GENERATION OF REALIZABLE CONFORMANCE TESTS UNDER TIMING CONSTRAINTS *

M. Ümit Uyar†, Mariusz A. Fecko, Adarshpal S. Sethi, Paul D. Amer
Computer and Information Science Department
University of Delaware, Newark, DE

Abstract

An optimization method is introduced for generating minimum-length test sequences taking into account timing constraints for FSM models of communication protocols. Due to active timers in many of today's protocols, the number of consecutive self-loops that can be traversed in a given state before a timeout occurs is limited. An example of a protocol where this constraint occurs is MIL-STD 188-220B. A test sequence that does not consider timing constraints will likely be unrealizable in a test laboratory, thereby potentially resulting in the incorrect failing of valid implementations. The solution uses a series of augmentations for a protocol's directed graph representation. The resulting test sequence is proven to be of minimum-length while not exceeding the tolerable limit of consecutive self-loops at each state. Although UIO sequences are used for state verification method, the results also are applicable to test generation that uses distinguishing or characterizing sequences.

1 Introduction

Due to interoperability requirements of heterogeneous devices in a complex communications network, each component must be tested for conformance against its specification. Automated generation of conformance tests based on the formal descriptions of general communication protocols has been an active research area [1] - [9]. Recently, these techniques have been considered for the test case generation of MIL-STD 188-220B [10]. These techniques, using a deterministic finite-state machine (FSM) model of a protocol specification, focus on the optimization of the test sequence length. If, however, there exist timing constraints imposed by a protocol's active timers and these constraints are not considered during test sequence generation, the generated test sequence may not be realizable in a test laboratory. This can result in the incorrect failing of valid implementations. It was during ATIRP-sponsored test case generation research for MIL-STD 188-220B that the problem of timing constraints was discovered and then studied [11].

In this paper, a solution is given to optimize the test sequence length and cost under the constraint that an Implementation Under Test (IUT) can remain only a limited amount of time in some states during testing, before a timer's expiration forces a state change. The solution augments original graph representation of the protocol FSM model. Then it formulates a Rural Chinese Postman Problem solution [12] to generate a minimum-length tour. In the final test sequence generated, the number of consecutive self-loops never exceeds any state's specified limit.

UIO sequences [13] are used for state verification throughout the paper. However, the results presented also are applicable to test generation that uses the distinguishing or characterizing sequences [14, 15]. Earlier results of this study, limited to verification sequences that are self-loops, are presented in [16]. This paper generalizes these earlier results to both self-loop and non-self-loop verification sequences.

Section 2 presents the practical motivation behind the optimization problem formulated in the paper. Two real protocols, U.S. Army MIL-STD 188-220B and Q.931 [17], demonstrate real examples of protocols with self-loop timing constraints. Section 3 provides the background information for FSM models and test generation. It also discusses the practical restrictions imposed on test sequences due to the timers. Section 4 presents an outline of the optimization problem. An outline and an example of a solution to this optimization problem are presented in Section 5.

2 Motivation

During testing, traversing each state transition of an IUT requires a certain amount of time. A test sequence that traverses too many *self-loops* (a *self-loop* is a state transition that starts and ends at the same state) in a given state will not be realizable in a test laboratory if the time to traverse the self-loops exceeds a timer limit as defined by another transition originating in this state. In this case, a timeout will inadvertently trigger forcing the IUT into a different state, and thereby disrupting the test sequence before all of the self-loops are traversed. If this unrealizable test sequence is not avoided during test generation, most IUTs will fail the test even when they meet the specification. Clearly, this is not the goal of testing. Therefore, a properly generated test sequence must take timer constraints into account.

Examples of protocols that contain many self-loop transitions in their FSM models include ISDN Q.931 for supplementary voice services, MIL-STD 188-220B for Combat Net Radio communication, and LAPD [18], the data link protocol for the ISDN's D channel.

In addition to the original self-loops of a specification model, additional self-loops are typically created when generated test sequences use state verification techniques such as unique input/output (UIO) sequences [13],

*This work is supported by ARO SPP administered by Battelle (DAAL03-91-C-0034), by ARO (DAAL03-91-G-0086), and by ATIRP Consortium sponsored by the ARL under the FedLab Program (DAAL01-96-2-0002).

†Dr. Uyar, a Research Professor with Department of Electrical Engineering, the City College of the City University of New York, is presently Visiting Associate Professor at University of Delaware.

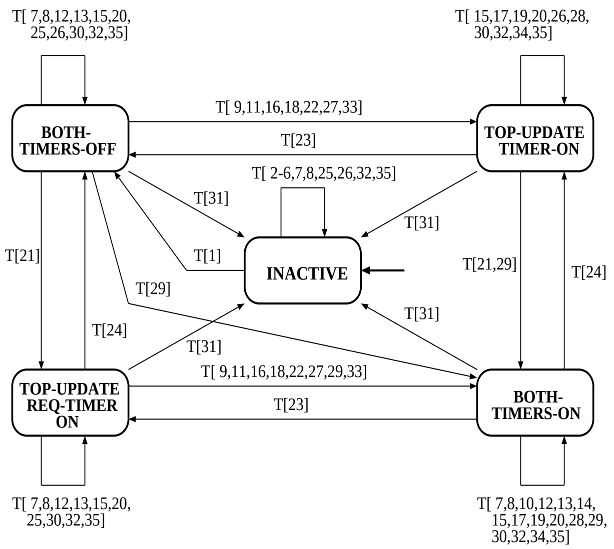

Figure 1: Extended FSM for Topology Update module of MIL-STD 188-220B.

distinguishing sequences [14, 15], or characterizing sequences [14, 15].

Example 2: Timing constraints in MIL-STD 188-220B

The University of Delaware's Protocol Engineering Laboratory is developing test scripts to be used by the U.S. Army CECOM in their MIL-STD 188-220B Conformance Tester. Tests are being generated for both the Data Link and Intranet Layers.

The tests are derived from an Estelle specification of the protocol. An extended FSM (i.e., FSM with memory) representing a portion of the Intranet Layer of 188-220B, called the Topology Update (TU), is shown in Figure 1 [19]. The equivalent FSM model of Topology Update has 10 states and 345 state transitions. In 8 of these states at least one timer is running in the implementation.

A timer's status (i.e., on or off) determines the behavior of the implementation. For example, when the topology information changes, the station is allowed to send a topology update message only if the *Topology_Update Timer* is not running. Otherwise, no message is sent. Based on this characteristic, the state names include the timer status in Figure 1.

There are 10 self-loop transitions defined for each of the states *TOP-UPDATE-REQ-TIMER-ON* and *TOP-UPDATE-TIMER-ON*, and 16 self-loops for state *BOTH-TIMERS-ON*. Depending on the timer expiration values, it may not be possible to execute all of the respective self-loop transitions during one visit to either state. Timing constraints due to the active timers must be taken into account to generate realizable test sequences for the Intranet Layer of 188-220B. Otherwise, valid implementations will fail the test sequence, which is not what the tester desires.

Example 1: ISDN Q.931

The portion of the Q.931 protocol that defines ISDN's basic voice services specifies 12 states and 16 different inputs for the user side. In the specification, there are 86 "normal" state transitions and 106 "inopportune" message transitions.

Each inopportune transition is modeled as a self-loop with a null output. In a test laboratory, an inopportune transition is tested by supplying its input to the IUT, and observing that the IUT does not generate any output. Usually, a timer is run by the tester to make sure that no output is generated. Then, to verify that the state of the IUT did not change, a *STATUS_INQUIRY* input is applied to the IUT, which generates an output called *STATUS*. The input of *STATUS_INQUIRY* and its output *STATUS* are self-loop transitions defined for each state.

Therefore, in Q.931, each state has an average of 9 inopportune transitions, which requires the traversal of 18 self-loop transitions during testing. The total ratio of self-loops to nonself-loop transitions is approximately 3 to 1 in the final test sequence. This ratio is even larger for the Q.931 supplementary voice services. LAPD, the ISDN data link layer protocol, demonstrates a similar characteristic: a high ratio of self-loop versus non-self-loop transitions.

A Q.931 implementation has several active timers that are running in certain states. For example, when an IUT moves from state *Null* to *Call Initiated*, a timer labeled as *T303* is started. When testing inopportune transitions in state *Call Initiated*, a tester has to consider a limited amount of time that can be used for inopportune tests before the timer expires. Other examples of timers in Q.931 are: timer *T304* running in state *Overlap Sending*, and timer *T310* in state *Outgoing Call Proceeding*.

3 Preliminaries and practical restrictions on test sequences

A *protocol* can be specified as a deterministic FSM [15, 20], which can be represented by a directed graph $G = (V, E)$. The set $V = \{v_1, \ldots, v_n\}$ of vertices correspond to the set of states S of the FSM. A directed edge from v_i to v_j with label $L_k = a_l/o_m$, and the *cost* to realize the edge during testing, corresponds to a state transition in the FSM from s_i to s_j by applying input a_l and observing output o_m. If the start and the end vertices of an edge are the same (i.e., $v_i = v_j$), the edge is called a *self-loop*. The *indegree* and *outdegree* of a vertex are the number of edges coming toward and directed away from it, respectively. If the indegree and outdegree of each vertex are equal, the graph is said to be *symmetric*.

A *tour* is a sequence of consecutive edges that starts and ends at the same vertex. An *Euler tour* is a tour that contains every edge of G exactly once. The so-called *Chinese Postman Problem* is defined as finding a minimum-cost tour of G that traverses every edge at least once [21]. The *Rural (Chinese) Postman Problem* is finding a (minimum-cost) tour for a subset of edges in G [12].

During conformance testing of a protocol implementation, the IUT is viewed as a *black box*, where only the inputs applied to the IUT and the outputs generated by the IUT can be controlled and observed, respectively. An IUT *conforms* to its specification if all state transitions defined in the specification are tested successfully. To test a single transition defined from state v_i to v_j, the following steps are needed:

- bring the IUT into state v_i;
- apply the required input and compare the output(s) generated with those defined by the specification;
- verify that the new state of the IUT is v_j by applying a state verification sequence.

Aho et al. introduced an optimization for the test sequence length (and cost) using UIO sequences [1] to perform the last step of the above single transition test. A UIO sequence of a state s_i is a sequence of edges starting at v_i such that the output sequence generated by these edges is unique for v_i.

The existing methods for conformance test generation [1, 6, 13, 14, 15, 20, 22, 23, 24, 25] emphasize optimizing the test sequence length and its cost, without considering any restrictions on the order in which the tests can be applied to an IUT. However, an optimization technique for generating realizable tests must consider the additional restriction that there is a limit on the number of self-loop transitions traversed consecutively.

This paper presents minimum-cost test sequence generation under the constraint that the number of consecutive self-loops that can be traversed during a visit to a given state is limited. In most cases, this test sequence will be longer than one without the constraint since limiting the number of self-loop traversals may require additional visits to a state which otherwise would have been unnecessary.

A minimum-cost test sequence generation method is presented in Section 5. The test sequence generated by the presented algorithm is longer than an absolute minimum-cost test sequence that can be obtained without the self-loop restriction. The limitation on how long an IUT can stay in a state may force the IUT to visit a state several times more than otherwise necessary in an absolute minimum-cost tour.

4 Problem formulation

Given the graph $G(V, E)$ representing the FSM for a certain protocol, let us define the following parameters:

- $d_{out}(v_i)$, $d_{in}(v_i)$ - the out-degree and in-degree of vertex v_i, respectively;
- $d_{self}(v_i)$ - the number of self-loops of vertex $v_i \in V$;
- $max\_self(v_i)$ - the maximum number of self-loops in a test sequence that can be traversed at each visit to v_i. As indicated in Section 3, attempting to remain in state v_i long enough to execute more than $max\_self(v_i)$ self-loops would result in disruption of a test sequence;
- $d_{min\_self}(v_i)$ - the minimum number of times a tour covering all edges in E must include vertex $v_i \in V$.

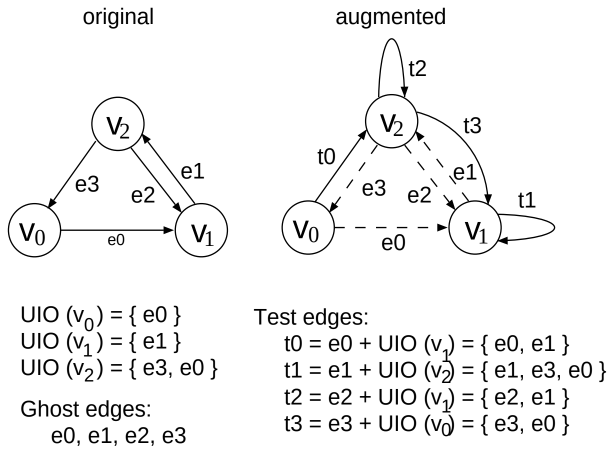

Figure 2: Augmenting a graph with test and ghost edges.

4.1 Formulation of Rural Chinese Postman Problem

Let each edge $(v_i, v_j) \in E$ in G be replaced by a *test edge* $(v_i, v_k) \in E_{test}$ and a *ghost edge* $(v_i, v_j) \in E_{ghost}$. The test edge (v_i, v_k) is a concatenation of edge (v_i, v_j) and $UIO(v_j)$, where $UIO(v_j)$ ends at v_k. The cost of (v_i, v_k) is the sum of the costs of (v_i, v_j) and $UIO(v_j)$ (see Figure 2 for an example of augmenting a graph with test and ghost edges).

Our goal is to build a minimum-cost tour of G such that all edges in E_{test} (and some edges in E_{ghost}, if needed) are traversed with the constraint that each vertex v_i can only tolerate $max\_self(v_i)$ consecutive self-loop traversals.

Let $\hat{E}_{test}$ be the set of all test edges that are a concatenation of a self-loop edge and a self-loop UIO sequence. Let $G^{'}(V^{'}, E^{'})$ be a graph containing all edges of G except for the test edges in $\hat{E}_{test}$ (edges in $\hat{E}_{test}$ will be added to a test sequence once it is found). The difference between the number of incoming and outgoing test edges of $v^{'} \in G^{'}$ is eliminated by duplicating some of the incoming and/or outgoing ghost edges of $v^{'}$, for all $v^{'} \in V^{'}$. The resulting graph $G^{''}(V^{''}, E^{''})$ is a rural symmetric augmentation of $G^{'}$. By definition, in $G^{''}$, the in-degree of any vertex $v_i^{''} \in V^{''}$ is equal to its out-degree. Also, the timing constraint requires that the in-degree of any vertex $v_i^{''}$ with a sel-loop UIO sequence be greater or equal to the value defined by $d_{min\_self}(v_i)$, where v_i is the corresponding vertex in V. [1]

Our goal is to build a Rural Chinese Postman tour in which the timing constraint due to timers is satisfied for each vertex $v_i \in V$. A Rural Chinese Postman tour is a minimum-cost tour covering each transition $e \in E_{test}^{'}$ exactly once, and each $e \in E_{ghost}$ zero or more times. Such a tour is equivalent to an Euler tour in a minimum cost

[1]Note that, unless stated otherwise, $v_i^{'}, v_i^{''}$ and v_i^* are used in this paper to denote the copies of a corresponding vertex $v_i \in V$ in graphs $G^{'}, G^{''}$ and G^*, respectively.

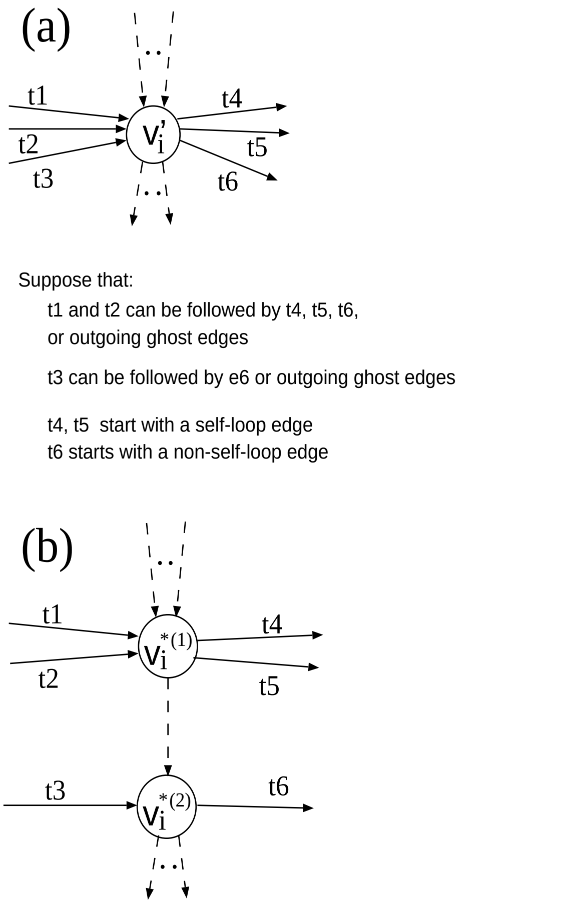

Figure 3: Conversion of $v_i^{'}$ in $G^{'}$ (part (a)) to $v_i^{*(1)}, v_i^{*(2)}$ in G^* (part (b)).

symmetric $G^{''}$. In other words, the objective is to obtain the graph $G^{''}$ as the minimum-cost rural symmetric augmentation of the graph $G^{'}$.

5 Minimum-cost solutions for constrained self-loop testing

The detailed description of an algorithm for finding the minimum-cost augmentation of $G^{'}$ as $G^{''}$ with the introduced self-loop constraint is presented in [16, 26]. The method uses several graph transformations and applies network flow techniques to obtain a minimum-cost solution. The transformations applied to vertex $v_i^{'}$ depend on the form of $UIO(v_i)$ [26].

Figure 3 shows the transformation of $v_i^{'}$ whose UIO sequence contains both self-loop and non-self-loop edges. All outgoing test edges of $v_i^{'}$ that start with a self-loop edge of $v_i^{'}$ have the same number of leading composite self-loop edges.

If the number of the ending self-loops of an incoming test edge (edges $t1$ and $t2$ in Figure 3) is less than $max\_self(v_i)$, the incoming test edge is made incident on $v_i^{*(1)}$. Each test edge incident on $v_i^{*(1)}$ will be included in the tour T such that it may be followed by any outgoing test edge or any outgoing ghost edge of $v_i^{'}$ (note that an outgoing test edge ($t4$, $t5$ or $t6$) may have at most one self-loop at the beginning).

On the other hand, if the number of the ending self-loops of an incoming test edge (edge $t3$ in Figure 3) is equal to or greater than $max\_self(v_i)$, the incoming test edge is made incident on $v_i^{*(2)}$. The incoming test edges of $v_i^{*(2)}$ will be followed only by the outgoing test or ghost edges of $v_i^{*(2)}$, which start with non-self-loops (e.g., an egde $t6$).. Therefore, the T will not be disrupted by timeouts when implemented as a test sequence.

Test edge	*Start vertex*	*End vertex*	*Edges included*
t0	v_0	v_2	e0, **e1, e5**
t1	v_1	v_2	e1, **e1, e5**
t2	v_1	v_2	e2, **e1, e5**
t3	v_1	v_2	e3, **e1, e5**
t4	v_1	v_2	e4, **e1, e5**
t5	v_1	v_2	e5, **e12**
t6	v_1	v_3	e6, **e13**
t7	v_2	v_3	e7, **e13**
t8	v_3	v_0	e8, **e0, e2**
t9	v_3	v_0	e9, **e0, e2**
t10	v_3	v_0	e10, **e0, e2**
t11	v_3	v_0	e11, **e0, e2**
t12	v_2	v_2	e12, **e12**
t13	v_3	v_3	e13, **e13**

Table 1: Test and ghost edges for the graph of Figure 4 (a)

Example: Consider an FSM whose UIO sequences belong to all three possible classes (Figure 4). Suppose that the maximum tolerable number of consecutive self-loop traversals is one for vertex v_0, two for v_1, and three for vertices v_2 and v_3. Let $e6$ and $e7$ be timeout transitions. When either of them is triggered, an IUT moves into state v_3. UIO sequences and the values of max_self and d_{min_self} are:

Vertex	*UIO sequence*	max_self	d_{min_self}
v_0	e0, e2	1	4
v_1	e1, e5	2	9
v_2	e12	3	5
v_3	e13	3	2

After replacing the original transitions by the ghost and test edges, we obtain the set of test edges for the graph of Figure 4 (a) as shown in Table 1.

Testing of $t8$ involves traversing one self-loop of v_1 (i.e., $e2$) as part of UIO sequence of v_0. Since $UIO(v_1)$ starts with a self-loop (i.e., $e1$) and $max\_self(v_1) = 2$, no self-loops of v_1 can be tested immediately after testing $t8$. This implies that test edges $t1, t2, t3$ and $t4$, which start from a self-loop of v_1, cannot follow $t8$ in a realizable test sequence. The same restriction also applies to $t9, t10$, and $t11$.

The following test sequence is obtained by applying the rural Chinese postman method [1] to the graph without

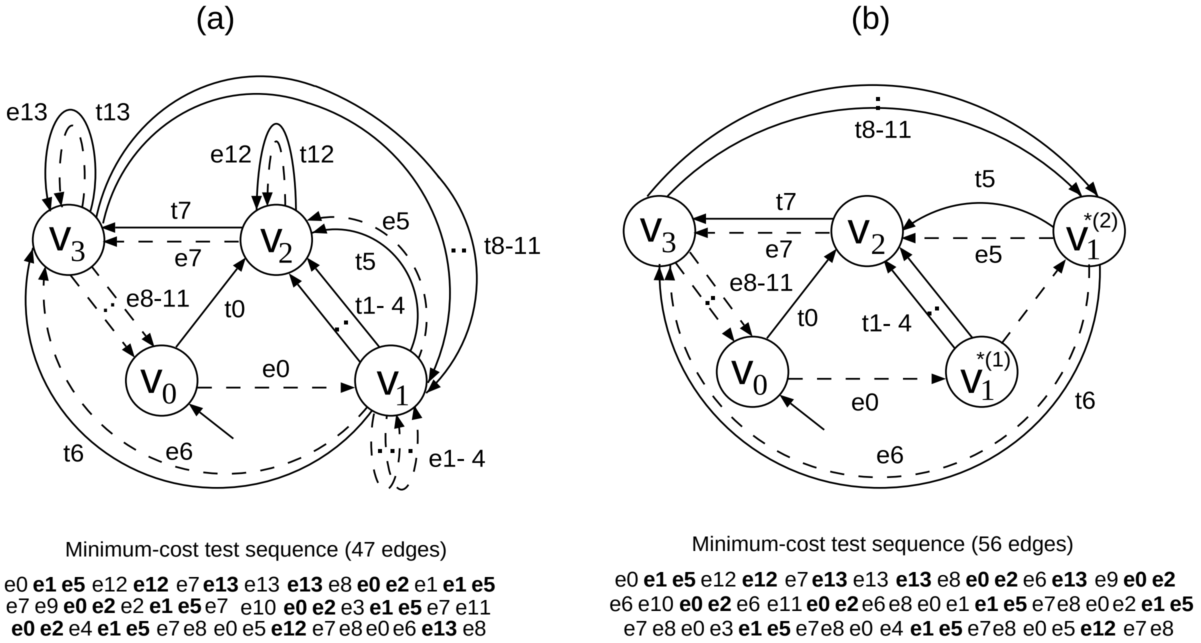

Figure 4: Minimum-cost test sequence without (a) and with (b) self-loop repetition constraint. Test and ghost edges appear in solid and dash lines, respectively.

self-loop repetition constraint:

$$\overbrace{e0, \mathbf{e1}, \mathbf{e5}}^{t0}, \overbrace{e12, \mathbf{e12}}^{t12}, \overbrace{e7, \mathbf{e13}}^{t7}, \overbrace{e13, \mathbf{e13}}^{t13}, \overbrace{e8, \mathbf{e0}, \mathbf{e2}}^{t8},$$
$$\overbrace{e1, \mathbf{e1}, \mathbf{e5}}^{t1}, e7, \overbrace{e9, \mathbf{e0}, \mathbf{e2}}^{t9}, \overbrace{e2, \mathbf{e1}, \mathbf{e5}}^{t2}, e7,$$
$$\overbrace{e10, \mathbf{e0}, \mathbf{e2}}^{t10}, \overbrace{e3, \mathbf{e1}, \mathbf{e5}}^{t3}, e7, \overbrace{e11, \mathbf{e0}, \mathbf{e2}}^{t11}, \overbrace{e4, \mathbf{e1}, \mathbf{e5}}^{t4},$$
$$e7, e8, e0, \overbrace{e5, \mathbf{e12}}^{t5}, e7, e8, e0, \overbrace{e6, \mathbf{e13}}^{t6}, e8 \qquad (1)$$

The test sequence contains 47 edges (the edges that are part of UIO sequences appear in bold).

The following part of the above test sequence

$$..., \overbrace{e8, \mathbf{e0}, \mathbf{e2}}^{t8}, \overbrace{e1, \mathbf{e1}, \mathbf{e5}}^{t1}, e7, ... \qquad (2)$$

requires that, after the IUT is brought into state v_1 via an edge $e0$, there should be enough time for at least three self-loop traversals before the IUT moves to another state. This part of the test sequence will fail after the second consecutive self-loop traversal. Since $max\_self(v_1) = 2$, the timeout edge $e6$ will be triggered instead of the required transition $e1$. The IUT will then move into v_3, thereby disrupting the test sequence. Further input/output exchanges are likely to fail even correct IUTs.

To avoid disruption of the above test sequence due to timeouts, edge $t1$ must be prevented from following $t8$. To meet this requirement, the graph of Figure 4 (a) is converted by the method outlined in Section 5 (see [26] for the detailed algorithm) to the graph shown in Figure 4 (b). As can be seen in Figure 4 (b), test edges $t8, t9, t10$, and $t11$ may be followed only by edges $t5, e5, t6$, and $e6$. To test $t1, t2, t3$, and $t4$, vertex v_1 must be entered through a ghost edge $e0$.

By limiting the number of consecutive self-loop traversals in a state to the maximum allowable, the following test sequence for the graph of Figure 4 (b) is obtained:

$$\overbrace{e0, \mathbf{e1}, \mathbf{e5}}^{t0}, \overbrace{e12, \mathbf{e12}}^{t12}, \overbrace{e7, \mathbf{e13}}^{t7}, \overbrace{e13, \mathbf{e13}}^{t13}, \overbrace{e8, \mathbf{e0}, \mathbf{e2}}^{t8},$$
$$\overbrace{e6, \mathbf{e13}}^{t6}, \overbrace{e9, \mathbf{e0}, \mathbf{e2}}^{t9}, e6, \overbrace{e10, \mathbf{e0}, \mathbf{e2}}^{t10}, e6,$$
$$\overbrace{e11, \mathbf{e0}, \mathbf{e2}}^{t11}, e6, e8, e0, \overbrace{e1, \mathbf{e1}, \mathbf{e5}}^{t1}, e7, e8, e0,$$
$$\overbrace{e2, \mathbf{e1}, \mathbf{e5}}^{t2}, e7, e8, e0, \overbrace{e3, \mathbf{e1}, \mathbf{e5}}^{t3}, e7, e8,$$
$$e0, \overbrace{e4, \mathbf{e1}, \mathbf{e5}}^{t4}, e7, e8, e0, \overbrace{e5, \mathbf{e12}}^{t5}, e7, e8 \qquad (3)$$

The test sequence contains 56 edges, an increase of 9 edges or almost 20%.

The test sequence in Figure 4 (b) is minimum-length given the self-loop constraint, although it is longer than the absolute minimum-length test sequence in Figure 4 (a). The maximum allowed number of self-loop traversals is not exceeded in any visit to a vertex, ensuring that the test sequence is realizable in a test laboratory.

6 Conclusion

This research has been motivated by UD's efforts to generate tests for MIL-STD 188-220B. In particular, optimization method based on the Rural Chinese Postman Problem is introduced to generate test sequences with timing constraints. Due to the active timers, the number of consecutive self-loops that can be traversed in a given state before a timeout occurs is limited. A test sequence must consider this constraint to be realizable in a test laboratory.

In the solution presented here, a series of augmentations are defined for the directed graph representation of the deterministic FSM model of a protocol. The resulting test sequence is proven to be of minimum-length while not exceeding the tolerable limit of consecutive self-loops at each state. In addition to the UIO sequences method, the solution described in this paper is also applicable to test sequences that use other state identification methods such as distinguishing sequences, and characterizing sequences.

Currently, this method is being implemented as a software tool and will be applied to MIL-STD 188-220B [10].

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 Trans. on Communications*, vol. 39, pp. 1604–1615, Nov 1991.

[2] B. S. Bosik and M. U. Uyar, "FSM-based formal methods in protocol conformance testing: from theory to implementation," *Computer Networks and ISDN Systems*, vol. 22, Sep 1991.

[3] W. Y. Chan and S. T. Vuong, "An improved protocol test generation procedure based on UIOs," in *Proc. ACM SIGCOMM*, Sep 1989.

[4] D. Lee and M. Yannakakis, "Principles and methods of testing finite state machines - a survey," *Proc. of the IEEE*, vol. 84, pp. 1090–1123, Aug 1996.

[5] R. J. Linn, "Conformance testing for OSI protocols," *Computer Networks and ISDN Systems*, vol. 18, pp. 203–219, 1990.

[6] R. E. Miller and S. Paul, "On the generation of minimal-length conformance tests for communication protocols," *IEEE/ACM Trans. on Networking*, vol. 2, pp. 116–129, Feb 1993.

[7] R. E. Miller and S. Paul, "Structural analysis of protocol specifications and generation of maximal fault coverage conformance test sequences," *IEEE/ACM Trans. on Networking*, vol. 2, pp. 457–470, Oct 1994.

[8] B. Sarikaya, G. von Bochmann, and E. Cerny, "A test design methodology for protocol testing," *IEEE Trans. Software Engineering*, vol. 13, pp. 518–531, May 1987.

[9] H. Ural and B. Yang, "A test sequence selection method for protocol testing," *IEEE Trans. on Communications*, vol. 39, no. 4, 1991.

[10] *Military Standard - Interoperability Standard for Digital Message Device Subsystems (MIL-STD 188-220B)*, Nov 1997.

[11] P. D. Amer, M. A. Fecko, A. S. Sethi, M. U. Uyar, T. J. Dzik, R. Menell, and M. McMahon, "Using Estelle to evolve MIL-STD 188-220," in *Proc. Estelle'98: Int'l Workshop on FDT Estelle*, (Evry, France), Nov 1998.

[12] J. K. Lenstra and A. H. G. R. Kan, "On general routing problems," *Networks*, vol. 6, pp. 273–280, 1976.

[13] K. K. Sabnani and A. T. Dahbura, "A protocol test generation procedure," *Computer Networks and ISDN Systems*, vol. 15, pp. 285–297, 1988.

[14] A. Bhattacharyya, *Checking Experiments in Sequential Machines*. New York, N.Y.: John Wiley & Sons, 1989.

[15] Z. Kohavi, *Switching and Finite Automata Theory*. New York, N.Y.: McGraw Hill, 1978.

[16] M. U. Uyar, M. A. Fecko, A. S. Sethi, and P. D. Amer, "Minimum-cost solutions for testing protocols with timers," in *Proc. Int'l Performance, Computing, and Communications Conf.*, (Phoenix, AZ), pp. 346–354, Feb 1998.

[17] AT&T 5E4 Generic Program, *AT&T 5ESS^{TM} Switch - ISDN Basic Rate Interface Specification*, Sep 1985.

[18] M. U. Uyar and M. H. Sherif, "Protocol modeling for conformance testing: Case study for the ISDN LAPD protocol," *AT&T Technical Journal*, vol. 69, Jan 1990.

[19] M. A. Fecko, P. D. Amer, A. S. Sethi, M. U. Uyar, T. Dzik, R. Menell, and M. McMahon, "Formal design and testing of MIL-STD 188-220A based on Estelle," in *Proc. MILCOM'97*, (Monterey, California), Nov 1997.

[20] Y. N. Shen, F. Lombardi, and A. T. Dahbura, "Protocol conformance testing using multiple UIO sequences," *IEEE Trans. on Communications*, vol. 40, pp. 1282–1287, Aug 1992.

[21] M. U. Uyar and A. T. Dahbura, "Optimal test sequence generation for protocols: the Chinese postman algorithm applied to Q.931," in *Proc. IEEE GLOBECOM*, pp. 68–72, Dec 1986.

[22] H. Ural and Y. Lu, "An improved method for test sequence generation," Tech. Rep. TR-90-12, Dept. of CSI, University of Ottawa, Mar 1990.

[23] M. S. Chen, Y. Choi, and A. Kershenbaum, "Minimal length test sequences for protocol conformance," in *Proc. First Network Management and Control Workshop*, (New York, NY), 1989.

[24] M. S. Chen, Y. Choi, and A. Kershenbaum, "Approaches utilizing segment overlap to minimize test sequences," in *Proc. PSTV X*, pp. 85–98, 1990.

[25] B. Yang and H. Ural, "Protocol conformance test generation using multiple UIO sequences with overlapping," in *Proc. ACM SIGCOMM'90*, pp. 118–125, 1990.

[26] M. U. Uyar, M. A. Fecko, A. S. Sethi, and P. D. Amer, "Test generation for protocols with timing constraints," Tech. Rep. TR-98-07, CIS Dept., University of Delaware, Newark, DE, 1997. (Submitted for publication).