A Temporal Approach for Testing Distributed Systems

Ahmed Khoumsi, *Member*, *IEEE*

Abstract—This paper deals with testing distributed software systems. In the past, two important problems have been determined for executing tests using a distributed test architecture: *controllability* and *observability* problems. A coordinated test method has subsequently been proposed to solve these two problems. In the present article: 1) we show that controllability and observability are indeed resolved if and only if the test system respects some timing constraints, even when the system under test is non-real-time; 2) we determine these timing constraints; 3) we determine other timing constraints which optimize the duration of test execution; 4) we show that the communication medium used by the test system has not to be necessarily FIFO; and 5) we show that the centralized test method can be considered just as a particular case of the proposed coordinated test method.

Index Terms—Conformance testing, distributed systems, controllability, observability, reaction time, waiting time.

1 Introduction

Testing, which aims to ensure the quality of the implementation, is realized by generating test sequences and applying them to the implementation which is referred to as an *Implementation Under Test* ($\mathcal{IUT}$). In this article, we consider the case when the $\mathcal{IUT}$ is distributed and we study the *test execution phase* (i.e., the phase when test sequences are applied). Here are the few works which inspired us:

- In [1], a distributed architecture for testing distributed $\mathcal{IUT}$s has been studied. In this architecture, the $\mathcal{IUT}$ contains several ports and the test system ($\mathcal{TS}$) consists of a local tester for each port of the $\mathcal{IUT}$. Each local tester communicates with the $\mathcal{IUT}$ through its corresponding port (see Fig. 1b). Two important problems which occur in the phase of *test execution* are determined in [1]: synchronization and fault detectability problems.
- In [2], [3], the problems of synchronization and fault detectability have been defined in terms of *controllability* and *observability*, respectively, and a *coordinated* test architecture is proposed to solve them (see Fig. 1c). The approach of resolution consists of allowing the local testers to exchange coordination messages with one another, through a reliable communication medium which is independent of the $\mathcal{IUT}$.
- In [4], certain timing constraints are given and it is stated, without any proof, that controllability and observability problems can actually be resolved if and only if the $\mathcal{TS}$ respects these timing constraints.

In this article, we propose a testing method which validates and improves [4] as follows:

1. Correctness of all timing constraints of [4] are proven. In other words, we prove that the timing constraints of [4] solve controllability and observability problems. Certain errors in [4] have also been corrected.
2. We determine other timing constraints which optimize duration of test execution. More precisely, we determine the minimal times the $\mathcal{TS}$ has to wait for expected outputs of the $\mathcal{IUT}$ before deducing whether the $\mathcal{IUT}$ is faulty.
3. We show that our test method does not require a FIFO communication medium.
4. We show that the centralized test method can be considered just as a particular case of the proposed coordinated test method. This implies that:
   - it is useless to do another study for a centralized test architecture and
   - the $\mathcal{TS}$ must respect timing constraints even when the test architecture is centralized.

Several other works have been written for testing distributed systems [5], [6], [7], [8], but they do not resolve the problems we will consider. In [5], the authors propose a coordination procedure between testers and they study how the test is affected by the transmission between the $\mathcal{TS}$ and the $\mathcal{IUT}$. In [6], the authors more thoroughly study the influence on testing of the transmission between the $\mathcal{TS}$ and the $\mathcal{IUT}$. There are several limitations of [5], [6] in comparison with our study. First, in [6], concrete results are obtained only when the $\mathcal{IUT}$ and the $\mathcal{TS}$ communicate through a *single* port. Second, in [5], [6], only the order of events is taken into account, while the causality relation between inputs and outputs is ignored. A consequence of this limitation is that controllability and observability problems are not resolved. Third, in [5], [6], the duration of test execution is not studied.

• *The author is with the Université de Sherbrooke, 2500, Boulevard de l'Université, Sherbrooke, Canada J1K 2R1.*
E-mail: ahmed.khoumsi@usherbrooke.ca.

Manuscript received 22 Dec. 2000; revised 15 June 2001; accepted 31 Jan. 2002.
Recommended for acceptance by Luqi.
For information on obtaining reprints of this article, please send e-mail to: tse@computer.org, and reference IEEECS Log Number 113355.

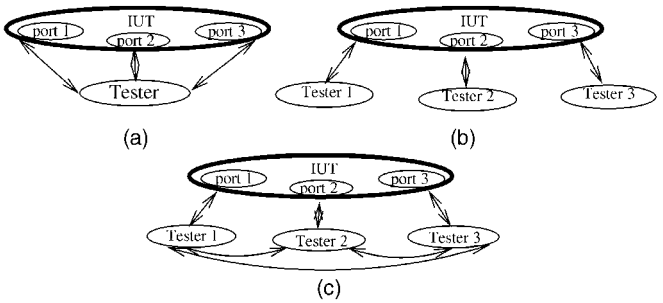

Fig. 1. Architectures for testing distributed systems. (a) Centralized test architecture. (b) Distributed test architecture. (c) Coordinated test architecture.

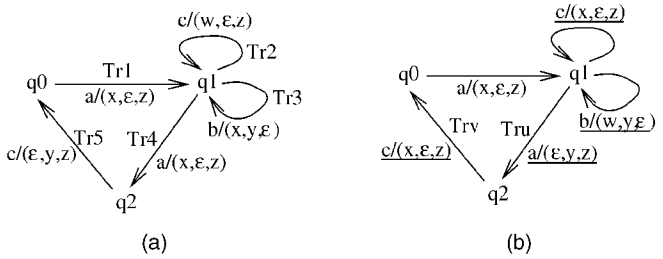

Fig. 2. Two examples of np-FSMs. (a) Specification and (b) faulty implementation.

In [7], the authors study the *generation* and *selection* of test cases which maximize resources utilization. In [8], the authors study the generation of test cases for a particular distributed routing protocol for Internet. In comparison with our article, in [7], [8], the *execution* of test cases and the *coordination* of testers are not studied.

The rest of this article is structured as follows: In Section 2, we introduce the communication model, the model used to describe the $\mathcal{IUT}$, and several hypotheses and concepts related to testing. In Section 3, we present in more detail the concepts of controllability and observability. In Section 4, we give a few definitions and present our objective. In Section 5, we present the coordinated test method, define reaction times and waiting times of the $\mathcal{TS}$, and define our objective more accurately. In the same section, we prove that our method is insensitive to the fact that the communication medium respects or does not respect the FIFO discipline (first-in, first-out). In Section 6, we present constraints of reaction times and waiting times of the $\mathcal{TS}$ which resolve controllability and observability problems and optimize duration of test execution, respectively. We also show that the centralized test method can be considered just as a particular case of the coordinated test method. In Section 7, we conclude by summarizing our contributions and discussing some future research issues. For clarity, the proofs of all results of Sections 5 and 6 have been put in Appendices A, B, and C.

2 Modeling and Testing Concepts

As we will see later, in this study, we adopt a coordinated test architecture. In the present section, we first present the communication model used in this architecture. Then, we present the np-FSM model that is used to describe the specification of the $\mathcal{IUT}$. After that, we present hypotheses and concepts related to testing.

2.1 Communication Model

A local $Tester_p$ communicates with port p of the $\mathcal{IUT}$ through a reliable communication medium denoted as CM_p. Two testers $Tester_p$ and $Tester_q$ communicate with one another through a reliable communication medium denoted as $CM^{ts}_{p,q}$. Each CM_p and $CM^{ts}_{p,q}$ is assumed reliable (i.e., no message loss and finite transmission delay).

The transfer times in all CM_p (respectively, $CM^{ts}_{p,q}$) are assumed bounded by a *finite* value TT^{max} (respectively, TT^{max}_{ts}) that can be determined. This hypothesis is realistic because the advent of real-time middlewares such as real-time CORBA [9] is foreseen, probably in a near future.

We assume that each tester and each port of the $\mathcal{IUT}$ uses its own local clock and that the local clocks are not synchronized, that is, there is no global clock. This implies that the transit time of a message (in CM_p or $CM^{ts}_{p,q}$) cannot be measured by reading the local clocks of the sender and of the receiver, at instants of sending and reception, respectively.

As we will prove it in Section 5.6, our test method guarantees that at any time each CM_p and $CM^{ts}_{p,q}$ contains at most a single message and, thus, our method is insensitive to the fact that the communication medium respects or does not respect the FIFO discipline.

2.2 Finite State Machine with n Ports np-FSM

A np-FSM is a 6-tuple $(Q, I, O, \delta, \sigma, q_0)$, where $n \geq 1$ and:

- Q is a finite set of states and $q_0 \in Q$ is the initial state.
- I is a n-tuple $(I_1, I_2, \ldots I_n)$, where I_i is a finite set of inputs of port i, $I_i \cap I_j = \emptyset$ for $i \neq j$ and $i, j = 1, \cdots, n$. Then, let $\mathcal{I} = I_1 \cup I_2 \cup \ldots \cup I_n$.
- O is a n-tuple $(O_1, O_2, \ldots O_n)$, where O_i is a finite set of outputs of port i, $O_i \cap O_j = \emptyset$ for $i \neq j$ and $i, j = 1, \cdots, n$. Then, let

$$\mathcal{O} = (O_1 \cup \{\varepsilon\}) \times (O_2 \cup \{\varepsilon\}) \times \ldots \times (O_n \cup \{\varepsilon\}),$$

  where ε stands for the empty output.
- δ is a transition function: $\mathcal{D} \longrightarrow Q$, and σ is an output function: $\mathcal{D} \longrightarrow \mathcal{O}$, where $\mathcal{D} \subseteq Q \times \mathcal{I}$.

Two 3p-FSMs are represented in Fig. 2, with $I_1 = \{a\}$, $I_2 = \{b\}$, $I_3 = \{c\}$, $O_1 = \{w, x\}$, $O_2 = \{y\}$, and $O_3 = \{z\}$. The nodes are the states and the directed edges are the transitions linking the states. A label α/γ on an edge linking q and q' means $\delta(q, \alpha) = q'$ and $\sigma(q, \alpha) = \gamma$. For example, if q_0 is the current state and the input a is received, then the state changes to q_1 and the outputs x and z are sent in ports 1 and 3, respectively.

Notation 1 (!x, ?x, Υ, y^k, !Υ, ?Υ). *The sending (respectively, reception) of an input or output x is denoted !x (respectively, ?x). Let $\Upsilon = (y_1, \cdots, y^n)$ be a n-tuple, where y^k is an output (possibly empty) in port k. Formally, $\Upsilon \in \mathcal{O}$. The sending by the $\mathcal{IUT}$ (respectively, reception by the $\mathcal{TS}$) of all the outputs of Υ, in any order, is denoted !Υ (respectively, ?Υ).*

2.3 Hypotheses and Concepts of Testing

Conformance testing consists of checking whether an $\mathcal{IUT}$ conforms to a specification SPEC.

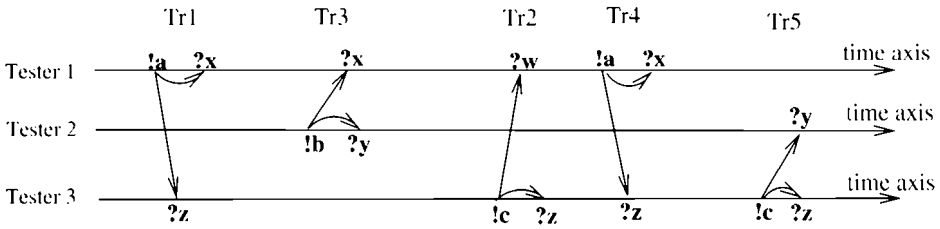

Fig. 3. Projection of a global test sequence.

Hypothesis 1. *Similarly to [1], [2], [3], SPEC is assumed to be described by a* deterministic np-FSM. *And similarly to [10], we assume that the behavior of the $\mathcal{IUT}$ (even when it is faulty) can be described by the model used to describe its specification SPEC, in our case a deterministic np-FSM, but the latter can be unknown.*

Property 1. *From the assumption that the $\mathcal{IUT}$ can be described by a np-FSM, we deduce that the $\mathcal{IUT}$ is* reactive, *that is, outputs are sent only in response to the reception of an input. In other terms, outputs are not spontaneous.*

Property 2. *From the assumption that the $\mathcal{IUT}$ can be described by a np-FSM, we deduce that the $\mathcal{IUT}$ reacts to an input by sending at most a single output in each port.*

In the following two definitions, we consider an np-FSM $A = (Q, I, O, \delta, \sigma, q_0)$ which describes SPEC (specification of the $\mathcal{IUT}$), and $\lambda = \langle x_1/\Upsilon_1\rangle\langle x_2/\Upsilon_2\rangle \cdots \langle x_t/\Upsilon_t\rangle$ which is a transition sequence of A, where $x_i \in \mathcal{I}$ and $\Upsilon_i \in \mathcal{O}$.

Definition 1 ($\phi(\lambda)$, conformance). *We consider the sequence $\phi(\lambda) = ?x_1!\Upsilon_1?x_2!\Upsilon_2 \cdots ?x_t!\Upsilon_t$. A $\mathcal{IUT}$ is conformant to λ iff: When the $\mathcal{IUT}$ receives the input sequence $x_1, x_2 \cdots x_t$, then it executes $\phi(\lambda)$.*

Definition 1 holds only with Hypothesis 1. An $\mathcal{IUT}$ is conformant to A iff it is conformant to every transition sequence of A.

Definition 2 ($\omega(\lambda)$, global test sequence (GTS)). *The sequence $\omega(\lambda) = !x_1?\Upsilon_1!x_2?\Upsilon_2 \cdots !x_t?\Upsilon_t$ is called global test sequence (GTS).*

$\phi(\lambda)$ and $\omega(\lambda)$ can be defined intuitively as follows:

- The $\mathcal{IUT}$ is conformant to λ if the $\mathcal{IUT}$ can execute $\phi(\lambda)$.
- With a centralized test method (see Fig. 1a), the $\mathcal{TS}$ deduces that the $\mathcal{IUT}$ is conformant to λ if the $\mathcal{TS}$ can execute $\omega(\lambda)$.

The centralized test method is therefore correct if we have the following equivalence: The $\mathcal{IUT}$ executes $\phi(\lambda)$ iff the $\mathcal{TS}$ executes $\omega(\lambda)$. Actually, the equivalence holds iff certain timing constraints are satisfied by the $\mathcal{TS}$. An example of timing constraint will be given in Example 4 (Section 5.5).

Let us consider the $\mathcal{IUT}$ of Fig. 2b, which contains four faults (which are underlined) with regard to the specification of Fig. 2a. Let us consider the following sequence which corresponds to the sequence of transitions Tr1 Tr3 Tr2 Tr4 Tr5 of Fig. 2a:

$$\lambda = \langle a/(x,\varepsilon,z)\rangle\langle b/(x,y,\varepsilon)\rangle\langle c/(w,\varepsilon,z)\rangle\langle a/(x,\varepsilon,z)\rangle\langle c/\varepsilon,y,z)\rangle. \quad (1)$$

Its corresponding GTS is:

$$\omega(\lambda) = !a?(x,\varepsilon,z)!b?(x,y,\varepsilon)!c?(w,\varepsilon,z)!a?(x,\varepsilon,z)!c?(\varepsilon,y,z). \quad (2)$$

With a centralized test method, the conformance of the faulty $\mathcal{IUT}$ to λ can be checked as follows, according to $\omega(\lambda)$. The $\mathcal{IUT}$ being initially in state q_0, the $\mathcal{TS}$ sends a and then receives the expected outputs x and z, in ports 1 and 3, respectively (transition Tr1). After that, the $\mathcal{TS}$ sends b and receives the outputs w and y, in ports 1 and 2, respectively. The nonconformance is detected because the expected output in port 1 is x instead of the received w.

Definition 3 (Local test sequence LTS in a distributed architecture). *With a distributed test architecture (see Fig. 1b), conformance to a sequence λ cannot be checked by using directly the corresponding GTS $\omega(\lambda)$. Instead, each $Tester_p$ (tester in port p) uses a local test sequence (LTS) which is obtained by projecting the GTS in port p. We might think that a $\mathcal{IUT}$ is conformant to a sequence λ iff each $Tester_p$ executes its LTS. We will see in Section 3 that this view is incorrect.*

The LTSs obtained from the GTS (2) are represented in Fig. 3, where each input is linked by arrows to the outputs of the same transition.

We obtain therefore the following LTSs ω_1, ω_2, and ω_3 for testers 1, 2, and 3, respectively:

$$\begin{cases} \omega_1 = !a?x?x?w!a?x \\ \omega_2 = !b?y?y \\ \omega_3 = ?z!c?z?z!c?z. \end{cases} \quad (3)$$

3 Controllability and Observability Issues

In this section, we present controllability and observability that are two important issues in testing because they have an effect on the capability of the $\mathcal{TS}$ to check the conformance of an $\mathcal{IUT}$. In this section, we assume a distributed test architecture and the use of LTSs. For clarity, we will consider the specification and the faulty $\mathcal{IUT}$ of Fig. 2.

Definition 4 (Controllability). *Controllability is the capability of the $\mathcal{TS}$ to force the $\mathcal{IUT}$ to receive inputs in a given order. During conformance testing to a given sequence $\lambda = \langle x_1/\Upsilon_1\rangle\langle x_2/\Upsilon_2\rangle \cdots \langle x_t/\Upsilon_t\rangle$, a controllability problem arises when the $\mathcal{TS}$ cannot guarantee that the $\mathcal{IUT}$ will receive x_i before x_{i+1}, for $i < t$. With a distributed test*

architecture, such a problem arises when there exists $i < t$ such that the port q of x_{i+1}: 1) is different from the port p of x_i (i.e., $q \neq p$) and *2) is not included in the set of ports of Υ_i (i.e., $y_i^q = \varepsilon$) [1].*

Example 1 (Controllability problem). The sequence (1) requires that when state q_1 is reached (see Fig. 2a), the $\mathcal{IUT}$ receives b (sent by $Tester_2$) before it receives c (sent by $Tester_3$). To guarantee this order, $Tester_3$ needs to receive a message informing it that b has been received by the $\mathcal{IUT}$. With a distributed architecture, such a message may only come from the $\mathcal{IUT}$. Since $Tester_3$ does not receive any output of the $\mathcal{IUT}$ in response to b (see Tr3 of Fig. 2a), then $Tester_3$ cannot receive the information it needs. In other terms, $Tester_3$ has no means to determine the order of inputs b and c. Here is an example which shows the effect of this problem on fault detectability. From state q_1, the three testers (i.e., the $\mathcal{TS}$) observe the same outputs in the following two situations: 1) a correct $\mathcal{IUT}$ receives c before b and 2) the faulty $\mathcal{IUT}$ receives b before c. The $\mathcal{TS}$ cannot deduce whether the $\mathcal{IUT}$ is correct or faulty because it is not aware of the order of inputs b and c.

Definition 5 (Observability). *Observability is the capability of the $\mathcal{TS}$ to observe the outputs of the $\mathcal{IUT}$ and to determine the input which is the cause of every output. During conformance testing to a given sequence $\lambda = \langle x_1/\Upsilon_1\rangle\langle x_2/\Upsilon_2\rangle \cdots \langle x_t/\Upsilon_t\rangle$, an observability problem arises when the $\mathcal{TS}$ receives $a \in \Upsilon_i$ and cannot determine whether a has been sent by the $\mathcal{IUT}$ after the latter has received x_i and before it (the $\mathcal{IUT}$) receives x_{i+1} [1]. With a distributed test architecture and with our model where a transition contains at most a single output for each port, such a problem arises when, for a port p, λ contains two consecutive transitions such that only one of the two transitions contains an output in port p [1].*

Example 2 (Observability problem). In Fig. 2b, let us consider the consecutive transitions Tru and Trv, from q_1 to q_2 and from q_2 to q_0, respectively. If we compare with the specification of Fig. 2a, x has been "shifted" from Tr4 to Tr5, and y has been "shifted" from Tr5 to Tr4. With a distributed architecture, these faults are not detected because, although the $\mathcal{IUT}$ is faulty, all the testers execute (and observe) exactly the LTSs (3) generated from GTS (2).

4 Certain Timing Definitions and Our Objective

Here are a few definitions and conditions which will be necessary to define our objective in a clear and concise manner. We consider a sequence

$$\lambda = \langle x_1/\Upsilon_1\rangle\langle x_2/\Upsilon_2\rangle \cdots \langle x_t/\Upsilon_t\rangle.$$

Definition 6 (Instant, τ_e and τ_E). *In this article, the term "instant" means "instant relatively to a fictitious global clock." This implies that the delay which separates two of any events (possibly distant) is the difference of their instants. Since the local clocks of the ports of $\mathcal{IUT}$ and of the testers are not assumed synchronized, the delay between two distant events cannot be measured just by reading the local clocks corresponding to the two events at their instants of occurrence.*

Let e be an event and E be a set (or n-tuple) of events. τ_e denotes the instant of e and τ_E denotes the instant when all the events of E have occurred.

Definition 7 (Reaction time of the $\mathcal{IUT}$, RT_{iut}). *Reaction time of the $\mathcal{IUT}$, denoted RT_{iut}, is an upper bound of the time separating: 1) any instant when an event e is received by the $\mathcal{IUT}$ and 2) the instant when the $\mathcal{IUT}$ has terminated to send* all *the outputs (if any) in response to the reception of e. We emphasize the word "all" (also in Definition 8) because the definition includes possible* unexpected *outputs (in the case of a nonconformant $\mathcal{IUT}$). In an execution conformant to λ, this definition implies, for $i \leq t$: $RT_{iut} \geq (\tau_{!\Upsilon_i} - \tau_{?x_i})$. RT_{iut}, which is guaranteed by the $\mathcal{IUT}$, is assumed finite. Intuitively, RT_{iut} quantifies the* promptness *of the $\mathcal{IUT}$ to react to an input.*

Definition 8 (Waiting time of the $\mathcal{IUT}$, WT_{iut}). *Waiting time of the $\mathcal{IUT}$, denoted WT_{iut}, is an upper bound of the time separating: 1) any instant when the $\mathcal{IUT}$ has terminated to send* all *the outputs (if any) in response to the reception of an input a and 2) the instant when the next input* must *be received by the $\mathcal{IUT}$. If there is no output in response to a, then WT_{iut} is an upper bound of the time separating the instant when the $\mathcal{IUT}$ receives a and the instant when it* must *receive the next input. In an execution conformant to λ, this definition implies, for $i < t$: if $\Upsilon_i \neq \epsilon$: $WT_{iut} \geq (\tau_{?x_{i+1}} - \tau_{!\Upsilon_i})$ and if $\Upsilon_i = \epsilon$: $WT_{iut} \geq (\tau_{?x_{i+1}} - \tau_{?x_i})$. WT_{iut}, which is required by the $\mathcal{IUT}$, is not assumed necessarily finite. Intuitively, WT_{iut} quantifies the* patience *of the $\mathcal{IUT}$ for receiving the next input after it has finished to sent its outputs (if any) or after it has received an input (if the latter causes no output). Therefore, an infinite WT_{iut} means that the $\mathcal{IUT}$ is infinitely patient.*

Definition 9 (Transfer time between the $\mathcal{IUT}$ and the $\mathcal{TS}$, $[TT^{min}; TT^{max}]$, $\triangle TT$). *Transfer time between the $\mathcal{IUT}$ and the $\mathcal{TS}$ is assumed to fall within a bounded interval $[TT^{min}; TT^{max}]$. Formally, $TT^{min} \leq (\tau_{?e} - \tau_{!e}) \leq TT^{max}$, where e is any input or output. We use notation $\triangle TT = (TT^{max} - TT^{min})$. Intuitively, the transfer time of every message exchanged between a tester and a port of the $\mathcal{IUT}$ is bounded by finite values.*

Definition 10 (Transfer time between testers, $[TT_{ts}^{min}; TT_{ts}^{max}]$, $\triangle TT_{ts}$). *Assuming a coordinated test architecture, transfer time between testers is the time which separates the sending and the reception of a message exchanged between two testers. This time is assumed to fall within a bounded interval $[TT_{ts}^{min}; TT_{ts}^{max}]$. We use notation $\triangle TT_{ts} = (TT_{ts}^{max} - TT_{ts}^{min})$. Intuitively, the transfer time of every message exchanged between two testers is bounded by finite values.*

Remark 1. $TT^{min}, TT^{max}, TT_{ts}^{min}, TT_{ts}^{max}$ make sense because the communication between testers and between the $\mathcal{IUT}$ and testers is assumed reliable. Recall that the FIFO assumption is not necessary for the reason given in Section 2.1.

Condition 1. During conformance testing to every sequence λ: 1) the $\mathcal{IUT}$ receives inputs in the desired order

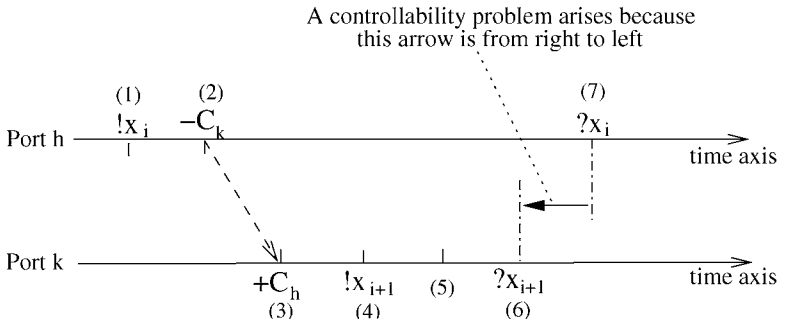

Fig. 4. Incorrect resolution of the problem of controllability.

(controllability) and 2) output faults are detected by the $\mathcal{TS}$ (observability). This condition means a resolution of controllability and observability problems.

Condition 2. Waiting time WT_{iut} (when it is finite) is a requirement of the $\mathcal{IUT}$ and must be guaranteed by the $\mathcal{TS}$. Note that this condition may be guaranteed only if TT^{max} is finite. More intuitively, we have to respect the upper bound of the patience of the $\mathcal{IUT}$.

Condition 3. The $\mathcal{TS}$ waits for every expected output of the $\mathcal{IUT}$ during a delay which is necessary and sufficient before to deduce whether the $\mathcal{IUT}$ is faulty. The term "necessary" means an optimization of the test duration. Note that this condition may be guaranteed only if RT_{iut} and TT^{max} are finite.

We can now define our objective as follows:

Hypothesis. *$RT_{iut}, WT_{iut}, [TT^{min}, TT^{max}]$, and $[TT^{min}_{ts}; TT^{max}_{ts}]$ are given.*

Objective 1. *To use a $\mathcal{TS}$ with a coordinated architecture (see Fig. 1c) and determine timing constraints of the $\mathcal{TS}$ which guarantee Conditions 1, 2, and 3.*

For simplicity, the objective presented in the abstract and in Section 1 did not include the guarantee of Condition 2.

5 Coordinated Test Method, Reaction, and Waiting Times of the $\mathcal{TS}$

In this section, we first present the coordinated test method which has been proposed in [2], [3] to solve controllability and observability problems which may arise in a distributed test architecture and we show that this method may generate incorrect results. Then, we propose a modification of coordination messages. After that, we define reaction times and waiting times of the $\mathcal{TS}$. We then propose another objective which guarantees Objective 1 presented at the end of Section 4. We terminate the section by proving that our method is insensitive to the fact that the communication medium respects or does not respect the FIFO discipline. In the present section, conformance is implicitly checked with regard to a given transition sequence $\lambda = \langle x_1/\Upsilon_1\rangle\langle x_2/\Upsilon_1\rangle \cdots \langle x_t/\Upsilon_t\rangle$.

5.1 Coordinated Test Method Proposed in [2], [3]

5.1.1 Approach to Solve the Controllability Problem

The controllability problem arises when, for a $i < t$, the tester which sends x_{i+1} cannot know whether x_i has been received by the $\mathcal{IUT}$. The solution proposed by [2], [3] can be explained as follows, for every $i < t$. Let $Tester_h$ and $Tester_k$ be the testers sending x_i and x_{i+1}, respectively. If $\Upsilon_i \neq \epsilon$, let $Tester_m$ be defined as follows:

- if $y^k_i \neq \varepsilon$: $Tester_m = Tester_k$;
- if $y^k_i = \varepsilon$: $Tester_m$ is any tester (arbitrarily selected) such that $y^m_i \neq \varepsilon$.

The controllability problem is then resolved by the use of a message C (Control) as follows:

- if (($\Upsilon_i = \epsilon$) and ($h \neq k$)): after it sends x_i, $Tester_h$ sends a message C to $Tester_k$;
- if (($\Upsilon_i \neq \epsilon$) and ($m \neq k$)): after it receives y^m_i, $Tester_m$ sends a message C to $Tester_k$.

In the above two cases, after it receives message C, $Tester_k$ sends x_{i+1}.

This approach may generate *incorrect* results. For example, when $\Upsilon_i = \epsilon$ this approach guarantees ($\tau_{!x_i} \leq \tau_{!x_{i+1}}$), while controllability problem is resolved if ($\tau_{?x_i} \leq \tau_{?x_{i+1}}$). This incorrectness is illustrated in Fig. 4, where each event is represented with its instant of occurrence relative to a fictitious global clock (see Definition 6). $-C_k$ means "send coordination message to $Tester_k$," and $+C_h$ means "receive coordination message from $Tester_h$." Intuitively, in certain cases, the above approach guarantees only the order in which inputs are *sent* by the $\mathcal{TS}$, instead of guaranteeing the order in which inputs are *received* by the $\mathcal{IUT}$. Recall that the local clocks of $Port_h$ and $Port_k$ are not synchronized and, thus, the incorrectness represented in Fig. 4 is not observable by reading the local clocks of $Port_h$ and $Port_k$ at the moment of time when $?x_i$ and $?x_{i+1}$ occur, respectively.

5.1.2 Approach to Solve the Observability Problem

The observability problem arises when, for a port p and $i < t$, *either* Υ_i *or* Υ_{i+1} (this is an exclusive OR) contains an output in port p [1]. The solution proposed in [2], [3] can be explained as follows, for every $i < t$. Let:

- $Tester_k$ be the tester sending x_{i+1};
- $Tester_p$ be any tester such that $p \neq k$, $y^p_i \neq \varepsilon$, $y^p_{i+1} = \varepsilon$, and, after it receives y^p_i, $Tester_p$ does not send a message C to $Tester_k$;

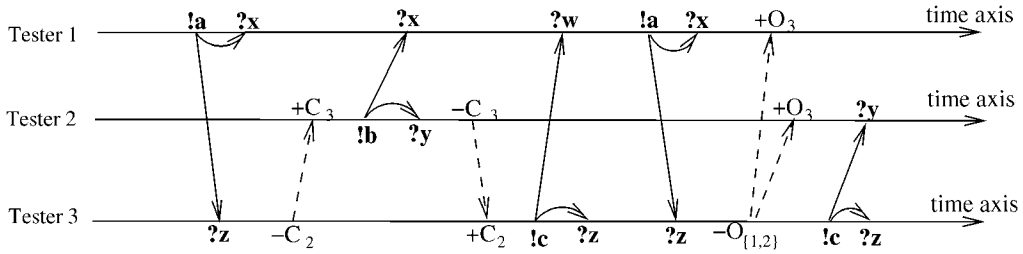

Fig. 5. Adding coordination messages.

- $Tester_q$ be any tester such that: $q \neq k$, $y_i^q = \varepsilon$, and $y_{i+1}^q \neq \varepsilon$.

Before it sends x_{i+1}, $Tester_k$ sends a message O (Observation) to every $Tester_p$ and $Tester_q$. In [2], the following proposition is stated:

Proposition 1. *In an execution conformant to λ, every $Tester_p$ receives y_i^p* before *it receives O and every $Tester_q$ receives y_{i+1}^q* after *it receives O.*

With Proposition 1, the $\mathcal{TS}$ can determine the input which is the cause of every output, i.e., output faults are detected. Henceforth, messages C and O are called *coordination* messages.

Example 3 (Adding coordination messages). For the example of Fig. 3, the obtained messages C and O are represented in Fig. 5. $-X_a$ (respectively, $+X_a$) means "message X is sent to (respectively, received from) $Tester_a$," for $X = C, O$.

This approach may generate *incorrect* results. In fact, with Proposition 1, the authors of [2], [3] assume (implicitly and unduly) that "crossings," like the one represented in the example of Fig. 6, are impossible. In this example, x_{i+1} is sent by $Tester_k$ to the $\mathcal{IUT}$, y_{i+1}^q is sent in $Port_q$ by the $\mathcal{IUT}$ in response to the reception of x_{i+1}, and we assume that $y_i^q = \varepsilon$. Since $Tester_q$ expects no output of Υ_i and an output of Υ_{i+1}, then $Tester_k$ sends O to $Tester_q$ just before sending x_{i+1} to the $\mathcal{IUT}$. The crossing of Fig. 6 illustrates the fact that Proposition 1 does not hold. We see that this crossing can be avoided if the delay between $-O_q$ and $+O_k$ is smaller than the delay between $!x_{i+1}$ and $?y_{i+1}^q$. Intuitively, this implies that crossings can be avoided if the transmission time between testers is always smaller than the response time of the $\mathcal{IUT}$. Since we do not use this (unrealistic) hypothesis, we consider that crossings are possible.

Therefore, the method in [2], [3] does not guarantee Condition 1. In Section 6, we will show how this condition can be guaranteed.

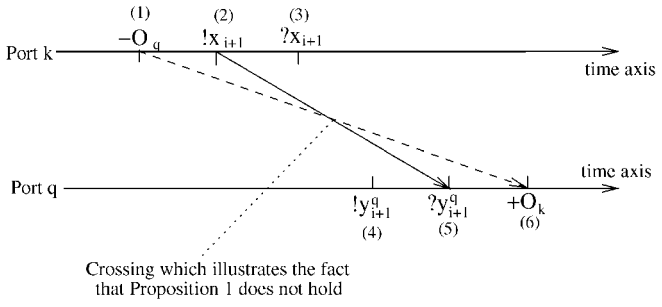

Fig. 6. Incorrect resolution of the problem of observability.

5.2 Enrichment of Coordination Messages with Information

Let us now enrich coordination messages with information as follows:

Messages C: The sending by $Tester_a$ of C is replaced by:

- the sending of $C1$ if it is preceded by $!x_i$ (corresponds to Case $\Upsilon_i = \epsilon$ of Section 5.1.1),
- the sending of $C2$ if it is preceded by $?y_i^a$ (corresponds to Case $\Upsilon_i \neq \epsilon$ of Section 5.1.1).

Messages O: The sending by $Tester_a$ of O is replaced by:

- the sending of $O1$ if it is preceded by $!x_i$,
- the sending of $O2$ if it is preceded by $?y_i^a$,
- the sending of $O3$ if it is preceded by the reception of $C1$, and
- the sending of $O4$ if it is preceded by the reception of $C2$.

Intuitively, a tester that receives an enriched coordination message X is informed about the type of event that precedes the sending of X. The interest of this enrichment is that it will allow us to obtain *weaker* timing constraints of the $\mathcal{TS}$ that guarantee Conditions 1 and 2.

5.3 Reaction Times of the $\mathcal{TS}$

During the testing of any $\lambda = \langle x_1/\Upsilon_1\rangle\langle x_2/\Upsilon_2\rangle \cdots \langle x_t/\Upsilon_t\rangle$, we have determined eight possible situations of $x_i, \Upsilon_i, x_{i+1}, \Upsilon_{i+1}$, for $i < t$. From these situations, we have determined 14 types of reaction times of the $\mathcal{TS}$. Each reaction time separates instants of events of a *same* tester. For a given $i < t$, let $Tester_h$ and $Tester_k$ be the testers sending x_i and x_{i+1}, respectively, and let $Tester_m$ (if any) be the tester which must receive an output $y_i^m \in \Upsilon_i$ and send a message C to $Tester_k$.

Situation 1 ($\Upsilon_i = \epsilon$, $h = k$, and $\forall p \neq k : y_{i+1}^p = \epsilon$). In this situation, there is neither message C nor message O (see Fig. 7a). We have determined one type of reaction time, illustrated in Fig. 7a by $rt_{!x,!x}$ which separates $!x_i$ and $!x_{i+1}$. Let $[RT_{!x,!x}^{min}, RT_{!x,!x}^{max}]$ denote an interval which contains $rt_{!x,!x}$.

Situation 2 ($\Upsilon_i = \epsilon$, $h = k$, and $\exists p \neq k$ such that $y_{i+1}^p \neq \epsilon$). In this situation, there is no message C and at least one message $O1$ (see Fig. 7b). We have determined two types

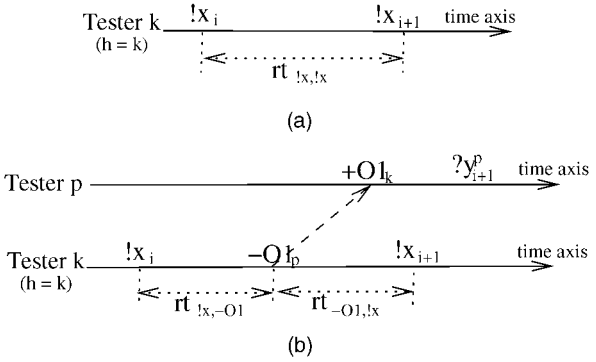

Fig. 7. Reaction times in Situations 1 and 2: $\Upsilon_i = \epsilon$ and there is no message C.

of reaction times, illustrated in Fig. 7b by 1) $rt_{!x,-O1}$ which separates $!x_i$ and $-O1_p$ and 2) $rt_{-O1,!x}$ which separates $-O1_p$ and $!x_{i+1}$. Let $[RT^{min}_{!x,-O1}; RT^{max}_{!x,-O1}]$ and

$$[RT^{min}_{-O1,!x}; RT^{max}_{-O1,!x}]$$

denote two intervals which contain $rt_{!x,-O1}$ and $rt_{-O1,!x}$, respectively.

Situation 3 ($y^k_i \neq \varepsilon$ and $\forall p \neq k : (y^p_i = \varepsilon) \Leftrightarrow (y^p_{i+1} = \varepsilon)$. In this situation, there is neither message C nor message O (see Fig. 8a). We have determined one type of reaction time, illustrated in Fig. 8a by $rt_{?y,!x}$ which separates $?y^k_i$ and $!x_{i+1}$. Let $[RT^{min}_{?y,!x}; RT^{max}_{?y,!x}]$ denote an interval which contains $rt_{?y,!x}$.

Situation 4 ($y^k_i \neq \varepsilon$ and $\exists p \neq k$ such that $(y^p_i = \varepsilon)\ XOR(y^p_{i+1} = \varepsilon)$ (XOR denotes the exclusive OR). In this situation, there is no message C and at least one message $O2$ (see Fig. 8b). We have determined two types of reaction times, illustrated in Fig. 8b by 1) $rt_{?y,-O2}$ which separates $?y^k_i$ and $-O2_p$ and 2) $rt_{-O2,!x}$ which separates $-O2_p$ and $!x_{i+1}$. Let $[RT^{min}_{?y,-O2}; RT^{max}_{?y,-O2}]$ and $[RT^{min}_{-O2,!x}; RT^{max}_{-O2,!x}]$ denote two intervals which contain $rt_{?y,-O2}$ and $rt_{-O2,!x}$, respectively.

Situation 5 ($\Upsilon_i = \epsilon$, $h \neq k$, and $\forall p \neq k : y^p_{i+1} = \varepsilon$). In this situation, there is a message $C1$ and no message O (see Fig. 9a). We have determined two types of reaction times, illustrated in Fig. 9a by 1) $rt_{!x,-C1}$ which separates $!x_i$ and $-C1_k$ and 2) $rt_{+C1,!x}$ which separates $+C1_h$ and $!x_{i+1}$. Let $[RT^{min}_{!x,-C1}; RT^{max}_{!x,-C1}]$ and $[RT^{min}_{+C1,!x}; RT^{max}_{+C1,!x}]$ denote two intervals which contain $rt_{!x,-C1}$ and $rt_{+C1,!x}$, respectively.

Situation 6 ($\Upsilon_i = \epsilon$, $h \neq k$, and $\exists p \neq k$ such that $y^p_{i+1} \neq \varepsilon$). In this situation, there is a message $C1$ and at least one message $O3$ (see Fig. 9b). We have determined three types of reaction times, illustrated in Fig. 9b by 1) $rt_{!x,-C1}$ which separates $!x_i$ and $-C1_k$, 2) $rt_{+C1,-O3}$ which separates $+C1_h$ and $-O3_p$, and 3) $rt_{-O3,!x}$ which separates $-O3_p$ and $!x_{i+1}$. Let $[RT^{min}_{!x,-C1}; RT^{max}_{!x,-C1}]$, $[RT^{min}_{+C1,-O3}; RT^{max}_{+C1,-O3}]$, and $[RT^{min}_{-O3,!x}; RT^{max}_{-O3,!x}]$ denote three intervals which contain $rt_{!x,-C1}$, $rt_{+C1,-O3}$, and $rt_{-O3,!x}$, respectively. Note that $rt_{!x,-C1}$ and $[RT^{min}_{!x,-C1}; RT^{max}_{!x,-C1}]$ have already been defined in Situation 5.

Situation 7 ($\Upsilon_i \neq \epsilon, y^k_i = \varepsilon$, and $\forall p \neq k$: $((y^p_i = \varepsilon) \Leftrightarrow (y^p_{i+1} = \varepsilon))$ or $(p = m)$. In this situation, there is a message $C2$ and no message O (see Fig. 10a). We have determined two types of reaction times, illustrated in Fig. 10a by 1) $rt_{?y,-C2}$ which separates $?y^m_i$ and $-C2_k$ and 2) $rt_{+C2,!x}$ which separates $+C2_m$ and $!x_{i+1}$. Let

$$[RT^{min}_{?y,-C2}; RT^{max}_{?y,-C2}]$$

and $[RT^{min}_{+C2,!x}; RT^{max}_{+C2,!x}]$ denote two intervals which contain $rt_{?y,-C2}$ and $rt_{+C2,!x}$, respectively.

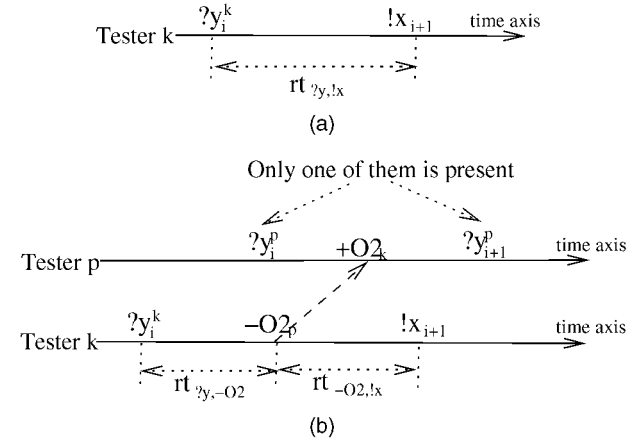

Fig. 8. Reaction times in Situations 3 and 4: $\Upsilon_i \neq \epsilon$ and there is no message C.

Situation 8 ($\Upsilon_i \neq \epsilon$, $y^k_i = \varepsilon$, and $\exists p \neq k$ such that $((y^p_i = \varepsilon) XOR (y^p_{i+1} = \varepsilon))$ and $(p \neq m)$. In this situation, there is a message $C2$ and at least one message $O4$ (see Fig. 10b). We have determined three types of reaction times, illustrated in Fig. 10b by 1) $rt_{?y,-C2}$ which separates $?y^m_i$ and $-C2_k$, 2) $rt_{+C2,-O4}$ which separates $+C2_m$ and $-O4_p$, and 3) $rt_{-O4,!x}$ which separates $-O4_p$ and $!x_{i+1}$. Let $[RT^{min}_{?y,-C2}; RT^{max}_{?y,-C2}]$, $[RT^{min}_{+C2,-O4}; RT^{max}_{+C2,-O4}]$, and $[RT^{min}_{-O4,!x}; RT^{max}_{-O4,!x}]$ denote three intervals which contain $rt_{?y,-C2}$, $rt_{+C2,-O4}$, and $rt_{-O4,!x}$, respectively. Note that $rt_{?y,-C2}$ and $[RT^{min}_{?y,-C2}; RT^{max}_{?y,-C2}]$ have already been defined in Situation 7.

The eight situations are illustrated in Fig. 11, by adding coordination messages to the transition sequence

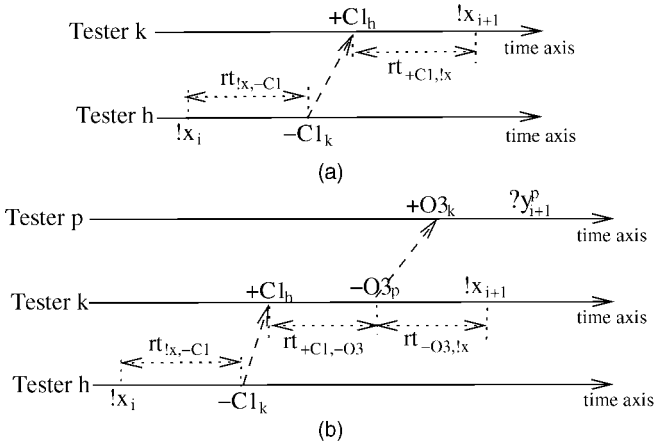

Fig. 9. Reaction times in Situations 5 and 6: $\Upsilon_i = \epsilon$ and there is a message C.

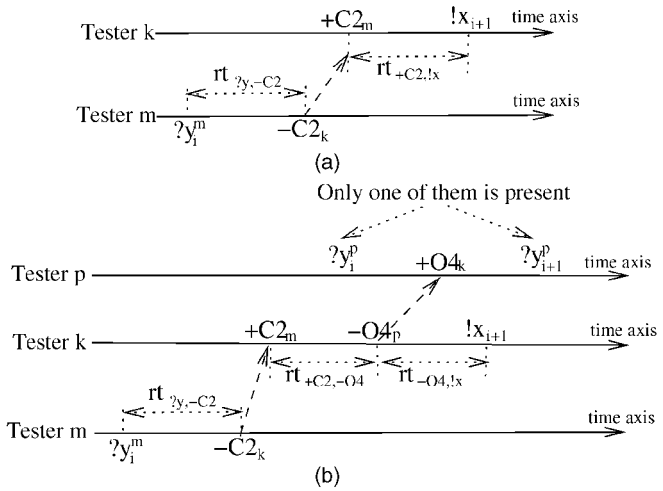

Fig. 10. Reaction times in Situations 7 and 8: $\Upsilon_i \neq \epsilon$ and there is a message C.

$$\lambda = \langle a^1/\varepsilon\rangle\langle b^1/\varepsilon\rangle\langle c^1/(\alpha^1,\beta^2)\rangle\langle d^1/(\gamma^1,\delta^2)\rangle \\ \langle e^1/\varepsilon\rangle\langle f^2/\varepsilon\rangle\langle g^1/(\phi^3)\rangle\langle h^1/(\mu^2,\rho^3)\rangle\langle k^1/\varepsilon\rangle,$$

where the superscript of each event indicates the tester which sends or receives it.

5.4 Waiting Times of the $\mathcal{TS}$

During a testing process, every tester may receive two types of events: outputs of the $\mathcal{IUT}$ and coordination messages. The aim here is to define the amounts of time testers have to wait for receptions before considering that an execution is nonconformant.

Theorem 1. *If a tester does not receive an expected coordination message, then there exists at least another tester which has not received an expected output of the $\mathcal{IUT}$.*

Let us prove this theorem ad absurdum.

Proof.

1. We assume that Theorem 1 does not hold, that is, the following two points 1a and 1b are satisfied.
   a. There exists a $Tester_{p_1}$ that does not receive an expected coordination message from a $Tester_{p_2}$.
   b. There exists no tester which does not receive an expected output of the $\mathcal{IUT}$.
2. A tester does not receive an expected coordination message X from another tester iff the latter does not send X.
3. A tester does not send an expected coordination message iff it has not received an expected output of the $\mathcal{IUT}$ or a coordination message.
4. Items 3 and 1b imply that a tester does not send an expected coordination message iff it has not received an expected coordination message.
5. Items 2 and 4 imply that any $Tester_{p_i}$ which has not received an expected coordination message is preceded chronologically by another $Tester_{p_{i+1}}$ which has not received an expected coordination message.
6. Items 1a and 5 imply that there exists an *infinite* chronological suite:

$$\ldots, Tester_{p_{i+1}}, Tester_{p_i}, Tester_{p_{i-1}}, \\ \ldots, Tester_{p_2}, Tester_{p_1}$$

   of testers which have not received expected coordination messages.
7. Item 6 is an absurdity because it means that the testing has begun in an infinite past. □

From Theorem 1 and the fact that the $\mathcal{IUT}$ is considered faulty iff at least one tester generates a verdict **fail**, we will only consider waiting times for the receptions of outputs of the $\mathcal{IUT}$. For a given test sequence $\lambda = \langle x_1/\Upsilon_1\rangle\langle x_2/\Upsilon_2\rangle\cdots\langle x_t/\Upsilon_t\rangle$, we will consider waiting times for the receptions of outputs of Υ_i, $i \leq t$. We have determined four situations. Let $Tester_h$ and $Tester_k$ be the testers sending x_i and x_{i+1}, respectively, and let $Tester_m$ be any tester which must receive an output $y_i^m \in \Upsilon_i$. In all of these situations, $y_i^m \neq \varepsilon$.

Situation A ($m \neq h$ and $i = 1$ (see Fig. 12a)). Let the starting instant be the instant of $!x_1$. We assume that there exists a mechanism which allows $Tester_m$ to know the starting instant. $WT_{\varepsilon,?y}$ is an upper bound of the time $wt_{\varepsilon,?y}$ separating the starting instant and $?y_1^m$. If $Tester_m$ cannot know the starting instant, a solution consists of transforming this Situation A into Situation C as follows: Before it sends x_1, $Tester_h$ sends a coordination message O (O can be $O1, O2, O3$, or $O4$) to $Tester_m$.

Situation B ($m = h$ (see Fig. 12b)). $WT_{!x,?y}$ is an upper bound of the time $wt_{!x,?y}$ separating $!x_i$ and $?y_i^m$.

Situation C ($m \neq h$, $i > 1$, and $y_{i-1}^m = \varepsilon$ (see Fig. 12c)). $WT_{+O,?y}$ is an upper bound of the time $wt_{+O,?y}$ separating $+O_h$ and $?y_i^m$.

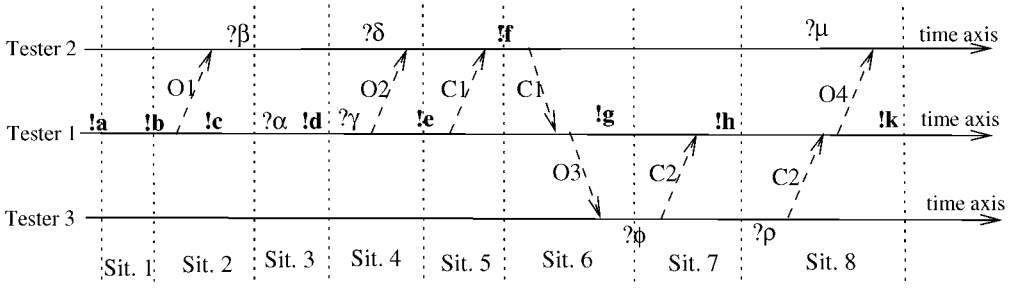

Fig. 11. The eight situations illustrated in one view.

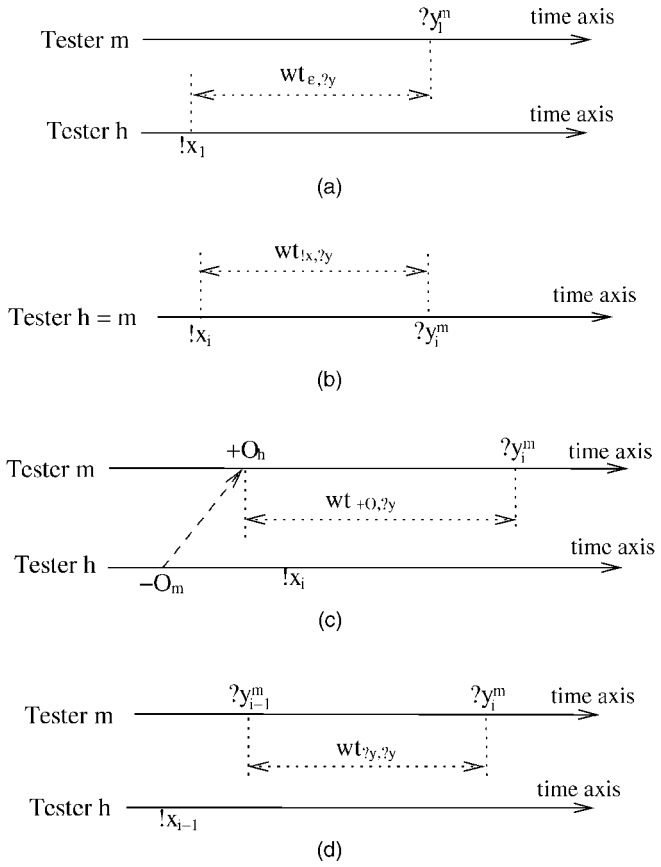

Fig. 12. Waiting times in Situations A, B, C, and D.

Situation D ($m \neq h$, $i > 1$, and $y_{i-1}^m \neq \varepsilon$ (see Fig. 12d)). $WT_{?y,?y}$ is an upper bound of the time $wt_{?y,?y}$ separating $?y_{i-1}^m$ and $?y_i^m$.

The four waiting times are illustrated in Fig. 13, by adding coordination messages to the transition sequence $\lambda = \langle a^1/(\alpha^1, \beta^2)\rangle\langle b^1/(\gamma^2, \delta^3)\rangle$.

5.5 Objective 2

Let us now propose another objective which guarantees Objective 1 (see end of Section 4).

Proposition 2. *The $\mathcal{TS}$ sends an input to the $\mathcal{IUT}$ only after it has received* all *the outputs (if any) of the $\mathcal{IUT}$ in response to the preceding input. We emphasize the word "all" because there may be possible* unexpected *outputs (in the case of a nonconformant $\mathcal{IUT}$). In an execution conformant to λ, this proposition implies for $i \leq t$: $\tau_{!x_{i+1}} \geq \tau_{?Y_i}$. When this proposition is not satisfied, controllability and observability problems may arise, even with a centralized test method (see Example 4).*

Example 4 (Fault detectability problem resolved by Proposition 2). We consider the specification and the faulty implementation of Fig. 14, where a and x (respectively, b and y) are the input and output of the $\mathcal{IUT}$ on port 1 (respectively, port 2). We assume a centralized test architecture and we consider the sequence: $\lambda' = \langle a/(x, \varepsilon)\rangle\langle b/(\varepsilon, y)\rangle$. With the faulty implementation, a possible "scenario" is the following:

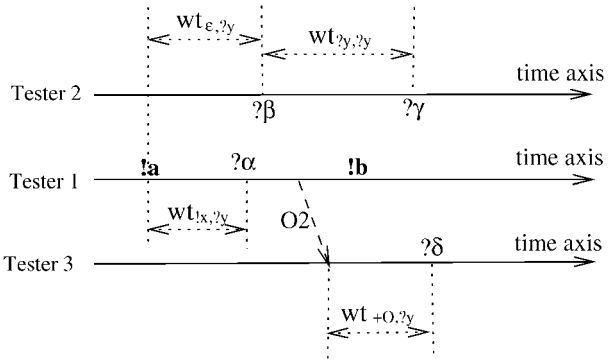

Fig. 13. The four waiting times illustrated in one view.

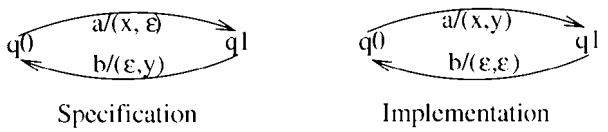

Fig. 14. Example of fault detectability problem resolved by Proposition 2.

instant	event		
0	!a	*	!a and ?a (resp. !b and ?b) denote
1	?a	+	the sending of a (resp. b) by the TS and
2	!x	+	its reception by the IUT, respectively.
3	?x	*	!x and ?x (resp. !y and ?y) denote
4	!b	*	the sending of x (resp. y) by the IUT and
5	!y	+	its reception by the TS, respectively.
6	?y	*	Events of TS are indicated by *
7	?b	+	Events of IUT are indicated by +

According to Definition 1, this execution is nonconformant to λ' because the $\mathcal{IUT}$ does not execute the sequence of events $\phi(\lambda') = ?a!x?b!y$ (see events "+"). The fault is not detected because the $\mathcal{TS}$ executes the GTS $\omega(\lambda') = !a?x!b?y$ (see events "*"). Intuitively, the $\mathcal{TS}$ cannot determine that y is a response of the $\mathcal{IUT}$ to a and not to b. When Proposition 2 holds, the above scenario never occurs because the $\mathcal{TS}$ sends b only *after* it has received *all* outputs in response to a, i.e., x (expected) and y (unexpected). The fault is detected because the $\mathcal{TS}$ receives an unexpected output.

Theorem 2. *With the coordinated test method, Propositions 1 and 2 guarantee Condition 1.*[1]

1. See proof in Appendix A.

Objective 2. With Theorem 2, Objective 1 (see end of Section 4) is achieved *if* Propositions 1 and 2 and Conditions 2 and 3 are guaranteed. Our new objective (which guarantees Objective 1) is as follows:

1. In Section 6.1, we determine constraints of reaction times of the $\mathcal{TS}$ which guarantee: Proposition 2 and Condition 2 in all cases, and Proposition 1 when a message O is used.
2. In Section 6.2, we determine waiting times of the $\mathcal{TS}$ which guarantee Condition 3.

5.6 FIFO Communication not Required

We have now all the elements which allow us to prove the following lemmas and theorem which were already introduced in Section 2.1.

Lemma 1. *Let CM_p denote the reliable communication medium through which $Tester_p$ communicates with port p of the $\mathcal{IUT}$. At any time, each CM_p contains at most a single message.*

Proof of Lemma 1. Let us prove that, between the sendings of any consecutive inputs x_i and x_{i+1}, each CM_* contains at most a single message. ("$*$" means "any value $\leq n$," where n is the number of ports). Let $Tester_p$ and $Tester_q$ be the testers which send x_i and x_{i+1}, respectively.

Proposition 2 implies that $Tester_p$ sends x_i to the $\mathcal{IUT}$ (through CM_p) only when all the CM_* (including CM_p) are empty. Therefore, Property 1 implies that, while x_i is in CM_p (i.e., before the reception of x_i by the $\mathcal{IUT}$), CM_p does not contain another message and all the other CM_* are empty. When CM_p becomes empty (i.e., when x_i is received by the $\mathcal{IUT}$), Property 2 implies that the $\mathcal{IUT}$ sends at most a single output to each tester and, thus, each CM_* will contain at most a single message. Proposition 2 implies that $Tester_q$ sends x_{i+1} to the $\mathcal{IUT}$ (through CM_q) only when all the nonempty CM_* become empty (i.e., after the receptions by the $\mathcal{TS}$ of all the outputs in response to x_i). □

Lemma 2. *Let $CM^{ts}_{p,q}$ denote the reliable communication medium through which $Tester_p$ and $Tester_q$ communicate with one another. At any time, each $CM^{ts}_{p,q}$ contains at most a single message.*

Proof of Lemma 2. During the testing of any

$$\lambda = \langle x_1/\Upsilon_1\rangle\langle x_2/\Upsilon_2\rangle \cdots \langle x_t/\Upsilon_t\rangle,$$

the eight possible situations of x_i, Υ_i, x_{i+1}, Υ_{i+1} are presented in Section 5.3 and illustrated in Figs. 7, 8, 9, and 10. We clearly see in these figures that in none of the eight situations does a $Tester_p$ send a message to a $Tester_q$ before the reception by $Tester_q$ of a previous message from $Tester_p$. This implies that, at any time, each $CM^{ts}_{p,p}$ contains at most a single message. □

Theorem 3. *From Lemmas 1 and 2, we deduce that our method is insensitive to the fact that the communication medium either respects the FIFO discipline or does not.*

Proof of Theorem 3. The FIFO discipline is not relevant because, at any time, the communication medium between a sender and a receiver contains at most a single message. □

6 Constraints of Reaction Times and Waiting Times of the $\mathcal{TS}$

6.1 Constraints of Reaction Times of the $\mathcal{TS}$ and Their Resolution

In this section, we reach Point 1 of Objective 2. The reader who is just interested by a systematic method to compute constraints of reaction times, without any explanation, may directly go to Section 6.1.9. In the following, for each situation, the reader may refer to the corresponding figure of Section 5.3. $\sup(a;b)$ denotes the greatest of a and b.

6.1.1 Situation 1: Constraints of $[RT^{min}_{!x,!x}; RT^{max}_{!x,!x}]$

Proposition 2 is guaranteed by[2]

$$RT^{min}_{!x,!x} \geq (TT^{max} + RT_{iut} + TT^{max}).$$

Condition 2 is guaranteed by[3]

$$RT^{max}_{!x,!x} + TT^{max} - TT^{min} \leq WT_{iut}.$$

These two inequations may therefore be combined as follows:

$$[RT^{min}_{!x,!x}; RT^{max}_{!x,!x}] \subseteq [RT_{iut} + 2TT^{max}; WT_{iut} - \triangle TT]. \quad (4)$$

Solutions exist for (4) iff:[4]

$$WT_{iut} \geq (RT_{iut} + 3TT^{max} - TT^{min}). \quad (5)$$

6.1.2 Situation 2: Constraints of $[RT^{min}_{!x,-O1}; RT^{max}_{!x,-O1}]$ and $[RT^{min}_{-O1,!x}; RT^{max}_{-O1,!x}]$

By analogy with Situation 1,[5] Proposition 2 and Condition 2 are guaranteed if:

$$[RT^{min}_{!x,-O1} + RT^{min}_{-O1,!x}; RT^{max}_{!x,-O1} + RT^{max}_{-O1,!x}]$$

satisfies the same constraints for $[RT^{min}_{!x,!x}; RT^{max}_{!x,!x}]$ (see (4)). Therefore, Proposition 2 and Condition 2 are guaranteed by:

$$\begin{aligned}&[RT^{min}_{!x,-O1} + RT^{min}_{-O1,!x}; RT^{max}_{!x,-O1} + RT^{max}_{-O1,!x}] \subseteq \\ &[RT_{iut} + 2TT^{max}; WT_{iut} - \triangle TT].\end{aligned} \quad (6)$$

Since there is a message O, we must also guarantee Proposition 1, i.e., $+O1_k$ is before $?y^p_{i+1}$. A sufficient condition for that is:[6]

$$RT^{min}_{-O1,!x} \geq (TT^{max}_{ts} - 2TT^{min}). \quad (7)$$

Solutions exist for (6) and (7) if and only if:[7]

$$\begin{aligned}&WT_{iut} \geq \sup \\ &(RT_{iut} + 3TT^{max} - TT^{min}; TT^{max}_{ts} + TT^{max} - 3TT^{min}).\end{aligned} \quad (8)$$

6.1.3 Situation 3: Constraints of $[RT^{min}_{?y,!x}; RT^{max}_{?y,!x}]$

Proposition 2 is guaranteed by[8]

$$(TT^{min} + RT^{min}_{?y,!x}) \geq (RT_{iut} + TT^{max}).$$

2. See proof in Appendix B.1.1
3. See proof in Appendix B.1.2.
4. See proof in Appendix B.4.
5. For details on the analogy, see Appendix B.2.1.
6. See proof in Appendix B.2.2.
7. See proof in Appendix B.2.3.
8. See proof in Appendix B.3.1.

Condition 2 is guaranteed by[9]

$$(TT^{max} + RT^{max}_{?y,!x} + TT^{max}) \leq WT_{iut}.$$

These two inequations may therefore be combined as follows:

$$[RT^{min}_{?y,!x}; RT^{max}_{?y,!x}] \subseteq [RT_{iut} + \triangle TT; WT_{iut} - 2TT^{max}]. \quad (9)$$

Solutions exist for (9) iff (5) of Section 6.1.1 holds.[10]

6.1.4 Situation 4: Constraints of $[RT^{min}_{?y,-O2}; RT^{max}_{?y,-O2}]$ and $[RT^{min}_{-O2,!x}; RT^{max}_{-O2,!x}]$

By analogy with Situation 3,[11] Proposition 2 and Condition 2 are guaranteed if:

$$[RT^{min}_{?y,-O2} + RT^{min}_{-O2,!x}; RT^{max}_{?y,-O2} + RT^{max}_{-O2,!x}]$$

satisfies the same constraints for $[RT^{min}_{?y,!x}; RT^{max}_{?y,!x}]$ (see (9)). Therefore, Proposition 2 and Condition 2 are guaranteed by:

$$\begin{aligned}&[RT^{min}_{?y,-O2} + RT^{min}_{-O2,!x}; RT^{max}_{?y,-O2} + RT^{max}_{-O2,!x}] \subseteq \\ &[RT_{iut} + \triangle TT; WT_{iut} - 2TT^{max}].\end{aligned} \quad (10)$$

Since there is a message O, we must also guarantee Proposition 1, i.e.,

- $+O2_k$ is after $?y^p_i$ (if any). A sufficient condition for that is:[12]

$$RT^{min}_{?y,-O2} \geq (RT_{iut} + \triangle TT - TT^{min}_{ts}). \quad (11)$$

- $+O2_k$ is before $?y^p_{i+1}$ (if any). A sufficient condition for that is:[13]

$$RT^{min}_{-O2,!x} \geq (TT^{max}_{ts} - 2TT^{min}). \quad (12)$$

Solutions exist for (10), (11), and (12) if and only if:[14]

$$\begin{aligned}WT_{iut} \geq \sup(&RT_{iut} + 3TT^{max} - TT^{min}; TT^{max}_{ts} + \\ &2\triangle TT; RT_{iut} + 3\triangle TT + \triangle TT_{ts}).\end{aligned} \quad (13)$$

6.1.5 Situation 5: Constraints of $[RT^{min}_{!x,-C1}; RT^{max}_{!x,-C1}]$ and $[RT^{min}_{+C1,!x}; RT^{max}_{+C1,!x}]$

By analogy with Situation 1,[15] Proposition 2 and Condition 2 are guaranteed if:

$$[RT^{min}_{!x,-C1} + TT^{min}_{ts} + RT^{min}_{+C1,!x}; RT^{max}_{!x,-C1} + TT^{max}_{ts} + RT^{max}_{+C1,!x}]$$

satisfies the same constraints for $[RT^{min}_{!x,!x}; RT^{max}_{!x,!x}]$. Therefore, Proposition 2 and Condition 2 are guaranteed by:

$$\begin{aligned}&[RT^{min}_{!x,-C1} + RT^{min}_{+C1,!x}; RT^{max}_{!x,-C1} + RT^{max}_{+C1,!x}] \subseteq \\ &[RT_{iut} + 2TT^{max} - TT^{min}_{ts}; WT_{iut} - \triangle TT - TT^{max}_{ts}].\end{aligned} \quad (14)$$

Solutions exist for (14) if and only if:[16]

$$\begin{aligned}WT_{iut} \geq \sup(&RT_{iut} + 3TT^{max} - TT^{min} + \\ &\triangle TT_{ts}; \triangle TT + TT^{max}_{ts}).\end{aligned} \quad (15)$$

6.1.6 Situation 6: Constraints of $[RT^{min}_{!x,-C1}; RT^{max}_{!x,-C1}]$, $[RT^{min}_{+C1,-O3}; RT^{max}_{+C1,-O3}]$, and $[RT^{min}_{-O3,!x}; RT^{max}_{-O3,!x}]$

By analogy with Situation 5,[17] Proposition 2 and Condition 2 are guaranteed if:

$$\begin{aligned}&[RT^{min}_{!x,-C1} + RT^{min}_{+C1,-O3} + RT^{min}_{-O3,!x}; \\ &RT^{max}_{!x,-C1} + RT^{max}_{+C1,-O3} + RT^{max}_{-O3,!x}]\end{aligned}$$

satisfies the same constraints for

$$[RT^{min}_{!x,-C1} + RT^{min}_{+C1,!x}; RT^{max}_{!x,-C1} + RT^{max}_{+C1,!x}]$$

(see (14)). Therefore, Proposition 2 and Condition 2 are guaranteed by:

$$\begin{aligned}&[RT^{min}_{!x,-C1} + RT^{min}_{+C1,-O3} + RT^{min}_{-O3,!x}; RT^{max}_{!x,-C1} + \\ &RT^{max}_{+C1,-O3} + RT^{max}_{-O3,!x}] \subseteq [RT_{iut} + 2TT^{max} - \\ &TT^{min}_{ts}; WT_{iut} - \triangle TT - TT^{max}_{ts}].\end{aligned} \quad (16)$$

Since there is a message O, and by analogy with Situation 2,[18] $RT^{min}_{-O3,!x}$ must satisfy the same constraint for $RT^{min}_{-O1,!x}$ (see (7)). Therefore, we obtain:

$$RT^{min}_{-O3,!x} \geq (TT^{max}_{ts} - 2TT^{min}). \quad (17)$$

Solutions exist for (16) and (17) if and only if:[19]

$$\begin{aligned}WT_{iut} \geq \sup(&RT_{iut} + 3TT^{max} - TT^{min} + \triangle TT_{ts}; \triangle TT + \\ &TT^{max}_{ts}; TT^{max} - 3TT^{min} + 2TT^{max}_{ts}).\end{aligned} \quad (18)$$

6.1.7 Situation 7: Constraints of $[RT^{min}_{?y,-C2}; RT^{max}_{?y,-C2}]$ and $[RT^{min}_{+C2,!x}; RT^{max}_{+C2,!x}]$

By analogy with Situation 3,[20] Proposition 2 and Condition 2 are guaranteed if:

$$[RT^{min}_{?y,-C2} + TT^{min}_{ts} + RT^{min}_{+C2,!x}; RT^{max}_{?y,-C2} + TT^{max}_{ts} + RT^{max}_{+C2,!x}]$$

satisfies the same constraints for $[RT^{min}_{?y,!x}; RT^{max}_{?y,!x}]$. Therefore, Proposition 2 and Condition 2 are guaranteed by:

$$\begin{aligned}&[RT^{min}_{?y,-C2} + RT^{min}_{+C2,!x}; RT^{max}_{?y,-C2} + RT^{max}_{+C2,!x}] \subseteq \\ &[RT_{iut} + \triangle TT - TT^{min}_{ts}; WT_{iut} - 2TT^{max} - TT^{max}_{ts}].\end{aligned} \quad (19)$$

Solutions exist for (19) if and only if:[21]

$$\begin{aligned}WT_{iut} \geq \sup(&RT_{iut} + 3TT^{max} - TT^{min} + \\ &\triangle TT_{ts}; 2TT^{max} + TT^{max}_{ts}).\end{aligned} \quad (20)$$

6.1.8 Situation 8: Constraints of $[RT^{min}_{?y,-C2}; RT^{max}_{?y,-C2}]$, $[RT^{min}_{+C2,-O4}; RT^{max}_{+C2,-O4}]$, and $[RT^{min}_{-O4,!x}; RT^{max}_{-O4,!x}]$

By analogy with Situation 7,[22] Proposition 2 and Condition 2 are guaranteed if:

9. See proof in Appendix B.3.2.
10. See proof in Appendix B.4.
11. For details on the analogy, see Appendix B.5.1.
12. See proof in Appendix B.5.2.
13. See proof in Appendix B.5.3.
14. See proof in Appendix B.5.4.
15. For details on the analogy, see Appendix B.6.1.
16. See proof in Appendix B.6.2.
17. For details on the analogy, see Appendix B.7.1.
18. For details on the analogy, see Appendix B.7.2.
19. See proof in Appendix B.7.3.
20. For details on the analogy, see Appendix B.8.1.
21. See proof in Appendix B.8.2.
22. For details on the analogy, see Appendix B.9.1.

$$[RT^{min}_{?y,-C2} + RT^{min}_{+C2,-O4} + RT^{min}_{-O4,!x};$$
$$RT^{max}_{?y,-C2} + RT^{max}_{+C2,-O4} + RT^{max}_{-O4,!x}]$$

satisfies the same constraints for

$$[RT^{min}_{?y,-C2} + RT^{min}_{+C2,!x}; RT^{max}_{?y,-C2} + RT^{max}_{+C2,!x}]$$

(see (19)). Therefore, Proposition 2 and Condition 2 are guaranteed by:

$$\begin{aligned}&[RT^{min}_{?y,-C2} + RT^{min}_{+C2,-O4} + RT^{min}_{-O4,!x};\\ &RT^{max}_{?y,-C2} + RT^{max}_{+C2,-O4} + RT^{max}_{-O4,!x}] \subseteq\\ &[RT_{iut} + \triangle TT - TT^{min}_{ts}; WT_{iut} - 2TT^{max} - TT^{max}_{ts}].\end{aligned} \tag{21}$$

Since there is a message O and by analogy with Situation 4:[23]

- $RT^{min}_{?y,-C2} + TT^{min}_{ts} + RT^{min}_{+C2,-O4}$ must satisfy the same constraint for $RT^{min}_{?y,-O2}$ (see (11)). A sufficient condition is therefore:

$$RT^{min}_{?y,-C2} + RT^{min}_{+C2,-O4} \geq (RT_{iut} + \triangle TT - 2TT^{min}_{ts}). \tag{22}$$

- $RT^{min}_{-O4,!x}$ must satisfy the same constraints for $RT^{min}_{-O2,!x}$ (see (12)). A sufficient condition is therefore:

$$RT^{min}_{-O4,!x} \geq (TT^{max}_{ts} - 2TT^{min}). \tag{23}$$

Solutions exist for (21), (22), and (23) if and only if:[24]

$$\begin{aligned}WT_{iut} \geq \sup(&RT_{iut} + 3TT^{max} - TT^{min} +\\ &\triangle TT_{ts}; 2TT^{max}_{ts} + 2\triangle TT; RT_{iut} +\\ &3\triangle TT + 2\triangle TT_{ts}; 2TT^{max} + TT^{max}_{ts}).\end{aligned} \tag{24}$$

6.1.9 A Scenario for Resolving Constraints of Reaction Times

We have determined seven conditions for existence of solutions: (5), (8), (13), (15), (18), (20), and (24). Actually, (24) is the global condition for existence of solutions because it implies the other six equations.[25] Here is a scenario of resolution of reaction times constraints.

Step 1: Check the existence of solutions. If (24) holds then continue, Else send a message "There is no solution !" and terminate.

Step 2: Resolve constraints of Situation 2. Compute intervals $[RT^{min}_{!x,-O1}; RT^{max}_{!x,-O1}]$ and $[RT^{min}_{-O1,!x}; RT^{max}_{-O1,!x}]$ which satisfy (6) and (7).

Step 3: Resolve constraints of Situation 1. Compute an interval $[RT^{min}_{!x,!x}; RT^{max}_{!x,!x}]$ which satisfies (4). By analogy with Situation 2, we can take:

$$[RT^{min}_{!x,!x}; RT^{max}_{!x,!x}] = [RT^{min}_{!x,-O1} + RT^{min}_{-O1,!x}; RT^{max}_{!x,-O1} + RT^{max}_{-O1,!x}].$$

Step 4: Resolve constraints of Situation 4. Compute intervals $[RT^{min}_{?y,-O2}; RT^{max}_{?y,-O2}]$ and $[RT^{min}_{-O2,!x}; RT^{max}_{-O2,!x}]$ which satisfy (10), (11), and (12).

Step 5: Resolve constraints of Situation 3. Compute an interval $[RT^{min}_{?y,!x}; RT^{max}_{?y,!x}]$ which satisfies (9). By analogy with Situation 4, we can take:

$$[RT^{min}_{?y,!x}; RT^{max}_{?y,!x}] = [RT^{min}_{?y,-O2} + RT^{min}_{-O2,!x}; RT^{max}_{?y,-O2} + RT^{max}_{-O2,!x}].$$

Step 6: Resolve constraints of Situation 6. Compute intervals $[RT^{min}_{!x,-C1}; RT^{max}_{!x,-C1}]$, $[RT^{min}_{+C1,-O3}; RT^{max}_{+C1,-O3}]$, and $[RT^{min}_{-O3,!x}; RT^{max}_{-O3,!x}]$ which satisfy (16) and (17).

Step 7: Resolve constraints of Situation 5. Compute an interval $[RT^{min}_{+C1,!x}; RT^{max}_{+C1,!x}]$ which satisfies (14). (Note that $[RT^{min}_{!x,-C1}; RT^{max}_{!x,-C1}]$ has been computed in Step 6). By analogy with Situation 6, we can take:

$$\begin{aligned}[RT^{min}_{+C1,!x}; RT^{max}_{+C1,!x}] =&[RT^{min}_{+C1,-O3} + RT^{min}_{-O3,!x};\\ &RT^{max}_{+C1,-O3} + RT^{max}_{-O3,!x}].\end{aligned}$$

Step 8: Resolve constraints of Situation 8. Compute intervals $[RT^{min}_{?y,-C2}; RT^{max}_{?y,-C2}]$, $[RT^{min}_{+C2,-O4}; RT^{max}_{+C2,-O4}]$, and $[RT^{min}_{-O4,!x}; RT^{max}_{-O4,!x}]$ which satisfy (21), (22), and (23).

Step 9: Resolve constraints of Situation 7. Compute an interval $[RT^{min}_{+C2,!x}; RT^{max}_{+C2,!x}]$ which satisfies (19). (Note that $[RT^{min}_{?y,-C2}; RT^{max}_{?y,-C2}]$ has been computed in Step 8). By analogy with Situation 8, we can take:

$$\begin{aligned}[RT^{min}_{+C2,!x}; RT^{max}_{+C2,!x}] =&[RT^{min}_{+C2,-O4} + RT^{min}_{-O4,!x};\\ &RT^{max}_{+C2,-O4} + RT^{max}_{-O4,!x}].\end{aligned}$$

Recall that the obtained constraints of reaction times of the $\mathcal{TS}$ must be guaranteed by the $\mathcal{TS}$. The problem was not simple because RT_{iut} requires *lower* bounds and WT_{iut} requires *upper* bounds on the reaction times of the $\mathcal{TS}$. The obtained conditions for the existence of solutions guarantee that the lower bounds are smaller than the upper bounds. The problem is significantly simplified in the particular case where the waiting time of the $\mathcal{IUT}$ (WT_{iut}) is infinite (i.e., the $\mathcal{IUT}$ is infinitely patient).

6.2 Resolution of Waiting Times of the $\mathcal{TS}$

In this section, we reach Point 2 of Objective 2 (presented at the end of Section 5.5).

Situation A ($WT_{\varepsilon,?y}$ (see Fig. 12a)). We assume here that $Tester_h$ sends x_1 and $Tester_m$ starts waiting for $?y^m_1$ at the same time. The least restrictive constraint of $WT_{\varepsilon,?y}$ is:[26]

$$WT_{\varepsilon,?y} = RT_{iut} + 2TT^{max}. \tag{25}$$

Situation B ($WT_{!x,?y}$ (see Fig. 12b)). Similarly to the preceding case, the least restrictive constraint of $WT_{!x,?y}$ is:[27]

$$WT_{!x,?y} = RT_{iut} + 2TT^{max}. \tag{26}$$

Situation C ($WT_{+O,?y}$ (see Fig. 12c)). The least restrictive constraint of $WT_{+O,?y}$ is:[28]

$$\begin{aligned}WT_{+O,?y} =&RT_{iut} + 2TT^{max} - TT^{min}_{ts} +\\ &\sup(RT^{max}_{-O1,!x}, RT^{max}_{-O2,!x}, RT^{max}_{-O3,!x}, RT^{max}_{-O4,!x}).\end{aligned} \tag{27}$$

23. For details on the analogy, see Appendix B.9.2.
24. See proof in Appendix B.9.3.
25. See proof in Appendix B.10.
26. See proof in Appendix C.1.
27. See proof in Appendix C.2.
28. See proof in Appendix C.3.

Situation D ($WT_{?y,?y}$ (see Fig. 12d)). The least restrictive constraint of $WT_{?y,?y}$ is:[29]

$$WT_{?y,?y} = 2RT_{iut} + 3TT^{max} - TT^{min} + TT^{max}_{ts} + RT^{max}_{?y,-C2} + RT^{max}_{+C2,!x}. \quad (28)$$

The obtained waiting times of the $\mathcal{TS}$ determine the delay the $\mathcal{TS}$ has to wait in each situation before declaring the $\mathcal{IUT}$ nonconformant (if the $\mathcal{TS}$ does not receive all the expected outputs).

Remark 2. If Situation A is replaced by Situation C by sending a first coordination message O before to send x_1 (see Section 5.4), the delay separating the sendings of O and of x_1 must respect (7). The similarity of (7), (12), (17), and (23) shows that this first O can be any of $O1, O2, O3$, or $O4$.

6.3 About the Centralized Test Method

With the centralized test method, a single tester communicates with all the ports of the distributed $\mathcal{IUT}$ (see Fig. 1a). As we will see, the centralized test method can be considered as just a particular case of the coordinated test method. In fact, with the centralized test method:

- Among Situations 1 to 8 of Section 5.3, the only situations which may occur are the situations where a single tester is involved, that is Situations 1 and 3. Two reaction times are therefore defined: $rt_{!x,!x} \in [RT^{min}_{!x,!x}; RT^{max}_{!x,!x}]$ and $rt_{?y,!x} \in [RT^{min}_{?y,!x}; RT^{max}_{?y,!x}]$. $rt_{!x,!x}$ has the same semantics as in Situation 1 and, therefore, the constraint of $[RT^{min}_{!x,!x}; RT^{max}_{!x,!x}]$ is (4). $rt_{?y,!x}$ has different semantics than in Situation 3. In fact, contrary to Situation 3 where $Tester_k$ receives a single output y^k_i, the centralized tester receives all the outputs of Υ_i. $rt_{?y,!x}$ is now the delay which separates the *last* output of Υ_i from $!x_{i+1}$. Despite this difference, the constraint of $[RT^{min}_{?y,!x}; RT^{max}_{?y,!x}]$ is (9) which is obtained by using the proofs of Situation 3 in Appendices B.3.1 and B.3.2.

  And similar to Situations 1 and 3, solutions exist for (4) and (9) iff (5) is satisfied.
- Among Situations A to D of Section 5.4, the only situation which may occur is the situation where a single tester is involved, that is Situation B. One waiting time $WT_{!x,?y}$ is therefore defined.

  $WT_{!x,?y}$ does not have the same semantics as in Situation B. In fact, contrary to Situation B where $Tester_m$ receives a single output y^m_i, the centralized tester receives all the outputs of Υ_i. $WT_{!x,?y}$ is now an upper bound of the time separating $!x_i$ from the *last* output of Υ_i. Despite this difference, $WT_{!x,?y}$ is resolved by (26) which is obtained by using the proof of Situation B in Appendix C.2.

7 Conclusion and Future Work

7.1 Contributions

In this study, we propose a test method for distributed systems. The main novelty aspects can be summarized by the following points:

1. Controllability and observability problems are resolved.
2. The finite waiting time of the $\mathcal{IUT}$ is respected.
3. The duration of test execution is minimized.
4. No global clock is required.
5. The communication medium used by the $\mathcal{TS}$ is not necessarily FIFO.

We also show that the centralized test method can be considered just as a particular case of the coordinated test method and correctness of all our results is proven.

Recall that the problem resolved in this article is much simpler when the waiting time of the $\mathcal{IUT}$ (WT_{iut}) is infinite, that is, if the $\mathcal{IUT}$ is infinitely patient. In fact, in this case, there is no required upper bounds for reaction times of the $\mathcal{TS}$. Therefore, conditions for the existence of solutions are always satisfied and do not need to be checked.

7.2 Future Work

In the near future, we intend to investigate the following research issues:

1. Theorem 2 states that Propositions 1 and 2 imply Condition 1. The fact that the inverse is not true, implies that Objective 2 (at the end of Section 5.5) is more restrictive than Objective 1 (at the end of Section 4). Our method allows us to obtain optimal solutions which guarantee Objective 2. These solutions guarantee Objective 1 but are not necessarily the optimal solutions which guarantee Objective 1. We intend to investigate the computation of optimal results which guarantee Objective 1.
2. The application of our study to complex examples. We intend to consider the areas of communications and robotics.
3. The extension of our study for testing *real-time* distributed $\mathcal{IUT}$s. Although the present test method uses a temporal approach, the $\mathcal{IUT}$ is not real-time, in the sense that there is no explicit timing constraints between transitions of the np-FSM that describes the specification of the $\mathcal{IUT}$.

Appendix A
Proof of Theorem 2

We consider the coordinated test method and we assume that Propositions 1 and 2 hold. Our aim here is therefore to prove that Condition 1 holds.

Let $\omega = \langle x_1/\Upsilon_1\rangle\langle x_2/\Upsilon_2\rangle \cdots \langle x_t/\Upsilon_t\rangle$ be the used GTS, and b be any response of the $\mathcal{IUT}$ to a given x_i.

1. $\forall \alpha : \tau_{!\alpha} \leq \tau_{?\alpha}$.
2. Since b is a response to x_i, $\tau_{?x_i} \leq \tau_{!b}$.
3. Proposition 2 implies $\tau_{?b} \leq \tau_{!x_{i+1}}$.
4. Items 1, 2, and 3 imply $\tau_{?x_i} \leq \tau_{!b} \leq \tau_{?b} \leq \tau_{!x_{i+1}} \leq \tau_{?x_{i+1}}$. Therefore, $\tau_{?x_i} \leq \tau_{?x_{i+1}}$.
5. Item 4 means that controllability is guaranteed.
6. Proposition 2 assumes that the $\mathcal{TS}$ observes all the outputs of the $\mathcal{IUT}$.
7. Items 1, 2, and 3 imply

$$\tau_{!x_i} \leq \tau_{?x_i} \leq \tau_{!b} \leq \tau_{?b} \leq \tau_{!x_{i+1}}.$$

   Therefore, $\tau_{!x_i} \leq \tau_{?b} \leq \tau_{!x_{i+1}}$.

29. See proof in Appendix C.4.

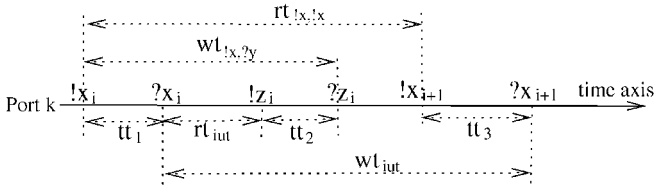

Fig. 15. Situation 1.

8. Proposition 2 and item 7 imply that b is caused by x_i.
9. Item 8 and Proposition 1 imply that, when the $\mathcal{TS}$ observes an output b, it can determine the input x_i which is the cause of b.
10. Items 6 and 9 mean that observability is guaranteed.
11. Items 5 and 10 mean that Condition 1 holds.

APPENDIX B

CONSTRAINTS OF REACTION TIMES

B.1 Situation 1 (Proof of (4))

For clarity, we use the diagram of Fig. 15, where there is no expected output and z_i represents a possible unexpected output (if the $\mathcal{IUT}$ is faulty). In this figure, $tt_1, tt_2, tt_3 \in [TT^{min}, TT^{max}]$, $wt_{!x,?y} \leq WT_{!x,?y}$, $rt_{iut} \leq RT_{iut}$, $wt_{!x,?y} = tt_1 + rt_{iut} + tt_2$, $tt_1 + wt_{iut} = rt_{!x,!x} + tt_3$, and

$$rt_{!x,!x} \in [RT^{min}_{!x,!x}; RT^{max}_{!x,!x}].$$

B.1.1 Proposition 2

The time separating $!x_i$ and $!x_{i+1}$ is $rt_{!x,!x}$. Since

$$rt_{!x,!x} \geq RT^{min}_{!x,!x},$$

then a lower bound of this time is $LB = RT^{min}_{!x,!x}$.

The time separating $!x_i$ and any unexpected $?z_i$ is $tt_1 + rt_{iut} + tt_2$. Since $tt_1, tt_2 \leq TT^{max}$, and $rt_{iut} \leq RT_{iut}$, then an upper bound of this time is $UB = TT^{max} + RT_{iut} + TT^{max}$. Proposition 2 is guaranteed by $UB \leq LB$.

B.1.2 Condition 2

The time separating $?x_i$ and $?x_{i+1}$ is $wt_{iut} = tt_3 + rt_{!x,!x} - tt_1$. Since $tt_3 \leq TT^{max}$, $tt_1 \geq TT^{min}$, and $rt_{!x,!x} \leq RT^{max}_{!x,!x}$, then an upper bound of this time is $UB = TT^{max} + RT^{max}_{!x,!x} - TT^{min}$.

Condition 2 is guaranteed by $UB \leq WT_{iut}$.

B.2 Situation 2

B.2.1 Analogy between Situations 1 and 2 (Proof of (6))

In Situation 1 (see Fig. 7a), Proposition 2 is guaranteed by constraints on the delay $rt_{!x,!x}$ (see (4)). In Situation 2 (see Fig. 7b), this delay is split by $-O1_p$ into two delays $rt_{!x,-O1}$ and $rt_{-O1,!x}$. Proposition 2 is therefore guaranteed if $rt_{!x,-O1} + rt_{-O1,!x}$ satisfies the same constraints as $rt_{!x,!x}$ in Situation 1. Therefore, $[RT^{min}_{!x,-O1} + RT^{min}_{-O1,!x}; RT^{max}_{!x,-O1} + RT^{max}_{-O1,!x}]$ must satisfy the same constraints as $[RT^{min}_{!x,!x}; RT^{max}_{!x,!x}]$.

B.2.2 Proposition 1 in Situation 2 (Proof of (7))

For clarity, we use the diagram of Fig. 16. In this figure, the time separating $-O1_p$ and $?y^p_{i+1}$ is $rt_{-O1,!x} + tt_2 + rt_{iut} + tt_3$. Since $tt_2, tt_3 \geq TT^{min}$, $rt_{iut} \geq 0$, and $rt_{-O1,!x} \geq RT^{min}_{-O1,!x}$, a lower bound of this time is:

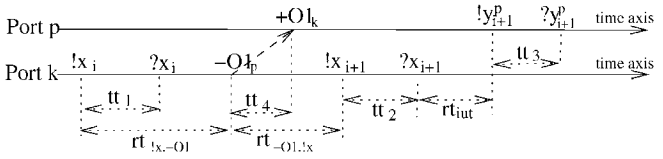

Fig. 16. Situation 2.

$$LB = RT^{min}_{-O1,!x} + TT^{min} + 0 + TT^{min}.$$

The time separating $-O1_p$ and $+O1_k$ is tt_4. Since $tt_4 \leq TT^{max}_{ts}$, an upper bound of this time is $UB = TT^{max}_{ts}$.

The fact that $UB \leq LB$ is a guarantee of Proposition 1, i.e., $+O1_k$ is before $?y^p_{i+1}$.

B.2.3 Existence of Solutions in Situation 2 (Proof of (8))

Let a, b, c be three constants and $\alpha, \beta, \gamma, \delta$ be four *positive* variables. There exist solutions for:

$$[\alpha + \gamma; \beta + \delta] \subseteq [a; b]$$
$$c \leq \gamma$$

if and only if $b \geq \sup(a; c; 0)$. If, for instance, $a \geq 0$, then the condition becomes $b \geq \sup(a; c)$. In (6) and (7),

- $a = (RT_{iut} + 2TT^{max}) \geq 0$, $b = (WT_{iut} - \triangle TT)$, and $c = (TT^{max}_{ts} - 2TT^{min})$;
- $\alpha = RT^{min}_{!x,-O1}$, $\beta = RT^{max}_{!x,-O1}$, $\gamma = RT^{min}_{-O1,!x}$, and

$$\delta = RT^{max}_{-O1,!x}.$$

Therefore, $b \geq \sup(a; c)$ is equivalent to (8) (in Section 6.1.2).

B.3 Situation 3 (Proof of (9))

For clarity, we use the diagram of Fig. 17 where y^k_i is the last output of Υ_i to be received by the $\mathcal{TS}$. In this figure, $tt_1, tt_2, tt_3 \in [TT^{min}; TT^{max}]$, $wt_{!x,?y} \leq WT_{!x,?y}$, $rt_{iut} \leq RT_{iut}$, $wt_{!x,?y} = (tt_1 + rt_{iut} + tt_2)$, $wt_{iut} = (tt_2 + rt_{?y,!x} + tt_3)$, and $rt_{?y,!x} \in [RT^{min}_{?y,!x}; RT^{max}_{?y,!x}]$.

B.3.1 Proposition 2

The time separating $!x_i$ and $!x_{i+1}$ is $tt_1 + rt_{iut} + tt_2 + rt_{?y,!x}$. Since $tt_2 \geq TT^{min}$, $rt_{iut} \geq 0$, and $rt_{?y,!x} \geq RT^{min}_{?y,!x}$ then, for a given tt_1, a lower bound of this time is:

$$LB = tt_1 + 0 + TT^{min} + RT^{min}_{?y,!x}.$$

The time separating $!x_i$ and $?y^k_i$ (possibly unexpected) is $tt_1 + rt_{iut} + tt_2$. Since $tt_2 \leq TT^{max}$ and $rt_{iut} \leq RT_{iut}$ then, for a given tt_1, an upper bound of this time is:

$$UB = tt_1 + RT_{iut} + TT^{max}.$$

Proposition 2 is guaranteed by $UB \leq LB$.

B.3.2 Condition 2

The time separating $!y^k_i$ and $?x_{i+1}$ is $wt_{iut} = tt_2 + rt_{?y,!x} + tt_3$. Since $tt_2, tt_3 \leq TT^{max}$ and $rt_{?y,!x} \leq RT^{max}_{?y,!x}$, then an upper bound of this time is $UB = (TT^{max} + RT^{max}_{?y,!x} + TT^{max})$.

Condition 2 is guaranteed by $UB \leq WT_{iut}$.

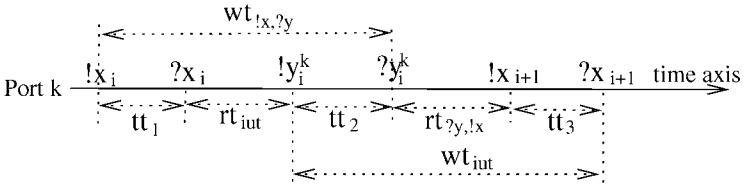

Fig. 17. Situation 3.

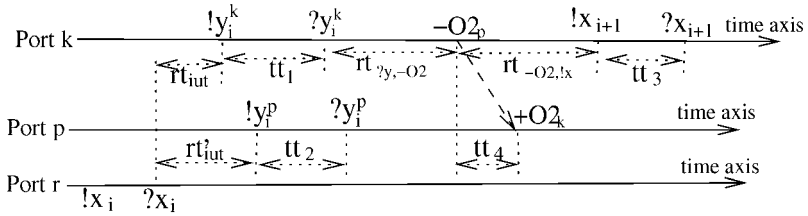

Fig. 18. Situation 4 when: $y_i^p \neq \varepsilon$ and $y_{i+1}^p = \varepsilon$.

B.4 Existence of Solutions in Situations 1 and 3 (Proof of (5))

Let a, b be two constants and α, β be two *positive* variables. There exist solutions for $[\alpha; \beta] \subseteq [a; b]$ if and only if: $b \geq \sup(a; 0)$. If, for instance, $a \geq 0$, then the condition becomes $b \geq a$.

In Equation 4 (Situation 1), $\alpha = RT^{min}_{!x,!x}$, $\beta = RT^{max}_{!x,!x}$, $a = (RT_{iut} + 2TT^{max}) \geq 0$, and $b = (WT_{iut} - \triangle TT)$.

In Equation 9 (Situation 3), $\alpha = RT^{min}_{?y,!x}$, $\beta = RT^{max}_{?y,!x}$, $a = (RT_{iut} + \triangle TT) \geq 0$, and $b = (WT_{iut} - 2TT^{max})$.

In both cases, $b \geq a$ is equivalent to (5) (in Section 6.1.1).

B.5 Situation 4

B.5.1 Analogy between Situations 3 and 4 (Proof of (10))

In Situation 3 (see Fig. 8a), Proposition 2 is guaranteed by constraints on the delay $rt_{?y,!x}$ (see (9)). In Situation 4 (see Fig. 8b), this delay is split by $-O2_p$ into two delays $rt_{?y,-O2}$ and $rt_{-O2,!x}$. Proposition 2 is therefore guaranteed if $rt_{?y,-O2} + rt_{-O2,!x}$ satisfies the same constraints as $rt_{?y,!x}$ in Situation 3. Therefore, $[RT^{min}_{?y,-O2} + RT^{min}_{-O2,!x}; RT^{max}_{?y,-O2} + RT^{max}_{-O2,!x}]$ must satisfy the same constraints as $[RT^{min}_{?y,!x}; RT^{max}_{?y,!x}]$.

B.5.2 Proposition 1 in Situation 4: First Case (Proof of (11))

For clarity, we use the diagram of Fig. 18. The time separating $?x_i$ and $+O2_k$ is $rt_{iut} + tt_1 + rt_{?y,-O2} + tt_4$. Since $tt_1 \geq TT^{min}$, $tt_4 \geq TT^{min}_{ts}$, $rt_{iut} \geq 0$, and $rt_{?y,-O2} \geq RT^{min}_{?y,-O2}$, then a lower bound of this time is $LB = 0 + TT^{min} + RT^{min}_{?y,-O2} + TT^{min}_{ts}$.

The time separating $?x_i$ and $?y_i^p$ is: $rt'_{iut} + tt_2$. Since $tt_2 \leq TT^{max}$ and $rt'_{iut} \leq RT_{iut}$, then an upper bound of this time is $UB = RT_{iut} + TT^{max}$.

The fact that $UB \leq LB$ is a guarantee of Proposition 1, i.e., $+O2_k$ is after $?y_i^p$.

B.5.3 Proposition 1 in Situation 4: Second Case (Proof of (12))

For clarity, we use the diagram of Fig. 19. The time separating $-O2_p$ and $?y_{i+1}^p$ is $rt_{-O2,!x} + tt_2 + rt'_{iut} + tt_3$. Since $tt_2, tt_3 \geq TT_{min}$, $rt'_{iut} \geq 0$, and $rt_{-O2,!x} \geq RT^{min}_{-O2,!x}$, then a lower bound of this time is:

$$LB = RT^{min}_{-O2,!x} + TT^{min} + 0 + TT^{min}.$$

The time separating $-O2_p$ and $+O2_k$ is tt_4. Since $tt_4 \leq TT^{max}_{ts}$, then an upper bound of this time is $UB = TT^{max}_{ts}$.

The fact that $UB \leq LB$ is a guarantee of Proposition 1, i.e., $+O2_k$ is before $?y^p_{i+1}$.

B.5.4 Existence of Solutions in Situation 4 (Proof of (13))

Let a, b, c, d be four constants and $\alpha, \beta, \gamma, \delta$ be four *positive* variables. There exist solutions for:

$$[\alpha + \gamma; \beta + \delta] \subseteq [a; b]$$
$$c \leq \alpha$$
$$d \leq \gamma.$$

if and only if $b \geq \sup(a; c; d; c + d; 0)$. If, for instance, $a \geq \sup(c; 0)$, then the condition becomes

$$b \geq \sup(a; d; c + d).$$

In (10), (11), and (12):

- $a = (RT_{iut} + \triangle TT)$, $b = (WT_{iut} - 2TT^{max})$,
$$c = (RT_{iut} + \triangle TT - TT^{min}_{ts}),$$
and $d = (TT^{max}_{ts} - 2TT^{min})$; (Note that $a \geq \sup(c; 0)$).
- $\alpha = RT^{min}_{?y,-O2}$, $\beta = RT^{max}_{?y,-O2}$, $\gamma = RT^{min}_{-O2,!x}$, and
$$\delta = RT^{max}_{-O2,!x}.$$

Therefore, $b \geq \sup(a; d; c + d)$ is equivalent to (13) (in Section 6.1.4).

B.6 Situation 5

B.6.1 Analogy between Situations 1 and 5 (Proof of (14))

In Situation 1 (see Fig. 7a), Proposition 2 is guaranteed by constraints on the delay $rt_{!x,!x}$ (see (4)). In Situation 5 (see Fig. 9a), this delay is split into three delays $rt_{!x,-C1}$, tt_{ts}, and $rt_{+C1,!x}$, where tt_{ts} is the delay separating $-C1_k$ and $+C1_h$. Proposition 2 is therefore guaranteed if $rt_{!x,-C1} + tt_{ts} + rt_{+C1,!x}$ satisfies the same constraints as $rt_{!x,!x}$ in Situation 1. Therefore,

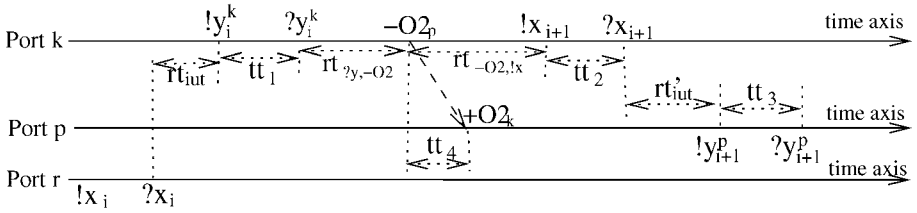

Fig. 19. Situation 4 when: $y_i^p = \varepsilon$ and $y_{i+1}^p \neq \varepsilon$

$$[RT^{min}_{!x,-C1} + TT^{min}_{ts} + RT^{min}_{+C1,!x}; RT^{max}_{!x,-C1} + TT^{max}_{ts} + RT^{max}_{+C1,!x}]$$

must satisfy the same constraints as $[RT^{min}_{!x,!x}; RT^{max}_{!x,!x}]$.

B.6.2 Existence of Solutions in Situation 5 (Proof of (15))

The proof here is quite similar to the proof in Section B.4. Let a, b be two constants and α, β be two *positive* variables. There exist solutions for $[\alpha; \beta] \subseteq [a; b]$ if and only if $b \geq \sup(a; 0)$. In (14),

- $a = (RT_{iut} + 2TT^{max} - TT^{min}_{ts})$ and
$$b = (WT_{iut} - \triangle TT - TT^{max}_{ts}).$$
- $\alpha = (RT^{min}_{!x,-C1} + RT^{min}_{+C1,!x})$ and
$$\beta = (RT^{max}_{!x,-C1} + RT^{max}_{+C1,!x}).$$

Therefore, $b \geq \sup(a; 0)$ is equivalent to (15) (in Section 6.1.5).

B.7 Situation 6

B.7.1 Analogy between Situations 5 and 6 (Proof of (16))

In Situation 5 (see Fig. 9a), Proposition 2 is guaranteed by constraints on the delay $rt_{!x,-C1} + rt_{+C1,!x}$ (see (14)). In Situation 6 (see Fig. 9b), the delay $rt_{+C1,!x}$ is split into two delays, $rt_{+C1,-O3}$ and $rt_{-O3,!x}$. Proposition 2 is therefore guaranteed if $rt_{!x,-C1} + rt_{+C1,-O3} + rt_{-O3,!x}$ satisfies the same constraints as $rt_{!x,-C1} + rt_{+C1,!x}$ in Situation 5. Therefore,

$$[RT^{min}_{!x,-C1} + RT^{min}_{+C1,-O3} + RT^{min}_{-O3,!x};$$
$$RT^{max}_{!x,-C1} + RT^{max}_{+C1,-O3} + RT^{max}_{-O3,!x}]$$

must satisfy the same constraints as

$$[RT^{min}_{!x,-C1} + RT^{min}_{+C1,!x}; RT^{max}_{!x,-C1} + RT^{max}_{+C1,!x}].$$

B.7.2 Analogy between Situations 2 and 6 (Proof of (17))

In Situation 2, Proposition 1 is guaranteed by a constraint on the lower bound $RT^{min}_{-O1,!x}$ of $rt_{-O1,!x}$ (see (7)). The proof (see Section B.2.2) depends only on instants of events occurring after $-O1_p$ relatively to the instant of $-O1_p$.

Situations 2 and 6 (see Figs. 7b and 9b) are similar from the instant when a message O is sent. The only difference is the use of messages $O1$ and $O3$, respectively.

Therefore, in Situation 6, the lower bound $RT^{min}_{-O3,!x}$ of $rt_{-O3,!x}$ must satisfy the same constraint as $RT^{min}_{-O1,!x}$ in Situation 2.

B.7.3 Existence of Solutions in Situation 6 (Proof of (18))

The proof here is quite similar to the proof of Section B.2.3 . Let a, b, c be three constants and $\alpha, \beta, \gamma, \delta$ be four *positive* variables. There exist solutions for:

$$[\alpha + \gamma; \beta + \delta] \subseteq [a; b]$$
$$c \leq \gamma$$

if and only if $b \geq \sup(a; c; 0)$. In (16) and (17),

- $a = (RT_{iut} + 2TT^{max} - TT^{min}_{ts})$,
$$b = (WT_{iut} - \triangle TT - TT^{max}_{ts}),$$
and $c = (TT^{max}_{ts} - 2TT^{min})$;
- $\alpha = (RT^{min}_{!x,-C1} + RT^{min}_{+C1,-O3})$,
$$\beta = (RT^{max}_{!x,-C1} + RT^{max}_{+C1,-O3}),$$
$\gamma = RT^{min}_{-O3,!x}$, and $\delta = RT^{max}_{-O3,!x}$.

Therefore, $b \geq \sup(a; c; 0)$ is equivalent to (18) (in Section 6.1.6).

B.8 Situation 7

B.8.1 Analogy between Situations 3 and 7 (Proof of (19))

In Situation 3 (see Fig. 8a), Proposition 2 is guaranteed by constraints on the delay $rt_{?y,!x}$ (see (9)). In Situation 7 (see Fig. 9a), this delay is split into three delays, $rt_{?y,-C2}$, tt_{ts}, and $rt_{+C2,!x}$, where tt_{ts} is the delay separating $-C2_k$ and $+C2_m$. Proposition 2 is therefore guaranteed if $rt_{?y,-C2} + tt_{ts} + rt_{+C2,!x}$ satisfies the same constraints as $rt_{?y,!x}$ in Situation 3. Therefore,

$$[RT^{min}_{?y,-C2} + TT^{min}_{ts} + RT^{min}_{+C2,!x};$$
$$RT^{max}_{?y,-C2} + TT^{max}_{ts} + RT^{max}_{+C2,!x}]$$

must satisfy the same constraints as $[RT^{min}_{?y,!x}; RT^{max}_{?y,!x}]$.

B.8.2 Existence of Solutions in Situation 7 (Proof of (20))

The proof here is quite similar to the proof in Section B.4. Let a, b be two constants and α, β be two *positive* variables. There exist solutions for $[\alpha; \beta] \subseteq [a; b]$ if and only if $b \geq \sup(a; 0)$. In (19),

- $a = (RT_{iut} + \triangle TT - TT^{min}_{ts})$ and
$$b = (WT_{iut} - 2TT^{max} - TT^{max}_{ts});$$
- $\alpha = (RT^{min}_{?y,-C2} + RT^{min}_{+C2,!x})$ and
$$\beta = (RT^{max}_{?y,-C2} + RT^{max}_{+C2,!x}).$$

Therefore, $b \geq \sup(a; 0)$ is equivalent to (20) (in Section 6.1.7).

B.9 Situation 8

B.9.1 Analogy between Situations 7 and 8 (Proof of (21))

In Situation 7 (see Fig. 10a), Proposition 2 is guaranteed by constraints on the delay $rt_{?y,-C2} + rt_{+C2,!x}$ (see (19)). In Situation 8 (see Fig. 10b), the delay $rt_{+C2,!x}$ is split into two

delays $rt_{+C2,-O4}$ and $rt_{-O4,!x}$. Proposition 2 is therefore guaranteed if $rt_{?y,-C2} + rt_{+C2,-O4} + rt_{-O4,!x}$ satisfies the same constraints as $rt_{?y,-C2} + rt_{+C2,!x}$ in Situation 7. Therefore,

$$[RT^{min}_{?y,-C2} + RT^{min}_{+C2,-O4} + RT^{min}_{-O4,!x};$$
$$RT^{max}_{?y,-C2} + RT^{max}_{+C2,-O4} + RT^{max}_{-O4,!x}]$$

must satisfy the same constraints as

$$[RT^{min}_{?y,-C2} + RT^{min}_{+C2,!x}; RT^{max}_{?y,-C2} + RT^{max}_{+C2,!x}].$$

B.9.2 Analogy between Situations 4 and 8 (Proofs of (22) and (23))

In Situation 4, Proposition 1 is guaranteed by constraints on lower bounds $RT^{min}_{?y,-O2}$ and $RT^{min}_{-O2,!x}$ of $rt_{?y-O2}$ and $rt_{-O2,!x}$, respectively (see (11) and (12)).

Constraint of $RT^{min}_{?y,-O2}$: We obtain Situation 8 from Situation 4 by splitting the delay $rt_{?y,-O2}$ into three delays $rt_{?y,-C2}$, tt_{ts}, and $rt_{+C2,-O4}$, where tt_{ts} is the delay separating $-C2_k$ and $+C2_m$. Therefore, in Situation 8, the lower bound $RT^{min}_{?y,-C2} + TT^{min}_{ts} + RT^{min}_{+C2,-O4}$ of $rt_{?y,-C2} + tt_{ts} + rt_{+C2,-O4}$ must satisfy the same constraint as $RT^{min}_{?y,-O2}$ in Situation 4.

Constraint of $RT^{min}_{-O2,!x}$ is computed by using an analogy between Situations 4 and 8 similar to the analogy between Situations 2 and 6 (see Section B.7.2). Therefore, in Situation 8, the lower bound $RT^{min}_{-O4,!x}$ of $rt_{-O4,!x}$ must satisfy the same constraint as $RT^{min}_{-O2,!x}$ in Situation 4.

B.9.3 Existence of Solutions in Situation 8 (Proof of (24))

The proof here is quite similar to the proof in Section B.5.4. Let a, b, c, d be four constants and $\alpha, \beta, \gamma, \delta$ be four *positive* variables. There exist solutions for:

$$[\alpha + \gamma; \beta + \delta] \subseteq [a; b]$$
$$c \leq \alpha$$
$$d \leq \gamma$$

if and only if, $b \geq \sup(a; c; d; c + d; 0)$. If, for instance, $a \geq c$, then the condition becomes $b \geq \sup(a; d; c + d; 0)$. In (21), (22), and (23):

- $$a = (RT_{iut} + \triangle TT - TT^{min}_{ts}),$$
  $$b = (WT_{iut} - 2TT^{max} - TT^{max}_{ts}),$$
  $$c = (RT_{iut} + \triangle TT - 2TT^{min}_{ts}),$$
  and $d = (TT^{max}_{ts} - 2TT^{min})$; (Note that $a \geq c$).
- $$\alpha = RT^{min}_{?y,-C2} + RT^{min}_{+C2,-O4},$$
  $$\beta = RT^{max}_{?y,-C2} + RT^{max}_{+C2-O4},$$
  $$\gamma = RT^{min}_{-O4,!x},$$
  and $\delta = RT^{max}_{-O4,!x}$.

Therefore, $b \geq \sup(a; d; c + d; 0)$ is equivalent to (24) (in Section 6.1.8).

B.10 Global Condition

For clarity, we first define the following parameters:

1. $K_1 = RT_{iut} + 3TT^{max} - TT^{min}$,
2. $K_2 = RT_{iut} + 3TT^{max} - TT^{min} + \triangle TT_{ts}$,
3. $K_3 = RT_{iut} + 3\triangle TT + \triangle TT_{ts}$,
4. $K_4 = RT_{iut} + 3\triangle TT + 2\triangle TT_{ts}$,
5. $L_1 = TT^{max}_{ts} + TT^{max} - 3TT^{min}$,
6. $L_2 = TT^{max}_{ts} + \triangle TT$,
7. $L_3 = TT^{max}_{ts} + 2\triangle TT$,
8. $L_4 = TT^{max}_{ts} + 2TT^{max}$,
9. $L_5 = 2TT^{max}_{ts} + TT^{max} - 3TT^{min}$,
10. $L_6 = 2TT^{max}_{ts} + 2\triangle TT$.

Equations (5), (8), (13), (15), (18), (20), and (24) may then be written as follows:

1. Equation 5: $WT_{iut} \geq K_1$,
2. Equation 8: $WT_{iut} \geq \sup(K_1; L_1)$,
3. Equation 13: $WT_{iut} \geq \sup(K_1; K_3; L_3)$,
4. Equation 15: $WT_{iut} \geq \sup(K_2; L_2)$,
5. Equation 18: $WT_{iut} \geq \sup(K_2; L_2; L_5)$,
6. Equation 20: $WT_{iut} \geq \sup(K_2; L_4)$,
7. Equation 24: $WT_{iut} \geq sup(K_2; K_4; L_4; L_6)$.

Therefore,

1. Equation 8 is stronger than (5) (trivial).
2. Equation 13 is stronger than (8) because $L_3 \geq L_1$.
3. Equation 15 is stronger than (8) because $K_2 \geq K_1$ and $L_2 \geq L_1$.
4. Equation 18 is stronger than (15) (trivial).
5. Equation 20 is stronger than (15) because $L_4 \geq L_2$.
6. Equation 24 is stronger than (13) because $K_2 \geq K_1$, $K_4 \geq K_3$, and $L_6 \geq L_3$.
7. Equation 24 is stronger than (18) because $L_4 \geq L_2$ and $L_6 \geq L_5$.
8. Equation 24 is stronger than (20) (trivial).

Using transitivity of relation "is stronger than," we can easily prove that (24) is stronger than (5), (8), (13), (15), (18), and (20).

APPENDIX C
CONSTRAINTS OF WAITING TIMES

For clarity, we use diagrams of Fig. 20.

C.1 Situation A (Proof of (25))

For clarity, we use the diagram of Fig. 20a. $WT_{\varepsilon,?y}$ is an upper bound of the time $wt_{\varepsilon,?y} = tt_1 + rt_{iut} + tt_2$ separating the starting instant (i.e., the instant of $!x_1$) and $?y^m_1$. Since $tt_1, tt_2 \leq TT^{max}$ and $rt_{iut} \leq RT_{iut}$, an upper bound of $wt_{\varepsilon,?y}$ is $TT^{max} + RT_{iut} + TT^{max}$.

C.2 Situation B (Proof of (26))

For clarity, we use the diagram of Fig. 20b. $WT_{!x,?y}$ is an upper bound of the time $wt_{!x,?y} = tt_1 + rt_{iut} + tt_2$ separating $!x_i$ and $?y^m_i$. Since $tt_1, tt_2 \leq TT^{max}$ and $rt_{iut} \leq RT_{iut}$, an upper bound of $wt_{!x,?y}$ is $TT^{max} + RT_{iut} + TT^{max}$.

C.3 Situation C (Proof of (27))

For clarity, we use the diagram of Fig. 20c. $WT_{+O,?y}$ is an upper bound of the time $wt_{+O,?y}$ separating $+O_h$ and $?y^m_i$, where O may be $O1, O2, O3$, or $O4$.

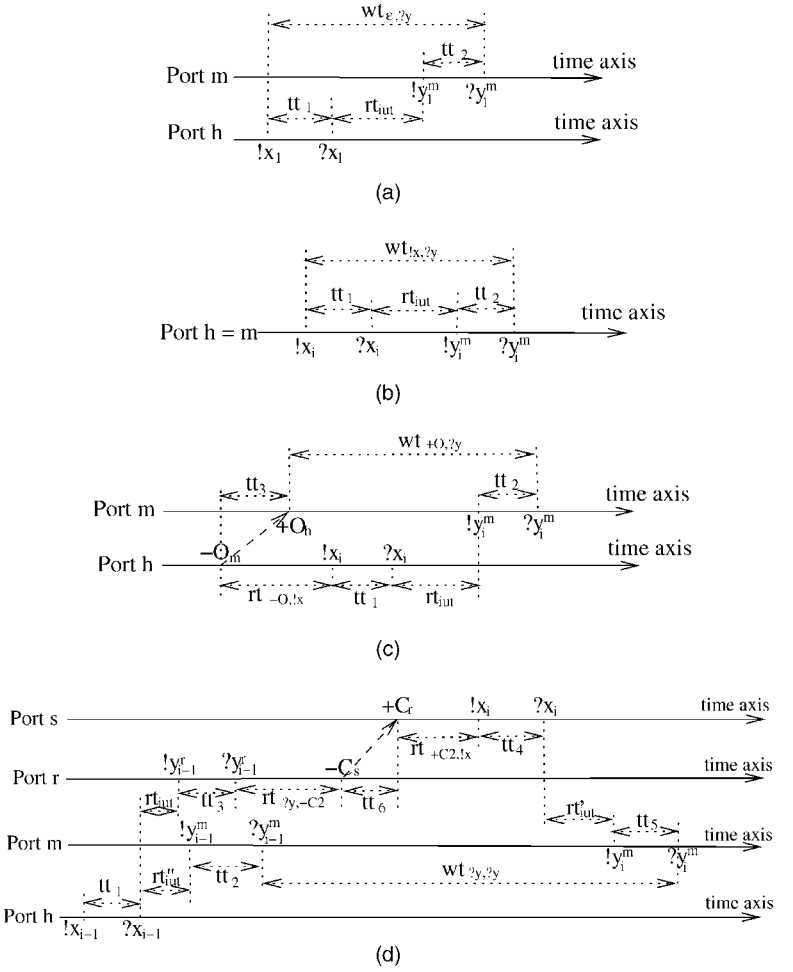

Fig. 20. Waiting times. (a) Situation A. (b) Situation B. (c) Situation C. (d) Situation D.

The time separating $-O_m$ and $?y_i^m$ is:

$$rt_{-O,!x} + tt_1 + rt_{iut} + tt_2.$$

Since $tt_1, tt_2 \leq TT^{max}$, $rt_{iut} \leq RT_{iut}$, and

$$rt_{-O,!x} \leq \sup(RT^{max}_{-O1,!x}; RT^{max}_{-O2,!x}; RT^{max}_{-O3,!x}; RT^{max}_{-O4,!x}),$$

an upper bound of this time is:

$$UB = \sup(RT^{max}_{-O1,!x}; RT^{max}_{-O2,!x}; RT^{max}_{-O3,!x}; RT^{max}_{-O4,!x}) + TT^{max} + RT_{iut} + TT^{max}.$$

The time separating $-O_m$ and $+O_h$ is tt_3. Since $tt_3 \geq TT^{min}_{ts}$, a lower bound of this time is $LB = TT^{min}_{ts}$.

The value $UB - LB$ is an upper bound of the time separating $+O_h$ and $?y_i^m$. We have:

$$UB - LB = \sup(RT^{max}_{-O1,!x}; RT^{max}_{-O2,!x}; RT^{max}_{-O3,!x}; RT^{max}_{-O4,!x}) + TT^{max} + RT_{iut} + TT^{max} - TT^{min}_{ts}.$$

C.4 Situation D (Proof of (28))

For clarity, we use the diagram of Fig. 20d. $WT_{?y,?y}$ is an upper bound of the time $wt_{?y,?y}$ separating $?y^m_{i-1}$ and $?y^m_i$.

The time separating $?x_{i-1}$ and $?y^m_i$ is:

$$rt_{iut} + tt_3 + rt_{?y,-C2} + tt_6 + rt_{+C2,!x} + tt_4 + rt'_{iut} + tt_5.$$

Since $tt_3, tt_4, tt_5 \leq TT^{max}$, $tt_6 \leq TT^{max}_{ts}$, $rt_{iut}, rt'_{iut} \leq RT_{iut}$, $rt_{?y,-C2} \leq RT^{max}_{?y,-C2}$, and $rt_{+C2,!x} \leq RT^{max}_{+C2,!x}$, an upper bound of this time is:

$$UB = (RT_{iut} + TT^{max} + RT^{max}_{?y,-C2} + TT^{max}_{ts} + RT^{max}_{+C2,!x} + TT^{max} + RT_{iut} + TT^{max}).$$

The time separating $?x_{i-1}$ and $?y^m_{i-1}$ is $rt''_{iut} + tt_2$. Since $tt_2 \geq TT^{min}$ and $rt''_{iut} \geq 0$, a lower bound of this time is $LB = (0 + TT^{min})$.

The value $UB - LB$ is an upper bound of the time separating $?y^m_{i-1}$ and $?y^m_i$. We have:

$$UB - LB = (RT_{iut} + TT^{max} + RT^{max}_{?y,-C2} + TT^{max}_{ts} + RT^{max}_{+C2,!x} + TT^{max} + RT_{iut} + TT^{max} - TT^{min}).$$

ACKNOWLEDGMENTS

The author thanks the reviewers for their suggestions and comments towards the improvement of this manuscript.

REFERENCES

[1] G. Luo, R. Dssouli, G.v. Bochmann, P. Venkataram, and A. Ghedamsi, "Test Generation with Respect to Distributed Interfaces," *Computer Standards and Interfaces*, vol. 16, pp. 119–132, 1994.

[2] L. Cacciari and O. Rafiq, "Controllability and Observability in Distributed Testing," *Information and Software Technology*, 1999.

[3] M. Benattou, L. Cacciari, R. Pasini, and O. Rafiq, "Principles an Tools for Testing Open Distributed Systems," *Proc. 12th Int'l Workshop Testing of Communicating Systems (IWTCS),* pp. 77–92, Sept. 1999.
[4] A. Khoumsi, "Timing Issues in Testing Distributed Systems," *Proc. Fourth IASTED Int'l Conf. Software Eng. Applications (SEA),* Nov. 2000.
[5] C. Jard, T. Jéron, H. Kahlouche, and C. Viho, "Towards Automatic Distribution of Testers for Distributed Conformance Testing," *Proc. Protocol Specification, Testing, and Verification—Formal Description Techniques Conf.,* Nov. 1998.
[6] C. Jard, T. Jéron, L. Tanguy, and C. Viho, "Remote Testing Can Be As Powerful As Local Testing," *Proc. Protocol Specification, Testing, and Verification—Formal Description Techniques Conf.,* Oct. 1999.
[7] J. Zhang, S.-C. Cheung, and S.T. Chanson, "Stress Testing of Distributed Multimedia Software Systems," *Proc. Protocol Specification, Testing, and Verification—Formal Description Techniques Conf.,* Oct. 1999.
[8] J. Bi and J. Wu, "A Formal Approach to Conformance Testing of Distributed Routing Protocols," *Proc. Protocol Specification, Testing, and Verification—Formal Description Techniques Conf.,* Oct. 1999.
[9] A.S. Gokhale and D.C. Schmidt, "Measuring and Optimizing CORBA Latency and Scalability over High-Speed Networks," *IEEE Trans. Computers,* vol. 47, no. 4, pp. 391–413, 1998.
[10] J. Tretmans, "A Formal Approach to Conformance Testing," PhD thesis, Univ. of Twente, The Netherlands, Dec. 1992.

Ahmed Khoumsi received the engineer degree in aeronautics and automation from the engineer school SUP'AERO (Toulouse, France), in 1984. From 1984 to 1988, he achieved his research activities in the LAAS, a CNRS Research Center, in Toulouse. In 1988, he received the PhD degree in robotics and automation from the University Paul Sabatier in Toulouse. From 1989 to 1992, he was an assistant professor in robotics and computer engineering at the engineer school ENSEM (Casablanca, Morocco). From 1993 to 1996, he was a postdoctoral fellow in the Communication Protocols Group at the University of Montreal. From 1996 to June 2000, he was an assistant professor and, currently, he is an associate professor in the Department of Electrical and Computer Engineering, at the University of Sherbrooke, in Canada. His present research activities include: testing and control of distributed and real-time systems and detection and resolution of feature interactions in telecommunications systems. He is a member of the IEEE.