A Passive Performance Testing approach for network protocols

Xiaoping CHE
CNRS UMR 5157 Samovar
Institut Mines-Telecom/Telecom SudParis
Evry, France
Email: Xiaoping.Che@telecom-sudparis.eu

Stephane MAAG
CNRS UMR 5157 Samovar
Institut Mines-Telecom/Telecom SudParis
Evry, France
Email: Stephane.Maag@telecom-sudparis.eu

Abstract—Complementary to performance evaluation, the performance testing of communicating protocols is a qualitative and quantitative test of a system, aiming at verifying whether the performance requirements of the protocol have been satisfied under certain conditions. Conformance testing of communicating protocols is a functional test which verifies whether the behaviours of the protocol satisfy defined requirements. It raises the interesting issue of how to accurately formalize the performance requirements and how to converge these two kinds of tests by using the same formal approach. In this paper, we introduce a novel logic-based approach to test the protocol performance using the same formal approach. In this paper, we introduce requirements and how to converge these two kinds of tests by using the same formal approach. In this paper, we introduce a novel logic-based approach to test the protocol performance through real execution traces and formally specified properties. And also a distributed testing framework is designed for network protocols.

Keywords—Session Initiation Protocol; Formal Methods; Passive Testing; Performance Testing

I. INTRODUCTION

In the recent years, many studies on checking the behaviour of an Implementation Under Test (IUT) have been performed. Important works are about the record of the observation during run-time and its comparison with the expected behaviour defined by either a formal model [1] or a set of formally specified properties [2] obtained from the requirements of the protocol. These approaches are commonly identified as Passive Testing approaches (or monitoring). With these techniques, the protocol messages observed in execution traces are generally modelled and analysed through their control parts [3]. In [4] and [5], a data-centric approach is proposed to test the conformance of a protocol by taking account the control parts of the messages as well as the data values carried by the message parameters contained in an extracted execution trace.

However, within the protocol testing process, conformance and performance testing are often associated. They are mainly applied to validate or verify the scalability and reliability of the system. Many benefits can be brought to the testing process if both inherit from the same approach. Our main objective is then to propose a passive distributed performance testing approach based on our formal conformance testing technique [5]. Although some crucial works have been done in conformance testing area [6], they study run-time verification of properties expressed either in linear-time temporal logic (LTL) or timed linear-time temporal logic (TTLT). Different from their work focusing on testing functional properties based on formal models, our work concentrates on formally testing non-functional properties without formal models. Also note that, our work is absorbed in the performance testing, not in performance evaluation. While performance evaluation of network protocols focuses on the evaluation of its performance, performance testing approaches aim at testing performance requirements that are expected in the protocol standard.

II. FORMAL METHODS

In our previous works [5] [4], data domains are defined either as atomic or compound. We defined a syntax based on Horn clauses to express properties that are checked on extracted traces, which will be briefly described in the following. Formulas in this logic can be defined with the introduction of terms and atoms. A substitution is a finite set of bindings \( \theta = \{ x_1/term_1, ..., x_k/term_k \} \). The relations between terms and atoms are stated by the definition of clauses. A clause is an expression of the form \( A_0 \leftarrow A_1 \land ... \land A_n \). A formula is defined by the following BNF:

\[
\phi ::= A_1 \land ... \land A_n \mid \phi \rightarrow \phi \mid \forall \phi \mid \forall \phi \mid \phi \land \psi \mid \exists \phi \mid \exists \phi \mid \exists \phi
\]

In our approach, the quantifiers are commonly defined as “it exists” (\( \exists \)) and “for all” (\( \forall \)). We can formalize the performance requirements to formulas by using the definitions described, and the truth values \( \{ T, \bot, ? \} \) will be given to evaluate them. The set of performance requirements and relative conformance requirements are defined by using ‘\( R_p \)’ and ‘\( R_c \)’ respectively, where \( R_c = \{ \psi_1, \psi_2, ..., \psi_n \} \), \( \psi = \{ A_1 \land A_2 \land ... \land A_m \} \in \mathbb{N} \) and \( R_p = \{ \phi_1, \phi_2, ..., \phi_m \} \), \( \phi = \{ \psi \land (A_1 \land A_2 \land A_m) \} \in \mathbb{N} \) where \( A_i = p(term_i, ..., term_m), term_i \in T \). By noting that most of performance requirements are based on relative conformance requirements, we defined a new operator ‘\( \ast \)’ for clearly differentiating the non-positive results. And then the final verdict can be obtained from the formula \( (\phi) \ast (\psi) \).

III. FRAMEWORK OF TESTING

A passive distributed testing architecture is designed for testing conformance and performance of the protocol. Initially, as the Figure 1 shows, the global monitor sends initial bindings to the sub testers. When the testers receive the information, they will initialize capturing packets and save the traces to readable files during each time slot. Once the readable files are generated, the testers will test the traces through the
formalized requirements formulas and send the results back to the global monitor for analysing. The testing procedures will keep running until the global monitor returns a Stop command to sub testers.

IV. EXPERIMENTS

With the syntax and semantics defined in the previous sections, we can formalize some complex performance properties, like in protocol SIP “Every successful INVITE request should be responded with a success response within 0.5s” which can be represented as $\forall (request(x) \land x.method = \text{'INVITE'}) \rightarrow \exists_{y>x}(success(y) \land responds(y,x) \land \text{withintime}(x,y,T))$. The testing results are shown in the following Table I. Besides, as the bar chart shows in Figure 2, for the performance requirement “For each successfully registration, the REGISTER request should be responded with a 200 response in 16s”, not only it can be formalized to $\forall (request(x) \land x.method = \text{'REGISTER'}) \rightarrow \exists_{y>x}(\text{nonProvisional}(y) \land responds(y,x))$ but also the number of ‘Pass’ verdicts in different time intervals can be obtained.

Moreover, through the designed testing framework, the performance properties can be tested distributively. After evaluating each formula $\phi$ on a trace $\rho$, $N_p, N_f$ and $N_{in}$ will be given to the trace as results representing the number of ‘Pass’, ‘Fail’ and ‘Inconclusive’ verdicts respectively. As Figure 3 depicts, our approach can precisely reflect the testing results of property “Registration Rates” ($N_p(\phi) / t_{test}$) from each tester in the distributed environment.

<table>
<thead>
<tr>
<th>Trace</th>
<th>No of Message</th>
<th>Pass</th>
<th>Fail</th>
<th>Inconclusive</th>
<th>Pass</th>
<th>Fail</th>
<th>Inconclusive</th>
</tr>
</thead>
<tbody>
<tr>
<td>1</td>
<td>500</td>
<td>60</td>
<td>10</td>
<td></td>
<td>60</td>
<td>10</td>
<td></td>
</tr>
<tr>
<td>2</td>
<td>1000</td>
<td>109</td>
<td>0</td>
<td>18</td>
<td>109</td>
<td>0</td>
<td>18</td>
</tr>
<tr>
<td>3</td>
<td>1500</td>
<td>139</td>
<td>0</td>
<td>37</td>
<td>139</td>
<td>0</td>
<td>37</td>
</tr>
<tr>
<td>4</td>
<td>2000</td>
<td>181</td>
<td>0</td>
<td>53</td>
<td>181</td>
<td>2</td>
<td>51</td>
</tr>
<tr>
<td>5</td>
<td>2500</td>
<td>267</td>
<td>0</td>
<td>51</td>
<td>267</td>
<td>2</td>
<td>49</td>
</tr>
</tbody>
</table>

TABLE I

<table>
<thead>
<tr>
<th>$\phi$</th>
<th>$\phi \lor \psi$</th>
<th>$\phi \land \psi$</th>
</tr>
</thead>
<tbody>
<tr>
<td>$\psi$</td>
<td>$\psi \lor \phi$</td>
<td>$\psi \land \phi$</td>
</tr>
</tbody>
</table>

V. CONCLUSION

This paper introduces a formal approach to passively test distributed conformance and performance of network protocol implementation. Our approach allows to define the conformance and performance properties that are evaluated on real protocol traces. The evaluation of the property returns a Pass, Fail or Inconclusive result, derived from the given trace. Our methodology has been implemented into a distributed framework which provides the possibility to test individual nodes of a complex network environment. The results from testing several properties on large traces have been obtained with success.

REFERENCES