Communication Protocols Testability Improvement by Narrow Input/Output (NIO) Sequences

Tao Huang and Anthony Chung
School of Computer Science, Telecommunications and Information Systems
DePaul University
Chicago, Illinois, U.S.A.
thuang@condor.depaul.edu, chung@cs.depaul.edu

Abstract: – Communication protocol conformance testing has become more complex as protocols have grown larger. The study of design for testability (DFT) is a research area to help overcoming the ever increasing complexity of testing distributed systems. In the previous research, we introduced a new metric for testability of communication protocols, demonstrated the usefulness of the metric for identifying faults that are more difficult to detect, and also presented the approach for improved testability without modifying the protocol structure by means of unique input/output (UIO) sequences. Since a UIO sequence may not exist for every state of some protocol specification finite state machines (FSMs), in this paper, we extend UIO sequence to a new general concept called narrow input/output (NIO) sequence. We demonstrate the effectiveness of NIO sequence application on the improvement of protocol testability.

Keywords: – Communication Protocols, Protocol Conformance Testing, Testability.

1. Introduction

The study of design for testability (DFT) is a research area in which researchers investigate design principles that will help to overcome the ever increasing complexity of testing distributed systems [13]. Testability in hardware design has been well studied and the idea has been extended to the field of software engineering [7]. Research in this area has resulted in metrics for testability of components and design rules for achieving testability. It is important to see how the idea can be extended to the field of communication protocols.

Testability metrics are essential for evaluating and comparing designs. In previous research [3], we proposed a new metric for protocol testability based on the *detection probability* of a fault [6], which is the probability of detecting the fault by a random input. We also introduced the *testability profile* of a protocol, which is a statistical measure of the "ease" or "difficulty" of testing a protocol. This profile can be used as a guide for test selection. With some empirical studies on both small and large or more realistic examples, we demonstrated the ways to generate the exact metric and approximation of the metric [3] [4]. As a result of these metrics, we were able to tell that certain faults in certain transitions are more difficult to detect. Then, we also proposed two approaches to improve the testability of communication protocols for a single tail state fault [5]. The first approach is to augment a protocol with transitions, similar to the insertion of debugging statements in computer programs. The second approach does not modify the structure of the protocol. When a transition tour is being generated, a state signature (a special input/output sequence that uniquely identifies a state) is applied to the tail state of a transition which is harder to detect if faulty. We used *unique input/output* (UIO) sequences in our study there.

However, there are some protocol specification Finite State Machines (FSMs) which do not possess UIO sequences for every state. In this paper, we extend UIO sequence to introduce a new concept, *narrow input/output* (NIO) sequence. An NIO sequence is always obtainable for every state of any FSM, and in fact, an NIO sequence for a state is also its UIO sequence in most cases. We want to investigate the effectiveness of NIO sequences on the improvement of the protocol testability.

The rest of the paper is organized as follows. In Section 2, general concepts of protocol conformance testing and our testability metric are introduced. Section 3 introduces the new NIO

sequence concept, describes the way to compute NIO sequences, and reports the experiment results on the application of NIO sequences. Section 4 concludes the paper.

2. Protocol Conformance Testing and the Testability Metric

Protocol conformance testing is generally done by applying a test sequence, which is constructed from the specification of the standard, to an implementation under test (IUT). The observed responses of the IUT are compared with the expected ones to detect errors.

In this paper, protocols are specified using the Finite State Machine (FSM) model. An FSM is a 5-tuple $M = (S, I, O, \delta, \lambda)$, where: (1) $S = \{s_i \mid 0 \le i \le n-1\}$ is a non-empty finite set of states, and there is an initial state s_0 among these states, (2) I is a non-empty set of inputs, (3) O is a non-empty set of outputs, (4) δ: $S \times I \rightarrow S$ is the state transition function, (5) λ: $S \times I \rightarrow O$ is the output function.

An FSM can be represented as a labeled directed graph $G = (V, E)$, where $V = \{v_i \mid 0 \le i \le n-1\}$ is a non-empty finite set of vertices and $E = \{(v_j, v_k; i_p/o_q) \mid v_j, v_k \in V; i_p \in I; o_q \in O\}$ is a non-empty finite set of directed edges. Each vertex v_i represents the corresponding state s_i of the FSM. Each directed edge $(v_j, v_k; i_p/o_q)$ from vertex v_j to vertex v_k with label i_p/o_q represents a state transition of the FSM from starting (head) state s_j to ending (tail) state s_k with input i_p and output o_q, that is, $\delta(s_j, i_p) = s_k$ and $\lambda(s_j, i_p) = o_q$.

An example protocol FSM is given in Figure 1. The double circled vertex is the initial state. The following is an example of a test sequence:

a/0 a/1 b/1 b/1 b/0 a/0 a/1

which visits each transition at least once. However, if an implementation has an ending state error (as shown in Figure 2), in which the ending state of a transition is different from that in the implementation, this sequence will not be able to detect the fault.

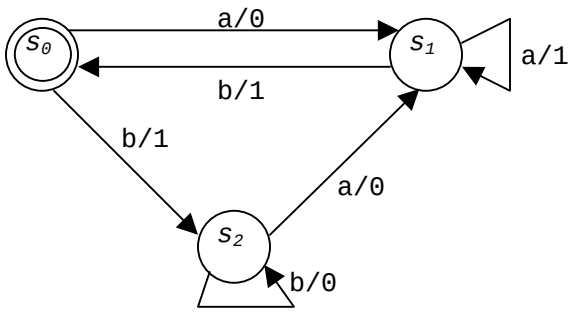

Figure 1: An Example FSM

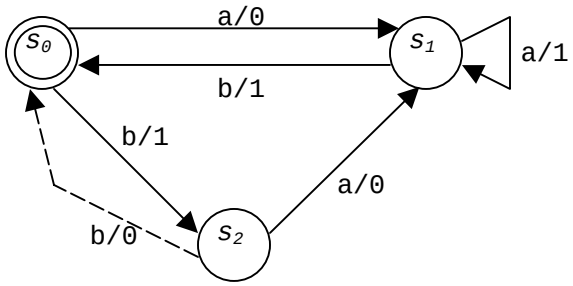

Figure 2: The Example FSM with a Tail State Error

In a previous paper [3] we propose a new metric for protocol testability. Before we briefly repeat the testability metric definition, we would like to stress the following points: (1) When a protocol FSM has transition sequence cycles (which is generally the case) the number of possible test sequences is infinite. Our testability metric is defined for a specific length of test sequences. (2) There are many different types of *fault models*. A common fault model used in the study of protocol test generations is a single ending state error. In this paper, we will focus on this fault model.

We define *DP(n)*, the *detection probability of a fault by a test sequence of length n*, to be ratio of test sequences of length *n* that can detect a fault to the total number of test sequences of length *n*. The *TP(n)* of a protocol is the probability density function of detection probabilities of its faults by test sequences of length *n*. We also define *T(n)* to be the weighted means of all the probabilities. To derive the quantities *DP(n), TP(n),* and *T(n)*, we focus on test sequences that visit each transition at least once. In [3], we illustrate the usefulness of the metric study for determining certain faults that

are harder to detect; we also see clearly that as the length of transition tours increases, the weighted average of detection probabilities also increases, and it approaches 1. In a follow-up paper [4] we demonstrate the generation of an approximation of the metric instead of the exact metric in case of large or more realistic protocol FSMs. That can be done by randomly applying a subset of faults and generating a subset of test sequences.

A *unique input/output* (UIO) sequence [11] for a state of an FSM is a sequence of input/output pairs such that this input/output behavior is only exhibited by starting from this state and not exhibited by starting from any other state of the FSM. For the example FSM in Figure 1, state s_0 has "b/1 b/0" as a UIO, state s_1 has "a/1" as a UIO, and state s_2 has "b/0" as a UIO. In another paper [5], we propose two approaches to improve the testability of communication protocols for a single tail state fault. One of the approaches does not modify the structure of the protocol. When a transition tour is being generated, a UIO sequence of the state is applied to the tail state of a transition that is known harder to detect if faulty. Results of the weighted means of all the probabilities $T(n)$ show that the testability of an example protocol is indeed improved with the approach.

3. Narrow Input/Output (NIO) Sequences for Testability Improvement

FSM tail state faults can be detected if a test sequence verifies which state the FSM is in at a certain point. This is accomplished by a sequence of inputs to be applied to the FSM starting from that state and observing the expected outputs. These are generally called *characterizing sequences* (CS). UIO sequence is one kind of CS. Others are *distinguishing sequence* (DS) [8], *characterization set* (called *W-set*) [2] and its variants. An FSM that does not have a DS may still have a UIO. Application of W-set requires a reliable reset transition which is able to bring the FSM from each state back to the initial state. Therefore, UIO sequence has been in popular use for tail state verification of an FSM transition by most test sequence generation optimization methods [1] [12] [9].

Unfortunately, not every state of an FSM has a UIO sequence. For instance, the state s_0 of an FSM shown in Figure 3 does not have a UIO sequence. If the input 'a' is applied to the FSM in state s_0 or state s_1 , its next state is state s_3 with the output '1'. This means that any UIO sequence for state s_0 cannot begin with the input 'a'. Similarly if the input 'b' is applied to the FSM in state s_0 or state s_2 or state s_3 , the next state is state s_2 with the output '0'. Thus, any UIO sequence for state s_0 cannot begin with the input 'b' either. It follows that the state s_0 of the FSM does not have a UIO sequence.

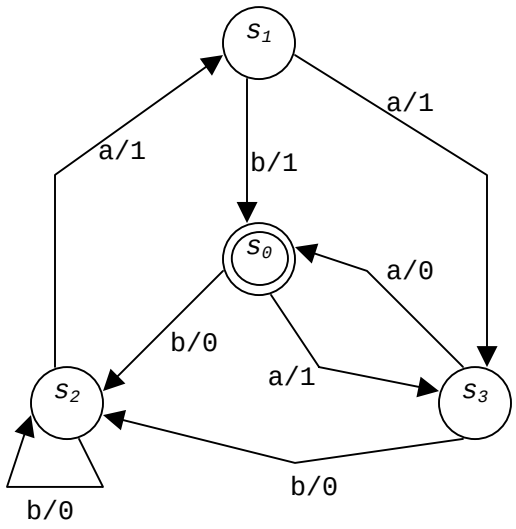

Figure 3: Another Example FSM with a State without UIO

To consider more general applicability, we extend UIO sequence to introduce a new concept, *narrow input/output* (NIO) sequence.

Definition: A narrow input/output (NIO) sequence for a state of an FSM is a sequence of input/output pairs such that this input/output behavior is exhibited by starting from this state and there are a minimum number of other states from which the same input/output behavior is also exhibited.

According to the above definition, if a state has a UIO sequence, then the UIO sequence is also an NIO sequence for that state. Thus, UIO sequence is actually a special case of NIO sequence. If the tail state of a transition has a UIO sequence, we

know that the UIO sequence can be used to uniquely identify the tail state reached by the transition. In case the tail state of a transition does not have a UIO sequence, intuitively, an NIO sequence of the state can be used to distinguish the state reached by the transition from as many other states as possible. We believe that the application of NIO sequence rather than otherwise no distinction being made at all for transition tail state in this situation will improve the overall fault detection capability of a test sequence.

To compute the NIO sequences for every state of an FSM, we may utilize the same method in [10] for construction of a UIO tree which we call NIO tree in our context. The NIO tree for the FSM in Figure 3 is shown in Figure 4. Each node in an NIO tree is represented by two rows of equivalent number of states: the top row lists the initial states, whereas the bottom row lists the corresponding current states respectively after a transition path which is the input/output sequence composed of all the labels from the root to this node. A state s_i in the top row with a corresponding s_i' in the bottom row of a node occurs in the top row with a corresponding s_i'' in the bottom row of its child node connected by an edge labeled i_p/o_q , if and only if there is a state transition in the FSM from state s_i' to state s_i'' with input i_p and output o_q , that is, $\delta(s_i', i_p) = s_i''$ and $\lambda(s_i', i_p) = o_q$. A terminal node is in one of the two conditions: either all its initial/current state pairs are totally contained in another ancestor node on the path from the root to this node; or all the states in its bottom row are identical, in which case it is called a homogeneous node. To get an NIO sequence for a state, we just locate a homogeneous terminal node with minimum number of initial/current state pair columns where the state is in the top row, then the label sequence along the path from the root to the terminal node is an NIO of this state.

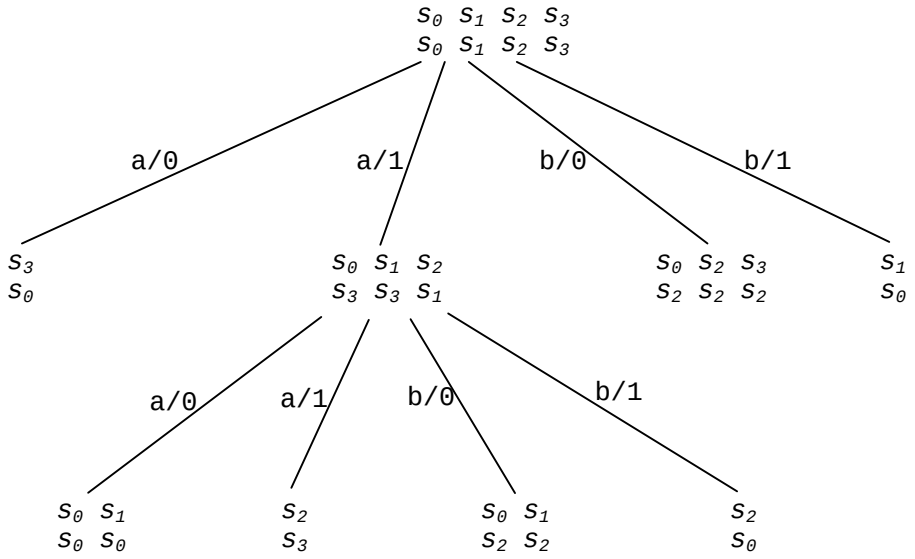

Figure 4: An NIO Tree

From the NIO tree in Figure 4, we can get the NIO sequences for the states of the FSM in Figure 3. They are listed in Table 1. Note that the NIO sequences for states s_1 , s_2 , and s_3 are also their UIO sequences, but this is not the case for state s_0 .

Table 1: NIO Sequences

s_0	a/1 a/0 a/1 b/0
s_1	b/1
s_2	a/1 a/1
s_3	a/0

To verify our original hypothesis about the effectiveness of NIO sequences, we conducted an empirical study on the NIO effect by means of our protocol testability metric. Specifically, first, we measured the testability metric quantities for the FSM in Figure 3 in the same way as we did in [3], by which the test sequences are all the possible transition tours with various lengths which traverse each transition at least once. Then, we did the metric measurement again by only applying the test sequence transition tours which satisfies the condition: for each transition which has the state s_0 as the tail state (from s_1 to s_0 with input/output label 'b/1', or from s_3 to s_0 with input/output label 'a/0'), at least one of its occurrence on the test sequence must be immediately followed by one of the NIO sequences for the state s_0 ("a/1 a/0" or "a/1 b/0"). The experiment results of the weighted means of all the probabilities $T(n)$ for n = 11 to 20 are summarized in Table 2. It is shown that the testability of the protocol has been improved after the NIO sequences were applied to the test sequences.

Table 2: Experiment Results of $T(n)$ for n = 11 to 20

	Before NIO Consideration	After NIO Consideration
T(11)	0.701003	0.808333
T(12)	0.724420	0.800725
T(13)	0.740794	0.824099
T(14)	0.757876	0.830293
T(15)	0.773753	0.842638
T(16)	0.787121	0.853363
T(17)	0.799650	0.862577
T(18)	0.811186	0.870811
T(19)	0.821638	0.878758
T(20)	0.831296	0.886031

4. Conclusions

We introduced new metric for testability of communication protocols based on the detection probability of a fault, and illustrated the usefulness of the metric for determining faults that are more difficult to detect in our previous research [3] [4]. Accordingly, test sequences can be chosen to focus more on the detection of those particular faults. In the follow-up research, we also demonstrated the approaches to improve the testability of communication protocols [5]. If the structure of a protocol is not modified, the testability of the protocol can be improved by applying the UIO sequences to the tail states of the transitions which are harder to detect to be faulty. However, there are some protocol specification FSMs which do not possess UIO sequences for every state. In this paper, we extend UIO sequence to introduce a new concept, *narrow input/output* (NIO) sequence. We also describe the way to compute NIO sequences. Through the protocol testability measurement experiment, we have demonstrated the effectiveness of NIO sequences on the improvement of the protocol testability. Due to its general applicability, the concept of NIO sequence is also very helpful for protocol conformance testing. Future research will include the study of NIO sequence improvement on protocol conformance test sequence generation methods.

References

[1] 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", *IEEE Transactions on Communications*, vol. 39, no. 11, pp. 1604-1615, Nov. 1991.

[2] T. S. Chow, "Testing Software Design Modeled by Finite-State Machines", *IEEE Transactions on Software Engineering*, vol. SE-4, no. 3, pp. 178-187, May 1978.

[3] Anthony Chung, and Tao Huang, "A New Metric for the Testability of Communication Protocols", International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications

(CSITeA 2002), Foz do Iguazu, Brazil, June 6-8, 2002.

[4] Anthony Chung, and Tao Huang, "Further Study of a Metric for the Testability of Communication Protocols", International Conference on Computer Science, Software Engineering, Information Technology, e-Business, and Applications (CSITeA 2003), Rio de Janeiro, Brazil, June 5-7, 2003.

[5] Anthony Chung, and Tao Huang, "Two Approaches for the Improvement in Testability of Communication Protocols", 4th Annual ACIS International Conference on Computer and Information Science (ICIS 2005), Jeju Island, South Korea, July 14-16, 2005.

[6] H. Farhat, "Test Generation for VLSI Circuits from Testability Profile Distribution", *14th International Conference on Computer Applications in Industry and Engineering*, Nov. 2001.

[7] R. S. Freedman, "Testability of Software Components", *IEEE Transactions on Software Engineering*, vol. 17, no. 6, pp. 553-564, June 1991.

[8] G. Gonenc, "A Method for the Design of Fault-Detection Experiments", *IEEE Trans. Comput.*, vol. C-19, pp. 551-558, June 1970.

[9] R. E. Miller, and S. Paul, "On the Generation of Minimal-Length Conformance Tests for Communication Protocols", *IEEE/ACM Transactions on Networking*, vol. 1, no. 1, pp. 116-129, Feb. 1993.

[10] K. Naik, "Efficient Computation of Unique Input/Output Sequences in Finite-State Machines", *IEEE/ACM Transactions on Networking*, vol. 5, no. 4, pp. 585-599, Aug. 1997.

[11] K. Sabnani, and A. Dahbura, "A Protocol Test Generation Procedure", *Computer Networks and ISDN Systems*, vol. 15, pp. 285-297, 1988.

[12] Y. N. Shen, F. Lombardi, and A. T. Dahbura, "Protocol Conformance Testing Using Multiple UIO Sequences", *IEEE Transactions on Communications*, vol. 40, no. 8, pp. 1282-1287, Aug. 1992.

[13] S. T. Vuong, A. A. F. Loureiro, and S. T. Chanson, "Design for Testability of Communication Protocols: A Framework and Perspectives", *Electronic Journal on Networks and Distributed Processing*, no. 2, pp. 95-119, Sept. 1995.