Protocol Conformance Testing

Anandi Giridharan
Department of Electrical Communication Engineering
Indian Institute of Science
Bangalore – 560012, India
SDL based tools for conformance testing

- **TESDL**
  - prototype tool for the automatic generation of test cases from SDL specifications in the context of the OSI CTMF.
  - TESDL implements a heuristic algorithm to derive behavior of a protocol as a tree called Asynchronous Communication Tree (ACT) which is based on restricted set of SDL diagram.
  - In ACT nodes represents global states. Global states contain information about states of all process in specification.
  - Tests are derived from ACT of specification by software tool called TESDL. Input is SDL specification and output is test case in TTCN notation.
- TTCN Link

  • environment for efficient development of TTCN test suites based on SDL specifications in SDT3.0 (SDL Description Tool). It assures consistency between SDL specification and TTCN suite. This tool is semi-automatic. The intended user of test suit TTCN test suit developer. His input are SDL specification and TTCN test suit structure with test purpose and his job is to develop abstract test suit based on Input.

- SAMSTAG

  • This was developed in research and development project Conformance testing for generation of test cases. Behavior of protocol was defied by SDL specification and purpose of test case is given by MSC. It includes algorithm for test case generation.

- TOPIC V2

  • works by cosimulating the SDL specification and an observer representing the test purpose. Observer is described in a language called GOAL (Geode Observation Automata Language)
- **Tveda V3**
  
  - tool for automatic test case generation which incorporates several features such as:
    - Modular architecture: That makes it possible to choose between the specification language, test description language, test selection strategy.
    - Semantic module: that is called from strategy modules to compute feasible paths.
    - Functional execution such as hyper text links between tests and specification etc.

- **TAG Tool**

  - generates test cases from SDL systems. EFSM corresponding to processes of SDL system are extracted. Files containing SDL declaration of interaction and channels are created. It generates automatic executable test cases for ECFSMS based systems.
SDL based conformance testing of MPLS

- Goal of test sequence generation is to minimize the number of tests with a guaranteed coverage. This test constructs 15 tests that cover all postions of protocol. This test generation method is based on local conformance test architecture. CT has 2 algorithms.

- Test sequence generation approach

- Alg. 1: Find a path that covers a maximum number of transitions:
  - Consider a initial state which makes transition to n states.
  - These n states may in turn make transitions to other states and so on.
  - Find out the maximum number of outgoing transitions from all states and retain those transitions that are of maximum length from each state discarding other transitions.
  - Finally initial state will choose the state path having maximum number of transitions.

- Use the generated test sequence and apply Alg. 2:
  - Give this test sequence to MPLS Simulation, see the behavior of the simulation.
  - Find again a path that covers a second maximum number of transitions by the method given above.
SDL based MPLS testing

- In the initial phase MPLS Extended Communication Finite State Machine (ECFSM) was transformed into an equivalent Formal SDL model.

- To view the typical messages exchanged between co-MPLS peers, the SDL model was simulated to view the TRANSPORT and MPLS messages exchanged between the two LDPs.

- During this exchange of messages that is witnessed in the Message Sequence Chart each of these MPLS nodes traverses through all States in the FSM thereby making associated transitions.

- Simulate a 3-node topology (ingress router, LSR1, egress router) and look into the stability issue of MPLS.

- After Initializing the system, each MPLS router gets to know about the current topology from their neighbors. Now that the topology is stabilized.

- Apply the message to ingress node, you will get the same message at egress node.

- Now MPLS 3-node topology is tested for its functionality.

- Find test sequences
A snapshot of SDL simulation

Output of sequence no. 1
A snapshot of SDL simulation

Output of sequence no. 2
LDP Message Exchange and Structure

2.1 When an LSR chooses to establish a session with another LSR learned via the Hello message, it uses the LDP initialization procedure over TCP transport.

2.2 Upon successful completion of the initialization procedure, the two LSRs are LDP peers, and may exchange advertisement messages.

2.3 The LSR requests a label mapping from a neighboring LSR when it needs one.

2.4 The LSR advertises a label mapping to a neighboring LSR when it wishes the neighbor to use a label.

2.5 LDP uses the TCP transport for session, advertisement and notification messages, i.e., for everything but the LDP-based label discovery mechanism.

3.3 We say that a particular address (match) is a particular address prefix if and only if that address begins with that prefix.

4.3 The Value part of a TLV-encoded object, or TLV for short, may itself contain one or more TLVs, (DUT Receiving TLV)

4.10 The Value part of a TLV-encoded object, or TLV for short, may itself contain one or more TLVs, (DUT Sending TLV)
<table>
<thead>
<tr>
<th>Case name</th>
<th>Case status</th>
<th>Case start time</th>
<th>Elapsed time (ms)</th>
<th>Case comment</th>
</tr>
</thead>
<tbody>
<tr>
<td>LDP-1.1</td>
<td>PASS</td>
<td>2003-12-11 14:57:47</td>
<td>17357</td>
<td></td>
</tr>
<tr>
<td>LDP-1.2</td>
<td>PASS</td>
<td>2003-12-11 14:58:05</td>
<td>16395</td>
<td></td>
</tr>
<tr>
<td>LDP-1.3</td>
<td>PASS</td>
<td>2003-12-11 14:58:22</td>
<td>19322</td>
<td></td>
</tr>
<tr>
<td>LDP-1.4</td>
<td>PASS</td>
<td>2003-12-11 14:58:41</td>
<td>24004</td>
<td></td>
</tr>
<tr>
<td>LDP-1.5</td>
<td>PASS</td>
<td>2003-12-11 14:59:05</td>
<td>31111</td>
<td></td>
</tr>
<tr>
<td>LDP-1.6</td>
<td>PASS</td>
<td>2003-12-11 14:59:26</td>
<td>33640</td>
<td></td>
</tr>
<tr>
<td>LDP-1.7</td>
<td>PASS</td>
<td>2003-12-11 15:00:10</td>
<td>33640</td>
<td></td>
</tr>
<tr>
<td>LDP-1.8</td>
<td>FAIL</td>
<td>2003-12-11 15:00:08</td>
<td>10398</td>
<td>No Label Request message received from 10.1.3.11</td>
</tr>
<tr>
<td>LDP-1.9</td>
<td>PASS</td>
<td>2003-12-11 15:01:10</td>
<td>23648</td>
<td></td>
</tr>
<tr>
<td>LDP-1.10</td>
<td>PASS</td>
<td>2003-12-11 15:01:34</td>
<td>42139</td>
<td></td>
</tr>
<tr>
<td>LDP-1.11</td>
<td>PASS</td>
<td>2003-12-11 15:02:16</td>
<td>45993</td>
<td></td>
</tr>
<tr>
<td>LDP-1.12</td>
<td>PASS</td>
<td>2003-12-11 15:03:02</td>
<td>29299</td>
<td></td>
</tr>
<tr>
<td>LDP-1.13</td>
<td>PASS</td>
<td>2003-12-11 15:03:30</td>
<td>29304</td>
<td></td>
</tr>
<tr>
<td>LDP-1.14</td>
<td>PASS</td>
<td>2003-12-11 15:03:58</td>
<td>31237</td>
<td></td>
</tr>
<tr>
<td>LDP-1.15</td>
<td>PASS</td>
<td>2003-12-11 15:04:22</td>
<td>40109</td>
<td>Did not receive Label Mapping Message from 10.1.3.11 for FEC 10.1</td>
</tr>
<tr>
<td>LDP-1.16</td>
<td>PASS</td>
<td>2003-12-11 15:05:02</td>
<td>38841</td>
<td></td>
</tr>
</tbody>
</table>