A Compositional Approach for Schedulability Analysis of Distributed Avionics Systems

Pujie Han Zhengjun Zhai
School of Computer Science and Engineering
Northwestern Polytechnical University
Xi'an, China
{hanpujie,zhaizjun}@mail.nwpu.edu.cn

Brian Nielsen Ulrik Nyman
Department of Computer Science
Aalborg University
Aalborg, Denmark
{bnielsen,ulrik}@cs.aau.dk

This work presents a compositional approach for schedulability analysis of Distributed Integrated Modular Avionics (DIMA) systems that consist of spatially distributed ARINC-653 modules connected by a unified AFDX network. We model a DIMA system as a set of stopwatch automata in UPPAAL to verify its schedulability by model checking. However, direct model checking is infeasible due to the large state space. Therefore, we introduce the compositional analysis that checks each partition including its communication environment individually. Based on a notion of message interfaces, a number of message sender automata are built to model the environment for a partition. We define a timed selection simulation relation, which supports the construction of composite message interfaces. By using assume-guarantee reasoning, we ensure that each task meets the deadline and that communication constraints are also fulfilled globally. The approach is applied to the analysis of a concrete DIMA system.

1 Introduction

The architecture of Distributed Integrated Modular Avionics (DIMA) has been successfully applied to the aviation industry. A DIMA system installs standardized computer modules in spatially distributed locations[19] that are connected by a unified bus system[3] such as an AFDX network. Avionics applications residing on the modules run in ARINC-653[1] compliant operating systems. The generic distributed structure of DIMA significantly improves performance and availability as well as reduces development and maintenance costs, while it also dramatically increases the complexity of schedulability analysis. A schedulable DIMA system should fulfil not only the temporal requirements of real-time tasks in each ARINC-653 module but also communication constraints among the distributed nodes. As a result, the system integrators need to consider both computation and communication when analyzing the schedulability of DIMA architecture.

Currently, model checking approaches have been increasingly developed in the schedulability analysis of complex real-time systems. However, we found no studies that analyzed the schedulability of distributed avionics systems as a whole including the network by model checking. The related research isolates computation modules from their underlying network, thereby considering these nodes as independent hierarchical scheduling systems or investigating the network in isolation, which possibly leads to pessimistic results. There have been works using model-checking to analyze the temporal behavior of individual avionics modules in various formal models such as Coloured Petri Nets (CPN)[10], preemptive Time Petri Nets (pTPN)[5], Timed Automata (TA)[2], and StopWatch Automata (SWA)[16, 8], and verify schedulability properties via state space exploration. Unfortunately, when being applied to concrete avionics systems, all of them suffer from an inevitable problem of state space explosion. For hierarchical scheduling systems, some studies[6, 18, 4] exploit the inherent temporal isolation of ARINC-653

Submitted to:
MeTRiD 2018

© P. Han, Z. Zhai B. Nielsen & U. Nyman
This work is licensed under the
Creative Commons Attribution License.

partitions[1] and analyze each partition separately, but they ignore the behavior of the underlying network or the interactions among partitions. Thus these methods are not applicable to DIMA environments in which multiple distributed ARINC-653 partitions communicate through a shared network to perform an avionics function together.

In this paper, we present a compositional approach for schedulability analysis of DIMA systems that are modeled as UPPAAL SWA, i.e. the TA extended with stopwatches. Compared with the clocks in TA, stopwatches can be blocked and resumed at any location and thus are effective in modeling task preemption. We decompose the system in such a way that we can check each ARINC-653 partition *including* a model of its communication environment individually and then assemble the local results together to derive conclusions about the schedulability of an entire system. Thereby, we verify a number of smaller, simpler, abstract systems rather than directly verifying a larger, more complex, concrete system including the details about all the partitions and the network. The main contributions of this paper are summarized as follows:

- *A compositional approach* performs assume-guarantee reasoning[12] to reduce the complexity of symbolic model-checking in the schedulability analysis of DIMA systems.
- *An abstraction relation*, timed selection simulation relation, allows users to create a set of abstract models that collectively describe the external behavior of a concrete model, thereby simplifying the abstraction in assume-guarantee reasoning.
- A notion of *message interfaces* decouples the communication dependencies between partitions. By composing any partition with its related message interfaces and verifying safety properties of the composition, we can conclude that these properties are still preserved at the global level.

The rest of the paper is organized as follows. Section 2 gives the necessary formal notions. The UPPAAL modeling of DIMA systems is presented in section 3. Section 4 gives the concept of timed selection simulation and its properties. In section 5, we detail the compositional analysis approach. Section 6 shows an experiment on a concrete DIMA system, and section 7 finally concludes.

2 Preliminaries

In this section, we present formal definitions including SWA with an input/output extension and its semantic object Timed I/O Transition Systems(TIOTSs)[9].

Suppose that C is a finite set of clocks and V is a finite set of integer variables. A *valuation* $u(x)$ with $x \in C \cup V$ denotes a mapping from C to $\mathbf{R}_{\geq 0}$ and from V to $\mathbf{N}$. Let $LC(C,V)$ be the set of linear constraints. A *guard* $g \in LC(C,V)$ is a linear constraint which is defined as a finite conjunction of atomic formulae in the form of $c \sim n$, $c - c' \sim n$ or $v \sim n$ with $c, c' \in C, v \in V, n \in \mathbf{N}$, and $\sim \in \{>,<,=\}$. Given any valuation u, we change the values of clocks and integer variables using an *update* operation $r(u) \in 2^R$ in the form of $c = 0$ or $v = n$ where $c \in C, v \in V$ and $n \in \mathbf{N}$, and R represents the set of update operations. In addition, we define an *action* set Σ. All the actions can be subsumed under two sets of unicast actions Σ^u and broadcast actions Σ^b. By contrast, $\tau \notin \Sigma$ denotes an internal action and $\Sigma^\tau = \Sigma \cup \{\tau\}$.

Definition 1 (Stopwatch Automaton[7]). *A stopwatch automaton is a tuple* $\langle Loc, l_0, C, V, E, \Sigma, Inv, drv \rangle$ *where* Loc *is a finite set of locations,* $l_0 \in Loc$ *is the initial location,* C *is a finite set of clocks,* V *is a finite set of integer variables,* $E \subseteq Loc \times LC(C,V) \times \Sigma^\tau \times 2^R \times Loc$ *is a set of edges,* $\Sigma = I \oplus O$ *is a finite set of actions divided into inputs(I) and outputs(O),* Inv *is a mapping* $Loc \to LC(C,V)$*, and* drv *is a mapping* $Loc \times C \to \{0,1\}$.

From a syntactic viewpoint, SWA belongs to the class of TA extended with drv, which can prevent part of the clocks from changing in specified locations semantically. We now shift the focus to the semantic object TIOTS of SWA.

In a TIOTS, there are two types of transitions: delay and action transitions. We use the set $D = \{\varepsilon(d) | d \in \mathbf{R}_{\geq 0}\}$ to denote the delay, and refer to the 0-delay $\varepsilon(0)$ as $\mathbf{0}$.

Definition 2 (Timed I/O Transition System). *A timed I/O transition system is a tuple $\mathscr{T} = \langle S, s_0, \Sigma, \rightarrow \rangle$ where S is an infinite set of states, s_0 is the initial state, $\Sigma = I \oplus O$ is a finite set of actions divided into inputs(I) and outputs(O), $I \cap O \subseteq \Sigma^u$, and $\rightarrow \subseteq S \times \Sigma^\tau \cup D \times S$ is a transition relation. $s \xrightarrow{a} s'$ represents $(s, a, s') \in \rightarrow$, which has the properties of time determinism, time reflexivity, and time additivity[9].*

For any SWA, a state is defined as a pair $\langle l, u \rangle$ where l is a location and u is a valuation over clocks and integer variables. On the basis of TIOTSs, the operational semantics of SWA is defined as follows.

Definition 3. *The operational semantics of a stopwatch automaton $A = \langle Loc, l_0, C, V, E, \Sigma, Inv, d \rangle$ is a timed I/O transition system $\mathscr{T}^A = \langle S, s_0, \Sigma, \rightarrow \rangle$ where S is the set of states of A, $s_0 = \langle l_0, u_0 \rangle$ is the initial state of A, Σ is the same set of actions as A, and $\rightarrow$ is the transition relation defined by*

- $\langle l, u \rangle \xrightarrow{a} \langle l', u' \rangle$ *iff* $\exists \langle l, g, a, r, l' \rangle \in E \ \ (u \models g \ \wedge \ u' = r(u) \ \wedge \ u' \models Inv(l'))$
- $\langle l, u \rangle \xrightarrow{\varepsilon(d)} \langle l', u' \rangle$ *iff* $l = l' \wedge (\forall v \in V \ \ u'(v) = u(v)) \ \wedge \ (\forall c \in C \ (drv(l, c) = 0 \Rightarrow u'(c) = u(c))) \ \wedge \ (\forall c \in C \ (drv(l, c) = 1 \Rightarrow u'(c) = u(c) + d)) \ \wedge \ u' \models Inv(l')$.

For any transition $s \xrightarrow{a} s'$, two symbols $a?$ and $a!$ denote the action a belonging to input I and output O respectively. Given $a \in \Sigma$, $s \xrightarrow{a}$ iff $\exists s' \in S$, s.t. $s \xrightarrow{a} s'$. $\xrightarrow{\tau}^*$ or $\xRightarrow{\mathbf{0}}$ denotes the reflexive and transitive closure of $\xrightarrow{\tau}$. $s \xRightarrow{\varepsilon(d)} s'$ iff $s \xrightarrow{\varepsilon(d)} s'$, or $\exists s_1, s_2, \ldots, s_n \in S$, s.t. $s \xrightarrow{\alpha_0} s_1 \xrightarrow{\alpha_1} s_2 \xrightarrow{\alpha_2} \cdots \xrightarrow{\alpha_{n-1}} s_n \xrightarrow{\alpha_n} s'$ and $\forall i \in \{0, \ldots, n\}$, s.t. $\alpha_i = \tau$ or $\alpha_i \in D$ and $d = \sum\{d_i | \alpha_i = \varepsilon(d_i)\}$.

The definition of parallel composition $\|$ of TIOTSs is similar to that in [9]. Given two TIOTSs $\mathscr{T}_i = \langle S_i, s_{i,0}, \Sigma_i, \rightarrow_i \rangle, i \in \{1, 2\}$, they are *compatible* iff they satisfy the following conditions:

- *(Unique output)* $O_1 \cap O_2 = \varnothing$.
- *(Deterministic-pair unicast)* $I_1 \cap I_2 \cap \Sigma^u = \varnothing$.

Note that broadcast actions in the composition of TIOTSs are *input-enabled*: $\forall s \in S_i \ \forall a \in I_i \cap \Sigma^b \ s \xrightarrow{a}$.

Definition 4 (Parallel Composition). *Suppose two timed I/O transition systems $\mathscr{T}_1 = \langle S_1, s_{1,0}, \Sigma_1, \rightarrow_1 \rangle$ and $\mathscr{T}_2 = \langle S_2, s_{2,0}, \Sigma_2, \rightarrow_2 \rangle$ are compatible. The parallel composition $\mathscr{T}_1 || \mathscr{T}_2$ is the timed I/O transition system $\langle S, s_0, \Sigma, \rightarrow \rangle$ where $S = S_1 \times S_2$, $s_0 = \langle s_{1,0}, s_{2,0} \rangle$, $\Sigma = I_{1||2} \oplus O_{1||2}$, $I_{1||2} = (I_1 \setminus (O_2 \cap \Sigma^b)) \cup (I_2 \setminus (O_1 \cap \Sigma^b))$, $O_{1||2} = O_1 \cup O_2$, and $\rightarrow$ is the largest relation generated by the following rules:*

- *INDEP-L:* $\dfrac{s_1 \xrightarrow{a} s_1' \quad a \in \{\tau\} \cup \Sigma_1 \setminus \Sigma_2}{\langle s_1, s_2 \rangle \xrightarrow{a} \langle s_1', s_2 \rangle}$ *INDEP-R:* $\dfrac{s_2 \xrightarrow{a} s_2' \quad a \in \{\tau\} \cup \Sigma_2 \setminus \Sigma_1}{\langle s_1, s_2 \rangle \xrightarrow{a} \langle s_1, s_2' \rangle}$
- *DELAY:* $\dfrac{s_1 \xrightarrow{\varepsilon(d)} s_1' \quad s_2 \xrightarrow{\varepsilon(d)} s_2' \quad d \in \mathbf{R}_{\geq 0}}{\langle s_1, s_2 \rangle \xrightarrow{\varepsilon(d)} \langle s_1', s_2' \rangle}$
- *SYNC-IN:* $\dfrac{s_1 \xrightarrow{a} s_1' \quad s_2 \xrightarrow{a} s_2' \quad a \in I_{1||2}}{\langle s_1, s_2 \rangle \xrightarrow{a} \langle s_1', s_2' \rangle}$
- *SYNC-BIO:* $\dfrac{s_1 \xrightarrow{a} s_1' \quad s_2 \xrightarrow{a} s_2' \quad a \in (I_1 \cap O_2) \cup (O_1 \cap I_2) \cap \Sigma^b}{\langle s_1, s_2 \rangle \xrightarrow{a} \langle s_1', s_2' \rangle}$

- *SYNC-UIO:* $\dfrac{s_1 \xrightarrow{a} s_1' \quad s_2 \xrightarrow{a} s_2' \quad a \in I_{1||2} \cap O_{1||2}}{\langle s_1, s_2 \rangle \xrightarrow{\tau} \langle s_1', s_2' \rangle}$.

We use Ω to denote the set of TA and SWA in our modeling framework. For any $A, B \in \Omega$, we define the composite model $C = A||B$ iff their TIOTSs satisfy $\mathcal{T}^C = \mathcal{T}^A || \mathcal{T}^B$.

3 Avionics System Modeling

We focus on a generic DIMA architecture including a set of ARINC-653 modules connected by an AFDX network, as shown in Fig.1. There is a three-layer structure in the DIMA system that consists of scheduling, task, and communication layers.

The *scheduling layer* is defined as the scheduling facilities for generic computation resources of a DIMA system, where standardized computer modules execute concurrent application tasks in partitioned operating systems. In this operating system, partitions are scheduled by a Time Division Multiplexing (TDM) scheduler and each partition also has its local scheduling policy, preemptive Fixed Priority (FP), to manage the internal tasks[1]. The scheduling layer is modeled as two TA templates `PartitionSupply` and `TaskScheduler` in UPPAAL[1]. The `PartitionSupply` depicted in Fig.2 provides the service of TDM partitioning for a particular partition `pid`. The `TaskScheduler` implementing FP scheduling allocates processor time to the task layer only when the partition is active.

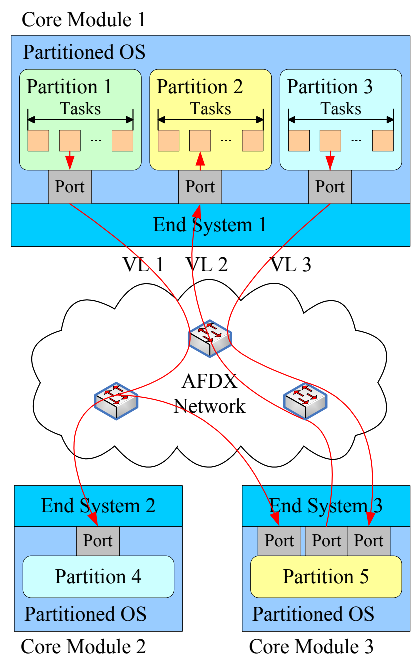


Figure 1: An Example of DIMA systems

The *task layer* contains all the application tasks executing avionics functions. A task is regarded as the smallest scheduling unit, each of which runs concurrently with other tasks in the same partition. The execution of a task is modelled as a sequence of commands that are either computing for a duration, locking/unlocking a resource, or sending/receiving a message. We consider two task types: *periodic tasks* and *sporadic tasks*. A periodic task has a fixed release period, while a sporadic task is characterized by a minimum separation between consecutive jobs. The task layer is instantiated from two SWA templates `PeriodicTask` and `SporadicTask` in UPPAAL. Since the tasks in a partition are scheduled by a task scheduler, we use a set of binary channels as scheduling actions to communicate between task models and `TaskScheduler`.

The *communication layer* carries out inter-partition communication over a common AFDX network. The AFDX protocol stack realized by an End System(ES) interfaces with the task layer through ARINC-653 ports. Based on the AFDX protocol structure, the communication layer is further divided into UDP/IP layer and Virtual Link layer, where a Virtual Link (VL) ensures an upper bound on end-to-end delay. In UPPAAL, the UDP/IP layer is divided into two TA templates `IPTx` and `IPRx`, which calculate the latency of the UDP/IP layer in a transmitting ES and a receiving ES respectively. Similarly, two TA templates `VLinkTx` and `VLinkRx` model the delay of a VL in opposite directions.

From a global view of the system, its schedulability is also affected by the communication layer. According to the ARINC-653 standard[1], there are two types of ARINC-653 ports, sampling ports

[1]Models available at `http://eptcs.web.cse.unsw.edu.au/paper.cgi?MARSVPT2018:2`

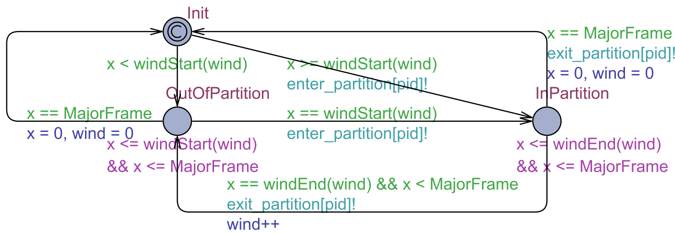


Figure 2: The UPPAAL Template of an ARINC-653 Partition Scheduler

and queuing ports. A sampling port can accommodate at most a single message that remains until it is overwritten by a new message. A refresh period is defined for each sampling port. This attribute provides a specified arrival rate of messages, regardless of the rate of receiving requests from tasks. In contrast, a queuing port is allowed to buffer multiple messages in a message queue with a fixed capacity. However, the operating system is not responsible for handling overflow from the message queue.

In this paper, we verify the following three typical schedulability properties:

- All the tasks meet their deadlines in each partition.
- The refresh period of any sampling port is guaranteed.
- The overflow from any queuing ports must be avoided.

The schedulability of an avionics system is described and verified as a safety property of the above TA/SWA models. We add a set *Err* of *error locations* to the templates. Once schedulability is violated, the related model will lead itself to one of the error locations immediately. Thus, the schedulability is replaced with this safety property φ:

$$\texttt{A[]}\ \neg(\bigvee_{loc \in Err} loc), \tag{1}$$

which belongs to a simplified subset of TCTL used in UPPAAL.

However, since the verification algorithm inside UPPAAL for SWA introduces a slight over-approximation[7][2], UPPAAL may sometimes give the verification result “Maybe satisfied” or “May not be satisfied”. To further refine the result in this case we manually analyse the possible counter example using UPPAAL’s concrete simulator to determine if the system is unschedulable. Alternatively, the statistical model-checking (SMC) engine could be invoked to attempt an automatic falsification. In our experiences, the result only appears when the system is on the very borderline of being schedulable.

4 Timed Selection Simulation

We propose a notion of timed selection simulation relation to support assume-guarantee reasoning. Compared with some other abstraction relations like timed simulation[15] and timed ready simulation[14], timed selection simulation only abstracts a selected subset of actions from the concrete model. Applying timed selection simulation to the abstraction of a concrete system, one can pay attention to part of the system, individually model the behavior of each component, and thereby obtain a composite abstract model rather than a monolithic one.

Considering the semantic object $\mathscr{T}^A$ of an automaton $A \in \Omega$, we denote the *error states* of $\mathscr{T}^A$ by the set $\mathscr{E} = \{\langle l,u \rangle | l \in Err\}$ where Err is the error-location set of A. Thus, for any TIOTS $\mathscr{T} = \langle S, s_0, \Sigma, \rightarrow \rangle$,

[2]Exact reachability for SWA with more than 3 stopwatches is known to be undecidable[7].

its error states are defined as a set $\mathscr{E} \subseteq S$, and the following function $g: S \to \{true, false\}$ indicates whether a state $s \in S$ has violated schedulability properties:

$$g(s) = \begin{cases} true & \text{if } s \in \mathscr{E} \\ false & \text{if } s \notin \mathscr{E}. \end{cases} \quad (2)$$

Given two compatible TIOTSs $\mathscr{T}_i, i \in \{1,2\}$ with the error-state set $\mathscr{E}_i$, their composition $\mathscr{T}_1 \| \mathscr{T}_2$ has the error-state set $\mathscr{E}_{\mathscr{T}_1 \| \mathscr{T}_2} = \{\langle s_1, s_2\rangle | s_1 \in \mathscr{E}_1 \vee s_2 \in \mathscr{E}_2\}$ and the function $g(\langle s_1, s_2\rangle) = g(s_1) \vee g(s_2)$.

Based on the function $g(s)$, the formal definition of timed selection simulation is given as follows.

Definition 5 (Timed Selection Simulation)**.** *Let $\mathscr{T}_1 = \langle S_1, s_{1,0}, \Sigma_1, \to_1\rangle$ and $\mathscr{T}_2 = \langle S_2, s_{2,0}, \Sigma_2, \to_2\rangle$ be two timed I/O transition systems with $\Sigma_2 \subseteq \Sigma_1$. Let R be a relation from S_1 to S_2. We call R a timed selection simulation from $\mathscr{T}_1$ to $\mathscr{T}_2$, written $\mathscr{T}_1 \preceq \mathscr{T}_2$ via R, provided $(s_{1,0}, s_{2,0}) \in R$ and for all $(s_1, s_2) \in R$, $g(s_1) = g(s_2)$ and*

1. *if $s_1 \xrightarrow{a?} s_1'$ for some $s_1' \in S_1$, $a \in \Sigma_2$, then $\exists s_2' \in S_2$ such that $s_2 \xRightarrow{a?} s_2'$ and $(s_1', s_2') \in R$*
2. *if $s_1 \xrightarrow{a!} s_1'$ for some $s_1' \in S_1$, $a \in \Sigma_2$, then $\exists s_2' \in S_2$ such that $s_2 \xRightarrow{a!} s_2'$ and $(s_1', s_2') \in R$*
3. *if $s_1 \xrightarrow{a} s_1'$ for some $s_1' \in S_1$, $a \in (\Sigma_1 \setminus \Sigma_2) \cup \{\tau\}$, then $\exists s_2' \in S_2$ such that $s_2 \xRightarrow{\mathbf{0}} s_2'$ and $(s_1', s_2') \in R$*
4. *if $s_1 \xrightarrow{\varepsilon(d)} s_1'$ for some $s_1' \in S_1$, $d > 0$, then $\exists s_2' \in S_2$ such that $s_2 \xRightarrow{\varepsilon(d)} s_2'$ and $(s_1', s_2') \in R$.*

Definition 6. *Let $A_i, i \in \{1,2\}$ be stopwatch automata. We say that $A_1 \preceq A_2$, if and only if their corresponding timed I/O transition systems $\mathscr{T}_i$ satisfy $\mathscr{T}_1 \preceq \mathscr{T}_2$.*

We now give some necessary properties of timed selection simulation.

Theorem 1. *Timed selection simulation $\preceq$ is a preorder.*

For any automaton $A \in \Omega$, by construction, the reachability of its error locations is equivalent to that of the error states in the corresponding TIOTS $\mathscr{T}^A$. Hence the following theorem shows that timed selection simulation can preserve the satisfaction of the safety properties in the form of Eq.(1).

Theorem 2 (Property preservation)**.** *Let $\mathscr{T}_i, i \in \{1,2\}$ be timed I/O transition systems and $\mathscr{E}_i$ be the set of error states of $\mathscr{T}_i$. Given a safety property $\varphi : \neg reach(\mathscr{E}_i)$ that any error states are not reachable, if $\mathscr{T}_1 \preceq \mathscr{T}_2$ and $\mathscr{T}_2 \models \varphi$, then $\mathscr{T}_1 \models \varphi$.*

Theorem 3 (Abstraction compositionality)**.** *Let $\mathscr{T}_i, i \in \{1,2,3\}$ be timed I/O transition systems. If $\mathscr{T}_1 \preceq \mathscr{T}_2$, $\mathscr{T}_1 \preceq \mathscr{T}_3$, and $\mathscr{T}_2$ and $\mathscr{T}_3$ are compatible, then $\mathscr{T}_1 \preceq \mathscr{T}_2 \| \mathscr{T}_3$.*

Theorem 4 (Compositionality)**.** *Let $\mathscr{T}_i = \langle S_i, s_{i,0}, \Sigma_i, \to_i\rangle$, $i \in \{1,2,3,4\}$ be timed I/O transition systems. Suppose $\mathscr{T}_1 \| \mathscr{T}_3$ and $\mathscr{T}_2 \| \mathscr{T}_4$ are the parallel compositions of compatible timed I/O transition systems. If (1) $\mathscr{T}_1 \preceq \mathscr{T}_2, \mathscr{T}_3 \preceq \mathscr{T}_4$, and (2) $O_1 \cap I_4 \subseteq \Sigma_2 \subseteq \Sigma^b, I_2 \cap O_3 \subseteq \Sigma_4 \subseteq \Sigma^b$, then $\mathscr{T}_1 \| \mathscr{T}_3 \preceq \mathscr{T}_2 \| \mathscr{T}_4$.*

5 Compositional Analysis

We apply assume-guarantee reasoning to the schedulability analysis, and describe the schedulability goal as a safety property φ (Eq.(1)). As shown in Fig.3, our compositional analysis is comprised of the following four steps:

1. *Decomposition:* The system is first decomposed into a set of communicating partitions modeled by TA and SWA. The global property φ is also divided into several local properties, each of which belongs to one partition.

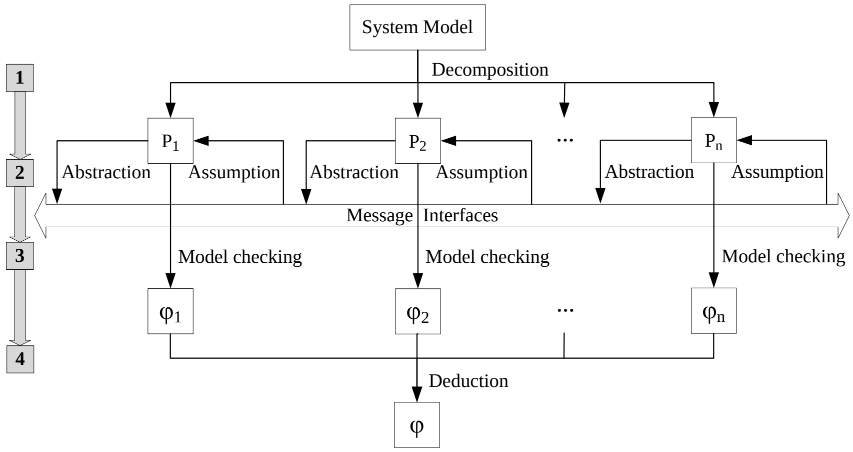


Figure 3: Compositional Analysis Procedure

2. *Construction of message interfaces:* We define message interfaces as the assumption and abstraction of the communication environment for each partition. In general, the templates of message interfaces should be built manually by the engineers.
3. *Model checking:* The local properties under the assumptions and the abstraction relations are verified by model checking.
4. *Deduction:* From the assume-guarantee rules, we finally derive the global property by combining all the local results.

The procedure can be performed automatically except for the first construction of message interfaces. We assume that a task never blocks while communicating with other partitions, which is commonly used in avionics systems[11, 6]. Otherwise a loop of communication dependency will cause circular reasoning, because the assumptions of a partition might be based on its own state recursively.

5.1 Decomposition

Assume that there are n constituent partitions in a system. Let $P_i, i \in \{1,2,\dots,n\}$ be the SWA composite model of a partition. Let Err_i be the error-location set of P_i. The safety property φ_i: $\texttt{A[]}\ \neg(\bigvee_{loc \in Err_i} loc)$ denotes the schedulability of P_i. The global property φ is therefore written as $\varphi_1 \wedge \varphi_2 \wedge \cdots \wedge \varphi_n$, and the goal of our schedulability analysis is expressed as the verification problem:

$$P_1 \| P_2 \| \cdots \| P_n \models \varphi \tag{3}$$

that can be further divided into n satisfaction relations:

$$P_1 \| P_2 \| \cdots \| P_n \models \varphi_i,\ i \in \{1,2,\dots,n\}. \tag{4}$$

Since the error-location set Err_i is only allowed to be manipulated by P_i, we check each partition model P_i independently for the corresponding *local property* φ_i instead of the original verification problem with a large and complex system. However, the communication environment of P_i, which denotes the behavior that P_i receives messages from other partitions, may affect the satisfaction of the schedulability property φ_i. Hence when performing the verification for partition P_i, one needs to give the *assumptions* of its communication environment and verifies the local property φ_i under these assumptions.

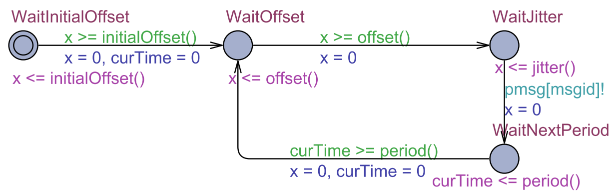


Figure 4: An Example of a Message Interface

5.2 Construction of message interfaces

A set of TA models is created to describe the message-sending behavior of a partition. Each of the TA is called a *message interface* of this partition and associated with a particular message type. Suppose there are a number of messages sent from partition P_j to another partition P_i and their corresponding message interfaces make up a composite TA model $A_{i,j}$. When we analyze P_i in the compositional way, it should be safe for $A_{i,j}$ to replace P_j. Hence, we say that a message interface of P_j is an *abstraction* of P_j.

Our abstraction of the message delivery between a partition and its underlying network is modelled using broadcast synchronization. A broadcast action represents a specific message types. Let $\Sigma_i = I_i \oplus O_i$ be the action set of a composite model for any partition P_i. An action $a_k \in I_i \cap \Sigma^b$(resp. $a_k \in O_i \cap \Sigma^b$) denotes that P_i receives(resp. sends) messages with the type msg_k from(resp. to) other partition(s). The symbol $j \triangleright i$ represents the condition that there exists a partition P_j sending messages to P_i via an action set $O_{j \rightharpoonup i} \subseteq I_i \cap O_j$.

Definition 7 (Message Interface)**.** *Let O_i be the output action set of a stopwatch automaton $P_i \in \Omega$. For any output action $a_k \in O_i \cap \Sigma^b$, the timed automaton A_i^k with an action set $\Sigma_i^k = O_i^k = \{a_k\}$ is a message interface of P_i if and only if there exists a timed selection simulation relation $\preceq$ on Ω such that*

$$P_i \preceq A_i^k. \tag{5}$$

We build the templates of message interfaces in accordance with the characteristics of message-sending actions. In practice, the structure of an interface can be designed straightforwardly from the task specification. The template in Fig.4 shows a message interface that sends messages periodically via the action array `pmsg`. Then we make an automatized binary search for the interface's parameters such as `offset` in the template and meanwhile check the satisfaction of timed selection simulation relation.

The message interfaces can serve as the assumptions of the communication environment of a partition. The composition $A_{i,j}$ of the message interfaces A_j^k for all $a_k \in O_{j \rightharpoonup i}$ provides P_i with a "complete" abstraction of P_j, which models the behavior of all the output actions from P_j to P_i. According to the abstraction compositionality (Theorem 3) of the preorder $\preceq$, we have

$$P_j \preceq A_{i,j}. \tag{6}$$

Considering all the partitions except P_i in the system, we describe the communication environment of P_i as the composite model $\|_{j=1, j\neq i}^{n} A_{i,j}$.

5.3 Model checking

In the third step, the local property φ_i of P_i under assumption $\|_{j=1, j\neq i}^{n} A_{i,j}$ can be verified by model checking. We denote these n subproblems by

$$P_i \parallel (\Big\|_{j=1, j\neq i}^{n} A_{i,j}) \models \varphi_i \quad i \in \{1, 2, \ldots, n\}. \tag{7}$$

Normally, $A_{i,j}$ in Eq.(7) has a much smaller model size than its corresponding partition model P_j in Eq.(4). Thus, the compositional approach allows us to verify a simpler abstract partition model instead of a complex concrete system model including the details about all the partitions.

In addition, we capture the computation time of each task as an interval between a best-case and worst-case execution time. When analyzing the schedulability of a partition, the model-checker explores all scheduling decisions that can be made in such an interval, and hence also examines possible cases of scheduling timing anomalies[17].

5.4 Deduction

We derive the global property φ by combining n local results in the last step. For any schedulable system, each property φ_i should be concluded from the satisfaction of Eq.(7) under assumptions and all the abstraction relations of Eq.(6). According to the compositionality (Theorem 4) and property preservation (Theorem 2) of timed selection simulation, we have the following assume-guarantee rule:

$$\frac{\begin{array}{c}\bigwedge_{\{j|j \triangleright i\}} P_j \preceq A_{i,j} \\ P_i \parallel (\big\|_{j=1, j\neq i}^{n} A_{i,j}) \models \varphi_i\end{array}}{P_1 \| P_2 \| \cdots \| P_n \models \varphi_i} \tag{8}$$

Note that this assume-guarantee rule only provides a sufficient schedulability condition, for abstract message interfaces might slightly over-approximate the external behavior of a partition.

A simplified DIMA system exemplifies the reasoning procedure. In the example, the system model is decomposed into three partitions $P_i, i \in \{1,2,3\}$. We divide the global property φ into three local properties $\varphi_i, i \in \{1,2,3\}$. Accordingly, the goal of the verification problem is to check

$$P_1 \| P_2 \| P_3 \models \varphi_1 \wedge \varphi_2 \wedge \varphi_3. \tag{9}$$

From Eq.(4), this problem can be replaced with three subproblems:

$$P_1 \| P_2 \| P_3 \models \varphi_i, i \in \{1,2,3\}. \tag{10}$$

Without loss of generality, we take the verification of φ_1 for example to show how the model-checking and deduction are carried out in the following steps.

Assume that P_2 sends P_1 two types of messages, msg_1 and msg_2, via two actions a_1 and a_2 respectively, and P_3 sends P_1 only a msg_3 with action a_3. We create one message interface $A_j^k, j \in \{2,3\}$(like Eq.(5)) for each message type $msg_k(k \in \{1,2,3\})$ received by P_1 in the system. The abstraction relations from Eq.(5) can be expressed as

$$P_2 \preceq A_2^1,\ P_2 \preceq A_2^2,\ P_3 \preceq A_3^3. \tag{11}$$

From abstraction compositionality of the preorder $\preceq$, we can obtain

$$P_2 \preceq A_2^1 \| A_2^2,\ P_3 \preceq A_3^3. \tag{12}$$

Then, from reflexivity and compositionality of the preorder $\preceq$, the composite model of the system satisfies

$$P_1 \| P_2 \| P_3 \preceq P_1 \| A_2^1 \| A_2^2 \| A_3^3. \tag{13}$$

Note that when we apply the compositionality to checking a partition P_i, any output actions sent to P_i will never be removed in abstraction relations (Eq.(12)), which satisfies the condition (2) of theorem 4.

With Eq.(13), we have from property preservation of the abstraction relation $\preceq$ that *if*

$$P_1 \| A_2^1 \| A_2^2 \| A_3^3 \models \varphi_1, \text{ then} \tag{14}$$

$$P_1 \| P_2 \| P_3 \models \varphi_1. \tag{15}$$

Since Eq.(15) covering all three partitions in the system has a higher complexity than Eq.(14), the techniques of model checking can be adopted to verify the simpler problem Eq.(14) instead of the original goal Eq.(15). The same steps will be repeated for local properties φ_2 and φ_3.

Consequently, we conclude all the local results of (10) according to the reasoning process from Eq.(11) to Eq.(15). When we analyze the partition P_1 and its communication environment, the local result of Eq.(15) can be deduced from Eq.(11) and Eq.(14) in the following assume-guarantee rule.

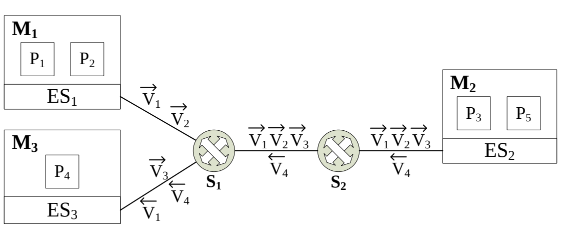
$$\frac{\begin{array}{r} P_2 \preceq A_2^1 \wedge P_2 \preceq A_2^2 \wedge P_3 \preceq A_3^3 \\ P_1 \| A_2^1 \| A_2^2 \| A_3^3 \models \varphi_1 \end{array}}{P_1 \| P_2 \| P_3 \models \varphi_1} \tag{16}$$

The local results are then combined to constitute the global result of Eq.(9).

6 Case Study

In this section, we applies the compositional approach to an avionics system which combines the workload of [6] and the AFDX configuration of [13]. The workload consists of 5 partitions, and further divided into 18 periodic tasks and 4 sporadic tasks. Considering the inter-partition messages in the workload, we assign each message type $Msg_i, i = \{1,2,3,4\}$ a separate VL with the same subscript. The messages of Msg_1 and Msg_2 are handled at the refresh period $50ms$ in sampling ports. Msg_3 and Msg_4 are configured to operate in queuing ports, each of which can accommodate a maximum of one message.

As shown in Fig.5, we consider the distributed architecture that comprises 3 ARINC-653 modules connected by an AFDX network. The module M_1 accommodates P_1 and P_2, the module M_2 executes P_3 and P_5, and the partition P_4 is allocated to M_3. There are 4 VLs V_1-V_4 connecting 3 ESs across 2 switches S_1 and S_2 in the AFDX network. The arrows above VLs' names indicate the direction of message flow.



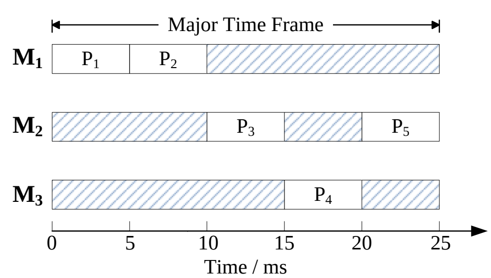


Figure 5: The Distributed Avionics Deployment and Partition Schedules (Times in Milliseconds)

The avionics system equips each of its processor cores with a partition schedule. Assume the modules in the experiment to be single-processor platforms. Fig.5 gives the partition schedules, which fix a common major time frame T_{mf} at $25ms$ and allocate $5ms$ to each partition within every T_{mf}. All the partition schedules are enabled at the same initial instant. The scheduling configuration keeps the temporal order of the partitions in [6]. Hence the partition schedules contain five disjoint windows $\langle P_1, 0, 5\rangle$,

$\langle P_2, 5, 5\rangle$, $\langle P_3, 10, 5\rangle$, $\langle P_4, 15, 5\rangle$, and $\langle P_5, 20, 5\rangle$, where the second parameter is the offset from the start of T_{mf} and last the duration.

We analyze the schedulability of this avionics system following the procedure in section 5:

(1) *Decomposition:* The system is first decomposed into five sets of SWA template instances corresponding to five partitions. The schedulability of any partition $P_i, i = \{1,2,3,4,5\}$ is described as the UPPAAL query q_i:

$$\texttt{A[] not perror[i]}, \tag{17}$$

where the boolean variable `perror[i]` should be assigned to `True` once any error locations are reached in P_i. When analyzing the schedulability of P_i, we *only instantiate* the set of SWA template instances of P_i into UPPAAL processes. This set contains two scheduler models coming from `PartitionSupply` and `TaskScheduler`, all the `PeriodicTask` and `SporadicTask` models in P_i, and the communication layer models from which P_i receives messages.

(2) *Construction of message interfaces:* The message interfaces are constructed from the template depicted in Fig.4, for all the messages originate in periodic tasks. There are four unknown parameters `period`, `initOffset`, `offset`, and `jitter` in the template. Initially, the parameters of a message interface are set to the same values as these of the source task. Then we employ a binary search to heuristically refine `offset` and `jitter`, meanwhile guaranteeing timed selection simulation relation exists.

(3) *Model checking:* The schedulability of five partitions is checked individually. After combining the models of P_i and its message interfaces, we verify the property q_i by model checking in UPPAAL. The verification was repeated for each partition to evaluate the schedulability of a complete system. The experiment was executed on the UPPAAL 4.1.19 64-bit version and an Intel Core i7-5600U laptop processor.

(4) *Deduction:* According to the assume-guarantee rule described in Eq.(8), we conclude the schedulability of the complete system from the results of the verification of five partitions.

Results of the Analysis

The result in Table 1 shows that each partition is separately schedulable (The results "Yes" of Case 1) except the partition P_3 (The result "No"). From a global view, we cannot conclude directly that the system is non-schedulable, because the compositional approach described in section 5 only provides a sufficient condition for schedulability. Nevertheless, we find a counter-example by simulation in UPPAAL, and thus it can be concluded that the current system is not schedulable. The counter-example shows that P_3 violates the constraint of the refresh period of Msg_2 due to network latency.

Considering the effect of network latency on the scheduling configuration, we updated the partition schedules by performing a swap of time slots between P_1 and P_2. The modified partition schedules provide five windows $\langle P_1, 5, 5\rangle$, $\langle P_2, 0, 5\rangle$, $\langle P_3, 10, 5\rangle$, $\langle P_4, 15, 5\rangle$, and $\langle P_5, 20, 5\rangle$. The compositional analysis of the updated system was executed again. The result (Case 2 in Table 1) shows that all the partitions of the updated system are individually schedulable. Thus, the updated system finally achieves the schedulability at the global level.

Table 1 also shows the performance in terms of execution time and memory usage. In both cases, the partition P_3 contains more instantiated models (19 processes) than the other four partitions. As a result, model-checking runs evidently slower and requires more memory than the others. Nevertheless, the compositional analysis could be performed on ordinary computers within an acceptable time.

Compared with the compositional way, global analysis based on the same UPPAAL models would require 51 processes including all the 22 task models, whose state space is much more complex than the others. This causes UPPAAL to run out of memory within a few minutes, and thus makes the global

Table 1: The Experiment Results (Result), Execution Time (Time/sec.) and Memory Usage (Mem/MB)

No.	Case 1			Case 2		
	Result	Time	Mem	Result	Time	Mem
P_1	Yes	7.46	146	Yes	6.07	105
P_2	Yes	0.95	46	Yes	1.10	52
P_3	No	42.94	664	Yes	256.48	3041
P_4	Yes	0.69	43	Yes	0.68	43
P_5	Yes	19.41	509	Yes	128.56	2041

analysis infeasible. In contrast, the compositional approach only requires at most 5 task models when we perform model checking, offering effective state space reduction.

7 Conclusion

In this paper, we present a compositional approach for schedulability analysis of DIMA systems, which are modeled as a set of stopwatch automata in UPPAAL, describing schedulability as safety properties of models. We check each ARINC-653 partition including its communication environment individually, thereby reducing the complexity of model-checking. The techniques presented in this paper are applicable to the design of DIMA scheduling systems. We have applied the compositional approach to a concrete DIMA system. As future work, we plan to develop a model-based approach to the automatic optimization and generation of the partition schedules of a DIMA system.

References

[1] AEEC (2010): *Avionics Application Software Standard Interface: Part 1 - Required Services*. ARINC Specification 653P1-3, Aeronautical Radio Inc.

[2] Tobias Amnell, Elena Fersman, Leonid Mokrushin, Paul Pettersson & Wang Yi: *TIMES: a tool for schedulability analysis and code generation of real-time systems*. In: *FORMATS 2003*.

[3] Björn Annighöfer & Frank Thielecke (2014): *A Systems Architecting Framework for Distributed Integrated Modular Avionics*. DGLR.

[4] Jalil Boudjadar, Kim Guldstrand Larsen, Jin Hyun Kim & Ulrik Nyman: *Compositional schedulability analysis of an avionics system using UPPAAL*. In: *AASE 2014*.

[5] Laura Carnevali, Giuseppe Lipari, Alessandro Pinzuti & Enrico Vicario: *A formal approach to design and verification of two-level Hierarchical Scheduling systems*. In: *RST 2011*.

[6] Laura Carnevali, Alessandro Pinzuti & Enrico Vicario (2013): *Compositional verification for hierarchical scheduling of real-time systems*. *IEEE Transactions on Software Engineering* 39(5), pp. 638–657.

[7] Franck Cassez & Kim Larsen (2000): *The impressive power of stopwatches*. In: *International Conference on Concurrency Theory*, Springer, pp. 138–152.

[8] Franco Cicirelli, Angelo Furfaro, Libero Nigro & Francesco Pupo: *Development of a schedulability analysis framework based on pTPN and UPPAAL with stopwatches*. In: *DSRA 2012*.

[9] Alexandre David, Kim G Larsen, Axel Legay, Ulrik Nyman & Andrzej Wasowski: *Timed I/O automata: a complete specification theory for real-time systems*. In: *HSCC 2010*.

[10] RB Dodd (2006): *Coloured petri net modelling of a generic avionics mission computer*. Technical Report, DTIC.

[11] Arvind Easwaran, Insup Lee, Oleg Sokolsky & Steve Vestal: *A compositional scheduling framework for digital avionics systems*. In: *RTCSA 2009*.

[12] Orna Grumberg & David Long (1994): *Model checking and modular verification*. *Toplas* 16(3), pp. 843–871.

[13] J Javier Gutiérrez, J Carlos Palencia & Michael González Harbour (2014): *Holistic schedulability analysis for multipacket messages in AFDX networks*. *Real-Time Systems* 50(2).

[14] Henrik Jensen (1999): *Abstraction-based verification of distributed systems*. Ph.D. thesis, Aalborg university.

[15] Henrik Jensen, Kim Larsen & Arne Skou: *Scaling up UPPAAL*. In: *FTRFS 2000*.

[16] Marius Mikučionis, Kim Larsen, Jacob Rasmussen, Brian Nielsen, Arne Skou, Steen Palm, Jan Pedersen & Poul Hougaard: *Schedulability analysis using Uppaal: Herschel-Planck case study*. In: *ISoLA 2010*.

[17] Jan Reineke, Björn Wachter & Stefan Thesing et al.: *A Definition and Classification of Timing Anomalies*. In: *WCET 2006*.

[18] Youcheng Sun, Giuseppe Lipari, Romain Soulat, Laurent Fribourg & Nicolas Markey: *Component-based analysis of hierarchical scheduling using linear hybrid automata*. In: *ERCSA 2014*.

[19] Guoqing Wang & Qingfan Gu: *Research on distributed integrated modular avionics system architecture design and implementation*. In: *DASC 2013*.

Appendix

The appendix consists of three sections. Appendix A gives the proofs of all the theorems in the paper. Appendix B details the avionics workload in the case study. The AFDX configuration is then presented in Appendix C.

A Theorem Proofs

Lemma 1. *Let $\mathcal{T}_i = \langle S_i, s_{i,0}, \Sigma_i, \rightarrow_i\rangle$, $i \in \{1,2\}$ be two TIOTSs. Assume that R is a timed selection simulation from $\mathcal{T}_1$ to $\mathcal{T}_2$. Then for all $(s_1,s_2) \in R$,*

1. *if $s_1 \overset{a?}{\Longrightarrow} s_1'$ for some $s_1' \in S_1$, $a \in \Sigma_2$, then there exists $s_2' \in S_2$ such that $s_2 \overset{a?}{\Longrightarrow} s_2'$ and $(s_1',s_2') \in R$*
2. *if $s_1 \overset{a!}{\Longrightarrow} s_1'$ for some $s_1' \in S_1$, $a \in \Sigma_2$, then there exists $s_2' \in S_2$ such that $s_2 \overset{a!}{\Longrightarrow} s_2'$ and $(s_1',s_2') \in R$*
3. *if $s_1 \overset{a}{\Longrightarrow} s_1'$ for some $s_1' \in S_1$, $a \in (\Sigma_1 \setminus \Sigma_2)$, then there exists $s_2' \in S_2$ such that $s_2 \overset{\mathbf{0}}{\Longrightarrow} s_2'$ and $(s_1',s_2') \in R$*
4. *if $s_1 \overset{\varepsilon(d)}{\Longrightarrow} s_1'$ for some $s_1' \in S_1$, $d \geq 0$, then there exists $s_2' \in S_2$ such that $s_2 \overset{\varepsilon(d)}{\Longrightarrow} s_2'$ and $(s_1',s_2') \in R$.*

Proof. Consider $\mathcal{T}_1$, $\mathcal{T}_2$, s_1, s_2, and R in Lemma 1. From 3 of Definition 5 it is trivially the fact that if $s_1 \overset{\tau}{\rightarrow}^* s_1', s_1' \in S_1$ then $s_2 \overset{\tau}{\rightarrow}^* s_2'$ for some $s_2' \in S_2$ such that $(s_1',s_2') \in R$. We denote this by $(*)$.

Suppose $s_1 \overset{a?}{\Longrightarrow} s_1'$, $s_1' \in S_1$, and $a \in \Sigma_2$. Thus $s_1 \overset{\tau}{\rightarrow}^* s_1'' \overset{a?}{\rightarrow} s_1''' \overset{\tau}{\rightarrow}^* s_1'$ for some $s_1'', s_1''' \in S_1$. From $(*)$ and 1 of Definition 5, we have that there exist $s_2', s_2'', s_2''' \in S_2$ such that $s_2 \overset{\tau}{\rightarrow}^* s_2'' \overset{a?}{\Longrightarrow} s_2''' \overset{\tau}{\rightarrow}^* s_2'$, i.e. $s_2 \overset{a?}{\Longrightarrow} s_2'$, where $(s_1',s_2'),(s_1'',s_2''),(s_1''',s_2''') \in R$. Hence 1 of Lemma 1 holds. Similarly 2 of Lemma 1 also holds.

Suppose $s_1 \overset{a}{\Longrightarrow} s_1'$ for some $s_1' \in S_1$, $a \in (\Sigma_1 \setminus \Sigma_2)$. Then $s_1 \overset{\tau}{\rightarrow}^* s_1'' \overset{a}{\rightarrow} s_1''' \overset{\tau}{\rightarrow}^* s_1'$ for some $s_1'', s_1''' \in S_1$. From $(*)$ and 3 of Definition 5, there exist $s_2', s_2'', s_2''' \in S_2$ such that $s_2 \overset{\tau}{\rightarrow}^* s_2'' \overset{\tau}{\rightarrow}^* s_2''' \overset{\tau}{\rightarrow}^* s_2'$ and $(s_1',s_2'),(s_1'',s_2''),(s_1''',s_2''') \in R$. Thus we have $s_2 \overset{\mathbf{0}}{\Longrightarrow} s_2'$ and 3 of Lemma 1 holds.

Finally, suppose $s_1 \overset{\varepsilon(d)}{\Longrightarrow} s_1'$ for some $s_1' \in S_1$, $d \geq 0$. First, if $d = 0$ then 4 of Lemma 1 holds because it is identical to $(*)$. Second, in the case of $d > 0$ we have $s_1 \overset{\tau}{\rightarrow}^* s_1^{1\prime} \xrightarrow{\varepsilon(d_1)} s_1^{1\prime\prime} \overset{\tau}{\rightarrow}^* s_1^{2\prime} \xrightarrow{\varepsilon(d_2)} s_1^{2\prime\prime} \overset{\tau}{\rightarrow}^* \cdots \overset{\tau}{\rightarrow}^* s_1^{n\prime} \xrightarrow{\varepsilon(d_n)} s_1^{n\prime\prime} = s_1'$ where $\Sigma_{i=1}^n d_i = d$. From From $(*)$ and 4 of Definition 5, there exist $s_2^{1\prime}, s_2^{1\prime\prime}, s_2^{2\prime}, s_2^{2\prime\prime}, \ldots, s_2^{n\prime}, s_2^{n\prime\prime} \in S_2$ such that $s_2 \overset{\tau}{\rightarrow}^* s_2^{1\prime} \overset{\varepsilon(d_1)}{\Longrightarrow} s_2^{1\prime\prime} \overset{\tau}{\rightarrow}^* s_2^{2\prime} \overset{\varepsilon(d_2)}{\Longrightarrow} s_2^{2\prime\prime} \overset{\tau}{\rightarrow}^* \cdots \overset{\tau}{\rightarrow}^* s_2^{n\prime} \overset{\varepsilon(d_n)}{\Longrightarrow} s_2^{n\prime\prime} = s_2'$ and $(s_1^{1\prime},s_2^{1\prime})$, $(s_1^{1\prime\prime},s_2^{1\prime\prime})$, $(s_1^{2\prime},s_2^{2\prime})$, $\cdots$, $(s_1^{n\prime},s_2^{n\prime})$, $(s_1^{n\prime\prime},s_2^{n\prime\prime}) \in R$. Hence we have $s_2 \overset{\varepsilon(d)}{\Longrightarrow} s_2'$ and 4 of Lemma 1 holds. □

Lemma 2. *Let $\mathcal{T}_i = \langle S_i, s_{i,0}, \Sigma_i, \rightarrow_i\rangle$, $i \in \{1,2\}$ be two compatible TIOTSs. Assume that $\mathcal{T}_{1\|2} = \langle S_{1\|2}, s_{1\|2,0}, \Sigma_{1\|2}, \rightarrow_{1\|2}\rangle = \mathcal{T}_1 \| \mathcal{T}_2$. Then for all $\langle s_1,s_2\rangle \in S_{1\|2}$,*

1. *if $s_1 \overset{a?}{\Longrightarrow} s_1'$ and $s_2 \overset{a?}{\Longrightarrow} s_2'$ for some $s_1' \in S_1$, $s_2' \in S_2$, $a \in \Sigma_1 \cap \Sigma_2$, then there exists $a \in \Sigma_{1\|2}$ such that $\langle s_1,s_2\rangle \overset{a?}{\Longrightarrow} \langle s_1',s_2'\rangle$ in $\mathcal{T}_{1\|2}$*
2. *if $s_1 \overset{a!}{\Longrightarrow} s_1'$ and $s_2 \overset{a?}{\Longrightarrow} s_2'$, or if $s_1 \overset{a?}{\Longrightarrow} s_1'$ and $s_2 \overset{a!}{\Longrightarrow} s_2'$, for some $s_1' \in S_1$, $s_2' \in S_2$, $a \in \Sigma_1 \cap \Sigma_2 \cap \Sigma^b$, then there exists $a \in \Sigma_{1\|2}$ such that $\langle s_1,s_2\rangle \overset{a!}{\Longrightarrow} \langle s_1',s_2'\rangle$ in $\mathcal{T}_{1\|2}$*

3. *if* $s_1 \overset{a!}{\Longrightarrow} s_1'$ *and* $s_2 \overset{a?}{\Longrightarrow} s_2'$*, or if* $s_1 \overset{a?}{\Longrightarrow} s_1'$ *and* $s_2 \overset{a!}{\Longrightarrow} s_2'$*, for some* $s_1' \in S_1$*,* $s_2' \in S_2$*,* $a \in \Sigma_1 \cap \Sigma_2 \cap \Sigma^u$*, then there exists* $\langle s_1, s_2 \rangle \overset{\mathbf{0}}{\Longrightarrow} \langle s_1', s_2' \rangle$ *in* $\mathscr{T}_{1\|2}$
4. *if* $s_1 \overset{a}{\Longrightarrow} s_1'$ *and* $s_2 \overset{\mathbf{0}}{\Longrightarrow} s_2'$*, or if* $s_1 \overset{\mathbf{0}}{\Longrightarrow} s_1'$ *and* $s_2 \overset{a}{\Longrightarrow} s_2'$*, for some* $s_1' \in S_1$*,* $s_2' \in S_2$*,* $a \in \Sigma_1 \oplus \Sigma_2$*, then there exists* $a \in \Sigma_{1\|2}$ *such that* $\langle s_1, s_2 \rangle \overset{a}{\Longrightarrow} \langle s_1', s_2' \rangle$ *in* $\mathscr{T}_{1\|2}$
5. *if* $s_1 \overset{\varepsilon(d)}{\Longrightarrow} s_1'$ *and* $s_2 \overset{\varepsilon(d)}{\Longrightarrow} s_2'$ *for some* $s_1' \in S_1$*,* $s_2' \in S_2$*,* $d \geq 0$*, then there exists* $\langle s_1, s_2 \rangle \overset{\varepsilon(d)}{\Longrightarrow} \langle s_1', s_2' \rangle$ *in* $\mathscr{T}_{1\|2}$*.*

Proof. Consider $\mathscr{T}_1$, $\mathscr{T}_2$, $s_1 \in S_1$, $s_2 \in S_2$, and $\langle s_1, s_2 \rangle \in S_{1\|2}$ in Lemma 2. From the rules "INDEP-L" and "INDEP-R" it is trivially the fact that if $s_1 \overset{\tau}{\rightarrow}^* s_1'$ and $s_2 \overset{\tau}{\rightarrow}^* s_2'$ for some $s_1' \in S_1$, $s_2' \in S_2$ then $\langle s_1, s_2 \rangle \overset{\tau}{\rightarrow}^* \langle s_1', s_2' \rangle$. We denote this by $(**)$.

Suppose $s_1 \overset{a?}{\Longrightarrow} s_1'$ and $s_2 \overset{a?}{\Longrightarrow} s_2'$ for some $s_1' \in S_1$, $s_2' \in S_2$, $a \in \Sigma_1 \cap \Sigma_2$. Then there exist s_1'', $s_1''' \in S_1$, s_2'', $s_2''' \in S_2$ such that $s_1 \overset{\tau}{\rightarrow}^* s_1'' \overset{a?}{\rightarrow} s_1''' \overset{\tau}{\rightarrow}^* s_1'$ and $s_2 \overset{\tau}{\rightarrow}^* s_2'' \overset{a?}{\rightarrow} s_2''' \overset{\tau}{\rightarrow}^* s_2'$. From $(**)$ and the rule "SYNC-IN", we have $\langle s_1, s_2 \rangle \overset{\tau}{\rightarrow}^* \langle s_1'', s_2'' \rangle \overset{a?}{\rightarrow} \langle s_1''', s_2''' \rangle \overset{\tau}{\rightarrow}^* \langle s_1', s_2' \rangle$. Hence $\langle s_1, s_2 \rangle \overset{a?}{\Longrightarrow} \langle s_1', s_2' \rangle$ and 1 of Lemma 2 holds.

Suppose $s_1 \overset{a!}{\Longrightarrow} s_1'$ and $s_2 \overset{a?}{\Longrightarrow} s_2'$ for some $s_1' \in S_1$, $s_2' \in S_2$, $a \in \Sigma_1 \cap \Sigma_2 \cap \Sigma^b$. Then there exist s_1'', $s_1''' \in S_1$, s_2'', $s_2''' \in S_2$ such that $s_1 \overset{\tau}{\rightarrow}^* s_1'' \overset{a!}{\rightarrow} s_1''' \overset{\tau}{\rightarrow}^* s_1'$ and $s_2 \overset{\tau}{\rightarrow}^* s_2'' \overset{a?}{\rightarrow} s_2''' \overset{\tau}{\rightarrow}^* s_2'$. From $(**)$ and the rule "SYNC-BIO", we have $\langle s_1, s_2 \rangle \overset{\tau}{\rightarrow}^* \langle s_1'', s_2'' \rangle \overset{a!}{\rightarrow} \langle s_1''', s_2''' \rangle \overset{\tau}{\rightarrow}^* \langle s_1', s_2' \rangle$. Hence $\langle s_1, s_2 \rangle \overset{a!}{\Longrightarrow} \langle s_1', s_2' \rangle$. Symmetrically we also have the same conclusion in the case of $s_1 \overset{a?}{\Longrightarrow} s_1'$ and $s_2 \overset{a!}{\Longrightarrow} s_2'$. Thus 2 of Lemma 2 holds. Similarly, when considering a to be a unicast action, we have that 3 of Lemma 2 holds.

Suppose $s_1 \overset{a}{\Longrightarrow} s_1'$ and $s_2 \overset{\mathbf{0}}{\Longrightarrow} s_2'$ for some $s_1' \in S_1$, $s_2' \in S_2$, $a \in \Sigma_1 \oplus \Sigma_2$. Then there exist s_1'', $s_1''' \in S_1$, $s_2'' \in S_2$ such that $s_1 \overset{\tau}{\rightarrow}^* s_1'' \overset{a}{\rightarrow} s_1''' \overset{\tau}{\rightarrow}^* s_1'$ and $s_2 \overset{\tau}{\rightarrow}^* s_2'' \overset{\tau}{\rightarrow}^* s_2'$. From $(**)$ and the rule "INDEP-L", we have $\langle s_1, s_2 \rangle \overset{\tau}{\rightarrow}^* \langle s_1'', s_2'' \rangle \overset{a}{\rightarrow} \langle s_1''', s_2'' \rangle \overset{\tau}{\rightarrow}^* \langle s_1', s_2' \rangle$. Hence $\langle s_1, s_2 \rangle \overset{a}{\Longrightarrow} \langle s_1', s_2' \rangle$. Symmetrically we also have the same conclusion in the case of $s_1 \overset{\mathbf{0}}{\Longrightarrow} s_1'$ and $s_2 \overset{a}{\Longrightarrow} s_2'$. Thus 4 of Lemma 2 holds.

Suppose $s_1 \overset{\varepsilon(d)}{\Longrightarrow} s_1'$ and $s_2 \overset{\varepsilon(d)}{\Longrightarrow} s_2'$ for some $s_1' \in S_1$, $s_2' \in S_2$, $d \geq 0$. From $(**)$ we have that 5 of Lemma 2 holds in the case of $d = 0$. Consider the case of $d \geq 0$. $s_1 \overset{\varepsilon(d)}{\Longrightarrow} s_1'$ is equivalent to $s_1 \overset{\tau}{\rightarrow}^* s_1^{1\prime} \xrightarrow{\varepsilon(d_1)} s_1^{1\prime\prime} \overset{\tau}{\rightarrow}^* s_1^{2\prime} \xrightarrow{\varepsilon(d_2)} s_1^{2\prime\prime} \overset{\tau}{\rightarrow}^* \cdots \overset{\tau}{\rightarrow}^* s_1^{n\prime} \xrightarrow{\varepsilon(d_n)} s_1^{n\prime\prime} \overset{\tau}{\rightarrow}^* s_1'$ where $n \in \mathbf{N}_+, \Sigma_{i=1}^n d_i = d$. We now prove 5 of Lemma 2 using mathematical induction.

Assume that $s_2 \overset{\varepsilon(d)}{\Longrightarrow} s_2'$ contains a transition chain $s_2 \overset{\tau}{\rightarrow}^* s_2^{1\prime} \xrightarrow{\varepsilon(d_1')} s_2^{1\prime\prime} \overset{\tau}{\rightarrow}^* s_2^{2\prime} \xrightarrow{\varepsilon(d_2')} s_2^{2\prime\prime} \overset{\tau}{\rightarrow}^* \cdots \overset{\tau}{\rightarrow}^* s_2^{m\prime} \xrightarrow{\varepsilon(d_m')} s_2^{m\prime\prime} \overset{\tau}{\rightarrow}^* s_2'$ where $m \in \mathbf{N}_+, \Sigma_{i=1}^m d_i' = d$.

If $n = 1$ then $s_1 \overset{\varepsilon(d)}{\Longrightarrow} s_1'$ will be equivalent to $s_1 \overset{\tau}{\rightarrow}^* s_1^{1\prime} \xrightarrow{\varepsilon(d)} s_1^{1\prime\prime} \overset{\tau}{\rightarrow}^* s_1'$. From time additivity of TIOTS, there exist $s_1^{2\prime}, s_1^{3\prime}, \ldots, s_1^{m\prime} \in S_1$ such that $s_1 \overset{\tau}{\rightarrow}^* s_1^{1\prime} \xrightarrow{\varepsilon(d_1')} s_1^{2\prime} \xrightarrow{\varepsilon(d_2')} \cdots \xrightarrow{\varepsilon(d_m')} s_1^{1\prime\prime} \overset{\tau}{\rightarrow}^* s_1'$. By $(**)$ and the rule "DELAY", we have the transition chain $\langle s_1, s_2 \rangle \overset{\tau}{\rightarrow}^* \langle s_1^{1\prime}, s_2^{1\prime} \rangle \xrightarrow{\varepsilon(d_1')} \langle s_1^{2\prime}, s_2^{1\prime\prime} \rangle \overset{\tau}{\rightarrow}^* \langle s_1^{2\prime}, s_2^{2\prime} \rangle \xrightarrow{\varepsilon(d_2')} \langle s_1^{3\prime}, s_2^{2\prime\prime} \rangle \overset{\tau}{\rightarrow}^* \cdots \overset{\tau}{\rightarrow}^* \langle s_1^{m\prime}, s_2^{m\prime} \rangle \xrightarrow{\varepsilon(d_m')} \langle s_1^{1\prime\prime}, s_2^{m\prime\prime} \rangle \overset{\tau}{\rightarrow}^* \langle s_1', s_2' \rangle$. Thus there exists $\langle s_1, s_2 \rangle \overset{\varepsilon(d)}{\Longrightarrow} \langle s_1', s_2' \rangle$ in $\mathscr{T}_{1\|2}$.

We assume that there exists $\langle s_1, s_2 \rangle \overset{\varepsilon(d)}{\Longrightarrow} \langle s_1', s_2' \rangle$ in $\mathscr{T}_{1\|2}$ if $n = t, t \in \mathbf{N}_+$. If $n = t+1$ then $s_1 \overset{\varepsilon(d)}{\Longrightarrow} s_1'$ should contain a transition chain $s_1 \overset{\tau}{\rightarrow}^* s_1^{1\prime} \xrightarrow{\varepsilon(d_1)} s_1^{1\prime\prime} \overset{\tau}{\rightarrow}^* s_1^{2\prime} \xrightarrow{\varepsilon(d_2)} s_1^{2\prime\prime} \overset{\tau}{\rightarrow}^* \cdots \overset{\tau}{\rightarrow}^* s_1^{t\prime} \xrightarrow{\varepsilon(d_t)} s_1^{t\prime\prime} \overset{\tau}{\rightarrow}^* s_1^{(t+1)\prime} \xrightarrow{\varepsilon(d_{t+1})} s_1^{(t+1)\prime\prime} \overset{\tau}{\rightarrow}^* s_1'$. Let $d' = d - d_{t+1}$. From time additivity of TIOTS, there exists $s_2^r \in S_2$ such that $s_2 \overset{\varepsilon(d')}{\Longrightarrow} s_2^r \overset{\varepsilon(d_{t+1})}{\Longrightarrow} s_2'$. By the assumption in the case of $n = t$, we have that $\langle s_1, s_2 \rangle \overset{\varepsilon(d')}{\Longrightarrow} \langle s_1^{t\prime\prime}, s_2^r \rangle$.

Consider the transitions $s_1^{t''} \xrightarrow{\tau}{}^* s_1^{(t+1)'} \xrightarrow{\varepsilon(d_{t+1})} s_1^{(t+1)''} \xrightarrow{\tau}{}^* s_1'$ and $s_2^r \xRightarrow{\varepsilon(d_{t+1})} s_2'$. From the conclusion under the assumption $n=1$, $\langle s_1^{t''}, s_2^r\rangle \xRightarrow{\varepsilon(d_{t+1})} \langle s_1', s_2'\rangle$ exists in $\mathscr{T}_{1\|2}$ and we also have $\langle s_1, s_2\rangle \xRightarrow{\varepsilon(d)} \langle s_1', s_2'\rangle$ in the case $n = t+1$. Hence 5 of Lemma 2 holds. □

Proof of Theorem 1. A preorder should be reflexive and transitive. For any TIOTS $\mathscr{T} = \langle S, s_0, \Sigma, \rightarrow\rangle$, the binary relation $R = \{(s,s)|s \in S\}$ trivially conforming to Definition 5 is a timed selection simulation from $\mathscr{T}$ to $\mathscr{T}$, i.e. $\mathscr{T} \preceq \mathscr{T}$. Hence reflexivity holds.

We now show the transitivity of timed selection simulation. Consider any three TIOTSs $\mathscr{T}_i = \langle S_i, s_{i,0}, \Sigma_i, \rightarrow_i\rangle$, $i \in \{1,2,3\}$. Assume that R_1 is a timed selection simulation from $\mathscr{T}_1$ to $\mathscr{T}_2$ and R_2 a timed selection simulation from $\mathscr{T}_2$ to $\mathscr{T}_3$. We prove that the new relation $R = R_1R_2$ is a timed selection simulation from $\mathscr{T}_1$ to $\mathscr{T}_3$.

From Definition 5 we have $(s_{0,1}, s_{0,2}) \in R_1$ and $(s_{0,2}, s_{0,3}) \in R_2$. Hence $(s_{0,1}, s_{0,3}) \in R$. For any $(s_1, s_3) \in R$, there exists $s_2 \in S_2$ such that $(s_1, s_2) \in R_1$ and $(s_2, s_3) \in R_2$. By Definition 5, $g(s_1) = g(s_2)$ and $g(s_2) = g(s_3)$. Thus $g(s_1) = g(s_3)$. Consider the four conditions of Definition 5.

Suppose $s_1 \xrightarrow{a?} s_1'$, $s_1' \in S_1$, and $a \in \Sigma_3$. Since $\mathscr{T}_2 \preceq \mathscr{T}_3$, we have $\Sigma_3 \subseteq \Sigma_2$ and thus $a \in \Sigma_2$. Since $\mathscr{T}_1 \preceq \mathscr{T}_2$, there exists $s_2' \in S_2$ such that $s_2 \xRightarrow{a?} s_2'$ and $(s_1', s_2') \in R_1$. Since $\mathscr{T}_2 \preceq \mathscr{T}_3$, by 1 of Lemma 1 there exists $s_3' \in S_3$ such that $s_3 \xRightarrow{a?} s_3'$ and $(s_2', s_3') \in R_2$. Thus $(s_1', s_3') \in R$ and condition 1 of Definition 5 holds. Similarly condition 2 of Definition 5 also holds.

Suppose $s_1 \xrightarrow{a} s_1'$, $s_1' \in S_1$, and $a \in (\Sigma_1 \setminus \Sigma_3)$. If $a \in \Sigma_2$, then $s_2 \xRightarrow{a} s_2', s_2' \in S_2$ and $(s_1', s_2') \in R_1$ for $\mathscr{T}_1 \preceq \mathscr{T}_2$ and thus $s_3 \xRightarrow{\mathbf{0}} s_3', s_3' \in S_3$ and $(s_2', s_3') \in R_2$ for $\mathscr{T}_2 \preceq \mathscr{T}_3$. Hence $(s_1', s_3') \in R$ in the case of $a \in \Sigma_2$. If $a \notin \Sigma_2$, then $s_2 \xRightarrow{\mathbf{0}} s_2', s_2' \in S_2$ and $(s_1', s_2') \in R_1$ for $\mathscr{T}_1 \preceq \mathscr{T}_2$. From Lemma 1 and $\mathscr{T}_2 \preceq \mathscr{T}_3$, we have $s_3 \xRightarrow{\mathbf{0}} s_3', s_3' \in S_3$ and $(s_2', s_3') \in R_2$. Thus $(s_1', s_3') \in R$ in this case.

Suppose $s_1 \xrightarrow{\varepsilon(d)} s_1'$, $s_1' \in S_1$, and $d \geq 0$. From $\mathscr{T}_1 \preceq \mathscr{T}_2$, $s_2 \xRightarrow{\varepsilon(d)} s_2', s_2' \in S_2$ and $(s_1', s_2') \in R_1$. From Lemma 1 and $\mathscr{T}_2 \preceq \mathscr{T}_3$, we have $s_3 \xRightarrow{\varepsilon(d)} s_3', s_3' \in S_3$ and $(s_2', s_3') \in R_2$. Thus $(s_1', s_3') \in R$ and both condition 3 and 4 of Definition 5 hold.

Therefore, R is a timed selection simulation from $\mathscr{T}_1$ to $\mathscr{T}_3$, and transitivity of timed selection simulation holds. □

Proof of Theorem 2. Let S_i be the state set of $\mathscr{T}_i$. Let R be a timed selection simulation from $\mathscr{T}_1$ to $\mathscr{T}_2$. Note that $\mathscr{T}_i \models \varphi$ iff for any reachable state $s_i \in S_i$ $g(s_i) = false$. We denote this by $(*)$.

From Definition 5 and $\mathscr{T}_1 \preceq \mathscr{T}_2$ we have that for each reachable state $s_1 \in S_1$, there exists a reachable state $s_2 \in S_2$ such that $(s_1, s_2) \in R$ and $g(s_1) = g(s_2)$. Since $\mathscr{T}_2 \models \varphi$ and $(*)$, $g(s_2) = false$ for each reachable state $s_2 \in S_2$. Thus $g(s_1) = false$ for any reachable state $s_1 \in S_1$. From $(*)$, we have $\mathscr{T}_1 \models \varphi$. □

Proof of Theorem 3. Let S_i be the state set of $\mathscr{T}_i$. Assume that R_1 and R_2 are timed selection simulations from $\mathscr{T}_1$ to $\mathscr{T}_2$ and from $\mathscr{T}_1$ to $\mathscr{T}_3$, respectively. Let R be a binary relation from S_1 to $S_2 \times S_3$ such that $(s_1, \langle s_2, s_3\rangle) \in R$ iff $(s_1, s_2) \in R_1$ and $(s_1, s_3) \in R_2$ for any $s_1 \in S_1, s_2 \in S_2, s_3 \in S_3$. We now prove R is a timed selection simulation relation.

Suppose $s_{i,0}$ is the initial state of $\mathscr{T}_i$. By assumption we have $(s_{1,0}, s_{2,0}) \in R_1$ and $(s_{1,0}, s_{3,0}) \in R_2$. Thus $(s_{1,0}, \langle s_{2,0}, s_{3,0}\rangle) \in R$ from the definition of R.

Whenever $(s_1, s_2) \in R_1$ and $(s_1, s_3) \in R_2$, $g(s_1) = g(s_2)$ and $g(s_1) = g(s_3)$ will hold. Hence, from the definition of the function g, we have $g(s_1) = g(\langle s_2, s_3\rangle)$ for any $(s_1, \langle s_2, s_3\rangle) \in R$.

Let Σ_i be the action set of $\mathscr{T}_i$. Let I_i and O_i be the input and output action set in Σ_i respectively. From the composition definition in [9], for any compositional TIOTS $\mathscr{T}_2\|\mathscr{T}_3$ we have $\Sigma_{2\|3} = I_{2\|3} \oplus O_{2\|3}$, $I_{2\|3} = (I_2 \setminus (O_3 \cap \Sigma^b)) \cup (I_3 \setminus (O_2 \cap \Sigma^b))$, and $O_{2\|3} = O_2 \cup O_3$. Since $\Sigma_2 \subseteq \Sigma_1$, $\Sigma_3 \subseteq \Sigma_1$ and $\mathscr{T}_2$ and $\mathscr{T}_3$ are compatible, we have

$$\begin{aligned}
&\Sigma_2 \cup \Sigma_3 \\
=&(I_2 \oplus O_2) \cup (I_3 \oplus O_3) \\
=&[(I_2 \cup O_2) \setminus (I_2 \cap O_2)] \cup [(I_3 \cup O_3) \setminus (I_3 \cap O_3)] \\
=&(I_2 \cup O_2 \cup I_3 \cup O_3) \setminus [(I_3 \cap O_3) \setminus (I_2 \cup O_2)] \setminus [(I_2 \cap O_2) \setminus (I_3 \cup O_3)] \\
\subseteq&\Sigma_1
\end{aligned}$$

Let $I_2' = I_2 \setminus (O_3 \cap \Sigma^b)$ and $I_3' = I_3 \setminus (O_2 \cap \Sigma^b)$.

$$\begin{aligned}
&\Sigma_{2\|3} \\
=&(I_2' \cup I_3') \oplus (O_2 \cup O_3) \\
=&(I_2' \cup O_2 \cup I_3' \cup O_3) \setminus [(I_2' \cup I_3') \cap (O_2 \cup O_3)] \\
=&(I_2' \cup O_2 \cup I_3' \cup O_3) \setminus (I_2' \cap O_2) \setminus (I_3' \cap O_3) \setminus (I_2' \cap O_3) \setminus (I_3' \cap O_2) \\
=&(I_2 \cup O_2 \cup I_3 \cup O_3) \setminus (I_2 \cap O_2) \setminus (I_3 \cap O_3) \setminus (I_2' \cap O_3) \setminus (I_3' \cap O_2) \\
\subseteq&(I_2 \cup O_2 \cup I_3 \cup O_3) \setminus (I_2 \cap O_2) \setminus (I_3 \cap O_3) \\
\subseteq&(I_2 \cup O_2 \cup I_3 \cup O_3) \setminus [(I_2 \cap O_2) \setminus (I_3 \cup O_3)] \setminus [(I_3 \cap O_3) \setminus (I_2 \cup O_2)] \\
=&\Sigma_2 \cup \Sigma_3
\end{aligned}$$

Thus $\Sigma_{2\|3} \subseteq \Sigma_1$.

Assume $(s_1, s_2) \in R_1$ and $(s_1, s_3) \in R_2$ for some $s_1 \in S_1, s_2 \in S_2, s_3 \in S_3$. Then $(s_1, \langle s_2, s_3 \rangle) \in R$. Consider each of the conditions in Definition 5.

Suppose $s_1 \xrightarrow{a?} s_1'$ for some $a \in \Sigma_{2\|3}$. Thus $a \in \Sigma_2 \cup \Sigma_3$. There are the following two cases:

Case 1: $a \in \Sigma_2 \cap \Sigma_3$. By simulation definition we have $s_2 \overset{a?}{\Longrightarrow} s_2'$ and $s_3 \overset{a?}{\Longrightarrow} s_3'$ for some $s_2' \in S_2, s_3' \in S_3$ such that $(s_1', s_2') \in R_1$ and $(s_1', s_3') \in R_2$. Hence $(s_1', \langle s_2', s_3' \rangle) \in R$, and from 1 of Lemma 2 there exists $\langle s_2, s_3 \rangle \overset{a?}{\Longrightarrow} \langle s_2', s_3' \rangle$ in $\mathscr{T}_2\|\mathscr{T}_3$.

Case 2: $a \in \Sigma_2 \oplus \Sigma_3$. By simulation definition we have that $s_2 \overset{a?}{\Longrightarrow} s_2'$ and $s_3 \overset{\mathbf{0}}{\Longrightarrow} s_3'$, or $s_2 \overset{\mathbf{0}}{\Longrightarrow} s_2'$ and $s_3 \overset{a?}{\Longrightarrow} s_3'$, for some $s_2' \in S_2, s_3' \in S_3$ such that $(s_1', s_2') \in R_1$ and $(s_1', s_3') \in R_2$. Hence $(s_1', \langle s_2', s_3' \rangle) \in R$, and from 4 of Lemma 2 there exists $\langle s_2, s_3 \rangle \overset{a?}{\Longrightarrow} \langle s_2', s_3' \rangle$ in $\mathscr{T}_2\|\mathscr{T}_3$.

Suppose $s_1 \xrightarrow{a!} s_1'$ for some $a \in \Sigma_{2\|3}$. There are also two cases:

Case 1: $a \in \Sigma_2 \cap \Sigma_3$. By simulation definition we have that $s_2 \overset{a!}{\Longrightarrow} s_2'$ and $s_3 \overset{a?}{\Longrightarrow} s_3'$, or $s_2 \overset{a?}{\Longrightarrow} s_2'$ and $s_3 \overset{a!}{\Longrightarrow} s_3'$, for some $s_2' \in S_2, s_3' \in S_3, a \in \Sigma^b$ such that $(s_1', s_2') \in R_1$ and $(s_1', s_3') \in R_2$. Hence $(s_1', \langle s_2', s_3' \rangle) \in R$, and from 2 of Lemma 2 there exists $\langle s_2, s_3 \rangle \overset{a!}{\Longrightarrow} \langle s_2', s_3' \rangle$ in $\mathscr{T}_2\|\mathscr{T}_3$.

Case 2: $a \in \Sigma_2 \oplus \Sigma_3$. By simulation definition we have that $s_2 \overset{a!}{\Longrightarrow} s_2'$ and $s_3 \overset{\mathbf{0}}{\Longrightarrow} s_3'$, or $s_2 \overset{\mathbf{0}}{\Longrightarrow} s_2'$ and $s_3 \overset{a!}{\Longrightarrow} s_3'$, for some $s_2' \in S_2, s_3' \in S_3$ such that $(s_1', s_2') \in R_1$ and $(s_1', s_3') \in R_2$. Hence $(s_1', \langle s_2', s_3' \rangle) \in R$, and from 4 of Lemma 2 there exists $\langle s_2, s_3 \rangle \overset{a!}{\Longrightarrow} \langle s_2', s_3' \rangle$ in $\mathscr{T}_2\|\mathscr{T}_3$.

Suppose $s_1 \xrightarrow{a} s_1'$ for some $a \in \Sigma_1 \setminus \Sigma_{2\|3}$. Since $\Sigma_{2\|3} \subseteq (\Sigma_2 \cup \Sigma_3) \subseteq \Sigma_1$, there are the following three cases:

Table 2: Transition Set 1

Premises	No.	$\mathscr{T}_1$			$\mathscr{T}_2$			$\mathscr{T}_3$			$\mathscr{T}_4$		
		st_1	Tr_1	st_1'	st_2	Tr_2	st_2'	st_3	Tr_3	st_3'	st_4	Tr_4	st_4'
$a \in \Sigma_2 \setminus \Sigma_4$	1	s_1	$\xrightarrow{a?}$	s_1'	s_2	$\xRightarrow{a?}$	s_2'	s_3	$\xrightarrow{a!}$	s_3'	s_4	$\xRightarrow{\mathbf{0}}$	s_4'
	2	s_1	$\xrightarrow{a!}$	s_1'	s_2	$\xRightarrow{a!}$	s_2'	s_3	$\xrightarrow{a?}$	s_3'	s_4	$\xRightarrow{\mathbf{0}}$	s_4'
	3	s_1	$\xrightarrow{a!}$	s_1'	s_2	$\xRightarrow{a!}$	s_2'	s_3	-	s_3	s_4	-	s_4
$a \in \Sigma_4 \setminus \Sigma_2$	4	s_1	$\xrightarrow{a?}$	s_1'	s_2	$\xRightarrow{\mathbf{0}}$	s_2'	s_3	$\xrightarrow{a!}$	s_3'	s_4	$\xRightarrow{a!}$	s_4'
	5	s_1	$\xrightarrow{a!}$	s_1'	s_2	$\xRightarrow{\mathbf{0}}$	s_2'	s_3	$\xrightarrow{a?}$	s_3'	s_4	$\xRightarrow{a?}$	s_4'
	6	s_1	-	s_1	s_2	-	s_2	s_3	$\xrightarrow{a!}$	s_3'	s_4	$\xRightarrow{a!}$	s_4'

Case 1: $a \in \Sigma_1 \setminus (\Sigma_2 \cup \Sigma_3)$. Thus $a \in \Sigma_1 \setminus \Sigma_2$ and $a \in \Sigma_1 \setminus \Sigma_3$. By simulation definition we have that $s_2 \xRightarrow{\mathbf{0}} s_2'$ and $s_3 \xRightarrow{\mathbf{0}} s_3'$ for some $s_2' \in S_2, s_3' \in S_3$ such that $(s_1', s_2') \in R_1$ and $(s_1', s_3') \in R_2$. Hence $(s_1', \langle s_2', s_3' \rangle) \in R$, and from 5 of Lemma 2 there exists $\langle s_2, s_3 \rangle \xRightarrow{\mathbf{0}} \langle s_2', s_3' \rangle$ in $\mathscr{T}_2 \| \mathscr{T}_3$.

Case 2: $a \in (\Sigma_2 \cup \Sigma_3) \setminus \Sigma_{2\|3}$ and $a \in \Sigma_2 \cap \Sigma_3$. Since $(\Sigma_2 \cup \Sigma_3) \setminus \Sigma_{2\|3} \subseteq (I_2' \cap O_2) \cup (I_3' \cap O_3) \cup (I_2' \cap O_3) \cup (I_3' \cap O_2)$, we ensure $a \in (I_2 \cap O_3) \cup (I_3 \cap O_2)$ and $a \in \Sigma^u$. However, from $\mathscr{T}_1 \preceq \mathscr{T}_2, \mathscr{T}_1 \preceq \mathscr{T}_3$ we have that $s_2 \xRightarrow{a!} s_2'$ and $s_3 \xRightarrow{a!} s_3'$, or $s_2 \xRightarrow{a?} s_2'$ and $s_3 \xRightarrow{a?} s_3'$, for some $s_2' \in S_2, s_3' \in S_3$. Thus $a \in (I_2 \cap I_3) \cup (O_2 \cap O_3)$, which contradicts the fact that $\mathscr{T}_2$ and $\mathscr{T}_3$ are compatible. Hence such an action a does not exist in this case.

Case 3: $a \in (\Sigma_2 \cup \Sigma_3) \setminus \Sigma_{2\|3}$ and $a \in \Sigma_2 \oplus \Sigma_3$. Without loss of generality, we assume that $a \in \Sigma_2$ and $a \notin \Sigma_3$. Thus $a \in (I_3 \cap O_3) \cup (I_2 \cap O_3) \cup (I_3 \cap O_2)$ and $a \in \Sigma^u$. Since $a \notin \Sigma_3$, we have $a \in (I_3 \cap O_3)$ or $a \notin I_3 \cup O_3$. Consider $a \in (I_3 \cap O_3)$. $a \in \Sigma_2$ implies $a \in I_2$ or $a \in O_2$, which contradicts the fact that $\mathscr{T}_1$ and $\mathscr{T}_2$ are compatible. If $a \notin I_3 \cup O_3$ then $(I_3 \cap O_3) \cup (I_2 \cap O_3) \cup (I_3 \cap O_2) = \varnothing$. Hence such an action a does not exist in this case.

Suppose $s_1 \xrightarrow{\varepsilon(d)} s_1'$ and $d \geq 0$. By simulation definition we have that $s_2 \xRightarrow{\varepsilon(d)} s_2'$ and $s_3 \xRightarrow{\varepsilon(d)} s_3'$ for some $s_2' \in S_2, s_3' \in S_3$ such that $(s_1', s_2') \in R_1$ and $(s_1', s_3') \in R_2$. Hence $(s_1', \langle s_2', s_3' \rangle) \in R$, and from 5 of Lemma 2 there exists $\langle s_2, s_3 \rangle \xRightarrow{\varepsilon(d)} \langle s_2', s_3' \rangle$ in $\mathscr{T}_2 \| \mathscr{T}_3$. All the conditions hold and thus $\mathscr{T}_1 \preceq \mathscr{T}_2 \| \mathscr{T}_3$. □

Proof of Theorem 4. Let S_i be the state set of $\mathscr{T}_i$. Assume that R_1 and R_2 are timed selection simulations from $\mathscr{T}_1$ to $\mathscr{T}_2$ and from $\mathscr{T}_3$ to $\mathscr{T}_4$, respectively. Let R be a binary relation from $S_1 \times S_3$ to $S_2 \times S_4$ such that $(\langle s_1, s_3 \rangle, \langle s_2, s_4 \rangle) \in R$ iff $(s_1, s_2) \in R_1$ and $(s_3, s_4) \in R_2$ for any $s_1 \in S_1, s_2 \in S_2, s_3 \in S_3, s_4 \in S_4$. We now prove R is a timed selection simulation relation.

Suppose $s_{i,0}$ is the initial state of $\mathscr{T}_i$. By assumption (1) we have $(s_{1,0}, s_{2,0}) \in R_1$ and $(s_{3,0}, s_{4,0}) \in R_2$. Thus $(\langle s_{1,0}, s_{3,0} \rangle, \langle s_{2,0}, s_{4,0} \rangle) \in R$ from the definition of R.

Whenever $(s_1, s_2) \in R_1$ and $(s_3, s_4) \in R_2$, $g(s_1) = g(s_2)$ and $g(s_3) = g(s_4)$ will hold. Hence, from the definition of the function g, we have $g(\langle s_1, s_3 \rangle) = g(\langle s_2, s_4 \rangle)$ for any $(\langle s_1, s_3 \rangle, \langle s_2, s_4 \rangle) \in R$.

Let Σ_i be the action set of $\mathscr{T}_i$. Let I_i and O_i be the input and output action set in Σ_i respectively. From the composition definition in [9], for any compositional TIOTS $\mathscr{T}_i \| \mathscr{T}_j$ we have $\Sigma_{i\|j} = I_{i\|j} \oplus O_{i\|j}$, $I_{i\|j} = (I_i \setminus (O_j \cap \Sigma^b)) \cup (I_j \setminus (O_i \cap \Sigma^b))$, and $O_{i\|j} = O_i \cup O_j$. Let $I_i' = I_i \setminus (O_j \cap \Sigma^b)$ and $I_j' = I_j \setminus (O_i \cap \Sigma^b)$. By assumption (1) and Definition 5 we have $\Sigma_2 \subseteq \Sigma_1$ and $\Sigma_4 \subseteq \Sigma_3$. Then $\Sigma_2 \cup \Sigma_4 \subseteq \Sigma_1 \cup \Sigma_3$. We now prove $\Sigma_{2\|4} \subseteq \Sigma_{1\|3}$.

Assume for the sake of contradiction that there exists $b \in \Sigma_{2\|4}$ but $b \notin \Sigma_{1\|3}$. $\Sigma_{1\|3} = (I_1 \cup O_1 \cup I_3 \cup O_3) \setminus (I'_1 \cap O_1) \setminus (I'_3 \cap O_3) \setminus (I'_1 \cap O_3) \setminus (I'_3 \cap O_1) \subseteq (I_1 \cup O_1 \cup I_3 \cup O_3) \setminus [(I_1 \cap O_1) \setminus (I_3 \cup O_3)] \setminus [(I_3 \cap O_3) \setminus (I_1 \cup O_1)] = \Sigma_1 \cup \Sigma_3$. Similarly, $\Sigma_{2\|4} \subseteq \Sigma_2 \cup \Sigma_4$. Since $b \in \Sigma_{2\|4}$ and $\Sigma_{2\|4} \subseteq \Sigma_2 \cup \Sigma_4 \subseteq \Sigma_1 \cup \Sigma_3$, $b \in \Sigma_1 \cup \Sigma_3$. Considering $b \notin \Sigma_{1\|3}$, we have $b \in (\Sigma_1 \cup \Sigma_3) \setminus \Sigma_{1\|3} \subseteq (I'_1 \cap O_1) \cup (I'_3 \cap O_3) \cup (I'_1 \cap O_3) \cup (I'_3 \cap O_1)$. From Definition 2, $(I'_1 \cap O_1) \cup (I'_3 \cap O_3) \subseteq \Sigma^u$. Consider $(I'_1 \cap O_3) \cup (I'_3 \cap O_1)$. $I'_1 \cap O_3 = I_1 \cap O_3 \cap \Sigma^u \subseteq \Sigma^u$ and $I'_3 \cap O_1 = I_3 \cap O_1 \cap \Sigma^u \subseteq \Sigma^u$. Hence $b \in \Sigma^u$, which contradicts the assumption that $\Sigma_2 \subseteq \Sigma^b, \Sigma_4 \subseteq \Sigma^b$, namely $b \in \Sigma_2 \cup \Sigma_4 \subseteq \Sigma^b$. Thus $b \in \Sigma_{2\|4}$ implies $b \in \Sigma_{1\|3}$, and we have that $\Sigma_{2\|4} \subseteq \Sigma_{1\|3}$.

Assume $(s_1, s_2) \in R_1$ and $(s_3, s_4) \in R_2$ for some $s_1 \in S_1, s_2 \in S_2, s_3 \in S_3, s_4 \in S_4$. Then $(\langle s_1, s_3\rangle, \langle s_2, s_4\rangle) \in R$. Consider each of the conditions in Definition 5.

Suppose $\langle s_1, s_3\rangle \xrightarrow{a?} \langle s'_1, s'_3\rangle$ for some $a \in \Sigma_{2\|4}$. Thus $a \in \Sigma_2 \cup \Sigma_4$. There are the following two cases:

Case 1: $a \in \Sigma_2 \cap \Sigma_4$. Since $\Sigma_2 \subseteq \Sigma_1$ and $\Sigma_4 \subseteq \Sigma_3$, $a \in \Sigma_1 \cap \Sigma_3$. Thus $s_1 \xrightarrow{a?} s'_1$ and $s_3 \xrightarrow{a?} s'_3$. By simulation definition we have $s_2 \overset{a?}{\Longrightarrow} s'_2$ and $s_4 \overset{a?}{\Longrightarrow} s'_4$ for some $s'_2 \in S_2, s'_4 \in S_4$ such that $(s'_1, s'_2) \in R_1$ and $(s'_3, s'_4) \in R_2$. Hence $(\langle s'_1, s'_3\rangle, \langle s'_2, s'_4\rangle) \in R$, and from 1 of Lemma 2 there exists $\langle s_2, s_4\rangle \overset{a?}{\Longrightarrow} \langle s'_2, s'_4\rangle$ in $\mathscr{T}_2 \| \mathscr{T}_4$.

Case 2: $a \in \Sigma_2 \oplus \Sigma_4$. Without loss of generality, we assume that $a \in \Sigma_2$ and $a \notin \Sigma_4$. Since $\Sigma_2 \subseteq \Sigma_1$ and $\Sigma_4 \subseteq \Sigma_3$, we have $a \in \Sigma_1 \cap \Sigma_3$ or $a \in \Sigma_1 \setminus \Sigma_3$. If $a \in \Sigma_1 \cap \Sigma_3$ then $s_1 \xrightarrow{a?} s'_1$ and $s_3 \xrightarrow{a?} s'_3$. By simulation definition we have that $s_2 \overset{a?}{\Longrightarrow} s'_2$ and $s_4 \overset{\mathbf{0}}{\Longrightarrow} s'_4$ for some $s'_2 \in S_2, s'_4 \in S_4$ such that $(s'_1, s'_2) \in R_1$ and $(s'_3, s'_4) \in R_2$. Hence $(\langle s'_1, s'_3\rangle, \langle s'_2, s'_4\rangle) \in R$, and from 4 of Lemma 2 there exists $\langle s_2, s_4\rangle \overset{a?}{\Longrightarrow} \langle s'_2, s'_4\rangle$ in $\mathscr{T}_2 \| \mathscr{T}_4$. Otherwise $a \in \Sigma_1 \setminus \Sigma_3$ then $s_1 \xrightarrow{a?} s'_1$ and $s_3 = s'_3$. From $\mathscr{T}_1 \preceq \mathscr{T}_2$, we have $s_2 \overset{a?}{\Longrightarrow} s'_2$ for some $s'_2 \in S_2$ such that $(s'_1, s'_2) \in R_1$. Hence $(\langle s'_1, s_3\rangle, \langle s'_2, s_4\rangle) \in R$. From 4 of Lemma 2 there exists $\langle s_2, s_4\rangle \overset{a?}{\Longrightarrow} \langle s'_2, s_4\rangle$ in $\mathscr{T}_2 \| \mathscr{T}_4$.

Suppose $\langle s_1, s_3\rangle \xrightarrow{a!} \langle s'_1, s'_3\rangle$ for some $a \in \Sigma_{2\|4}$. There are also two cases:

Case 1: $a \in \Sigma_2 \cap \Sigma_4$. Since $\mathscr{T}_1$ and $\mathscr{T}_3$ are compatible and $\Sigma_2 \cap \Sigma_4 \subseteq \Sigma_1 \cap \Sigma_3$, $a \in \Sigma_1 \cap \Sigma_3 \cap \Sigma^b$. Thus we have that $s_1 \xrightarrow{a?} s'_1$ and $s_3 \xrightarrow{a!} s'_3$, or $s_1 \xrightarrow{a!} s'_1$ and $s_3 \xrightarrow{a?} s'_3$. By simulation definition we have $s_2 \overset{a?}{\Longrightarrow} s'_2$, $s_4 \overset{a!}{\Longrightarrow} s'_4$, and $s_2 \overset{a!}{\Longrightarrow} s'_2$, $s_4 \overset{a?}{\Longrightarrow} s'_4$ respectively, for some $s'_2 \in S_2, s'_4 \in S_4$ such that $(s'_1, s'_2) \in R_1$ and $(s'_3, s'_4) \in R_2$. Hence $(\langle s'_1, s'_3\rangle, \langle s'_2, s'_4\rangle) \in R$, and from 2 of Lemma 2 there exists $\langle s_2, s_4\rangle \overset{a!}{\Longrightarrow} \langle s'_2, s'_4\rangle$ in $\mathscr{T}_2 \| \mathscr{T}_4$.

Case 2: $a \in \Sigma_2 \oplus \Sigma_4$. Table 2 shows the possible transitions in $\mathscr{T}_1$ and $\mathscr{T}_3$. From the assumption that $I_2 \cap O_3 \subseteq \Sigma_4$ and $O_1 \cap I_4 \subseteq \Sigma_2$, there exist $a \in \Sigma_4$ in No. 1 and $a \in \Sigma_2$ in No. 5, which contradict their premises $a \in \Sigma_2 \setminus \Sigma_4$ and $a \in \Sigma_4 \setminus \Sigma_2$ respectively. Thus the cases of No. 1 and No. 5 will not exist. Consider the other cases in Table 2. By $\mathscr{T}_1 \preceq \mathscr{T}_2$ and $\mathscr{T}_3 \preceq \mathscr{T}_4$ we have that (st_2, Tr_2, st'_2) in $\rightarrow_2$ and (st_4, Tr_4, st'_4) in $\rightarrow_4$ for some $st'_2 \in S_2, st'_4 \in S_4$ such that $(st'_1, st'_2) \in R_1$ and $(st'_3, st'_4) \in R_2$. Hence $(\langle st'_1, st'_3\rangle, \langle st'_2, st'_4\rangle) \in R$, and from Lemma 2 there exists $\langle st_2, st_4\rangle \overset{a!}{\Longrightarrow} \langle st'_2, st'_4\rangle$ in $\mathscr{T}_2 \| \mathscr{T}_4$.

Suppose $\langle s_1, s_3\rangle \xrightarrow{a} \langle s'_1, s'_3\rangle$ for some $a \in \Sigma_{1\|3} \setminus \Sigma_{2\|4}$. Since $\Sigma_{2\|4} \subseteq (\Sigma_2 \cup \Sigma_4)$, there are the following two cases:

Case 1: $a \in \Sigma_{1\|3} \setminus (\Sigma_2 \cup \Sigma_4)$. Table 3 shows the possible transitions in $\mathscr{T}_1$ and $\mathscr{T}_3$. By $\mathscr{T}_1 \preceq \mathscr{T}_2$ and $\mathscr{T}_3 \preceq \mathscr{T}_4$ we have that (st_2, Tr_2, st'_2) in $\rightarrow_2$ and (st_4, Tr_4, st'_4) in $\rightarrow_4$ for some $st'_2 \in S_2, st'_4 \in S_4$ such that $(st'_1, st'_2) \in R_1$ and $(st'_3, st'_4) \in R_2$. Hence $(\langle st'_1, st'_3\rangle, \langle st'_2, st'_4\rangle) \in R$, and from Lemma 2 there exists $\langle st_2, st_4\rangle \overset{\mathbf{0}}{\Longrightarrow} \langle st'_2, st'_4\rangle$ in $\mathscr{T}_2 \| \mathscr{T}_4$.

Case 2: $a \in (\Sigma_2 \cup \Sigma_4) \setminus \Sigma_{2\|4}$. Since $(\Sigma_2 \cup \Sigma_4) \setminus \Sigma_{2\|4} \subseteq (I'_2 \cap O_2) \cup (I'_4 \cap O_4) \cup (I'_2 \cap O_4) \cup (I'_4 \cap O_2)$, we have $a \in (I_2 \cap O_4) \cup (I_4 \cap O_2)$ and $a \in \Sigma^u$, which contradicts the fact that $\Sigma_2 \subseteq \Sigma^b$ and $\Sigma_4 \subseteq \Sigma^b$. Hence such an action a does not exist in this case.

Table 3: Transition Set 2

Premises	$\mathscr{T}_1$			$\mathscr{T}_2$			$\mathscr{T}_3$			$\mathscr{T}_4$		
	st_1	Tr_1	st_1'	st_2	Tr_2	st_2'	st_3	Tr_3	st_3'	st_4	Tr_4	st_4'
$a \in I_{1\|3}$	s_1	$\xrightarrow{a?}$	s_1'	s_2	$\xRightarrow{\mathbf{0}}$	s_2'	s_3	-	s_3	s_4	-	s_4
	s_1	-	s_1	s_2	-	s_2	s_3	$\xrightarrow{a?}$	s_3'	s_4	$\xRightarrow{\mathbf{0}}$	s_4'
	s_1	$\xrightarrow{a?}$	s_1'	s_2	$\xRightarrow{\mathbf{0}}$	s_2'	s_3	$\xrightarrow{a?}$	s_3'	s_4	$\xRightarrow{\mathbf{0}}$	s_4'
$a \in O_{1\|3}$	s_1	$\xrightarrow{a!}$	s_1'	s_2	$\xRightarrow{\mathbf{0}}$	s_2'	s_3	-	s_3	s_4	-	s_4
	s_1	-	s_1	s_2	-	s_2	s_3	$\xrightarrow{a!}$	s_3'	s_4	$\xRightarrow{\mathbf{0}}$	s_4'
	s_1	$\xrightarrow{a?}$	s_1'	s_2	$\xRightarrow{\mathbf{0}}$	s_2'	s_3	$\xrightarrow{a!}$	s_3'	s_4	$\xRightarrow{\mathbf{0}}$	s_4'
	s_1	$\xrightarrow{a!}$	s_1'	s_2	$\xRightarrow{\mathbf{0}}$	s_2'	s_3	$\xrightarrow{a?}$	s_3'	s_4	$\xRightarrow{\mathbf{0}}$	s_4'

Table 4: Transition Set 3

No.	$\mathscr{T}_1$			$\mathscr{T}_2$			$\mathscr{T}_3$			$\mathscr{T}_4$		
	st_1	Tr_1	st_1'	st_2	Tr_2	st_2'	st_3	Tr_3	st_3'	st_4	Tr_4	st_4'
1	s_1	$\xrightarrow{\tau}$	s_1'	s_2	$\xRightarrow{\mathbf{0}}$	s_2'	s_3	-	s_3	s_4	-	s_4
2	s_1	-	s_1	s_2	-	s_2	s_3	$\xrightarrow{\tau}$	s_3'	s_4	$\xRightarrow{\mathbf{0}}$	s_4'
3	s_1	$\xrightarrow{a?}$	s_1'	s_2	$\xRightarrow{\mathbf{0}}$	s_2'	s_3	$\xrightarrow{a!}$	s_3'	s_4	$\xRightarrow{\mathbf{0}}$	s_4'
4	s_1	$\xrightarrow{a!}$	s_1'	s_2	$\xRightarrow{\mathbf{0}}$	s_2'	s_3	$\xrightarrow{a?}$	s_3'	s_4	$\xRightarrow{\mathbf{0}}$	s_4'

Suppose $\langle s_1, s_3\rangle \xrightarrow{\tau} \langle s_1', s_3'\rangle$. Table 4 shows the possible transitions in $\mathscr{T}_1$ and $\mathscr{T}_3$. Note that $a \in \Sigma^u$ and $a \notin \Sigma_2 \cup \Sigma_4 \subseteq \Sigma^b$ in No. 3 and No. 4. By $\mathscr{T}_1 \preceq \mathscr{T}_2$ and $\mathscr{T}_3 \preceq \mathscr{T}_4$ we have that (st_2, Tr_2, st_2') in $\rightarrow_2$ and (st_4, Tr_4, st_4') in $\rightarrow_4$ for some $st_2' \in S_2, st_4' \in S_4$ such that $(st_1', st_2') \in R_1$ and $(st_3', st_4') \in R_2$. Hence $(\langle st_1', st_3'\rangle, \langle st_2', st_4'\rangle) \in R$, and from Lemma 2 there exists $\langle st_2, st_4\rangle \xRightarrow{\mathbf{0}} \langle st_2', st_4'\rangle$ in $\mathscr{T}_2 \| \mathscr{T}_4$.

Suppose $\langle s_1, s_3\rangle \xrightarrow{\varepsilon(d)} \langle s_1', s_3'\rangle$ and $d > 0$. Thus $s_1 \xrightarrow{\varepsilon(d)} s_1'$ and $s_3 \xrightarrow{\varepsilon(d)} s_3'$. By simulation definition we have that $s_2 \xRightarrow{\varepsilon(d)} s_2'$ and $s_4 \xRightarrow{\varepsilon(d)} s_4'$ for some $s_2' \in S_2, s_4' \in S_4$ such that $(s_1', s_2') \in R_1$ and $(s_3', s_4') \in R_2$. Hence $(\langle s_1', s_3'\rangle, \langle s_2', s_4'\rangle) \in R$, and from 5 of Lemma 2 there exists $\langle s_2, s_4\rangle \xRightarrow{\varepsilon(d)} \langle s_2', s_4'\rangle$ in $\mathscr{T}_2 \| \mathscr{T}_4$. All the conditions hold and thus $\mathscr{T}_1 \| \mathscr{T}_3 \preceq \mathscr{T}_2 \| \mathscr{T}_4$. $\square$

B Avionics Workload

As shown in Table 5, the workload is comprised of 5 partitions ($P_1 - P_5$), and further divided into 18 periodic tasks and 4 sporadic tasks. The type of a task depends on its *release* interval. A periodic task has a deterministic period, whereas the release time of a sporadic task is only bounded by a minimum separation. The execution of a task is characterized as a sequence of *chunks*. Each chunk involves the description of a non-deterministic *execution time*, required resources and message-passing operations. There are 3 intra-partition locks(column *mutex*) and 4 inter-partition message types defined in the task set. The columns *output* and *input* indicate transfer direction of messages.

Table 5: The Workload of the Avionics System[6, 11](Times in Milliseconds)

No.	Task	Release	Offset	Jitter	Deadline	Priority	Execution Chunks			
							Time	Mutex	Output	Input
P_1	Tsk_1^1	[25,25]	2	0	25	2	[0.8,1.3]	-	-	-
							[0.1,0.2]	-	-	-
	Tsk_2^1	[50,50]	3	0	50	3	[0.2,0.4]	-	Msg_1	-
	Tsk_3^1	[50,50]	3	0	50	4	[2.7,4.2]	-	-	-
	Tsk_4^1	[50,50]	0	0	50	5	[0.1,0.2]	Mux_1^1	-	-
	Tsk_5^1	[120,∞)	0	0	120	6	[0.6,0.9]	-	-	-
							[0.1,0.2]	Mux_1^1	-	-
P_2	Tsk_1^2	[50,50]	0	0.5	50	2	[1.9,3.0]	-	-	-
	Tsk_2^2	[50,50]	2	0	50	3	[0.7,1.1]	-	Msg_2	-
	Tsk_3^2	[100,100]	0	0	100	4	[0.1,0.2]	Mux_1^2	-	-
	Tsk_4^2	[100,∞)	10	0	100	5	[0.8,1.3]	-	-	-
							[0.2,0.3]	Mux_1^2	-	-
P_3	Tsk_1^3	[25,25]	0	0.5	25	2	[0.5,0.8]	-	-	Msg_1
	Tsk_2^3	[50,50]	0	0	50	3	[0.7,1.1]	-	-	Msg_2
	Tsk_3^3	[50,50]	0	0	50	4	[1.0,1.6]	-	-	Msg_3
	Tsk_4^3	[100,∞)	11	0	100	5	[0.7,1.0]	-	-	-
							[0.1,0.3]	-	-	-
P_4	Tsk_1^4	[25,25]	3	0.2	25	2	[0.7,1.2]	-	-	-
	Tsk_2^4	[50,50]	5	0	50	3	[1.2,1.9]	-	Msg_3	Msg_1
	Tsk_3^4	[50,50]	25	0	50	4	[0.1,0.2]	-	-	Msg_4
	Tsk_4^4	[100,100]	11	0	100	5	[0.7,1.1]	-	-	-
	Tsk_5^4	[200,200]	13	0	200	6	[3.7,5.8]	-	-	-
P_5	Tsk_1^5	[50,50]	0	0.3	50	1	[0.7,1.1]	-	-	Msg_1
	Tsk_2^5	[50,50]	2	0	50	2	[1.2,1.9]	-	Msg_4	Msg_2
	Tsk_3^5	[200,200]	0	0	200	3	[0.4,0.6]	-	-	-
							[0.2,0.3]	Mux_1^5	-	-
	Tsk_4^5	[200,∞)	14	0	200	4	[1.4,2.2]	-	-	-
							[0.1,0.2]	Mux_1^5	-	-

C AFDX Configuration

The AFDX configuration in Table 6 is based on the case of [13]. There are four message types $Msg_i, i = \{1,2,3,4\}$, each of which is allocated to a separate VL with the same subscript shown in column "VL". The column "Length" indicates the length of a message sent from an ARINC-653 partition. For any VL in the configuration, the columns "BAG" and "L_{max}" denote its Bandwidth Allocation Gap and Maximum packet Length respectively. The source and destination partition(s) are given in the columns "Source" and "Destination" respectively.

Table 6: The AFDX Configuration in the Case Study (Times in Milliseconds and Sizes in Bytes)

Message	Length	VL	BAG	L_{max}	Source	Destinations
Msg_1	306	V_1	8	200	P_1	P_3,P_4,P_5
Msg_2	953	V_2	16	1000	P_2	P_3,P_5
Msg_3	453	V_3	32	500	P_4	P_3
Msg_4	153	V_4	32	200	P_5	P_4