A Formal Approach to Protocol Performance Testing

XU Mingwei (徐明伟) and WU Jianping (吴建平)

Department of Computer Science, Tsinghua University, Beijing 100084, P.R. China
E-mail: xmw@csnet1.cs.tsinghua.edu.cn; jianping@cernet.edu.cn
Received August 25, 1997; revised December 11, 1997.

Abstract This paper proposes a formal approach to protocol performance testing based on the extended concurrent TTCN. To meet the needs of protocol performance testing, concurrent TTCN is extended, and the extended concurrent TTCN's operational semantics is defined in terms of Input-Output Labeled Transition System. An architecture design of protocol performance test system is described, and an example of test cases and its test result are given.

Keywords protocol performance testing, extended concurrent TTCN, operational semantics, IOLTS

1 Introduction

Protocol testing is an important means to ensure the interconnection and interoperation between protocol products from different vendors. Current test activities for protocols can be classified into three classes according to their test purpose: conformance testing, interoperability testing and performance testing. Conformance testing and interoperability testing are functional test, and performance testing, however, is different from them. Its purpose is to test the characteristic parameters of protocol implementation, such as packet transfer delay and throughput, so as to evaluate the efficiency of protocol implementation.

Protocol performance testing is a complex test procedure, needing several test components to coordinate and run test cases in parallel, which can be specified by concurrent TTCN. Concurrent TTCN allows more than one active test component to participate in the execution of a test case. All test components run in parallel and coordinate their behavior by exchanging coordination messages. The advantage of concurrent TTCN is that the description of test cases for complex test environment becomes easier.

However, concurrent TTCN cannot satisfy all needs of protocol performance testing. In protocol performance testing, it is necessary to obtain the accurate time when a test event just starts or stops. So the sequential operation of a test event and its corresponding operation, reading time, cannot be interrupted by other processes. Moreover, the traffic operation and timer operation should be extended in concurrent TTCN.

The aim of this paper is to discuss a formal approach to protocol performance testing based on the extended concurrent TTCN. To meet the needs of protocol performance testing, we extend concurrent TTCN, and formally define operational semantics for the extended concurrent TTCN in terms of Input-Output Labeled Transition System (IOLTS). Moreover, we describe a protocol performance test system based on the extended concurrent TTCN's operational semantics, and give a test case written in the extended concurrent TTCN.

The remainder of the paper proceeds as follows. Section 2 summarizes the major features of concurrent TTCN, and extends it to meet the needs of protocol performance testing. Section 3 formally defines the operational semantics of the extended concurrent TTCN. In Section 4, we describe the design of a protocol performance test system based on the...
extended concurrent TTCN and give an example of test cases and its test result. Finally, we give some conclusions in Section 5.

2 Extended Concurrent TTCN

This section gives a short introduction to concurrent TTCN, and then extends concurrent TTCN to meet the needs of protocol performance testing. Also, a conceptual model for the extended concurrent TTCN is elaborated, whose semantic representation is discussed in Section 3.

2.1 Concurrent TTCN

The Tree and Tabular Combined Notation (TTCN)\[3\] is recommended by ISO to describe abstract test suite. Concurrent TTCN is an extension of TTCN. The concern of TTCN is a single test component executing a test case. Concurrent TTCN, however, allows to execute a test case by several test components (TC) running in parallel. A conceptual model of test components is depicted in Fig. 1. A tester consists of exactly one main test component (MTC) and any number of parallel test components (PTCs). TCs are linked by coordination points (CPs) capable of conveying coordination messages (CMs). Communication of TCs with the environment, such as the (N - 1) service provider or the implementation under test (IUT), takes place at points of control and observation (PCOs).

Execution of a test case starts with the execution of MTC. It is the concern of MTC to set up all PTCs, to manage all PCOs and CPs to be connected to, and to compute the final verdict. PTCs can be created by MTC on demand. A ‘create’ operation associates a PTC with a behavior tree. The newly created PTC starts execution of its assigned behavior tree concurrently with MTC. MTC may explicitly terminate a PTC by executing a ‘terminate’ operation.

2.2 Extensions to Concurrent TTCN

Concurrent TTCN allows to execute a test case by several test components running in parallel. This is important to protocol performance testing. There are, however, still some requirements in protocol performance testing, e.g. atomic operation, traffic operation and some timer operation, that cannot be provided by concurrent TTCN. According to the needs of protocol performance testing, we extend concurrent TTCN as follows.

2.2.1 Atomic Operation

In protocol performance testing, some test cases about time parameters, such as delay, are indispensable. We can use TTCN to specify the test case, for example,

\[\begin{align*}
!a & \text{readtimer}(t1) \\
?b & \text{readtimer}(t2) \\
& \text{(Delay}:= t2 - t1) 
\end{align*}\]