not larger than the number of jobs $|\overline{E}|$ in the evaluated job chain. We show that MRT and MDA can be expressed as 1-partitioned job chains and $\overline{E}$-partitioned job chains, respectively.
- In Section 5, we discuss the equivalence between MRT and MDA using partitioned job chains and show that the timing behavior is independent of p, i.e., any arbitrary p can be chosen to compute MDA and MRT.
- The implication of our results in practice is discussed in Section 6.
- We discuss how to apply our results to a reduced version of MRT and MDA in Section 7.1 and how this equivalence can be transferred to a definition of MRT and MDA based on valid job chains Section 7.2.
- We evaluate our results considering randomly generated periodic tasks, communicating via Logical Execution Time (LET) [15] in Section 8. In particular, we show that by the right choice of p the required time for computation is reduced significantly.
- To validate our theoretical results, we examine MRT, MDA on a ROS2 system with non-periodic tasks under implicit communication in Section 9.

2 System Model and Problem Definition

This section introduces the definitions and notations for task model, communication model, and cause-effect chains, as well as our problem definition. For our analysis, a very general task and communication model with very few assumptions is sufficient. Yet, we introduce the notion of *periodic* and *sporadic* tasks as well as the communication policies *implicit communication* and *logical execution time (LET)*, as they are utilized to provide intuitive examples, in the empirical evaluation with synthesized tasks sets in Section 8, and for the case study in Section 9. Our notation is summarized in Table 1.

Jobs and Tasks. A *job* is a program instance that produces output based on its input. The aggregation of all jobs of the same program is called a *task*, denoted by τ. We denote the (countably many) jobs of task τ by $(\tau(m))_{m\in\mathbb{N}^+}$, and denote the induced ordering by $\tau(i) \preccurlyeq \tau(j)$ if and only if $i \leq j$. The set of all tasks in the system is denoted by $\mathbb{T}$.

For our analysis, we need no further assumption on the job releases. Specifically, we make no assumptions regarding the first time a job of a task arrives or on the inter-arrival pattern of jobs. However, the most common task models, namely the periodic and the sporadic model, both fulfill our assumptions.

A **periodic task** τ is described as $\tau = (C_\tau, T_\tau, \phi_\tau) \in \mathbb{R}^3$, where $C_\tau \geq 0$ is the worst-case execution time (WCET), $T_\tau > 0$ is the period, and ϕ_τ is the phase of the task. The first job $\tau(1)$ of τ is released at time ϕ_τ and subsequent jobs are released every T_τ time units, i.e., $\tau(m)$ is released at time $\phi_\tau + (m-1)T_\tau$. Every job of τ executes for at most C_τ time units.

A **sporadic tasks** τ is described by the tuple $\tau = (C_\tau, T_\tau) \in \mathbb{R}^2$, where $C_\tau \geq 0$ is the worst-case execution time (WCET), and $T_\tau > 0$ is the minimum inter-arrival time between two jobs. The release of two subsequent jobs of τ are separated by at least T_τ time units and each job executes for at most C_τ time units.

Communication. Jobs communicate by receiving (reading) their input from a shared resource and handing over (writing) their output to a shared resource. We denote the time that the *read-event* of a job J takes place by $\mathrm{re}(J) \in \mathbb{R}$ and the time that the *write-event* of J takes place by $\mathrm{we}(J) \in \mathbb{R}$. We assume the following two requirements are met:

- **(R1)** For each task $\tau \in \mathbb{T}$, the read- and write-events of its jobs are ordered in the sense that $\mathrm{re}(\tau(m)) < \mathrm{re}(\tau(m+1))$, $\mathrm{we}(\tau(m)) < \mathrm{we}(\tau(m+1))$, and $\mathrm{re}(\tau(m)) \leq \mathrm{we}(\tau(m))$ for all $m \in \mathbb{N}$.
- **(R2)** The sets $\{\mathrm{re}(\tau(m)) \,|\, m \in \mathbb{N}\}$ and $\{\mathrm{we}(\tau(m)) \,|\, m \in \mathbb{N}\}$ have no accumulation point, i.e., the number of read- and write-events in each bounded time interval is finite.

These not very restrictive requirements are fulfilled by commonly considered communication semantics, e.g., for *logical execution time* and *implicit communication*.

Logical Execution Time (LET): Under logical execution time [15], each task τ is assigned an arbitrary deadline D_τ, and the read-event and write-event of each job J of τ is set to its release time r_J and its absolute deadline $r_J + D_\tau$, respectively.

Implicit Communication: Under implicit communication, each job has its read-event at the first time that it is executed, i.e., when the job starts, and the job has its write-event at the last time it is executed, i.e., when the job finishes.

Cause-effect chains. A *cause-effect chain* $E = (\tau_1 \to \tau_2 \to \cdots \to \tau_k)$, with $k \in \mathbb{N}$, describes the path of data through different programs by a finite sequence of tasks $\tau_i \in \mathbb{T}$. The number of entries in the sequence E is denoted as $|E|$ and $E(i)$ is the task at the i-th entry of E for $i \in \{1, \ldots, |E|\}$. This definition of cause-effect chains is inspired by event-chains of the AUTOSAR Timing Extensions [1], which represent chains of more general functional dependency. We assume implicit sampling, where the sampling for a cause-effect chain E happens at the read-event of each job of $E(1)$. However, we can easily model any kind of sampling by adding a *sampling task* τ^{sample} to the system, where each job $\tau^{sample}(1), \tau^{sample}(2), \ldots$ reads and writes data at a time when the sampling happens.

If the data dependency is described by a directed acyclic graph (DAG) with several sources and sinks, we follow the typical approach from the literature and analyze each cause-effect chain (i.e., path through the DAG from a source to a sink) individually.

Problem definition. For a given cause-effect chain $\overline{E}$, we discuss the equivalence of its maximum reaction time and maximum data age. Specifically, we answer the question: *To* *which extent can the maximum reaction time and maximum data age (and their variations) be derived from each other?* We answer this question for large variety of systems, including periodic or sporadic task systems with LET or implicit job communication.

3 Maximum Reaction Time & Maximum Data Age

This section specifies the analyzed End-to-End latencies, namely, the *maximum reaction time* (MRT), which is the length of the longest time interval from an external cause until the earliest time where this external cause is fully processed (i.e., the maximum button to action delay), and the *maximum data age* (MDA), which is the length of the longest time interval starting from sampling a value until an actuation based on that sampled value takes place (i.e., the data freshness). The definitions are based on *job chains* which represent the path of data through the schedule. We recap the job chain definitions for arbitrary cause-effect chains E of the task set $\mathbb{T}$ as stated by Günzel et al. [13].

▶ **Definition 1** (Job chain [13])**.** Let E be a cause effect chain of the task set $\mathbb{T}$. A job chain $c = (J_1, \dots, J_{|E|})$ for E is a sequence of jobs where the following two conditions are fulfilled:

- J_i is a job of task $E(i)$ for all $i \in \{1, 2, \dots, |E|\}$.
- Each job in the chain reads the data not before it was written by the previous job in the chain. That is, $\mathrm{we}(J_{i-1}) \leq \mathrm{re}(J_i)$ for all $i \in \{2, 3, \dots, |E|\}$.

The *length of a job chain c* is the length of the time interval between the read-event of the first job J_1 in the chain and the write-event of the last job $J_{|E|}$ in the chain, i.e.,

$$\ell(J_1, J_2, \dots, J_{|E|}) = \mathrm{we}(J_{|E|}) - \mathrm{re}(J_1). \tag{1}$$

We denote the i-th job in c as $c(i)$ for $i \in \{1, 2, \dots, |E|\}$, hence, $\ell(c) = \mathrm{we}(c(|E|)) - \mathrm{re}(c(1))$.

Like in previous work [9,13], our analysis for MRT and MDA is built on two related types of jobs chains: *Immediate forward* job chains and *immediate backward* job chains.

▶ **Definition 2** (Immediate forward job chain [13])**.** Let E be a cause-effect chain for task set $\mathbb{T}$. A job chain $c = (J_1, J_2, \dots, J_{|E|})$ for E is *immediate forward* if for all $i \in \{2, \dots, |E|\}$ the job J_i is the *earliest* job of task $E(i)$ with read-event no earlier than the write-event of J_{i-1}. That is, J_i is the earliest job that fulfills the properties from Definition 1.
We denote the m-th immediate forward job chain for E (i.e., $J_1 = E(1)(m)$) by $\vec{c}^{E}_{m}, m \in \mathbb{N}^+$.

▶ **Definition 3** (Immediate backward job chain [13])**.** Let E be a cause-effect chain for task set $\mathbb{T}$. A job chain $c = (J_1, J_2, \dots, J_{|E|})$ for E is *immediate backward* if for all $i \in \{|E| - 1, \dots, 1\}$ the job J_i is the *latest* job of task $E(i)$ with write-event no later than the read-event of J_{i+1}. That is, J_i is the latest job that fulfills the properties from Definition 1.
For $m \in \mathbb{N}^+$, if there is an immediate backward job chain with $J_{|E|} = E(|E|)(m)$, then we call it the m-th immediate backward job chain $\overleftarrow{c}^{E}_{m}$.

For an example system with three tasks and implicit communication, the forward arrows in Figure 1 mark all job chains starting at the job $\tau_1(1)$. The immediate forward job chain starting at $\tau_1(1)$ is marked red. The immediate backward job chain starting at $\tau_3(6)$ is marked with a dotted blue arrow.

To account for the time of the external activity (z) and the actuation (z'), we consider augmented job chains.

▶ **Definition 4** (Immediate forward augmented job chain [13])**.** Let $m \in \mathbb{N}^+$. We define the m-th immediate forward augmented job chain for $\overline{E}$ by $\vec{ac}_m = (z, J_1, \dots, J_{|\overline{E}|}, z')$, where

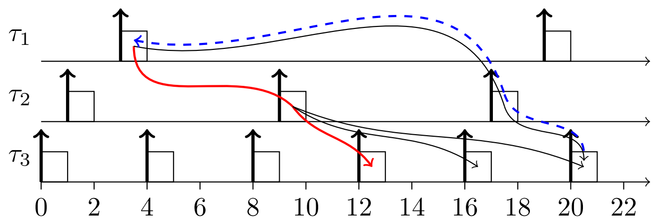

Figure 1 Cause-effect chain $E = (\tau_1 \to \tau_2 \to \tau_3)$ for three tasks with implicit communication. Forward arrows mark the job chains with first job $\tau_1(1)$, the *immediate forward* job chain $\vec{c}_1^E$ is marked red. The dashed blue arrow marks the *immediate backward* job chain $\overleftarrow{c}_6^E$ starting at $\tau_3(6)$.

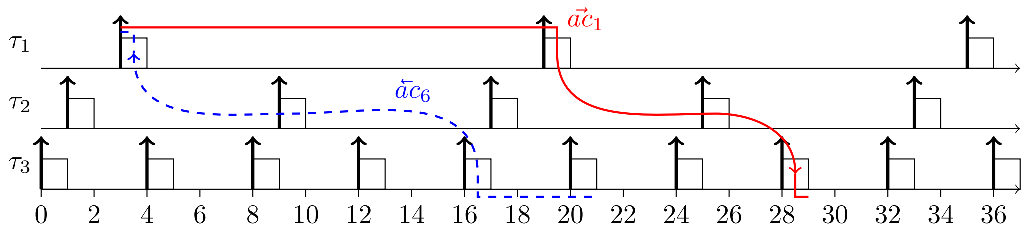

Figure 2 Cause-effect chain $E = (\tau_1 \to \tau_2 \to \tau_3)$ with implicit communication. The immediate forward augmented job chain $\vec{ac}_1 = (3, \tau_1(2), \tau_2(4), \tau_3(8), 29)$ is marked red, and the immediate backward augmented job chain $\overleftarrow{ac}_6 = (3, \tau_1(1), \tau_2(2), \tau_3(5), 21)$ is marked by a dashed blue arrow.

$z \in \mathbb{R}$ is just after the read-event of the m-th job of task $\overline{E}(1)$, $(J_1, \dots, J_{|\overline{E}|})$ is the $(m+1)$-th immediate forward job chain for $\overline{E}$, and z' is at the write-event of $J_{|\overline{E}|}$.

▶ **Definition 5** (Immediate backward augmented job chain [13]). Let $m \in \mathbb{N}^+$. If $\overleftarrow{c}_{m-1}^{\overline{E}}$ exists, then we define the m-th immediate backward augmented job chain for $\overline{E}$ by $\overleftarrow{ac}_m = (z, J_1, \dots, J_{|\overline{E}|}, z')$, where z' is just before the write-event of the m-th job of task $\overline{E}(|\overline{E}|)$, $(J_1, \dots, J_{|\overline{E}|})$ is the $(m-1)$-th immediate backward job chain for $\overline{E}$, and z is at the read-event of job J_1.

Examples of an immediate forward augmented job chain and an immediate backward augmented job chain are illustrated in Figure 2. Note that $\overleftarrow{ac}_m$ can only be constructed if $\overleftarrow{c}_{m-1}^{E}$ exists. Moreover, if $\overleftarrow{ac}_m$ exists, then for all $\tilde{m} \geq m$, $\overleftarrow{ac}_{\tilde{m}}$ exists as well. Our notation related to job chains is summarized in Table 2.

Symbol	Name	Defined in
c	Job chain	Sec. 3, Def. 1
$\vec{c}$	Immediate forward job chain	Sec. 3, Def. 2
$\overleftarrow{c}$	Immediate backward job chain	Sec. 3, Def. 3
$\vec{ac}$	Immediate forward augmented job chain	Sec. 3, Def. 4
$\overleftarrow{ac}$	Immediate backward augmented job chain	Sec. 3, Def. 5
pc	Partitioned job chain	Sec. 4, Def. 9

Table 2 Job chains and extensions defined in this paper.

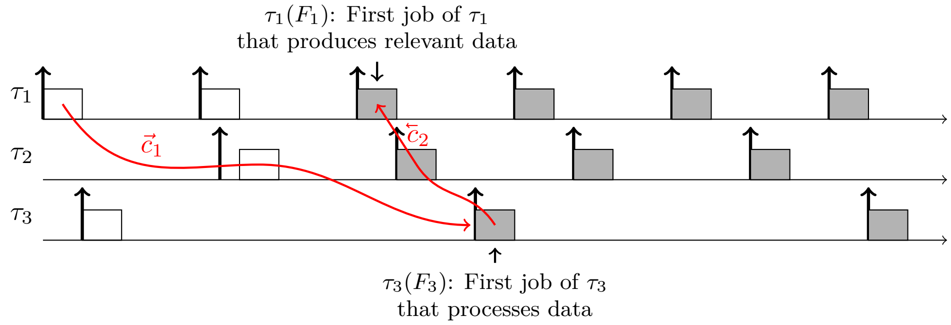

Figure 3 Schedule for the cause-effect chain $\overline{E} = (\tau_1 \to \tau_2 \to \tau_3)$. Data arrives at job $\tau_3(2)$. The warm-up covers the white jobs. MRT and MDA of the gray jobs coincides.

MRT and MDA can be defined based on Definition 4 and Definition 5 directly. However, if not all tasks in the chain are released at system start, data may be processed by some tasks in the chain but not by all and no actuation based on this data can happen. For example, in Figure 3 data written by $\tau_1(1)$ is overwritten by $\tau_1(2)$. Similarly, data written by $\tau_1(2)$ is used only by $\tau_2(1)$ but never reaches the last task because $\tau_2(2)$ overwrites the data written by $\tau_2(1)$. Hence, intuitively, only the gray jobs that process data that might be used in an actuation should be considered when determining MRT and MDA. We say that the system has *warmed up* when the complete chain processes data for the first time.

The first job of task $\overline{E}(|\overline{E}|)$ that reads data processed by all jobs in $\overline{E}$ can be determined by one immediate forward job chain, i.e., it is the last entry of $\vec{c}_1$. The immediate backward job chain based on that job exists and determines the first job of $\overline{E}(1)$ which processes data that reaches task $\overline{E}(|\overline{E}|)$. The procedure is summarized in Figure 3. We formalize this by specifying the *warm-up* period of the system.

▶ **Definition 6** (Warm-up)**.** Let $F \in \mathbb{N}^+$ such that $\vec{c}_1^{\,\overline{E}}(|\overline{E}|)$ is the F-th job of $\overline{E}(|\overline{E}|)$. Then we construct the immediate backward job chain $\overleftarrow{c}_F^{\overline{E}}$ and denote by $F_1, \ldots, F_{|\overline{E}|} \in \mathbb{N}^+$ the job number of each job in $\overleftarrow{c}_F^{\overline{E}}$, i.e., $\overleftarrow{c}_F^{\overline{E}} = (\overline{E}(1)(F_1), \ldots, \overline{E}(|\overline{E}|)(F_{|\overline{E}|}))$.[1]

The *warm-up* covers all jobs of $\overline{E}(i)$ before $\overline{E}(i)(F_i)$, $i = 1, \ldots, |\overline{E}|$. In particular, those jobs are not considered for the maximum reaction time or maximum data age.

Only augmented job chains with z no earlier than the read-event of $\overline{E}(1)(F_1)$ should be considered for the end-to-end latency. These are all $\vec{ac}_m$ with $m \geq F_1$ and all $\overleftarrow{ac}_m$ with $m \geq F_{|\overline{E}|} + 1$. For the following definitions, the length of an immediate forward or an immediate backward augmented job chain is the length of the time interval from z to z', i.e., $\ell(z, J_1, \ldots, J_{|\overline{E}|}, z') = z' - z$.

▶ **Definition 7** (Maximum reaction time)**.** The maximum reaction time for $\overline{E}$ is defined as

$$\mathrm{MRT} = \sup_{m \geq F_1} \ell(\vec{ac}_m) \tag{2}$$

[1] The existence of the immediate forward job chain $\vec{c}_1^{\,\overline{E}}$ ensures that $\overleftarrow{c}_F^{\overline{E}}$ can be fully constructed, i.e., during the backwards construction a job that writes data early enough can always be found. This is captured by Lemma 13 and not further discussed here to improve the reading flow.

▶ **Definition 8** (Maximum data age). The maximum data age for $\overline{E}$ is defined as

$$\text{MDA} = \sup_{m \geq F_{|\overline{E}|}+1} \ell(\overleftarrow{ac}_m) \tag{3}$$

In this work, for MRT the time from the external event z is included in the definition, as it is also done by Feiertag et al. [10], Dürr et al. [9], and Günzel et al. [13]. For the MDA, however, z' is included by Günzel et al. [13] but not by Feiertag et al. [10] and Dürr et al. [9]. We refer to this definition as maximum reduced data age (MRDA). These definitions cover slightly different scenarios: (i) The MRDA assumes that the actuation is directly triggered by the write event of the last task in the chain, while (ii) the MDA assumes the actuation is not directly triggered; thus, additional time for the actuation has to be included, which, in the worst case, may happen directly before the next write event of the last task in the chain. How MDA relates to MRDA is discussed in Section 7.1. Note, while not specifically stating this, AUTOSAR [1] considers MDA, as explained at the end of Section 7.1.

The definitions of MRT and MDA are similar in the work by Günzel et al. [13]. However, their calculation of MRT and MDA starts as soon as the augmented job chains become *valid*, which potentially includes part of the warm-up period. We discuss in Section 7.2 how our results can be extended to the definition of [13].

4 Partitioned Job chains

In this section, for a given task chain $\overline{E}$, we define p-partitioned job chains ($p \in \{1, 2, \ldots, |\overline{E}|\}$). We show that these p-partitioned job chains allow maximum reaction time (MRT) and maximum data age (MDA) definitions that are equivalent to the ones based on augmented job chains stated in Section 3. In particular, a 1-partitioned job chain is equivalent to an immediate forward augmented job chain and an $|\overline{E}|$-partitioned job chain is equivalent to an immediate backward augmented job chain. In Section 5, we utilize these p-partitioned job chains to show the equivalence between MRT and MDA by discussing the difference of p-partitioned and $(p+1)$-partitioned job chains for $p = 1, \ldots, |\overline{E}| - 1$.

A p-partitioned job chain is a combination of (i) two subsequent jobs J_p and $\tilde{J}_p$ of task $\overline{E}(p)$, (ii) an immediate backward job chain that starts at J_p, and (iii) an immediate forward job chain that starts at $\tilde{J}_p$. In Figure 4, the dashed chain pc_1^2 is a 2-partitioned job chain consisting of an immediate backward job chain with last entry $\tau_2(1)$, an immediate forward job chain with first entry $\tau_2(2)$, and the connection between $\tau_2(1)$ and $\tau_2(2)$.

We start by formally defining partitioned job chains.

▶ **Definition 9** (Partitioned job chain). Let $p \in \{1, \ldots, |\overline{E}|\}$ and $m \in \mathbb{N}^+$. Moreover, let $\overline{E}_p^{first} = (\overline{E}(1) \to \cdots \to \overline{E}(p))$, and let $\overline{E}_p^{last} = (\overline{E}(p) \to \cdots \to \overline{E}(|E|))$. If $\overleftarrow{c}_m^{\overline{E}_p^{first}}$ exists, then we define the m-th p-partitioned job chain pc_m^p by

$$pc_m^p = (J_1, \ldots, J_p, \tilde{J}_p, \ldots, \tilde{J}_{|\overline{E}|}) \tag{4}$$

where $(J_1, \ldots, J_p) = \overleftarrow{c}_m^{\overline{E}_p^{first}}$ is the m-th immediate backward job chain for the cause-effect chain $\overline{E}_p^{first}$ and $(\tilde{J}_p, \ldots, \tilde{J}_{|E|}) = \vec{c}_{m+1}^{\overline{E}_p^{last}}$ is the $(m+1)$-th immediate forward job chain for $\overline{E}_p^{last}$. In particular, $J_p = \overline{E}(p)(m)$ and $\tilde{J}_p = \overline{E}(p)(m+1)$.

The length of a partitioned job chain is $\ell(J_1, \ldots, J_p, \tilde{J}_p, \ldots, \tilde{J}_{|\overline{E}|}) = \text{we}(\tilde{J}_{|\overline{E}|}) - \text{re}(J_1)$.

Since pc_m^1 is composed of $\vec{c}_{m+1}^{\overline{E}}$ and one additional job at the beginning (for the external activity), it is equivalent to $\vec{ac}_m$. Since $pc_m^{|\overline{E}|}$ is composed of $\overleftarrow{c}_m^{\overline{E}}$ and one additional job at

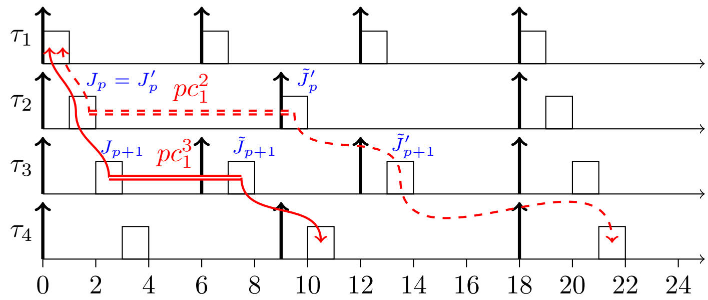

Figure 4 Four tasks with cause-effect chain ($\tau_1 \to \tau_2 \to \tau_3 \to \tau_4$) communicating via implicit communication. The red line depicts the chain pc_1^3 whereas the dashed red line depicts pc_1^2. The length of pc_1^3 is upper bounded by the length of pc_1^2. The blue job annotations illustrate the proof of Lemma 12 ($\geq$-relation).

the end (for the actuation), it is equivalent to $\overleftarrow{ac}_{m+1}$. For instance, in Figure 2 the chain $\vec{ac}_1$ is equivalent to the 1-partitioned job chain $pc_1^1 = (\tau_1(1), \tau_1(2), \tau_2(4), \tau_3(8))$, and $\overleftarrow{ac}_6$ is equivalent to the $|\overline{E}|$-partitioned job chain $pc_5^{|\overline{E}|} = (\tau_1(1), \tau_2(2), \tau_3(5), \tau_3(6))$. Note that the m-th $|\overline{E}|$-partitioned job chain $pc_m^{|\overline{E}|}$ is equivalent to the $(m+1)$-th augmented backward job chain $\overleftarrow{ac}_{m+1}$, i.e., the index is shifted by 1. Thus, their length is also the same:

▶ **Corollary 10.** *For all $m \in \mathbb{N}^+$ the following two properties hold.*
1. $\ell(pc_m^1) = \ell(\vec{ac}_m)$.
2. $pc_m^{|\overline{E}|}$ *exists if and only if* $\overleftarrow{ac}_{m+1}$ *exists. If they exist, then* $\ell(pc_m^{|\overline{E}|}) = \ell(\overleftarrow{ac}_{m+1})$.

We restate the definitions of MRT and MDA using p-partitioned job chains.

▶ **Definition 11** (MRT and MDA by partitioned job chains)**.** The maximum reaction time (MRT) and the maximum data age (MDA) of the cause-effect chain $\overline{E}$ can be expressed by partitioned job chains as follows:

$$\mathrm{MRT} = \sup_{m \geq F_1} \ell(pc_m^1) \tag{5}$$

$$\mathrm{MDA} = \sup_{m \geq F_{|\overline{E}|}} \ell(pc_m^{|\overline{E}|}) \tag{6}$$

5 Equivalence of MRT and MDA

In this section, we show the equivalence of maximum reaction time (MRT) and maximum data age (MDA). More precisely, we prove that, for a given cause-effect chain $\overline{E}$, the maximum length of a p-partitioned job chain is the same for all $p \in \{1, \dots, |\overline{E}|\}$. Since MRT and MDA can be expressed by 1-partitioned and by $|\overline{E}|$-partitioned job chains, respectively, this directly shows the equivalence of MRT and MDA.

We prove that the maximum length of a p-partitioned job chain is the same for all $p \in \{1, \dots, |\overline{E}|\}$ in two steps: (i) We show that for all $p \in \{|\overline{E}|, \dots, 2\}$ the length of every p-partitioned job chain is upper bounded by the length of a $(p-1)$ partitioned job chain. This scenario is depicted in Figure 4, where the length of pc_1^3 is upper bounded by the length of pc_1^2. (ii) Conversely, we show that for all $p \in \{1, \dots, |\overline{E}| - 1\}$ the length of every p-partitioned job chain is upper bounded by the length of a $(p+1)$-partitioned job chain. This is depicted in Figure 5, where the length of pc_2^2 is upper bounded by the length of pc_1^3.

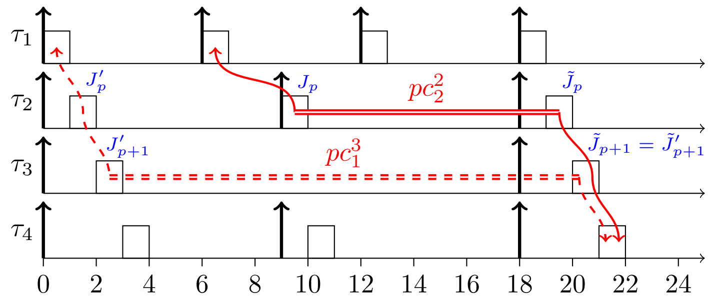

Figure 5 Four tasks with cause-effect chain ($\tau_1 \to \tau_2 \to \tau_3 \to \tau_4$) communicating via implicit communication. The red line depicts the chain pc_2^2 whereas the dashed red line depicts pc_1^3. The length of pc_2^2 is upper bounded by the length of pc_1^3. The blue job annotations illustrate the proof of Lemma 12 ($\geq$-relation).

One problem we have to consider during step (ii) is that some $(p+1)$-partitioned job chains may not be fully constructed. For example, if the first job of τ_1 in Figure 5 would be missing, then pc_1^3 could not be fully constructed and there would be no 3-partitioned job chain which provides an upper bound on the length of pc_2^2. This, however, does not impact our result, since we show that this scenario never occurs for p-partitioned job chains pc_m^p after the warm-up, i.e., if $m \geq F_p$ with F_p from Definition 6.

The following lemma indicates that p-partitioned job chains for different p can be used equivalently for the computation of MDA and MRT according to their description by partitioned job chains from Definition 11.

▶ **Lemma 12.** *For all $p \in \{1, \dots, |\overline{E}| - 1\}$, we have*

$$\sup \{\ell(pc_m^p) \mid m \geq F_p\} = \sup \left\{\ell(pc_m^{p+1}) \,\middle|\, m \geq F_{p+1}\right\}. \tag{7}$$

To prove this lemma, we apply fundamental properties of immediate forward and immediate backward job chains (which are part of the partitioned job chains). In particular:

- If an immediate forward job chain $\vec{c}$ starts at or before another job chain c, then no job of c can be *before* the job of $\vec{c}$ because in the forward chain always the earliest possible job is chosen during construction.
- If an immediate backward job chain $\overleftarrow{c}$ ends at or after another job chain c, then no job of c can be *after* the job of $\overleftarrow{c}$ because in the backward chain always the latest possible job is chosen during construction.

We formalize those two properties in the following lemma that we utilize to prove Lemma 12. It is formulated for an arbitrary cause-effect chain E which can be $\overline{E}$ or a sub-chain of $\overline{E}$.

▶ **Lemma 13.** *Let E be a cause-effect chain in $\mathbb{T}$, let c be a job chain for E, let $\vec{c}$ be an immediate forward job chain for E, and let $\overleftarrow{c}$ be an immediate backward job chain for E.*

1. *If $i \in \{1, \dots, |E|\}$ exists such that $\vec{c}(i) \preccurlyeq c(i)$, then $\vec{c}(j) \preccurlyeq c(j)$ for all $j \in \{i, \dots, |E|\}$.*
2. *If $i \in \{1, \dots, |E|\}$ exists such that $c(i) \preccurlyeq \overleftarrow{c}(i)$, then $c(j) \preccurlyeq \overleftarrow{c}(j)$ for all $j \in \{1, \dots, i\}$.*

The proof for Lemma 13 is provided in the appendix. We now provide the proof of Lemma 12. We split the proof into two steps, showing the $\geq$-relation and the $\leq$-relation, from which the equality directly follows.

To proof idea for the $\geq$-relation is as follows. First, we pick any p-partitioned and the related $(p+1)$-partitioned job chain that have the job of $\overline{E}(p)$ in common. Second, we show (i) that their immediate backward job chains both end at the same job of $\overline{E}(1)$, and (ii) that the immediate forward job chain related to the $p+1$-partitioned job chain ends not later than the one related to the p-partitioned job chain. For instance, in Figure 4, pc_1^3 and pc_1^2 have job $\tau_2(1)$ in common, both immediate backwards job chains end at $\tau_1(1)$, and pc_1^3 ends at $\tau_4(2)$ which is not later than the end of pc_1^2 at $\tau_4(3)$. Hence, $\ell(pc_1^2) \geq \ell(pc_1^3)$.

Proof of Lemma 12, $\geq$-relation. Let $m \geq F_{p+1}$. We denote the jobs of the partitioned job chain pc_m^{p+1} by $pc_m^{p+1} = (J_1, \dots, J_p, J_{p+1}, \tilde{J}_{p+1}, \dots, \tilde{J}_{|\overline{E}|})$. Let $\xi \in \mathbb{N}$ such that J_p is the ξ-th job of $\overline{E}(p)$, i.e., $\overline{E}(p)(\xi) = J_p$. We prove that the length of pc_m^{p+1} is upper bounded by the length of $pc_\xi^p = (J'_1, \dots, J'_p, \tilde{J}'_p, \dots, \tilde{J}'_{|\overline{E}|})$ by showing $\mathrm{we}(\tilde{J}'_{|\overline{E}|}) - \mathrm{re}(J'_1) \geq \mathrm{we}(\tilde{J}_{|\overline{E}|}) - \mathrm{re}(J_1)$ in two substeps: First, we show that J_1 and J'_1 are the same (i.e., $J_1 = J'_1$), and second, that pc_m^{p+1} finishes not later than pc_ξ^p (i.e., $\tilde{J}_{|\overline{E}|} \preccurlyeq \tilde{J}'_{|\overline{E}|}$). Important jobs of this proof are illustrated in Figure 4.

Step 1 ($J_1 = J'_1$): By definition, $J_p = J'_p$. Since $(J_1, \dots, J_p)$ and $(J'_1, \dots, J'_p)$ are immediate backward job chains with the same last entry, they coincide. Hence, $J_1 = J'_1$.

Step 2 ($\tilde{J}_{|\overline{E}|} \preccurlyeq \tilde{J}'_{|\overline{E}|}$): Since $(J_1, \dots, J_{p+1})$ is an immediate backward job chain, this means that $J_p = \overline{E}(p)(\xi)$ is the latest job with write-event no later than the read-event of J_{p+1}. Therefore, the write-event of the subsequent job of the same task, which is $\overline{E}(p)(\xi+1) = \tilde{J}'_p$, must be after the read-event of J_{p+1}, i.e., $\mathrm{re}(J_{p+1}) < \mathrm{we}(\tilde{J}'_p)$.

The job $\tilde{J}_{p+1}$ of $\overline{E}(p+1)$ subsequent to J_{p+1} either has its read-event before $\mathrm{we}(\tilde{J}'_p)$ as well (i.e., $\mathrm{re}(\tilde{J}_{p+1}) < \mathrm{we}(\tilde{J}'_p)$) or is the earliest job of $\overline{E}(p+1)$ with $\mathrm{re}(\tilde{J}_{p+1}) \geq \mathrm{we}(\tilde{J}'_p)$. In both cases $\tilde{J}_{p+1} \preccurlyeq \tilde{J}'_{p+1}$ as $\mathrm{re}(\tilde{J}'_{p+1}) \geq \mathrm{we}(\tilde{J}'_p)$. Since $(\tilde{J}_{p+1}, \dots, \tilde{J}_{|\overline{E}|})$ and $(\tilde{J}'_{p+1}, \dots, \tilde{J}'_{|\overline{E}|})$ are both immediate forward job chains and $\tilde{J}_{p+1} \preccurlyeq \tilde{J}'_{p+1}$, we know $\tilde{J}_{|\overline{E}|} \preccurlyeq \tilde{J}'_{|\overline{E}|}$ by Lemma 13.

Combining Step 1 and 2, we get $\ell(pc_m^{p+1}) = \mathrm{we}(\tilde{J}_{|\overline{E}|}) - \mathrm{re}(J_1) \leq \mathrm{we}(\tilde{J}'_{|\overline{E}|}) - \mathrm{re}(J'_1) = \ell(pc_\xi^p)$. Since $(J_1, \dots, J_{p+1})$ is an immediate backward job chain and $E(p+1)(F_{p+1}) \preccurlyeq J_{p+1}$, by Lemma 13, $E(p)(F_p) \preccurlyeq J_p$ holds as well. As Step 1 shows $J_p = J'_p = E(p)(\xi)$, we conclude that $\xi \geq F_p$. Hence, $\ell(pc_m^{p+1}) \leq \ell(pc_\xi^p) \leq \sup \left\{ \ell(pc_\eta^p) \,\middle|\, \eta \geq F_p \right\}$.

Since $m \geq F_{p+1}$ is arbitrarily chosen, we obtain $\sup \left\{ \ell(pc_m^{p+1}) \,\middle|\, m \geq F_{p+1} \right\} \leq \sup \left\{ \ell(pc_\eta^p) \,\middle|\, \eta \geq F_p \right\}$, i.e., the relation $\geq$ holds for Equation (7) ◀

Similarly, we show the $\leq$-relation by picking any p-partitioned and the related $(p+1)$-partitioned job chain that have the job of $\overline{E}(p+1)$ in common. Second, we show (i) that their immediate forward job chains both end at the same job of $\overline{E}(|\overline{E}|)$, and (ii) that the immediate backward job chain related to the $p+1$-partitioned job chain ends not later than the one related to the p-partitioned job chain. For instance, in Figure 5, pc_1^3 and pc_1^2 have job $\tau_3(2)$ in common, both immediate forward job chains ends $\tau_4(2)$, and pc_1^3 ends at $\tau_1(1)$ which is not later then the end of pc_2^2 at $\tau_1(2)$. Hence, $\ell(pc_2^2) \leq \ell(pc_1^3)$.

Proof of Lemma 12, $\leq$-relation. Let $m \geq F_p$. We denote the jobs of the partitioned job chain pc_m^p by $pc_m^p = (J_1, \dots, J_p, \tilde{J}_p, \tilde{J}_{p+1}, \dots, \tilde{J}_{|\overline{E}|})$. Let $\xi \in \mathbb{N}$ such that $\tilde{J}_{p+1}$ is the ξ-th job of $\overline{E}(p+1)$, i.e., $\overline{E}(p+1)(\xi) = \tilde{J}_{p+1}$.

As an additional step, we must show that $\xi - 1 \geq F_{p+1}$ holds for the previous job of $\overline{E}(p+1)(\xi-1)$. Assume for contradiction that $\xi - 1 < F_{p+1}$. Then $\xi \leq F_{p+1}$. Therefore, $\tilde{J}_{p+1} \preccurlyeq \overline{E}(p+1)(F_{p+1})$ holds. Since $(\overline{E}(1)(F_1), \dots, \overline{E}(|\overline{E}|)(F_{|\overline{E}|}))$ is an immediate backward job chain, by Lemma 13 we obtain that $\tilde{J}_p \preccurlyeq \overline{E}(p)(F_p)$. Furthermore, since $\tilde{J}_p = \overline{E}(p)(m+1)$ by definition of pc_m^p, we obtain $m+1 \leq F_p$, i.e., $m < F_p$ which contradicts that $m \geq F_p$.

Since $\xi - 1 \geq F_{p+1}$ and for F_{p+1} an immediate backward job chain exists, the immediate backward job chain $\overleftarrow{c}_{\xi-1}^{\overline{E}_{p+1}^{first}}$ can be fully constructed and $pc_{\xi-1}^{p+1}$ exists. We now prove that the length of pc^p_m is upper bounded by the length of $pc^{p+1}_{\xi-1} = (J'_1, \dots, J'_{p+1}, \tilde{J}'_{p+1}, \dots, \tilde{J}'_{|\overline{E}|})$, i.e., we show that $\mathrm{we}(\tilde{J}'_{|\overline{E}|}) - \mathrm{re}(J'_1) \geq \mathrm{we}(\tilde{J}_{|\overline{E}|}) - \mathrm{re}(J_1)$. Specifically, we show $\tilde{J}_{|\overline{E}|} = \tilde{J}'_{|\overline{E}|}$ and $J'_1 \preccurlyeq J_1$ in two individual steps. Important jobs of this proof are illustrated in Figure 5.

Step 1 ($\tilde{J}'_{|\overline{E}|} = \tilde{J}_{|\overline{E}|}$): By definition, we have $\tilde{J}'_{p+1} = \overline{E}(p+1)(\xi) = \tilde{J}_{p+1}$. Since the immediate forward job chains $(\tilde{J}'_{p+1}, \dots, \tilde{J}'_{|\overline{E}|})$ and $(\tilde{J}_{p+1}, \dots, \tilde{J}_{|\overline{E}|})$ both have the same job as the first entry, these job chains coincide. In particular, $\tilde{J}'_{|\overline{E}|} = \tilde{J}_{|\overline{E}|}$.

Step 2 ($J'_1 \preccurlyeq J_1$): Since $(\tilde{J}_p, \dots, \tilde{J}_{|\overline{E}|})$ is an immediate forward job chain, the job $\tilde{J}_{p+1} = \overline{E}(p+1)(\xi)$ is the earliest job with read-event no earlier than $\mathrm{we}(\tilde{J}_p)$. Thus, the read-event of $\overline{E}(p+1)(\xi-1) = J'_{p+1}$ must be before the write-event of $\tilde{J}_p$, i.e., $\mathrm{we}(\tilde{J}_p) > \mathrm{re}(J'_{p+1})$. For the job of $\overline{E}(p)$ previous to $\tilde{J}_p$, which is J_p, we either have $\mathrm{we}(J_p) > \mathrm{re}(J'_{p+1})$ as well, or J_p is the latest job of $\overline{E}(p)$ with $\mathrm{we}(J_p) \leq \mathrm{re}(J'_{p+1})$. In both cases $J'_p \preccurlyeq J_p$ because $\mathrm{we}(J'_p) \leq \mathrm{re}(J'_{p+1})$. Since $(J'_1, \dots, J'_p)$ and $(J_1, \dots, J_p)$ are both immediate backward job chains and $J'_p \preccurlyeq J_p$, we have $J'_1 \preccurlyeq J_1$ as well by Lemma 13.

Therefore, we obtain $\ell(pc^p_m) = \mathrm{we}(\tilde{J}_{|\overline{E}|}) - \mathrm{re}(J_1) \leq \mathrm{we}(\tilde{J}'_{|\overline{E}|}) - \mathrm{re}(J'_1) = \ell(pc^{p+1}_{\xi-1})$. We already showed that $\xi - 1 \geq F_{p+1}$, hence $\ell(pc^p_m) \leq \ell(pc^{p+1}_{\xi-1}) \leq \sup\left\{\ell(pc^{p+1}_\eta) \,\middle|\, \eta \geq F_{p+1}\right\}$.

Since $m \geq F_p$ is arbitrarily chosen, the relation $\leq$ holds for Equation (7) as $\sup\{\ell(pc^p_m) \mid m \geq F_p\} \leq \sup\left\{\ell(pc^{p+1}_\eta) \,\middle|\, \eta \geq F_{p+1}\right\}$. ◀

We have shown that p-partitioned job chains for different p can be utilized equivalently. The equivalence of MRT and MDA follows directly by applying Lemma 12 multiple times.

▶ **Theorem 14** (Equivalence). *The maximum reaction time and maximum data age are equivalent, i.e.,*

$$\mathrm{MRT} = \sup\{\ell(pc^p_m) \mid m \geq F_p\} = \mathrm{MDA} \tag{8}$$

for all $p \in \left\{1, \dots, |\overline{E}|\right\}$.

6 Implication in Practice

The provided result is much stronger than the observation made in the AUTOSAR timing specification, namely, that "*without over- and undersampling, age and reaction are the same*" [1, Section 7.2, p. 149]. Specifically, in Section 5 we show that MRT and MDA are *always* the same. Since we only assume that each task releases a countably infinite number of jobs, this equivalence holds for a variety of scenarios. This covers, for example:

- Systems with over- and undersampling
- Implicit communication or communication by logical execution time (LET)
- Fixed-priority or Dynamic-priority schedulers
- Tasks scheduled by the Robot Operating System 2 (ROS2), as demonstrated in Section 9
- Synchronized or asynchronized distributed systems
- Periodic or sporadic task systems

This implies that for many industrial applications MRT and MDA can be used and analyzed equivalently. In particular, any guarantee for one metric holds for the other one as well. Furthermore, end-to-end timing specification in industrial systems only needs to consider one instead of two different latencies. This eases the verification of timing constraints.

The model of p-partitioned job chains, introduced in Section 4, and the description of MRT and MDA independent of p can be used to improve end-to-end latency analysis. For instance, a significant speedup of the latency calculation of periodic tasks communicating via LET can be obtained by choosing the right p, as demonstrated in Section 8.

7 Extension to Alternative Definitions

In this section we discuss how the results from Section 5 can be used for alternative definitions of MRT and MDA. In particular, we discuss *reduced* end-to-end latencies used by Dürr et al. [9] and Feiertag et al. [10] in Section 7.1, and we discuss a definition based on *valid* job chains used Günzel et al. [13] in Section 7.2.

7.1 Maximum Reduced Data Age and Reaction Time

The reaction time definition by, e.g., Dürr et al. [9] and Feiertag et al. [10] coincides with the maximum reaction time specified in Section 3. However, their definition of maximum data age does not include the additional time between the last job in the chain and the actuation (at time z'); they use MRDA instead. In this section, we discuss how our results can be transferred to this alternative definition of the end-to-end latencies.

We start by defining the reduced length of a job chain providing a maximum reduced reaction time (MRRT), analogously to the MRDA definition by Günzel et al. [13]. The MRT assumes a similar scenario.

▶ **Definition 15** (Reduced length)**.** For an immediate forward or immediate backward augmented job chain $(z, J_1, \dots, J_{|\overline{E}|}, z')$, we define the reduced length ℓ^* as the length of the intermediate job sequence, i.e., $\ell^*(z, J_1, \dots, J_{|\overline{E}|}, z') = \ell(J_1, \dots, J_{|\overline{E}|})$.

This MRDA definition assumes that the actuation is directly triggered by the write event of the last task in the chain, while the MDA definition assumes that the actuation is based on the calculated value but not triggered directly.

▶ **Definition 16** (Maximum reduced data age)**.** The maximum reduced data age (MRDA) is: $\mathrm{MRDA} = \sup_{m \geq F_{|\overline{E}|}+1} \ell^*(\overleftarrow{ac}_m)$.

Similarly, one can distinguish between the MRT, assuming a scenario where an external cause happens at any time and is registered at the read-event of the first task in the chain, and the MRRT, where read-event and cause happen at the same time.

▶ **Definition 17** (Maximum reduced reaction time (MRRT))**.** We define the maximum reduced reaction time (MRRT) as: $\mathrm{MRRT} = \sup_{m \geq F_1} \ell^*(\vec{ac}_m)$.

Since the reduced length can be computed from the length of a cause-effect chain by removing the additional time at the beginning or at the end, there is the following relation between MRT (MDA, respectively) and MRRT (MRDA, respectively).

▶ **Theorem 18.** *Let $\rho^+(\tau)$ be the* maximal *time between two subsequent read-events of a task τ, and let $\rho^-(\tau)$ be the* minimal *time between two subsequent read-events of a task τ. Then the following two bounds hold:*

$$\mathrm{MRT} - \mathrm{MRRT} \in [\rho^-(\overline{E}(1)), \rho^+(\overline{E}(1))] \tag{9}$$

$$\mathrm{MDA} - \mathrm{MRDA} \in [\rho^-(\overline{E}(|\overline{E}|)), \rho^+(\overline{E}(|\overline{E}|))] \tag{10}$$

We omit the proof, as the relation directly follows from the definition. The above relations can be utilized to transfer the statements from Theorem 14 to MRDA and MRRT. We note that for periodic or sporadic systems MRT = MDA > MRDA holds, i.e., MRT ≠ MRDA in all scenarios. In particular, AUTOSAR considers MDA since they observe that "*without over- and undersampling, age and reaction are the same*" [1, Section 7.2, p. 149].

7.2 MRT and MDA Based On Valid Chains

In this section, we extend the equivalence results from the Section 5 to MRT and MDA as defined by Günzel et al. [13] based on *valid* immediate forward and *valid* immediate backward augmented job chains. In particular, valid job chains are defined in [13] as follows.

▶ **Definition 19** (Valid [13]). Let $(z, J_1, \dots, J_{|\overline{E}|}, z')$ be an immediate forward or immediate backward augmented job chain. Moreover, let $j \in \mathbb{N}^+$ such that $z = \mathrm{re}(\overline{E}(1)(j))$, i.e., z is at the read-event of the j-th job of $\overline{E}(1)$. Then $(z, J_1, \dots, J_{|\overline{E}|}, z')$ is called *valid*, if and only if $j \geq v^{\overline{E}}$, where $v^{\overline{E}}$ be the smallest integer such that $\mathrm{re}(\overline{E}(1)(v^{\overline{E}}+1)) > \max_{i \in \{1,\dots,|\overline{E}|\}} \mathrm{re}(\overline{E}(i)(1))$.

We denote the MRT and MDA based on valid chains by $\mathrm{MRT}^{\mathrm{V}}$ and $\mathrm{MDA}^{\mathrm{V}}$, respectively. For the MRT based on valid chains, instead of $m \geq F_1$ all $m \in \mathbb{N}^+$ such that $\vec{ac}_m$ is valid are considered. Similarly, for the MDA based on valid chains, instead of $m + 1 \geq F_{|E|}$ all $m \in \mathbb{N}^+$ such that $\overleftarrow{ac}_m$ exists and is valid are considered.

$$\mathrm{MRT}^{\mathrm{V}} = \sup \left\{ \ell(\vec{ac}_m) \,\middle|\, m \in \mathbb{N}^+, \vec{ac}_m \text{ valid} \right\} \tag{11}$$

$$\mathrm{MDA}^{\mathrm{V}} = \sup \left\{ \ell(\overleftarrow{ac}_m) \,\middle|\, m \in \mathbb{N}^+, \overleftarrow{ac}_m \text{ exists and valid} \right\} \tag{12}$$

In the following we discuss the equivalence of $\mathrm{MRT}^{\mathrm{V}}$ and $\mathrm{MDA}^{\mathrm{V}}$. To that end, we first define $V_1, \dots, V_E$ through an immediate backward job chain as follows.

▶ **Definition 20.** Let $V \in \mathbb{N}^+$ such that $\overleftarrow{c}_V^{\overline{E}}$ is the first immediate backward job chain with $\overline{E}(1)(v^{\overline{E}}) \preccurlyeq \overleftarrow{c}_V^{\overline{E}}(1)$, with $v^{\overline{E}}$ defined as in Definition 19. We denote by $V_1, \dots, V_{|\overline{E}|} \in \mathbb{N}^+$ the job number of each job in $\overleftarrow{c}_V^{\overline{E}}$, i.e., $\overleftarrow{c}_V^{\overline{E}} = (\overline{E}(1)(V_1), \dots, \overline{E}(|\overline{E}|)(V_{|\overline{E}|}))$.

With an analogous proof as in Lemma 12 we obtain that for all $p \in \left\{1, \dots, |\overline{E}| - 1\right\}$, we have $\sup \{\ell(pc_m^p) \,|\, m \geq V_p\} = \sup \left\{\ell(pc_m^{p+1}) \,\middle|\, m \geq V_{p+1}\right\}$. As a result,

$$\sup_{m \geq V_1} \ell(\vec{ac}_m) = \sup_{m \geq V_1} \ell(pc_m^1) = \sup_{m \geq V_p} \ell(pc_m^p) = \sup_{m \geq V_{|\overline{E}|}} \ell(pc_m^{|\overline{E}|}) = \sup_{m \geq V_{|\overline{E}|}+1} \ell(\overleftarrow{ac}_m). \tag{13}$$

By the definition of $V_{|\overline{E}|}$ it can be shown that $\mathrm{MDA}^{\mathrm{V}} = \sup_{m \geq V_{|\overline{E}|}+1} \ell(\overleftarrow{ac}_m)$. For $\mathrm{MRT}^{\mathrm{V}}$ one additional immediate augmented forward job chain has to be included to account for the m in $\left\{v^{\overline{E}}, \dots, V_1\right\}$. In particular, $\mathrm{MRT}^{\mathrm{V}} = \max(\ell(\vec{ac}_{v^{\overline{E}}}), \sup_{m \geq V_1} \ell(\vec{ac}_m))$. We conclude that for the alternative definition of MRT and MDA based on valid chains, an equivalence can be obtained up to the first immediate forward augmented job chain, i.e.,

$$\mathrm{MRT}^{\mathrm{V}} = \sup(\ell(\vec{ac}_{v^{\overline{E}}}), \mathrm{MDA}^{\mathrm{V}}), \tag{14}$$

where $\vec{ac}_{v^{\overline{E}}}$ is the first valid immediate forward augmented job chain. The proof of this equivalence is provided in the appendix.

8 Application: Analysis of Chains with Periodic LET Tasks

In this section, we experimentally evaluated the correctness of the relations from Theorem 14 and Theorem 18 by comparing the results to the state of the art by Günzel et al. [13]. In particular, we applied them to periodic tasks scheduled on one electronic control unit (ECU) and communicating under Logical Execution Time (LET) to see if the correct latency values were calculated. Furthermore, we examined whether our approach resulted in faster computation time than the state of the art.

Considered approaches. We compared our approach to Günzel et al. [13].

Approach by Günzel et al. [13] (G21): Let $\Phi(\overline{E}) := \max\left\{\phi_{\overline{E}(i)} \,\middle|\, i = 1, \dots, |\overline{E}|\right\}$ be the maximal phase of the tasks in $\overline{E}$ and let $H(\overline{E}) := \mathrm{lcm}(\left\{T_{\overline{E}(i)} \,\middle|\, i = 1, \dots, |\overline{E}|\right\})$ be the hyperperiod of all tasks in $\overline{E}$. Then under LET the immediate forward and immediate backward augmented job chains with external activity during the interval $[\Phi(\overline{E})+H(\overline{E}), \Phi(\overline{E})+2H(\overline{E}))$ repeat each hyperperiod. In particular only those immediate forward and immediate backward augmented job chains with external activity no later than $\Phi(\overline{E}) + 2H(\overline{E})$ have to be constructed and compared to obtain the MDA, MRDA, MRT and MRRT.

Our approach (Our): According to Theorem 14 it is sufficient to examine the p-partitioned job chains for one $p \in \{1, \dots, |\overline{E}|\}$ to compute the maximum data age. To minimize the calculation time, we considered the task $E(p)$ with the highest period. By this choice of p, the number of constructed job chains can be reduced significantly. MRT and MDA are computed by Theorem 14. By applying Theorem 18, the MRDA and the MRRT can be computed from the MDA and MRT, respectively, since $\rho^-(\tau) = T_\tau = \rho^+(\tau)$ for all LET tasks τ.

Task set generation. We randomly generated 10000 task sets $\mathbb{T}$, each with a random cardinality of $n_{\mathbb{T}} \in [50, 100] \cap \mathbb{N}$. To evaluate the approaches with tasks similar to a real-world application, we generated tasks according to the Automotive Benchmark by Kramer et al. [19]. In particular, for each task τ we drew a period T_τ from the set $\{1, 2, 5, 10, 20, 50, 100, 200, 1000\}$ according to the related share[2] of these periods in [19, Table III, IV and V]. We assumed implicit deadlines, i.e., the relative deadline D_τ was set to the period T_τ, and the phase ϕ_τ was set to 0 for all tasks. Since the read- and write-events under logical execution time are independent from the execution behavior, no execution time of the task was synthesized.

Cause-effect chain generation. For each task set $\mathbb{T}$, we generated a cause-effect chain E according to Kramer et al. [19, Section IV-E]. In particular, we applied the following steps:

1. The number of involved activation patterns $P_E \in \{1, 2, 3\}$, i.e., the number of unique periods of the tasks in the cause-effect chain E, was drawn according to the distribution in [19, Table VI].
2. A set S_E of P_E unique periods was uniformly drawn from the periods in $\mathbb{T}$.
3. For each period in S_E, we drew 2 to 5 tasks at random (without replacement) according to the distribution in [19, Table VII] from the tasks in $\mathbb{T}$ with the respective period. The cause-effect chain E consists of these tasks in random order.

If there were not sufficient tasks with required period in Step 3, we discarded the set and randomly drew another task set until a cause-effect chain was successfully created.

Evaluation results. For each cause-effect chain, we applied **(G21)** to compute MDA, MRT, MRDA, and MRRT. Additionally, we applied **(Our)** to calculate the same values using Theorem 14 and Theorem 18. For the runtime measurements, we use a machine equipped with 2x AMD EPYC 7742 running Linux, i.e., in total 256 threads with 2.25GHz and 256GB RAM. Each measurement runs on one independent thread.

Observation 1: For all 10000 cause-effect chains all latency values obtained by **(Our)** coincide with the corresponding values obtained by **(G21)**. In particular, the equivalence of MRT and MDA holds for all scenarios, even with over- and undersampling.

Observation 2: Our results can reduce the required time for computing the end-to-end latencies significantly. Specifically, Figure 6 depicts the time ratio, defined by $\frac{\text{time\_our}}{\text{time\_g21}}$, where

[2] The sum of probabilities in [19] is only 85%. The remaining 15% are reserved for angle-synchronous tasks that we do not consider here. Therefore, all share values were divided by 0.85.

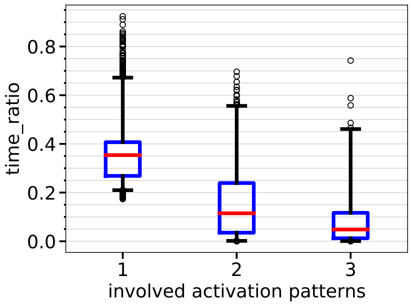

Figure 6 The time_ratio for different number of involved activation patterns. Red line depicts median, blue box 50% of the data, and the whiskers all data except the highest and the lowest 1%.

involved patterns	1	2	3
number values	8109	1436	455
min speed-up	1.08	1.44	1.35
median speed-up	2.83	8.70	20.71
mean speed-up	3.03	48.13	96.27
max speed-up	5.76	1476.02	1623.50

Table 3 Speed-up for different number of activation patterns.

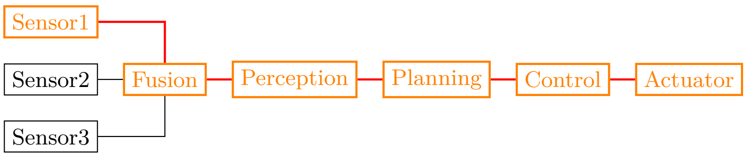

Figure 7 ROS2 basic navigation system.

time_our and time_g21 is the time needed by **(Our)** and **(G21)**, respectively, to derive all four latency values (considering the minimal runtime over 1000 runs for each of the 10000 cause effect chains). On the x-axis the number of different activation patterns, i.e., the number of different periods in the cause-effect chain, is shown. Additionally, Table 3 shows the speed-up, defined as $\frac{\text{time\_g21}}{\text{time\_our}}$.

We observe that **(Our)** reduces the required time significantly compared to **(G21)**. In particular, when all tasks in the cause-effect chain have the same period, then the number of computed chains, and hence the runtime, is halved on average. When there is more than one activation pattern, then **(Our)** reduces the number of constructed chains by choosing p such that $E(p)$ is the largest period, and a much larger speed-up is observed such cases.

9 Case Study: ROS2

In this section, we validate Theorem 18 and Theorem 14 considering a basic navigation system, as shown in Figure 7, and apply the scheduling mechanism of the Robot Operating System 2 [23] (ROS2) on a single ECU. The navigation system includes three sensors, whose data is combined and processed for the perception of the environment, planning the route, controlling the vehicle, and sending the output to the vehicle interfaces via an actuator.

A system in ROS2 consists of nodes and topics. Each node represents one component of the system, which can communicate with other nodes via topics, that implement a publish-subscribe architecture. Nodes are represented by tasks and each execution of a node can be considered as the execution of a job. The nodes follow an implicit communication policy, i.e., the read-event of a node is at its start and the write-event of a node is at its finish. Nodes are either time-triggered and event-triggered, i.e., some tasks have an aperiodic behavior. ROS2 has a non-standard custom scheduler that executes tasks instances under a round-robin scheduling approach. Specifically, the scheduler repeatedly collects at most one job of each task for the round, after which it executes them according to their priority. However, the

Component	Type	Period	WCET
Sensor	time-trig.	$100ms$	$10ms$
Fusion	time-trig.	$100ms$	$100ms$
Fusion	event-trig.	-	$5ms$
Perception	event-trig.	-	$200ms$
Planning	event-trig.	-	$100ms$
Control	event-trig.	-	$50ms$
Actuator	event-trig.	-	$5ms$

Table 4 Periods and WCET of the nodes.

MRT	4.0
MDA	4.0
MRRT	3.5
MRDA	3.5
$\ell(\vec{ac}_{m'})$	3.5

$\rho^+(\overline{E}(1))$	0.5
$\rho^-(\overline{E}(1))$	0.5
$\rho^+(\overline{E}(\lvert\overline{E}\rvert))$	0.5
$\rho^-(\overline{E}(\lvert\overline{E}\rvert))$	0.5

Table 5 Measurements of the ROS2 system.

results from this work can still be applied as the basic assumptions (R1) and (R2) for the read- and write-events in Section 2 are met. We do not provide details of the ROS2 scheduling approach here, as we only focus on the timing behavior of the resulting system.

The system depicted in Figure 7 has three sensors whose data is combined in the fusion node. The perception, planning, and control node process the data and supply the actuator node with instructions. For the simulation of the ROS2 scheduling behavior we assume that all jobs have a fixed execution time. Table 4 gives an overview of the WCET of each component and of the period of the time-triggered components.

For the first chain $\overline{E} = (\text{Sensor1}, \dots, \text{Actuator})$ marked in orange in Figure 7, we measured the MRT, MDA, MRRT, and MRDA by determining the longest immediate forward and immediate backward augmented job chains. The measured values are summarized in Table 5, showing that Theorem 14 and Theorem 18 hold.

10 Related Work

The first end-to-end latency analysis for maximum reaction time of a cause-effect chain was developed by Davare et al. [8] in 2007. Later, in 2009, the end-to-end semantics were further extended by Feiertag et al. [10]. Specifically, a formal definition of maximum data age and maximum reaction time is proposed by *timed paths* to distinguish the semantics for the data propagation delay. Based on forward reachability and overwriting of data, *first-to-first* and *last-to-last* data propagation semantics are proposed, coinciding with the maximum reaction time and the maximum data age, respectively. In addition, they are also known as First-in-First-out (FIFO) and Last-in-Last-out (LILO), respectively [21, 22].

The semantics of maximum reaction time and maximum data age have been widely studied in the literature. Several techniques have been derived through formal verification and compiler to verify the data propagation delay for periodic task systems. Rajeev et al. [24] develop a model-checking based technique to compute the end-to-end delay under both semantics for periodic tasks. Forget et al. [28] propose a language-based approach to verify end-to-end delay at the model level. Klaus et al. [16] extend the design flow of the Real-Time Systems Compiler (RTSC) to take data propagation delay into account but only focus on the maximum data age.

The analysis approaches for end-to-end delays can be classified into two paradigm: active approaches [5, 7, 12, 25], where the release of jobs in the chain is actively controlled depending on the production of data to ensure the data coherence for read and write, and passive approaches [2, 3, 6, 8–11, 13, 17, 24, 26], that focus on how the data is produced and consumed among the jobs of the recurrent tasks in the cause-effect chain provided that the release of jobs of subsequent tasks is independent from the production of data. The approaches proposed in this work are passive ones, but the equivalence of both end-to-end semantics holds in general; that is, the equivalence is not limited to the passive paradigm.

Some recent results focus on end-to-end analysis under the Logical Execution Time (LET) communication model, proposed by Kirsch and Sokolova [15]. Martinez et al. [20] analyzed its effect on the end-to-end delay. Becker et al. [4] further extend their previous analysis from implicit communication to LET. Kordon and Tang [18] develop a framework to calculate the maximum data age under the LET communication based on a given general task dependency graph. Hamann et al. [14] propose a model transformation method to increase the expressiveness of current timing analysis. They address heterogeneous communication semantics including LET. Our analysis is also applicable for LET communication.

11 Conclusion

Over the last years different timing metrics for the end-to-end latency of cause-effect chains were considered. Of particular interest are the maximum reaction time and the maximum data age. AUTOSAR observed that both metrics coincide if no over- and undersampling occurs. However, it was assumed that both metrics differ for the general case.

In this paper, we show the equivalence of maximum reaction time and maximum data age. To this end, we introduce p-partitioned job chains and show that both timing metrics can be determined by constructing p-partitioned job chains for the same p.

The impact of the equivalence is twofold:

1. Analytical literature results for one metric can immediately be used for the other.
2. The choice of $p \in \{2, \dots, |\overline{E}| - 1\}$ allows novel analysis approaches and a reduced runtime of current approaches by reducing the number of necessary job chains under consideration.

We demonstrate the impact of this work by considering the case of periodic tasks communicating under LET. In particular, we show that the end-to-end analysis can be performed up to 1600 times faster in this scenario.

The equivalence holds for almost any scheduling mechanism and even for task systems which do not adhere to the typical periodic or sporadic task model. We support this statement with a case study based on ROS2.

Appendix: Proof of Lemma 13

We first restate Lemma 13.

▶ **Lemma 13.** *Let E be a cause-effect chain for the task set $\mathbb{T}$. Moreover, let c be a job chain for E, let $\vec{c}$ be an immediate forward job chain for E, and let $\overleftarrow{c}$ be an immediate backward job chain for E.*

1. *If there exists $i \in \{1, \dots, |E|\}$ such that $\vec{c}(i) \preccurlyeq c(i)$, then $\vec{c}(j) \preccurlyeq c(j)$ for all $j \in \{i, \dots, |E|\}$.*
2. *If there exists $i \in \{1, \dots, |E|\}$ such that $c(i) \preccurlyeq \overleftarrow{c}(i)$, then $c(j) \preccurlyeq \overleftarrow{c}(j)$ for all $j \in \{1, \dots, i\}$.*

Proof. We prove 1) by induction over $j = i, \dots, |E|$.
Initial case: For $j = i$, $\vec{c}(j) \preccurlyeq c(j)$ by assumption.
Induction step: If $\vec{c}(j) \preccurlyeq c(j)$ for $j \in \{i, \dots, |E| - 1\}$, then this means that the write-event of the job $\vec{c}(j)$ is no later than the write-event of the job $c(j)$. Since the read-event of the job $c(j+1)$ is no earlier than the write-event of $c(j)$ by definition of a job chain, the read-event of $c(j+1)$ is also no earlier than the write-event of $\vec{c}(j)$. Since $\vec{c}(j+1)$ is the earliest job with read-event no earlier than the write-event of $\vec{c}(j)$, we conclude that $\vec{c}(j+1) \preccurlyeq c(j+1)$.

We prove 2) by induction over $j = i, \dots, 1$.
Initial case: For $j = i$, $c(j) \preccurlyeq \overleftarrow{c}(j)$ by assumption.
Induction step: If $c(j) \preccurlyeq \overleftarrow{c}(j)$ for $j \in \{i, \dots, 2\}$, then this means that the read-event of the job $c(j)$ is no later than the read-event of the job $c(j)$. Since the write-event of the job $c(j-1)$ is no later than the read-event of $c(j)$ by definition of a job chain, the write-event of $c(j-1)$ is also no later than the read-event of $\overleftarrow{c}(j)$. Since $\overleftarrow{c}(j-1)$ is the latest job with write-event no later than the read-event of $\overleftarrow{c}(j)$, we conclude that $c(j-1) \preccurlyeq \overleftarrow{c}(j-1)$. ◀

Appendix: Proof of Extension to Valid Chains

In Section 7.2 it is shown that $\sup_{m\geq V_1} \ell(\overleftarrow{ac}_m) = \sup_{m \geq V_{|\overline{E}|}+1} \ell(\overleftarrow{ac}_m)$. It is left to show how this relates to $\mathrm{MDA}^{\mathrm{V}}$ and $\mathrm{MRT}^{\mathrm{V}}$. We start with $\mathrm{MDA}^{\mathrm{V}}$.

▶ **Lemma 14.** *We have* $\mathrm{MDA}^{\mathrm{V}} = \sup_{m \geq V_{|\overline{E}|}+1} \ell(\overleftarrow{ac}_m)$.

Proof. By definition in Equation (12), $\mathrm{MDA}^{\mathrm{V}} = \sup\{\ell(\overleftarrow{ac}_m) \mid m \in \mathbb{N}^+, \overleftarrow{ac}_m \text{ exists and valid}\}$. We divide the proof in two steps: $\mathrm{MDA}^{\mathrm{V}} \geq \sup_{m \geq V_{|\overline{E}|}+1} \ell(\overleftarrow{ac}_m)$ and $\mathrm{MDA}^{\mathrm{V}} \leq \sup_{m \geq V_{|\overline{E}|}+1} \ell(\overleftarrow{ac}_m)$.

Step 1 ($\mathrm{MDA}^{\mathrm{V}} \geq \sup_{m \geq V_{|\overline{E}|}+1} \ell(\overleftarrow{ac}_m)$): We know that $\overleftarrow{ac}_{V_{|\overline{E}|}+1}$ is composed of $z' = \mathrm{we}(\overline{E}(|\overline{E}|)(V_{|\overline{E}|}+1))$, the immediate backward job chain $\overleftarrow{c}^{\overline{E}}_{V_{|\overline{E}|}}$, and $z \geq \mathrm{re}(\overline{E}(1)(v^{\overline{E}}))$. Therefore, $\overleftarrow{ac}_{V_{|\overline{E}|}+1}$ is valid. Consequently, all $\overleftarrow{ac}_m$ with $m \geq V_{|\overline{E}|}+1$ are valid and $\mathrm{MDA}^{\mathrm{V}} \geq \sup_{m \geq V_{|\overline{E}|}+1} \ell(\overleftarrow{ac}_m)$ holds.

Step 2 ($\mathrm{MDA}^{\mathrm{V}} \leq \sup_{m \geq V_{|\overline{E}|}+1} \ell(\overleftarrow{ac}_m)$): We prove this step by contradiction and assume that there exists a valid $\overleftarrow{ac}_m = (z, J_1, \dots, J_{|\overline{E}|}, z')$ with $m < V_{|\overline{E}|}+1$. In that case $J_{|\overline{E}|}$ must be a job *before* $\overline{E}(V_{|\overline{E}|})$. Since by Definition 20, $\overleftarrow{c}^{\overline{E}}_{V_{|\overline{E}|}}$ is the first chain with $\overline{E}(1)(v^{\overline{E}}) \preccurlyeq \overleftarrow{c}^{\overline{E}}_V(1)$, the job J_1 must be earlier than $\overline{E}(1)(v^{\overline{E}})$. Hence, $\overleftarrow{ac}_m = (z, J_1, \dots, J_{|\overline{E}|}, z')$ is not valid, which is a contradiction. ◀

For the MRT, we may need to account for additional immediate forward augmented job chains before V_1. By definition, valid immediate forward augmented job chains $\overrightarrow{ac}_m$ with $v^{\overline{E}} \leq m < V_1$ include the immediate forward job chain $\overrightarrow{c}^{\overline{E}}_{m+1}$ with $v^{|\overline{E}|} < m+1 \leq V_1$. The following lemma examines those immediate forward job chains further, showing that all of them end at the same job, which means that the longest $\overrightarrow{ac}^1_m$ with $m < F_1$ is the first one.

▶ **Lemma 15.** *Let $m \in \mathbb{N}^+$ with $v^{\overline{E}} \leq m \leq V_1$. The last entry of the m-th immediate forward job chain $\overrightarrow{c}^{\overline{E}}_m$ is the job where $\overleftarrow{c}^{\overline{E}}_V$ ends, i.e., $\overrightarrow{c}^{\overline{E}}_m(|\overline{E}|) = \overline{E}(|\overline{E}|)(V_{|\overline{E}|})$.*

Proof. Let $\xi \in \mathbb{N}^+$ such that $\overrightarrow{c}^{\overline{E}}_m(|\overline{E}|) = \overline{E}(|\overline{E}|)(\xi)$. In the following, we show that $\xi = V$.

Since there exist job chains with last job $\overline{E}(|\overline{E}|)(\xi)$, this means that $\overleftarrow{c}^{\overline{E}}_\xi$ exists as well because for the backward construction of that chain in each step a job can be chosen. Moreover, since $\overleftarrow{c}^{\overline{E}}_\xi$ is immediate backward and $\overrightarrow{c}^{\overline{E}}_m$ is a job chain with the same job as last entry, by Lemma 13 $\overrightarrow{c}^{\overline{E}}_m(1) \preccurlyeq \overleftarrow{c}^{\overline{E}}_\xi(1)$. Hence, $E(1)(v^{|\overline{E}|}) \preccurlyeq \overleftarrow{c}^{\overline{E}}_\xi(1)$. Since $\overleftarrow{c}^{\overline{E}}_V$ is the earliest immediate backward job chain with this property, we obtain $\xi \geq V$.

Since $\overrightarrow{c}^{\overline{E}}_m(1) \preccurlyeq \overline{E}(|\overline{E}|)(V_1) = \overleftarrow{c}^{\overline{E}}_V(1)$ and $\overrightarrow{c}^{\overline{E}}_m$ is immediate forward, we can apply Lemma 13 and obtain $\overrightarrow{c}^{\overline{E}}_m(|\overline{E}|) \preccurlyeq \overleftarrow{c}^{\overline{E}}_V(|\overline{E}|) = \overline{E}(|\overline{E}|)(V_{|\overline{E}|})$, i.e., $\xi \leq V$. ◀

Since the last job of every $\overrightarrow{c}^{\overline{E}}_m$ with $v^{|\overline{E}|} \leq m \leq V_1$ is $\overline{E}(|\overline{E}|)(V_{|\overline{E}|})$, the last job of every $\overrightarrow{ac}_m$ with $v^{|\overline{E}|} \leq m < V_1$ is $\overline{E}(|\overline{E}|)(V_{|\overline{E}|})$ as well. Furthermore, since the $\overrightarrow{ac}^1_m$ with the lowest

m has the earliest first job, it is the longest one, i.e., to obtain $\mathrm{MRT}^{\mathrm{V}}$ only the first valid immediate forward augmented job chain is considered in addition to $\sup_{m \geq V_1} \ell(\vec{ac}_m)$.

▶ **Lemma 16.** *We have* $\mathrm{MRT}^{\mathrm{V}} = \max(\ell(\vec{ac}_{v^{\overline{E}}}), \sup_{m \geq V_1} \ell(\vec{ac}_m))$, *where* $\vec{ac}_{v^{\overline{E}}}$ *is the first valid immediate forward augmented job chain.*

Proof. By definition, $V_1 \geq v^{\overline{E}}$. Moreover, 1-partitioned job chains $\vec{ac}_m$ are valid if and only if $m \geq v^{\overline{E}}$. Therefore, $\mathrm{MRT}^{\mathrm{V}}$ is the maximum of $\sup\{\ell(\vec{ac}_m) \,|\, m \geq V_1\}$ and $\max\left\{\ell(\vec{ac}_m) \,\middle|\, V_1 > m \geq v^{\overline{E}}\right\}$. Moreover, by Lemma 15 $\max\left\{\ell(\vec{ac}_m) \,\middle|\, V_1 > m \geq v^{\overline{E}}\right\}$ is either 0 if $V_1 = v^{\overline{E}}$, or it is the length of $\vec{ac}_{v^{\overline{E}}}$. ◀

We summarize the relation between $\mathrm{MDA}^{\mathrm{V}}$ and $\mathrm{MRT}^{\mathrm{V}}$.

▶ **Proposition 17.** *The maximum reaction time* $\mathrm{MRT}^{\mathrm{V}} = \sup(\ell(\vec{ac}_{v^{\overline{E}}}), \mathrm{MDA}^{\mathrm{V}})$, *where* $\vec{ac}_{v^{\overline{E}}}$ *is the first valid immediate forward augmented job chain.*

The $\mathrm{MRT}^{\mathrm{V}}$ can also be formulated as $\mathrm{MRT}^{\mathrm{V}} = \max(\ell(pc^1_{v^{\overline{E}}}), \mathrm{MDA}^{\mathrm{V}})$, where $pc^1_{v^{\overline{E}}}$ is the first valid immediate forward augmented job chain, since by Lemma 10 $\ell(pc^1_{v^{\overline{E}}}) = \ell(\vec{ac}_{v^{\overline{E}}})$. Moreover, $\mathrm{MDA}^{\mathrm{V}}$ can be computed through p-partitioned job chains as $\mathrm{MDA}^{\mathrm{V}} = \sup_{m \geq V_p} \ell(pc^p_m)$ for any $p \in \left\{1, \ldots, |\overline{E}|\right\}$.

References

1 AUTOSAR. Specification of timing extensions (AUTOSAR CP R21-11). `https://www.autosar.org/fileadmin/user_upload/standards/classic/21-11/AUTOSAR_TPS_TimingExtensions.pdf`, 2021. Accessed: 2022-10-18.

2 Matthias Becker, Dakshina Dasari, Saad Mubeen, Moris Behnam, and Thomas Nolte. Mechaniser-a timing analysis and synthesis tool for multi-rate effect chains with job-level dependencies. In *Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS)*, 2016.

3 Matthias Becker, Dakshina Dasari, Saad Mubeen, Moris Behnam, and Thomas Nolte. Synthesizing job-level dependencies for automotive multi-rate effect chains. In *International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 159–169, 2016. `doi:10.1109/RTCSA.2016.41`.

4 Matthias Becker, Dakshina Dasari, Saad Mubeen, Moris Behnam, and Thomas Nolte. End-to-end timing analysis of cause-effect chains in automotive embedded systems. *J. Syst. Archit.*, 80(C):104–113, oct 2017. `doi:10.1016/j.sysarc.2017.09.004`.

5 Matthias Becker, Saad Mubeen, Dakshina Dasari, Moris Behnam, and Thomas Nolte. A generic framework facilitating early analysis of data propagation delays in multi-rate systems. In *International Conference on Embedded and Real-Time Computing Systems and Applications (RTCSA)*, pages 1–11, 2017. `doi:10.1109/RTCSA.2017.8046323`.

6 Albert Benveniste, Paul Caspi, Paul Le Guernic, Hervé Marchand, Jean-Pierre Talpin, and Stavros Tripakis. A protocol for loosely time-triggered architectures. In *EMSOFT*, pages 252–265, 2002. `doi:10.1007/3-540-45828-X\_19`.

7 Hyunjong Choi, Mohsen Karimi, and Hyoseung Kim. Chain-based fixed-priority scheduling of loosely-dependent tasks. In *International Conference on Computer Design (ICCD)*. IEEE, 2020.

8 Abhijit Davare, Qi Zhu, Marco Di Natale, Claudio Pinello, Sri Kanajan, and Alberto L. Sangiovanni-Vincentelli. Period optimization for hard real-time distributed automotive systems. In *Design Automation Conference, DAC*, pages 278–283, 2007. `doi:10.1145/1278480.1278553`.

9 Marco Dürr, Georg von der Brüggen, Kuan-Hsun Chen, and Jian-Jia Chen. End-to-end timing analysis of sporadic cause-effect chains in distributed systems. *ACM Trans. Embedded Comput. Syst. (Special Issue for CASES)*, 18(5s):58:1–58:24, 2019. doi:10.1145/3358181.

10 Nico Feiertag, Kai Richter, Johan Nordlander, and Jan 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*, 2009.

11 Julien Forget, Frédéric Boniol, and Claire Pagetti. Verifying end-to-end real-time constraints on multi-periodic models. In *ETFA*, pages 1–8, 2017. doi:10.1109/ETFA.2017.8247612.

12 Alain Girault, Christophe Prevot, Sophie Quinton, Rafik Henia, and Nicolas Sordon. Improving and estimating the precision of bounds on the worst-case latency of task chains. *IEEE Trans. on CAD of Integrated Circuits and Systems, (Special Issue for EMSOFT)*, 37(11):2578–2589, 2018. doi:10.1109/TCAD.2018.2861016.

13 Mario Günzel, Kuan-Hsun Chen, Niklas Ueter, Georg von der Brüggen, Marco Dürr, and Jian-Jia Chen. Timing analysis of asynchronized distributed cause-effect chains. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 40–52, 2021. doi:10.1109/RTAS52030.2021.00012.

14 Arne Hamann, Dakshina Dasari, Simon Kramer, Michael Pressler, and Falk Wurst. Communication centric design in complex automotive embedded systems. In *Euromicro Conference on Real-Time Systems, ECRTS*, pages 10:1–10:20, 2017.

15 Christoph M. Kirsch and Ana Sokolova. The logical execution time paradigm. In *Advances in Real-Time Systems*, pages 103–120. Springer, 2012. doi:10.1007/978-3-642-24349-3_5.

16 Tobias Klaus, Florian Franzmann, Matthias Becker, and Peter Ulbrich. Data propagation delay constraints in multi-rate systems: Deadlines vs. job-level dependencies. In *Proceedings of the 26th International Conference on Real-Time Networks and Systems*, pages 93–103. ACM, 2018.

17 Tomasz Kloda, Antoine Bertout, and Yves Sorel. Latency analysis for data chains of real-time periodic tasks. In *IEEE International Conference on Emerging Technologies and Factory Automation, ETFA*, pages 360–367, 2018. doi:10.1109/ETFA.2018.8502498.

18 Alix Munier Kordon and Ning Tang. Evaluation of the Age Latency of a Real-Time Communicating System Using the LET Paradigm. In Marcus Völp, editor, *32nd Euromicro Conference on Real-Time Systems (ECRTS 2020)*, volume 165 of *Leibniz International Proceedings in Informatics (LIPIcs)*, pages 20:1–20:20, Dagstuhl, Germany, 2020. Schloss Dagstuhl–Leibniz-Zentrum für Informatik. URL: https://drops.dagstuhl.de/opus/volltexte/2020/12383, doi:10.4230/LIPIcs.ECRTS.2020.20.

19 Simon Kramer, Dirk Ziegenbein, and Arne Hamann. Real world automotive benchmarks for free. In *International Workshop on Analysis Tools and Methodologies for Embedded and Real-time Systems (WATERS)*, 2015.

20 Jorge Martinez, Ignacio Sañudo, and Marko Bertogna. End-to-end latency characterization of task communication models for automotive systems. *Real-Time Syst.*, 56(3):315–347, jul 2020. doi:10.1007/s11241-020-09350-3.

21 Saad Mubeen, Jukka Mäki-Turja, and Mikael Sjödin. Implementation of end-to-end latency analysis for component-based multi-rate real-time systems in rubus-ice. In *Factory Communication Systems (WFCS), 2012 9th IEEE International Workshop on*, pages 165–168. IEEE, 2012.

22 Saad Mubeen, Jukka Mäki-Turja, and Mikael Sjödin. Translating end-to-end timing requirements to timing analysis model in component-based distributed real-time systems. *SIGBED Review*, 9(4):17–20, 2012. doi:10.1145/2452537.2452539.

23 Open Robotics. Ros 2 documentation: Foxy, May 2022. https://docs.ros.org/en/foxy.

24 AC Rajeev, Swarup Mohalik, Manoj G Dixit, Devesh B Chokshi, and S Ramesh. Schedulability and end-to-end latency in distributed ecu networks: formal modeling and precise estimation. In *International Conference on Embedded Software*, pages 129–138, 2010.

25 Johannes Schlatow and Rolf Ernst. Response-time analysis for task chains in communicating threads. In *IEEE Real-Time and Embedded Technology and Applications Symposium (RTAS)*, pages 245–254, 2016. doi:10.1109/RTAS.2016.7461359.

26 Johannes Schlatow, Mischa Möstl, Sebastian Tobuschat, Tasuku Ishigooka, and Rolf Ernst. Data-age analysis and optimisation for cause-effect chains in automotive control systems. In *IEEE International Symposium on Industrial Embedded Systems (SIES)*, pages 1–9, 2018.

27 Harun Teper, Mario Günzel, Niklas Ueter, Georg von der Brüggen, and Jian-Jia Chen. End-to-end timing analysis in ros2. In *43rd IEEE Real-Time Systems Symposium (RTSS)*, 2022.

28 Rémy Wyss, Frédéric Boniol, Claire Pagetti, and Julien Forget. End-to-end latency computation in a multi-periodic design. In *Proceedings of the 28th Annual ACM Symposium on Applied Computing, SAC*, pages 1682–1687, 2013. doi:10.1145/2480362.2480678.