

On conformance test and fault resolution of protocols based on FSM model

T. Ramalingam[†] Anindya Das[‡] K. Thulasiraman[†]

[†]Department of Electrical & Computer Engineering, Concordia University, 1455 De Maisonneuve West, Montreal, Canada H3G 1M8.

[‡]Département d'informatique et de recherche opérationnelle, Université de Montréal, Montreal, Canada.

Abstract

In this paper we study the problem of designing a Conformance Resolution Test (CRT) for computer communication protocols based on the Finite State Machine model. We have examined the existing conformance testing methods with reference to their fault resolution capabilities. Among these, the W_p -method [FBK91] is found to have the best fault resolution capability when the implementation has almost one fault. We present a CRT method with a fault resolution capability better than the W_p -method when the implementation has almost one fault and the specification meets certain W_p conditions. This method is based on the W_p -method. We report results of applying this method on a subset of a transport protocol.

Keyword Codes: C.2.2; D.2.1; D.2.5

Keywords: Computer-Communication Networks, Network protocols;
Software Engineering, Requirements/Specifications; Testing and Debugging

1. INTRODUCTION

Conformance Testing(CT) of a protocol is to check if an implementation conforms to its specification[R87]. During conformance testing the implementation is viewed as a black box with input and output ports. Conformance testing is done by stimulating the implementation with certain inputs and observing its behavior through the output ports. The resulting output sequence is compared with the expected output sequence with respect to the specification. If they match then the implementation is said to pass the conformance test; otherwise it is said to fail. The sequence of input and the expected output pairs used for testing the implementation is known as a *test sequence*.

Test sequences are generated using the specification. Methods are available in the literature to transform a protocol specified in formal specification languages LOTOS, ESTELLE, or SDL into an extended finite state machine. Both control flow and data flow aspects of a protocol have to be tested in order to certify an implementation. In this paper we will consider only the control flow aspects of testing. The control flow part of any protocol can be represented by a finite state machine(FSM). Different methods are available for

generating test sequences from an FSM representation of a protocol [DSU90-1].

Four types of testing are distinguished according to the extent to which they provide an indication of conformance: *basic interconnection tests*, *capability tests*, *behavior tests*, and *Conformance Resolution Tests (CRT)* [R87]. The CRT is to provide diagnostic answers, as near to definitive as possible, to the resolution of whether an implementation satisfies specified requirements. So far, there are no formal methods known for generating conformance resolution tests. This paper aims at developing such a test. We have analyzed the existing FSM based conformance testing methods with reference to their fault resolution capabilities. We have observed that as these methods improved to provide better fault coverage their fault resolution capabilities also improved. Among these methods, the W_p -method of Fujiwara et al [FBK91] is found to have the best fault resolution capability for implementations with atmost one fault. In this paper we present a CRT method with a fault resolution capability better than the W_p -method when the implementation has atmost one fault and when certain conditions are satisfied by the specification. This CRT method is based on the W_p -method. We have omitted the proofs of lemmas and theorems in this paper. However the proofs can be seen in [RDT92].

2. PRELIMINARIES

As discussed earlier we model the control flow aspects of a protocol (henceforth referred to as protocol for simplicity) specification as well as its implementation by an FSM.

An FSM M can be formally defined as a 5-tuple $M=(S, s_0, I, O, T)$ where S is the nonempty set of states of M in which s_0 is a designated state called the *initial state*. I and O are nonempty sets of possible inputs and outputs of the protocol, respectively. The transition function T is a partial function defined as $T: S \times I \rightarrow S \times O$. $T(s_i, a_k)=(s_j, o_l)$ means that the FSM M at state s_i makes a transition to state s_j when the input a_k is applied producing the output o_l . Graphically this is also represented as $s_i -a_k/o_l \rightarrow s_j$. We call an FSM M *fully specified* if at each state s_i in M and for each input a_k in I , there is an outgoing transition from s_i with input a_k .

An FSM $M=(S, s_0, I, O, T)$ can also be represented by a directed labelled graph $G=(V, E)$, where $S=V$ and each transition $s_i -a_k/o_l \rightarrow s_j$ corresponds to an edge in E directed from s_i to s_j with label a_k/o_l . Thus an edge in E is specified by a triple $(s_i, s_j; a_k/o_l)$.

An FSM is said to have *reset capability* if for each state s_i in S there exists a transition $(s_i, s_0; r/-)$, called a *reset transition* which resets the FSM to its initial state where 'r' denotes the 'reset' command and '-' denotes the 'null' output. An FSM can be modified into a fully specified one by using what is called a *completeness assumption* [DS88]. The completeness assumption requires that a self loop transition with input a_k and output '-' be added for each state s_i and for each input a_k if the state s_i does not have an outgoing transition with input a_k .

A sequence of input-output pairs is said to be *applicable* at a state of an FSM if the output part of the sequence is observed on applying the input part of the sequence to the FSM at that state. We use the operators '•' and '@' for concatenating input-output / input symbols, and input-output / input sequences respectively.

In this paper we assume that the FSM of the specification of a protocol (henceforth SPEC for simplicity) and the FSM of the corresponding implementation (henceforth referred to as IUT) have the same number of states. We consider two types of faults, namely, label fault and tailstate fault. A transition $(s_i, s_j; a_k/o_l)$ of the SPEC is said to have a *label fault* if the corresponding transition in the IUT is $(s_i, s_j; a_k/o_p)$ where $o_p \neq o_l$. A transition $(s_i, s_j; a_k/o_l)$ of the SPEC is said to have a *tailstate fault* if the corresponding transition in

the IUT is $(s_i, s_j; a_k/o_l)$ where $p \neq j$. The *fault coverage* of a conformance test method is the percentage of faulty IUTs the method can detect from the set of all IUTs with label faults and/or tailstate faults.

A CRT method has *t-fault resolution capability of level k* if for any IUT with atmost t faulty transitions, the method can localize each of the t faults to within k transitions. A CRT method is said to have *t-fault location capability* if it has t-fault resolution capability of level 1. In this paper we consider CRT methods with 1-fault resolution capability.

3. FAULT RESOLUTION CAPABILITY OF CONFORMANCE TESTING METHODS

In this section we analyze the existing CT methods. For more detailed discussions of these methods and illustrations of our claims [NT81] [ADL88] [CVI89] [FBK91] and [RDT92] may be consulted. Our focus is on the fault resolution capability of these methods. We have examined the extent to which a fault can be localized by these methods. In this paper the terms 'fault resolution' and 'fault diagnosis' are used interchangeably. Test subsequence corresponding to transition $(s_i, s_j; a_k/o_l)$ is denoted by $TEST(s_i, s_j; a_k/o_l)$. $T(s_i, s_j)$ denotes the sequence along a path from state s_i to state s_j .

3.1. Transition tour method

The transition tour method (T-method, in short) described in [NT81] assumes that the SPEC is fully specified. The test sequence is generated based on a minimal transition tour which traverses each transition in the SPEC atleast once. Here the test subsequence corresponding to a transition is simply its label. It should be noted that this method does not verify the intermediate states in the IUT as it traverses the transitions. Hence the method does not have the capability of detecting tailstate faults, as noted in [SL89], [DSU90-1]. For the same reason this method can not diagnose faults in the IUT with tailstate faults even if it certifies the IUT as faulty. Thus we conclude that the T-method only has the 1-fault resolution capability of level $|E|$, where $|E|$ is the number of transitions in the SPEC.

3.2. Unique input output sequence method

The Unique Input Output (UIO) sequence method (in short, the U-method) introduced in [SD85] requires that the SPEC be strongly connected. This method also assumes for each state the existence of an input-output sequence which uniquely identifies that state. Such sequences are called *UIO-sequences*. Formally, a UIO-sequence for state s_i of an FSM M denoted by UIO_i is an input-output sequence of minimum length such that UIO_i is applicable at state s_i and it is not applicable at any other state in M . The U-method tests the transitions as follows:

To test a transition, say $(s_i, s_j; a_k/o_l)$, the IUT is first put in state s_i . Then the input a_k is applied and the output is checked to verify that it is o_l as expected. Finally the input part of UIO_i is applied to the current state of the IUT and the resulting output sequence is examined to check whether the current state of the IUT is in fact s_j as expected.

In order to minimize the length of the test sequence, the U-method uses a technique for solving the Rural Postman Problem (RPP) [K62]. A polynomial algorithm for solving the RPP for an augmented graph of a SPEC with either the reset capability or with a self loop at each state is presented in [ADL88]. The algorithm is known as *Rural Chinese Postman (RCP)-algorithm*.

Although the fault coverage of this method is better than the T-method, its fault resolution capability is affected due to the following reasons.

- (1) While applying the UIO-sequences, or transferring the IUT from one state to another state through a set of transitions, it might traverse transitions which have not yet been tested.
- (2) An UIO-sequence of a state in the SPEC need not be an UIO-sequence of the corresponding state in the IUT. This fact is also reported in [CVI89].

A bound on the 1-fault resolution capability level of this method will be presented after discussing the Improved UIO sequence method.

3.3. Improved UIO Sequence Method

As its name suggests, the improved UIO sequence method (in short, the U_v -method) is an improvement over the U-method. The improvement was suggested by Chan et al in [CVI89]. In addition to the assumption made in the U-method, this method assumes that both the SPEC and the IUT are fully specified. It also assumes that the IUT is strongly connected. The method consists of two phases. In the first phase, the method checks whether the selected UIO-sequences of the states of the SPEC are also UIO-sequences of the corresponding states in the IUT. In the second phase, all transitions are tested as in the U-method. Though Chan et al suggested the need for verifying the UIO-sequences in the IUT, they did not provide a method for achieving this requirement. In general, finding a method to meet this requirement seems to be difficult. We observe that though the fault coverage of the U_v -method is better than the U-method, the method does not guarantee complete fault coverage. As in the U-method, the U_v -method may use some transitions for putting the IUT in the start state of the transition under test. Such transitions constitute a *preamble* for the transition under test. This preamble may contain faulty transitions which are yet to be tested. Also some of the transitions which constitute the UIO-sequence of the tailstate of the transition under test may be faulty. Thus even when the method detects a faulty IUT, it may not be able to identify the faulty transitions. The following lemma shows the levels of 1-fault resolution capability of the U- and the U_v - methods under certain conditions.

Lemma 1:

Suppose the IUT has at most one fault and if the UIO-sequence of each state in the SPEC is also an UIO-sequence of the corresponding state in the IUT then both the U- and the U_v -methods have the 1-fault resolution capability of level $n + 2l_u$, where n is the number of states in the SPEC and l_u is the length of a longest UIO-sequence considered in these methods.

3.4. W_p -method

The W_p -method introduced by Fujiwara et al [FBK91] is based on the W-method [C78]. The W_p -method assumes the following about a SPEC and its IUT:

- (1) the SPEC is minimal;
- (2) both the SPEC and its IUT are strongly connected, and fully specified;
- (3) the SPEC has a reset capability which is correctly implemented in the IUT;
- (4) both the SPEC and its IUT have the same input set;
- (5) the number of states in the IUT is bounded by a number which may be larger than the number of states in the SPEC.

For state identification purposes, the W_p -method uses what is called a *characterization set*. A set W of input sequences is a characterization set if no two states in the SPEC have the same set of output sequences when W is applied to them. When W is singleton, the unique sequence of W is called a *distinguishing sequence* [K78]. As in the U_v -method, the W_p -method also has two phases. In the first phase, the characterization set W is verified in the IUT. In order to reach different states from the initial state (for verifying W), the method uses what is called a *state cover*. A state cover is a set Q of input sequences such that for each state s_i there is an input sequence q in Q taking the SPEC to s_i from the initial state. Starting from the initial state each transition in the state cover is also tested in an incremental fashion as they are used in reaching the state for verifying the characterization set. In the second phase the remaining transitions are tested using what is called a *transition cover*. A set P of input sequences is called a transition cover of the SPEC if for each transition $(s_i, s_j; a_k/o_l)$ there exist two input sequences p and $p@a_k$ in P such that $s_0=p \Rightarrow s_i$ where $s_0=p \Rightarrow s_j$ denotes that the SPEC goes from state s_0 to the state s_i when the input sequence p is applied. P is also chosen in such a way that it contains Q . In this phase, however, only subsets (*identification sets*) of W are used in verifying the tailstates of transitions. Formally, a subset W_i of W is called an identification set of the state s_i if the set of output sequences obtained by applying W_i at s_i is different from that obtained by applying W_i at any other state in the SPEC and no subset of W_i has this property. The W_p -method is described in the following procedure. In this procedure the symbol ' θ ' denotes the concatenation operator for concatenating two sets of input sequences. Formally, if A and B are two sets of input sequences then $A \theta B = \{x @ y \mid x \in A \text{ and } y \in B\}$. Note that in the W_p -method test-sequences are input sequences rather than input-output sequences.

procedure tseqwp()

Phase 1:

Let P be a transition cover of the SPEC.
 Let Q be a state cover of the SPEC such that $Q \subseteq P$.
 Let W be a characterization set of the SPEC.
 Apply $Q \theta W$ to the IUT.

Phase 2:

Let $R = P - Q$.
 Define $R \theta W = \{p \theta W_i \mid p \in R \text{ and } s_0=p \Rightarrow s_j \text{ and } W_i \text{ is the identification set of } s_j\}$.
 Apply $R \theta W$ to the IUT.

end tseqwp.

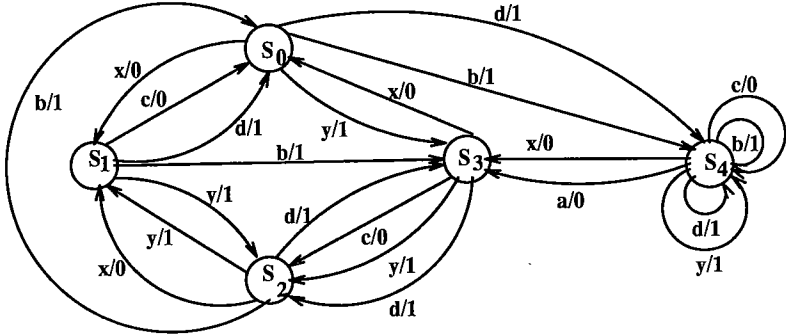
Unlike the T-, U-, and U_v -methods, the W_p -method guarantees complete fault coverage. It is claimed in [FBK91] that if an IUT passes phase 1 successfully, then

- (1) all transitions in the state cover are implemented correctly in the IUT,
- (2) for each state s_i , W_i is an identification set for s_i in the IUT.

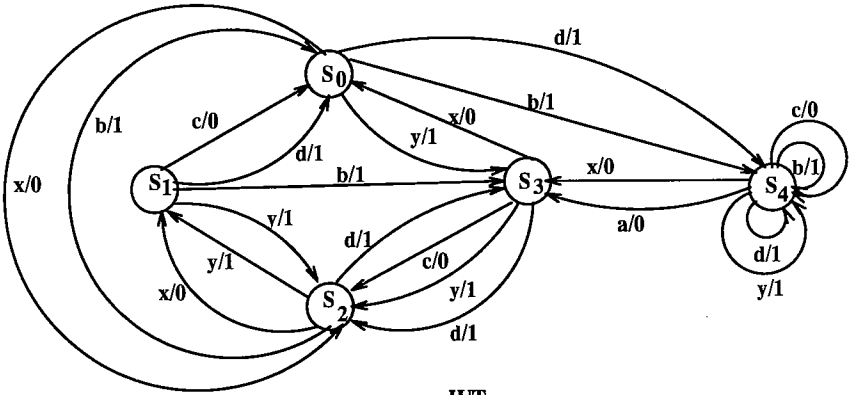
These claims are not always true as illustrated in the following example. (However, as we shall see in the following section, the correctness of these claims will help in achieving better fault resolution capability.) Consider the SPEC and an IUT of an abstract protocol shown in Figure 1. The reset transitions are not shown in the figure. The state cover, the characterization set and the identification set are as follows:

$$\begin{aligned}
 Q &= \{x, x \bullet y, y, b\} \\
 W &= \{y \bullet x \bullet b \bullet b, b \bullet x \bullet x \bullet c, d \bullet x \bullet x \bullet c, x \bullet b \bullet a, a\} \\
 W_0 &= \{y \bullet x \bullet b \bullet b\}
 \end{aligned}$$

- $W_1 = \{b \bullet x \bullet x \bullet c\}$
- $W_2 = \{d \bullet x \bullet x \bullet c\}$
- $W_3 = \{x \bullet b \bullet a\}$
- $W_4 = \{a\}$



SPEC



IUT

(reset transitions are not shown explicitly)

Figure 1. An example for the Wp-method

Note that with the above state cover Q and the identification set W the IUT passes phase 1 successfully. However, the transition $(s_0, s_1; x/0)$ of the SPEC which is a part of the state cover has a tailstate fault in the IUT. Also, in the IUT the sets W_1 and W_2 are not identification sets for s_1 and s_2 , respectively. The transition cover for the transition $(s_2, s_3; d/1)$ is $x \bullet y \bullet d$. Therefore, $TEST(s_2, s_3; d/1) = r \bullet x \bullet y \bullet d \bullet x \bullet b \bullet a$. Since $W_3 = x \bullet b \bullet a$ is not applicable at the tail state of the transition $(s_2, s_3; d/1)$ in the IUT, one could be led to conclude that the transition has the tail state fault. However, the actual fault corresponds to the transition $(s_0, s_1; x/0)$ of the SPEC. Note that the above results are valid even if the SPEC and the IUT are modified to be fully specified using the completeness assumption.

The level of 1-fault resolution capability of the W_p -method is presented in the following lemma.

Lemma 2:

Suppose the IUT has atmost one fault, then the W_p -method has 1-fault resolution capability of level $n + l_w$, where n is the number of states in the SPEC and l_w is the length of a longest sequence in W .

Such a large bound on the level of 1-fault resolution capability is due to the fact that the method may not always produce a correct state cover for the IUT at the end of the first phase. This is also the case in the above example.

Summarising, in all the CT methods discussed in this section, the test subsequence corresponding to a transition, say $(s_i, s_j; a_k/o_l)$ is a concatenation of the following subsequences:

- preamble: this is to put the IUT in the state s_i
- a_k/o_l : the input-output of the transition under test
- state identification: this is to verify the tailstate of the transition in the IUT
- postamble: this is to put the system back to the initial state s_0 .

For the purpose of optimization, in some of the CT methods some of these subsequences may be overlapped or omitted.

Suppose a CRT-method can test and confirm the correctness of all transitions in the preamble, in the tailstate identification sequence, and in the postamble of any transition, prior to the testing using the W_p -method. Then the method will have the fault location capability. As we have pointed out, none of the CT methods considered in this paper meet this requirement. In the next section, we present a CRT method which aims at meeting this requirement thereby achieving 1-fault resolution capability with a lower level .

4. CONFORMANCE RESOLUTION TEST METHOD

In this section we consider the problem of designing a CRT method with 1-fault resolution capability. Obviously, the method can be used for performing the behavior tests and it will guarantee complete fault coverage when the IUT has atmost one fault. Our method is based upon the W_p -method. Our method has better fault resolution capability than the W_p -method. Nevertheless, the test sequence generated using our method is shorter than the one generated using the W_p -method. All the basic assumptions made for the W_p -method need to be satisfied to apply our method. We assume that the SPEC and its IUT have the same number of states. However, as in [FBK91] our method can be extended for IUTs with more states than that of the SPEC. For identifying the states we use UIO-sequences. Our method can very easily be adapted if one wishes to use the characterization set, distinguishing sequence, or any other suitable set of sequences for identifying the states.

A set $U = \{ UIO_i \mid s_i \in SPEC \}$ is defined as an *UIO-set*. A directed spanning tree T rooted at the initial state of the SPEC is referred to as a *state cover tree*. Note that the set of input-output sequences corresponding to the paths in T from the initial state to every state in the SPEC forms a state cover. A transition is called a *T-transition* if it is on the state cover tree T . A transition is called a *U-transition* if it is part of an UIO-sequence from the UIO-set U . Like the W_p -method, our method also has two phases. In the first phase all the transitions in a state cover tree, say T , are required to be tested. Also, UIO-sequences in a UIO-set, say U , are required to be verified in the IUT. Meeting these requirements is difficult as T-transition testing requires verified UIO-sequences and UIO-sequence verification requires tested T-transitions. We achieve these requirements by systematically traversing the state cover tree and assuming the following *Tree UIO-sequence Label Disjoint (TULD) property* between T and U .

For each transition $(s_i, s_j; a_k/o_l)$ in T , the label a_k/o_l does not appear in the UIO-sequence UIO_j of U corresponding to the state s_j .

We have analyzed a number of protocols (reported later) and found that they have T and U with the TULD property. However, there may exist some protocols which do not have such T and U .

Phase 2 consists of two steps. In the first step all U -transitions which are not T-transitions are tested. We use the unique path in T for reaching the U -transitions. In the second step, we test all transitions which are neither T -transitions nor U -transitions. Here we apply the RCP-algorithm to obtain optimal test subsequences for these transitions. A detailed description of the method is given below. At any step of the following procedure, if the observed output is different from the one expected, the procedure terminates.

procedure U-CRT()

Phase 1:

- (1) Construct a state cover tree T and an UIO-set U of the SPEC such that T and U satisfy the TULD property.
/* This step will be elaborated upon later */
- (2) Test the state cover tree T in a breadth-first fashion as follows until all T -transitions are tested:
 - (i) Let $(s_i, s_j; a_k/o_l)$ in T be the current transition;
 - (ii) Use the unique path in T to reach s_i from s_0 .
 - (iii) Apply a_k and observe o_l ;
 - (iv) Apply UIO_j at the tailstate to check that UIO_j is applicable;
 - (v) Apply $U - \{UIO_j\}$ at the tailstate and check that they are not applicable. Use the state cover tree T and the reset transitions to reach the state s_j ;

Phase 2:

- (1) Repeat the following until all the U -transitions are covered:

Let $(s_i, s_j; a_k/o_l)$ be an untested U -transition;
 Apply r to observe $-$;
 Use the unique path in T to reach s_i from s_0 ;
 Apply a_k and observe o_l .
 Apply UIO_j at the tailstate and check that it is applicable;

- (2) Let $G'=(V', E')$ be the graph where $V'=V, E'=E(T)\cup E(U)\cup E_c$. Here, $E_c = \{ (s_i, s_p; a_k/o_l @ UIO_j) \mid (s_i, s_j; a_k/o_l) \in E-E(T)-E(U) \}$ and $TAIL_c(UIO_j) = s_p^j$, where $E(T)$, and $E(U)$ denote the set of all T -transitions and the set of all U -transitions respectively.

Apply the RCP algorithm for finding an optimal tour in G' which traverses each transition in E_c atleast once.

end U-CRT.

We will shortly give a procedure for generating a state cover tree T and an UIO-set U satisfying the TULD property. We can apply the RCP algorithm of [ADL88] only if the induced graph $G[E_c]$ is a weakly connected spanning subgraph of G' . This condition is satisfied here since our SPEC as well as the IUT have the reset capability.

As an example to illustrate our CRT method, consider the SPEC and its IUT shown in Figure 1. The set MU of UIO-sequences is given below.

$MU = \{UIO_0, UIO_1, UIO_2, UIO_3, UIO_4\}$, where
 $UIO_0 = y/1 \bullet x/0 \bullet b/1 \bullet b/1$
 $UIO_1 = b/1 \bullet x/0 \bullet x/0 \bullet c/0$
 $UIO_2 = d/1 \bullet x/0 \bullet x/0 \bullet c/0$
 $UIO_3 = x/0 \bullet b/1 \bullet a/0$
 $UIO_4 = a/0$

At the end of the first step of phase 1 we obtain the state cover tree T shown in Figure 2 and the UIO-set U (same as MU) with the TULD property. In step 2 of phase 1, the T -transition $(s_3, s_2; y/1)$ is tested in the IUT with the subsequence

$r/-\bullet y/1 \bullet y/1 \bullet d/1 \bullet x/0 \bullet x/0 \bullet c/0$.

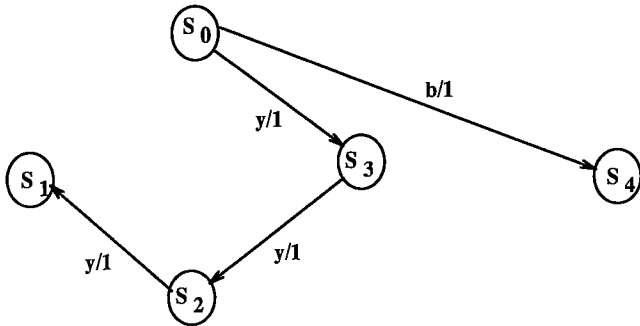


Figure 2. State cover tree of the SPEC given in figure 1

Since in the IUT $d/1 \bullet x/0 \bullet x/0 \bullet c/0$ is not applicable at the tailstate of $(s_3, s_2; y/1)$, phase 1 terminates. We can conclude that the transition $(s_3, s_2; y/1)$ or a transition in the sequence UIO_2 is faulty in the IUT.

The following theorem shows that the successful completion of phase 1 of the procedure U-CRT will guarantee a fault-free state cover tree and a verified set of UIO-sequences for the IUT.

Theorem 1:

Suppose that an IUT has atmost 1 fault and it passes phase 1 successfully then the following are true:

- (1) The state cover tree T obtained from the SPEC is fault-free in the IUT.
- (2) The UIO-sequence of each state of the SPEC from the set U is also an UIO-sequence of the corresponding state in the IUT.

The following theorem gives the fault resolution capability of our CRT-method.

Theorem 2 :

Suppose an IUT has atmost one fault, then the CRT-method has 1-fault resolution capability of level $1+l_u$ where l_u is the length of a longest UIO-sequence in the set U .

As noted in [SD88], for most of the known protocols $l_u \leq 5$. Therefore, from Theorem 2 we can deduce that for most of the known protocols, our method localizes the fault within six transitions.

Next, we consider the problem of extracting a state cover tree T and a UIO-set U with the TULD property. Let MU_i be a set of multiple UIO-sequences for the state s_i . Let MU be the collection of all the UIO-sequences in MU_i for every state s_i . We shall consider a simple procedure for constructing T and U given G and MU . The procedure is given below:

procedure constreeuio(MU,G);

 VT \leftarrow $\{s_0\}$;

 NT \leftarrow $\{s_0\}$;

 U \leftarrow $\{UIO_0\}$, where UIO_0 is an arbitrary member of MU_0 ;

 repeat

 delete a state s_i from NT;

for each outgoing transition $(s_i, s_j; a_k/o_l)$ at s_i such that $s_j \notin$ VT **do**

for each UIO_j in MU_j **do**

if the label a_k/o_l is not in UIO_j **then**

begin

 U \leftarrow U \cup $\{UIO_j\}$;

 VT \leftarrow VT \cup $\{s_j\}$;

 NT \leftarrow NT \cup $\{s_j\}$;

 T \leftarrow T \cup $\{(s_i, s_j; a_k/o_l)\}$;

break;

end

until (VT = V) or (NT = \emptyset)

 /* if NT = \emptyset and VT \neq V then there is no T and U

 satisfying the required condition for the given G and MU */

end constreeuio.

The algorithm guarantees the construction of the required state cover tree and a UIO-set if they exist for the given SPEC, and MU . The complexity of the procedure *constreeuo* is $O(|E| n^2 \mu_{\max})$, where n , $|E|$, μ_{\max} are the number of states in the SPEC, the number of transitions in the SPEC and the maximum number of multiple UIO-sequences (from the set MU) of any state in the SPEC, respectively. Here we have used the result that an UIO-sequence is of length atmost $2n^2$ [SD88]. However, as pointed out earlier the UIO-sequences for most of the practical protocols are of length atmost 5. In such cases, the complexity of our algorithm is $O(|E| \mu_{\max})$.

5. AN EXAMPLE: CONFORMANCE RESOLUTION TEST OF TRANSPORT PROTOCOL

We applied the CRT method on a subset of the class 4 transport protocol, NBS TP4 developed by the National Bureau of Standards [TP83]. In [SL89], Sidhu et al analyzed this protocol for studying different formal methods of conformance testing based on FSM model. The protocol has 15 states and 61 core transitions. It is found that the protocol has a state cover tree and a UIO-set satisfying the TULD property. They are obtained using the procedure *constreeuo*. The test sequence generated using the procedure U-CRT contains 1145 input interactions. Using this test sequence any faulty transition in the state cover tree is located to within 3 transitions. Any fault in transitions outside the state cover tree is located exactly. The detailed illustration can be referred in [RDT92].

6. CONCLUSION

In this paper, we have examined the existing formal conformal testing methods with reference to their fault resolution capabilities. We found that as the methods improved for better fault coverage, their fault resolution capabilities also improved. Among all these conformance testing methods, the W_p -method is found to have the best fault resolution capability for implementations with atmost one fault. Based on the W_p -method we have presented a CRT method with a fault resolution capability better than the W_p -method when the IUT has atmost one fault and the SPEC meets certain conditons. This method can be used for conformance testing of the IUT as well as for localizing the fault in the IUT. We have also reported results of applying the method on a subset of the transport protocol NBS TP4 presented in [SL89].

Our CRT-method achieves 1-fault resolution capability on a SPEC if the SPEC has a state cover tree, and a UIO-set with the TULD property. Interestingly such a tree and a UIO-set exist for the simplified transport protocol which we have used for illustration. We have also found that a few other protocols such as ISDN-BRI-D-Channel signaling protocol (network-interface side, originating end) [DSU90-2], and the alternating bit protocol [SD88] satisfy the required conditions for applicability of our CRT method.

An interesting open problem is to determine necessary and sufficient conditions on the structure of the SPEC for which our CRT-method will ensure 1-fault location capability. We are also investigating CRT-methods with multiple fault resolution capability.

7. REFERENCES

- [ADL88] A.V.Aho, A.T.Dahbura, D.Lee, and M.U.Uyar, "An optimization technique for protocol conformance test generation based on UIO sequences and rural chinese postman tours", Symposium of Protocol Specification, Testing and Verification 1988, pp. 75-86.
- [C78] T.Chow, "Testing software design modeled by finite state machine", IEEE Tr. Soft. Engg., Vol. SE-4, March 1978, pp. 178-187.
- [CVI89] W.Y.L.Chan, S.T.Vuong, and M.R.Ito, "An improved protocol test generation procedure based on UIOs", ACM SIGCOMM, 1989.
- [DS88] A.T.Dahbura and K.K.Sabnani, "An experience in estimating fault coverage of a protocol test", Proc. IEEE INFOCOM, March, 1988, pp. 71-79.
- [DSU90-1] A.T.Dahbura, K.K.Sabnani, and M.U.Uyar, "Formal methods for generating protocol conformance test sequences", Proc. IEEE, vol. 78, No. 8, 1990, pp. 1317-1326.
- [DSU90-2] A.T.Dahbura, K.K.Sabnani, and M.U.Uyar, "Algorithmic generation of protocol conformance tests", AT&T Technical Journal, January/February 1990, pp. 101-118.
- [FBK91] S.Fujiwara, G.v.Bochmann, F.Khendek, M.A.Amalou, and A.Ghedamsi, "Test selection based on finite state model", IEEE Tr. Soft. Engg., Vol. SE-17, Jun 1991, pp. 591-603.
- [K62] M.-K.Kuan, "Graphic Programming Using Odd or Even Points", Chinese Math vol. 1, 1962, pp.273-277.
- [K78] Z.Kohavi, *Switching and Finite Automata Theory*, New York, NY: McGraw Hill, 1978.
- [NT81] S.Naito and M.Tsunoyama, "Fault detection for sequential machines by transition tours", Proc. IEEE Fault Tolerant Computing Conference, 1981.
- [R87] D.Rayner, "OSI conformance testing", Computer Networks and ISDN systems 14, 1987, pp. 79-98.
- [RDT92] T.Ramalingam, Anindya Das and K.Thulasiraman, "On conformance test and fault resolution of protocols based on FSM model", Technical Report, Department of Electrical Engineering, Concordia University, Montreal, Canada.
- [SD85] K.K.Sabnani and A.T.Dahbura, "A new technique for generating protocol tests", Proc. ninth Data Communication Symp., IEEE Computer Society press, Sep 1985, pp. 36-43.
- [SD88] K.Sabnani and A.Dahbura, "A protocol test generation procedure", Computer Networks and ISDN systems, 15, 1988, pp. 285-297.
- [SL89] D.P.Sidhu and T.K.Leung, "Formal methods for protocol testing: a detailed study", IEEE Tr. Soft. Engg., vol. 15, No. 4, 1989, pp. 413-426.
- [TP83] "Specification of a transport protocol for computer communications, vol. 3: class 4 protocol", Nat. Bureau Standards, Washington, DC., Rep. ICST/HNLP-83-4 Jan., 1983.