Journal of Zhejiang University SCIENCE A
ISSN 1673-565X (Print); ISSN 1862-1775 (Online)
www.zju.edu.cn/jzus; www.springerlink.com
E-mail: jzus@zju.edu.cn

Timing issues in distributed testing

HUANG Chuan-dong, JIANG Fan

(*Department of Computer Science, University of Science and Technology of China, Hefei 230027, China*)

E-mail: yellow9611@gmail.com; fjiang@ustc.edu.cn

Received Feb. 20, 2006; revision accepted Nov. 28, 2006

Abstract: The objective of conformance testing is to determine whether an implementation under test (IUT) conforms to its specification. In distributed test architecture where there are multiple remote testers, the objective can be complicated by the fact that testers may encounter controllability and observability problems during the application of a test sequence. A certain amount of work has been done in the area of generating test sequence that is free from these problems. However, few researchers investigate them from the aspect of test execution. This work studies the test execution phase when test sequences are applied to the implementation and it is pointed out that controllability and observability problems can be resolved if and only if the test system implements some timing constraints. When determining these constraints, the dynamic time information during test is taken into account, which reduces the test execution time and improves test efficiency further.

Key words: Conformance testing, Distributed system, Controllability, Observability, Reaction time

doi:10.1631/jzus.2007.A0522 **Document code:** A **CLC number:** TP311

INTRODUCTION

Testing aimed at ensuring the quality of the implementation, is often carried out by generating test sequences from the specification and applying them to the implementation in a test architecture. When testing a distributed system, a distributed test architecture (Luo *et al*., 1994) as illustrated in Fig.1 is needed. In this architecture, the implementation under test (IUT) contains a number of separate interfaces, called ports and the test system consists of a local tester for each port of the IUT. Each local tester communicates with the IUT through its corresponding port.

During the application of a test sequence in distributed testing, the existence of multiple testers raises the possibility of coordination problems among testers known as controllability and observability problems (Rafiq and Cacciari, 2003). These problems occur if a tester cannot determine when to apply a particular input to the IUT or whether a particular output from the IUT is generated in response to a specific input, respectively.

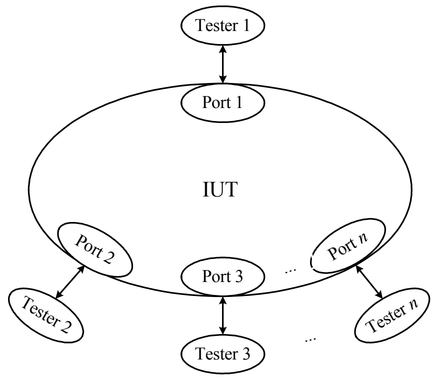

Fig.1 Distributed test architecture

Controllability and observability problems have great influence on several aspects of the testing activity, such as the execution of test sequences, the fault detectability of test system and the interpretation of testing results. To resolve these problems, it is often necessary for testers to exchange coordination messages directly through reliable communication channels which are independent of the IUT (Rafiq and Cacciari, 2003). Many researches have been done in the area of generating test sequence that is free from controllability and observability problems and that either uses no coordination messages or uses a minimum number of coordination messages (Hierons, 2001; Liu *et al*., 2003; Luo *et al*., 1994; Rafiq and Cacciari, 2003; Ural and Whittier, 2003). All of them made a simplifying assumption that the time required for a coordination message to travel from a tester to another, is greater than the reaction time of the IUT, i.e., the time elapsed between the reception of an input by the IUT and the sending of the corresponding output by the IUT. If not, the test system may produce incorrect test results such as hiding a possible fault or detecting a non-existing one (Rafiq and Cacciari, 2003; Khoumsi, 2002). Unlike previous researchers, Khoumsi (2002) investigated the timing issues in distributed testing from the aspect of test execution and determined some timing constraints. Khoumsi (2002) showed that controllability and observability problems are indeed resolved if and only if the test system observes those timing constraints. However, there are some deficiencies in his method, especially as the constraints are statically determined before test execution and the dynamic time information during test execution was not considered, which brings out the possibility that the tester system cannot send inputs to the IUT as quickly as possible. Hence, this paper mainly copes with the problem of dynamically determining those timing constraints.

The rest of the paper is organized as follows: Section 2 gives the preliminaries. In Section 3, we present our motivation and objectives. In Section 4, we determine the timing constraints for reaction time of the test system. Section 5 illustrates the application of those constraints to an example. Finally, conclusions are given.

PRELIMINARIES

Communication model

Consider the distributed test architecture illustrated in Fig.1. In our communication model, each local tester communicates with the IUT through reliable communication medium. Two testers can also exchange coordination messages through reliable communication channels. For simplicity, we assume that the transfer time between tester and IUT is zero. The transfer time between testers is assumed to fall within a bounded interval $[TT_{ts}^{min}, TT_{ts}^{max}]$. This hypothesis is realistic because the advent of real-time middleware such as real CORBA is foreseen, probably in the future.

We also assume that each tester uses its local clock and that the local clocks are not synchronized, i.e., there is no global clock. This implies that the transmit time of a coordination message cannot be measured by reading the local clocks of the sender and the receiver respectively.

IUT model

We assume that the reaction time of the IUT is bounded by a finite value RT_{iut}. Similar to (Khoumsi, 2002), we also assume that the behaviors of the IUT, even if it is faulty, can be described by the multi-port finite state machine model.

A multi-port finite state machine with n ports (np-FSM) is a 7-tuple $M=(S, P, \Sigma, \Gamma, \delta, \lambda, s_0)$. S is a finite set of states and $s_0\in S$ is the initial state. $P=\{1, 2, \ldots, n\}$ is the set of ports. $\Sigma=(\Sigma_1, \Sigma_2, ..., \Sigma_n)$, where Σ_k is the input alphabet of port k, and $\Sigma_i\cap\Sigma_j=\varnothing$, for $i\neq j$, i, j, $k\in P$. Let $I=\Sigma_1\cup\Sigma_2\cup...\cup\Sigma_n$. $\Gamma=(\Gamma_1, \Gamma_2, ..., \Gamma_n)$, where Γ_k is the output alphabet of port k, and $\Gamma_i\cap\Gamma_j=\varnothing$, for $i\neq j$, i, j, $k\in P$. Let $O=(\Gamma_1\cup\{\varepsilon\})\times(\Gamma_2\cup\{\varepsilon\})\times...\times(\Gamma_n\cup\{\varepsilon\})$, where ε stands for the null output. δ is the transition function: $D\rightarrow S$, and λ is the output function $D\rightarrow O$, where $D\subseteq S\times I$.

A transition of an np-FSM M is a triple $(s_j, s_k; x/y)$ where $s_j, s_k\in S$, $x\in I$, $y\in O$, such that $\delta(s_j, x)=s_k$ and $\lambda(s_j, x)=y$. An np-FSM M can be represented by a directed graph $G=(V, E)$ where V represents the set S of states of M and E represents all specified transitions of M. An example of 2p-FSM is given in Fig.2, where $S=\{s_0, s_1, s_2\}$, $P=\{1, 2\}$, $\Sigma_1=\{\alpha\}$, $\Sigma_2=\{\beta\}$, $\Gamma_1=\{a\}$, $\Gamma_2=\{b\}$. In this figure, the transition t_1 denotes that if s_0 is the current state and the input α is received, then state changes to s_1 and the outputs a and b are sent in ports 1 and 2 respectively.

Remark 1 Given $x\in I$, $p\in P$, $y=(y_1, y_2, ..., y_n)\in O$, let $port(x)$ denote the port associated with input x, $Tester_p$ denote the tester at port p and $ports(y)$ denote the set of ports associated with values from y that are not null. Given two consecutive transitions $t=(s_i, s_j; x/y)$ and $t'=(s_j, s_k; x'/y')$, let $U(y, y', x')$ denote the set of ports: $ports(y)\backslash(ports(y')\cup\{port(x')\})$.

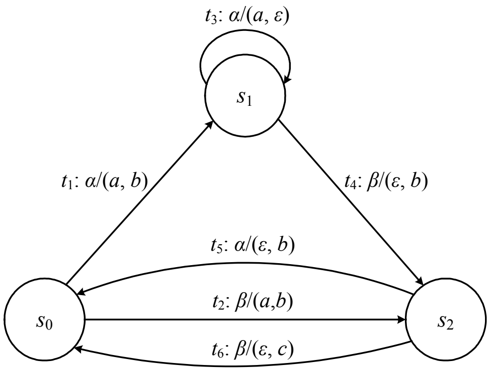

Fig.2 An example of 2p-FSM

Controllability and observability problems

During the application of a test sequence in distributed test architecture, a controllability (synchronization) problem arises if a tester cannot determine when to apply a particular input to the IUT because it is not involved in the previous transition, i.e., it does not send the input or receive any output in the previous transition. Formally, for any two consecutive transitions $t=(s_i, s_j; x/y)$ and $t'=(s_j, s_k; x'/y')$, a controllability (synchronization) problem occurs if $port(x')\notin(ports(y)\cup\{port(x)\})$.

Observability refers to the ease of determining which input triggers a particular output. An observability problem arises when a tester is expecting to receive an output from the IUT in response to the previous input or the current input, not knowing when to start or stop waiting for the output because it does not send the current input. An instance of the observability problem manifests itself as a potentially undetectable output-shifting fault (Luo *et al*., 1994).

Definition 1 Given two consecutive transitions $t=(s_i, s_j; x/y)$ and $t'=(s_j, s_k; x'/y')$ where $y=(y_1, y_2, ..., y_n)$ and $y'=(y_1', y_2', ..., y_n')$, there is a potential output-shifting fault at port p if $(p\neq port(x'))\wedge(y_p\neq\varepsilon \text{ XOR } y_p'\neq\varepsilon)$, $p\in P$.

Due to the lack of a global clock in distributed testing, it is difficult to determine the input which is the cause of a particular output. Even if the behaviors of all the ports are the same as expected, the output-shifting faults may still stay in the IUT (Luo *et al*., 1994; Khoumsi, 2002).

In general, for any two consecutive transitions $t=(s_i, s_j; x/y)$ and $t'=(s_j, s_k; x'/y')$, the following rule guarantees the synchronization of multiple testers and the detection of output-shifting faults.

Rule 1

(1) If $port(x')\notin(ports(y)\cup\{port(x)\})$, then $Tester_{port(x)}$ waits time Δ and sends a control message C to $Tester_{port(x')}$ after sending the input x. Message C may help $Tester_{port(x)}$ and $Tester_{port(x')}$ synchronize.

(2) If $U(y, y', x')\cup U(y', y, x')\neq\varnothing$, then $Tester_{port(x')}$ waits time Δ' and sends an observation message O to each tester at a port from $U(y, y', x')\cup U(y', y, x')$ after receiving expected output in response to x from the IUT or control message C from another tester, or sending the previous input x (in the case of $port(x')=port(x)$). Message O may help detect the potential output-shifting faults.

(3) $Tester_{port(x')}$ waits time Δ'' and sends the next input x' to the IUT.

Remark 2 The sending and reception of an input, output, or message x are denoted by $!x$ and $?x$, respectively. The duration between two events (sending or reception of an input/output or message) e_1 and e_2 is denoted by $\Delta t(e_1, e_2)$.

MOTIVATION AND OBJECTIVE

Khoumsi (2002) showed that the controllability and observability problems are indeed resolved if and only if the wait time Δ, Δ' and Δ'', which are called the reaction time of the test system, satisfy some constraints. In order to determine these timing constraints, Khoumsi (2002) presented the following theorem which is the basis of our work.

Theorem 1 In distributed testing involving multiple testers, if:

Condition 1 Test system sends an input to the IUT only after it has received all the outputs (expected or unexpected) in response to the previous input, and

Condition 2 For any two consecutive transitions $t=(s_i, s_j; x/y)$ and $t'=(s_j, s_k; x'/y')$ where $y=(y_1, y_2, ..., y_n)$ and $y'=(y_1', y_2', ..., y_n')$, if the IUT is correct, every $Tester_p$ receives y_p before it receives observation message O and every $Tester_q$ receives y_q' after it receives O, $p\in U(y, y', x')$, $q\in U(y', y, x')$,

then the multiple testers would be synchronizable and output-shifting faults could be detected, i.e., the controllability and observability problems are indeed resolved.

Proof of the theorem can be referred to (Khoumsi, 2002). Based on Theorem 1, Khoumsi (2002) determined some timing constraints. However, there are some deficiencies in his method, especially the constraints are statically determined before test execution, i.e., the dynamic time information during test execution is not utilized. For example, consider two consecutive transitions $t=(s_i, s_j; x/y)$ and $t'=(s_j, s_k; x'/y')$. Suppose that $port(x)=port(x')=q$, $y_q\neq\varepsilon$, $U(y, y', x')=\varnothing$, and $U(y', y, x')=\varnothing$. In this situation, there is one type of reaction time Δ'' as illustrated in Fig.3. Based on Theorem 1, we only need guarantee $\Delta t(!x, !x')\geq RT_{\text{iut}}$. Because $\Delta t(!x, !x')=\Delta''+\Delta t(!x, ?y_q)$, we obtain: $\Delta''\geq RT_{\text{iut}}-\Delta t(!x, ?y_q)$. Khoumsi (2002) conservatively estimates $\Delta t(!x, ?y_q)$, the reaction time of the IUT, at zero. But in fact, $\Delta t(!x, ?y_q)$ can be locally measured or calculated by $Tester_q$ during test execution and it usually does not equal zero. Obviously, if the dynamic time information $\Delta t(!x, ?y_q)$ is taken into consideration, the test system can send inputs to the IUT at a higher speed, which reduces test execution time and improves test efficiency. Moreover, the extra overhead for that is trivial. Only little time information such as $\Delta t(!x, ?y_q)$ needs to be calculated during the test. Consequently, making use of the dynamic time information during the test execution and determining the constraints for the reaction time of the test system are our main objectives.

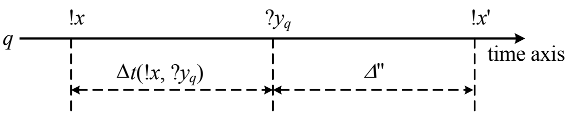

Fig.3 Reaction time of test system in Case 1(1)

CONSTRAINTS OF THE REACTION TIME

For any two consecutive transitions $t=(s_i, s_j; x/y)$ and $t'=(s_j, s_k; x'/y')$ where $y=(y_1, y_2, ..., y_n)$ and $y'=(y_1', y_2', ..., y_n')$, suppose that $port(x)=p$ and $port(y)=q$. In this section, for each possible case of t and t', we determine the timing constraints.

Case 1 $y_q\neq\varepsilon$, $U(y, y', x')\cup U(y', y, x')=\varnothing$

In this case, there is one type of reaction time $\Delta''=\Delta t(?y_q, !x')$. The timing constraint given by Khoumsi (2002) is

$$\Delta''\geq RT_{\text{iut}}. \qquad (1)$$

As we pointed out in Section 3, we only need guarantee $\Delta t(!x, !x')\geq RT_{\text{iut}}$, i.e., $\Delta''+\Delta t(!x, ?y_q)\geq RT_{\text{iut}}$. In the following, $\Delta t(!x, ?y_q)$ is calculated in different situations. Suppose that $'t=(s_h, s_i; 'x/'y)$ where $'y=('y_1, 'y_2, ..., 'y_n)$ is the previous transition of t and $port('x)=m$.

(1) $p=q$

In this situation, as illustrated in Fig.3, the events $!x$ and $?y_q$ both occur at port q. $\Delta t(!x, ?y_q)$ can be measured by $Tester_q$ using its local clock during the test execution. Therefore,

$$\Delta''\geq RT_{\text{iut}}-\Delta t(!x, ?y_q). \qquad (2)$$

(2) $p\neq q$, $'y_q=\varepsilon$

In this situation, as illustrated in Fig.4, $\Delta t(!x, ?y_q) =\Delta t(!O, ?O)-\Delta t(!O, !x)+\Delta t(?O, ?y_q)$. $\Delta t(?O, ?y_q)$ can be measured by $Tester_q$ using its local clock and the duration $\Delta t(!O, !x)$ can be piggybacked by the observation message O. Note that $\Delta t(!O, ?O)\geq TT_{\text{ts}}^{\min}$, then we obtain

$$\Delta''\geq RT_{\text{iut}}-\Delta t(?O, ?y_q)+\Delta t(!O, !x)-TT_{\text{ts}}^{\min}. \qquad (3)$$

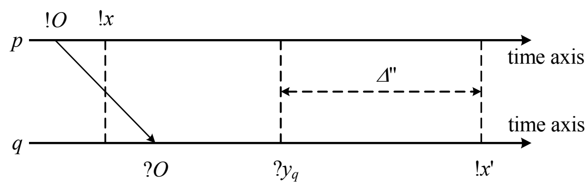

Fig.4 Reaction time of the test system in Case 1(2)

(3) $p\neq q$, $'y_q\neq\varepsilon$, $'y_p\neq\varepsilon$

In this situation, as illustrated in Fig.5, $\Delta t(!x, ?y_q)=\Delta t(?'y_q, ?y_q)-\Delta t(!'x, ?'y_p)-\Delta t(?'y_p, !x)+\Delta t(!'x, ?'y_q)$, where $\Delta t(?'y_q, ?y_q)$ can be locally measured by $Tester_q$ but $\Delta t(?'y_p, !x)$ cannot. We replace $\Delta t(?'y_p, !x)$ with the duration $\Delta t^s(?'y_p, !x)$ which is determined by Khoumsi (2002) before test execution. Since $\Delta t^s(?'y_p, !x)\geq \Delta t(?'y_p, !x)$, $\Delta t(!'x, ?'y_p)\leq RT_{\text{iut}}$ and $\Delta t(!'x, ?'y_q)\geq 0$, Condition 1 is guaranteed by $\Delta''\geq 2RT_{\text{iut}}-\Delta t(?'y_q, ?y_q) +\Delta t^s(?'y_p, !x)$. Combining it with inequality (1), we only require

$$\Delta''\geq \min(RT_{\text{iut}}, 2RT_{\text{iut}}-\Delta t(?'y_q, ?y_q)+\Delta t^s(?'y_p, !x)). \qquad (4)$$

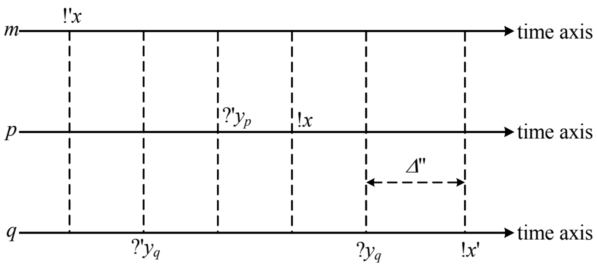

Fig.5 Reaction time of the test system in Case 1(3)

(4) $p\neq q$, $'y_q\neq\varepsilon$, $'y_p=\varepsilon$

In this situation, as illustrated in Fig.6, $\Delta t(!x, ?y_q)=\Delta t(?'y_q, ?y_q)-\Delta t(!'x, !C)-\Delta t(!C, ?C)-\Delta t(?C, !x)+\Delta t(!'x, ?'y_q)$. $\Delta t(?'y_q, ?y_q)$ can be locally measured by $Tester_q$, but $\Delta t(!'x, !C)$ and $\Delta t(?C, !x)$ cannot. We replace $\Delta t(!'x, !C)$ and $\Delta t(?C, !x)$ with the statically determined duration $\Delta t^s(!'x, !C)$ and $\Delta t^s(?C, !x)$, respectively. Note that $\Delta t^s(!'x, !C)\geq\Delta t(!'x, !C)$, $\Delta t^s(?C, !x)\geq\Delta t(?C, !x)$, $\Delta t(!C, ?C)\leq TT_{ts}^{max}$, $\Delta t(!'x, ?'y_q)\geq 0$. Then Condition 1 is guaranteed by $\Delta''\geq RT_{iut}+TT_{ts}^{max}-\Delta t(?'y_q, ?y_q)+\Delta t^s(!'x, !C)+\Delta t^s(?C, !x)$. Combining it with inequality (1), we only require

$$\Delta''\geq\min(RT_{iut}, RT_{iut}+TT_{ts}^{max}-\Delta t(?'y_q, ?y_q)+\Delta t^s(!'x, !C)+\Delta t^s(?C, !x)). \quad (5)$$

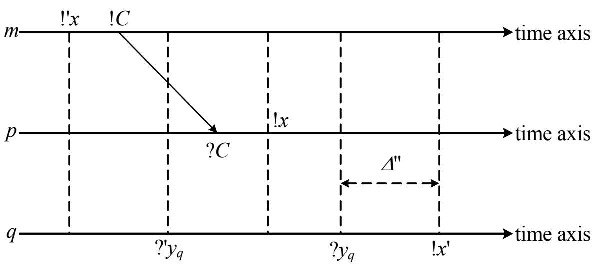

Fig.6 Reaction time of the test system in Case 1(4)

Case 2 $y_q\neq\varepsilon$, $U(y, y', x')\cup U(y', y, x')\neq\varnothing$

In this case, there are two types of reaction time (Δ' and Δ'') as illustrated in Fig.7. In the same way as Case 1, the constraints for Δ' can be dynamically determined. We list the constraints in Table 1. The constraint for Δ'' can be statically determined as $\Delta''\geq TT_{ts}^{max}$.

Case 3 $y_q=\varepsilon$

In this case, there is no dynamic time information that we can obtain during the test execution. We get the same timing constraints as Khoumsi (2002), which are listed in Table 2.

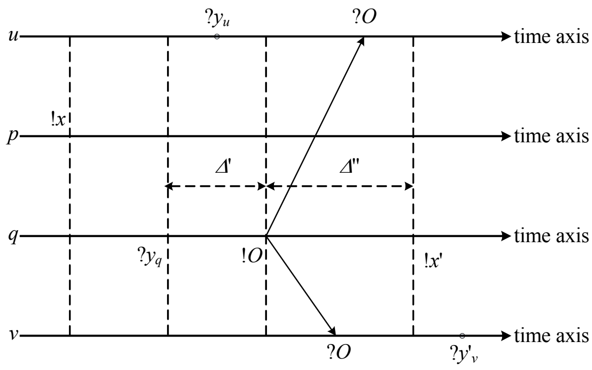

Fig.7 Reaction time of the test system in Case 2. $u\in U(y, y', x')$, $v\in U(y', y, x')$

Table 1 Constraints of reaction time in Case 2

Situation	Timing constraint	
$p=q$	$\Delta'\geq RT_{iut}-TT_{ts}^{min}-\Delta t(!x,?y_q)$	(6)
$p\neq q$, $'y_q=\varepsilon$	$\Delta'\geq RT_{iut}-2TT_{ts}^{min}-\Delta t(?O,?y_q)+\Delta t(!O,!x)$	(7)
$p\neq q$, $'y_q\neq\varepsilon$, $'y_p\neq\varepsilon$	$\Delta'\geq\min(RT_{iut}-TT_{ts}^{min}, 2RT_{iut}-\Delta t(?'y_q,?y_q)+\Delta t^s(?'y_p,!x)-TT_{ts}^{min})$	(8)
$p\neq q$, $'y_q\neq\varepsilon$, $'y_p=\varepsilon$	$\Delta'\geq\min(RT_{iut}-TT_{ts}^{min}, RT_{iut}+TT_{ts}^{max}-TT_{ts}^{min}-\Delta t(?'y_q,?y_q)+\Delta t^s(!'x,!C)+\Delta t^s(?C,!x))$	(9)

Table 2 Constraints of reaction time in Case 3. $W=U(y, y', x')\cup U(y', y, x')$

Situation	Timing constraint	
$p=q$, $W=\varnothing$	$\Delta''=\Delta t(!x,!x')\geq RT_{iut}$	(10)
$p\neq q$, $W=\varnothing$	$\begin{cases}\Delta=\Delta t(!x,!C)\\ \Delta''=\Delta t(?C,!x')\\ \Delta+\Delta''\geq RT_{iut}-TT_{ts}^{min}\end{cases}$	(11)
$p=q$, $W\neq\varnothing$	$\begin{cases}\Delta'=\Delta t(!x,!O)\geq RT_{iut}-TT_{ts}^{min}\\ \Delta''=\Delta t(!O,!x')\geq TT_{ts}^{max}\end{cases}$	(12)
$p\neq q$, $W\neq\varnothing$	$\begin{cases}\Delta=\Delta t(!x,!C)\\ \Delta'=\Delta t(?C,!O)\\ \Delta''=\Delta t(!O,!x')\geq TT_{ts}^{max}\\ \Delta+\Delta'\geq RT_{iut}-2TT_{ts}^{min}\end{cases}$	(13)

CASE STUDY

In this section, we will illustrate the application of our study to the testing of the 2p-FSM in Fig.2.

Consider the test sequence t_4, t_5, t_2. Expected behavior of each tester is shown in Fig.8.

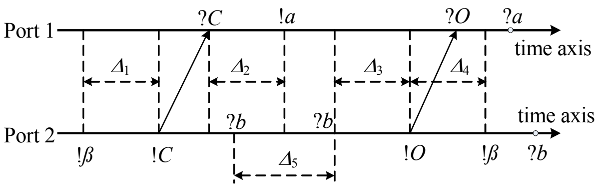

Fig.8 Expected behaviour of each tester

First, Δ_1 and Δ_2 should satisfy the inequality (11), i.e. $\Delta_1+\Delta_2 \geq RT_{\text{iut}}-TT_{\text{ts}}^{\min}$. One solution is $\Delta_1=\Delta_2=(RT_{\text{iut}}-TT_{\text{ts}}^{\min})/2$.

Second, Δ_3 and Δ_4 should satisfy the constraint (9), i.e. $\Delta_4 \geq TT_{\text{ts}}^{\max}$ and $\Delta_3 \geq \min(RT_{\text{iut}}-TT_{\text{ts}}^{\min},\ RT_{\text{iut}}+TT_{\text{ts}}^{\max}-TT_{\text{ts}}^{\min}-\Delta_5+\Delta_1^s+\Delta_2^s)$.

In this case, $\Delta_1^s=\Delta_1, \Delta_2^s=\Delta_2$. If $TT_{\text{ts}}^{\max}=TT_{\text{ts}}^{\min}$, we obtain $\Delta_3 \geq \min(RT_{\text{iut}}-TT_{\text{ts}}^{\min},\ RT_{\text{iut}}-\Delta_5+RT_{\text{iut}}-TT_{\text{ts}}^{\min})$. Furthermore, if the reaction time of the IUT is uniformly distributed in the interval [0, RT_{iut}], the probability of $RT_{\text{iut}} \leq \Delta_5$ will equal 1/2.

In Appendix A, we describe the behaviour of each tester in the standardized test specification language, i.e. TTCN-3 (ETSI, 2001).

CONCLUSION

The increasing significance of distributed systems has led to much interest in issues relating to the test of such systems. In distributed testing, the existence of multiple testers complicates testing because remote testers may encounter controllability and observability problems during the application of a test sequence. This paper investigates these problems from the aspect of test execution and points out that controllability and observability problems are indeed resolved if and only if the test system respects some timing constraints. When determining these constraints, the dynamic time information during test is taken into consideration, which can further reduce the test execution time and improve test efficiency. Moreover, the overhead for that is trivial. Only little time information needs to be calculated during the test. The future work is to apply our study to the testing of practical communication protocols such as SIP (Session Initiation Protocol).

References

ETSI, 2001. The Testing and Test Control Notation, Version 3, Part 1: TTCN-3 Core Language.

Hierons, R.M., 2001. Testing a distributed system: generating minimal synchronised test sequences that detect output-shifting faults. *Inf. Software Technol.*, **43**(9):551-560. [doi:10.1016/S0950-5849(01)00169-0]

Khoumsi, A., 2002. A temporal approach for testing distributed systems. *IEEE Trans. Software Eng.*, **28**(11):1085-1103. [doi:10.1109/TSE.2002.1049406]

Liu, H., Wu, J.P., Yin, X., 2003. Generating External Synchronizable Test Sequences that Detect Output-Shifting Faults. Proc. 7th International Conference on Telecommunications, p.565-571.

Luo, G., Dssouli, R., Bochmann, G.V., Venkataram, P., Ghedamsi, A., 1994. Test generation with respect to distributed interfaces. *Computer Standards and Interfaces*, **16**(2):119-132. [doi:10.1016/0920-5489(94)90003-5]

Rafiq, O., Cacciari, L., 2003. Coordination algorithm for distributed testing. *J. Supercomput.*, **24**(2):203-211. [doi:10.1023/A:1021759127956]

Ural, H., Whittier, D., 2003. Distributed testing without encountering controllability and observability problems. *Inf. Processing Lett.*, **88**(3):133-141. [doi:10.1016/S0020-0190(03)00364-8]

APPENDIX A

Behavior of *Tester1*

channel.receive(*C*); // delay Δ_2
timer *timer1*;
timer1.start(Δ_2);
timer1.timeout;

port1.send(α);
channel.receive(*O*);
port1.receive(*a*);
setverdict(*pass*);

Behavior of *Tester2*

port2.send(β);

float *begin*=0.0; // for calculating Δ_5
float *end*=0.0; // for calculating Δ_5

```
timer timer3=Δ1;     // for delaying
timer3.start;
interleave {
        [] port2.receive(b) {    // receive the 1st b
              begin=externalGetTime();
        }
        [] timer3.timeout {     // delay Δ1
              channel.send(C);
        }
}
port2.receive(b) {    // receive the 2nd b
end=externalGetTime();

Δ5=end−begin;
//  find proper Δ3 based on inequality (9)
Δ3=calculateDelta3(Δ5)
//  delay Δ3
timer timer6=Δ3;
timer6.start;
timer6.timeout;

channel.send(O);

//   delay Δ4
timer timer7=Δ4;
timer7.start;
timer7.timeout;

port2.send(β);
port2.receive(b);    // receive the 3rd b

setverdict(pass);
```