FORMAL DESIGN AND TESTING OF MIL-STD 188-220A BASED ON ESTELLE

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

M. Ümit Uyar
Department of Electrical Engineering
City College of New York, New York, NY

Ted Dzik, Raymond Menell
US Army CECOM, Fort Monmouth, NJ 07703

Mike McMahon
ARINC, Inc., Shrewsbury, NJ 07702

Abstract

This paper describes the Estelle specification of MIL-STD 188-220A Intranet Layer as well as a methodology for generating test sequences for checking the conformance of a protocol implementation to its specification. The methodology for deriving test cases from an Estelle specification, which serves as input to test generation techniques, is presented. A Chinese postman tour *is used to determine a minimum-cost tour of the transition graph for various transition types. Finally, the paper discusses several controllability and optimization issues that need to be addressed in test cases generation for the intranet and datalink layers of MIL-STD 188-220A.*

1 Introduction

One of the University of Delaware's (UD) major contributions to the Advanced Telecommunications/Information Distribution Research Program (ATIRP) is the study of formal specification languages in the specification and testing of Army communication protocols. In 1989, Estelle was approved as one of two ISO International Standard Formal Description Techniques (FDT) for the specification of computer communication protocols [3] [10]. Based on communicating extended finite state machines, Estelle has a formal, mathematical, implementation-independent semantics. It is an expressive, well-defined, well-structured language that is capable of specifying distributed, concurrent information processing systems in a complete, consistent, concise and unambiguous manner. An Estelle specification aims at discovering and re solving ambiguities in the original English document that would cause interpretation problems for implementors. The Estelle specification as a model of a communication protocol then can be used as input to conformance test generation techniques. Since Estelle makes it possible to create a complete and unambiguous protocol model, the test cases generated from it achieve higher fault coverage than hand generated ones.

* "This work supported, in part, by the US Army Research Office Scientific Services Program administered by Battelle (DAAL03-91-C-0034), by the US Army Research Office (DAAL03-91-G-0086), and through collaborative participation in t he Advanced Telecommunications/Information Distribution Research Program (ATIRP) Consortium sponsored by the U.S. Army Research Laboratory under the Federated Laboratory Program, Cooperative Agreement DAAL01-96-2-0002." Dr. Uyar, a Research Professor with CCNY, is presently Visiting Associate Professor at University of Delaware.

2 Specifying MIL-STD 188-220A Data Link and Intranet Layers

For the past ten years, faculty and students at UD's Protocol Engineering Lab have been researching problems and developing tools that facilitate the general use of Estelle in designing and bringing communication network protocols to market. Most recently, we have been investigating one specific suite of protocols: MIL-STD 188-220A [8].

Originally designated 188-220 [9], this protocol suite was a joint services interoperability standard for digital message transfer device subsystems. It evolved into 188-220A to become the standard for interoperability of command, control, communications, computers, and intelligence (C4I) over Combat Net Radio (CNR). 188-220A is a key component of the Army Technical Architecture (ATA) for the digitized battlefield, and will likely become so in the Joint Technical Architecture (JTA). There are several synergistic efforts to design 188-220A to be complete, correct, unambiguous, and performing suitably well.

To ensure that the 188-220A standard is free from ambiguities which might cause implementation problems, we used Estelle to create an unambiguous specification of the Data Link Layer as specified in the 27Jul95 version of the standard [2]. This specification effort was divided into three subprojects specifying: Type 1: Connectionless (CL) Operation (unack'ed and coupled

0-7803-4249-6/97/$10.00

ack'ed); Type 2 Connection-mode Operation, and Type 4: Decoupled Ack'ed Connectionless Operation.

Most recently, UD's Protocol Engineering Lab has been working on specifying 188-220A's Intranet Layer which resides above Data Link and below IP. All radios tuned to the same radio channel make up an Intranet. Conceptually when any one radio transmits, its signal is physically broadcast to all of the others. However, due to geographic (or other) obstacles, some radios within an Intranet may be unable to communicate with others. These 'down' links and the Intranet topology in general may be temporary, particularly during highly dynamic battlefield situations. As a result, routing issues exist both within an Intranet and between Intranets (i.e., IP).

In the process of developing the Estelle specifications of the Data Link layer and, most recently, the Intranet Layer, a number of problems (> 50) in the original English specification have been documented. These problems have been reported back to the CNR Implementation Working Group, the official group that meets roughly bi-monthly at Ft. Monmouth, NJ and is responsible for the evolving 188-220 document.

Examples range from ambiguities such as:

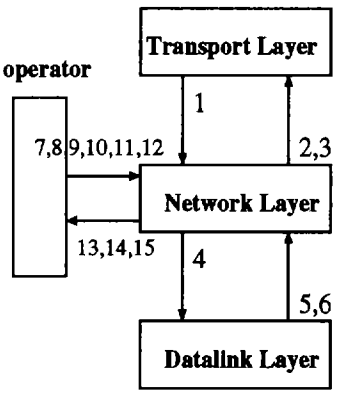
- "... a station shall wait for some period of time *bounded by the probability* of the remote ack time expiration."
- The Intranet Layer allows a station to enter *Quiet mode* whereas the Data Link layer refers to a station being in *response mode off*. It was unclear how these two terms differ, if at all.

to more serious examples of correctness/completeness such as:

- Intranet routing was originally defined based on spanning trees of the Intranet topology. However the draft standard's examples did not comply with the mathematical definition of a spanning tree.
- The phrase "may report to the higher layer protocol, and may initiate appropriate error recovery action" was added in several locations when the datalink layer identified an error condition such as a lack of acknowledgement after the maximum allowed number of retransmissions.

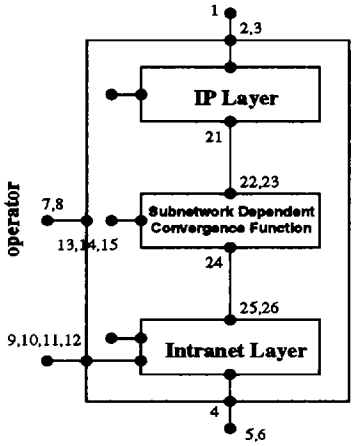
At the meeting in June 1997 the US Army Combat Net Radio Working Group approved including Delaware's Estelle specifications as an official part of the soon to be released MIL-STD 188-220B - Interoperability Standard for Digital Message Device Subsystems.

3 Estelle Specification of the Intranet Layer



1. NL-Unitdata.Req
2. NL-Unitdata.Ind
3. NL-Status.Ind
4. DL-Unitdata.Req
5. DL-Unitdata.Ind
6. DL-Status.Ind
7. OP-join-net.Req
8. OP-leave-net.Req
9. OP-response-mode
10. OP-relay-mode
11. OP-topology-precedence
12. OP-min-update-per
13. OP-unable-to-join-net.Ind
14. OP-join-net.Ind
15. OP-leave-net.Ind

Figure 1: Network Layer Interface



7. OP-join-net.Req
8. OP-leave-net.Req
9. OP-response-mode
10. OP-relay-mode
11. OP-topology-precedence
12. OP-min-update-per
13. OP-unable-to-join-net.Ind
14. OP-join-net.Ind
15. OP-leave-net.Ind
21. SNDCF-Unitdata.Req
22. SNDCF-Unitdata.Ind
23. SNDCF-Status.Ind
24. IL-Unitdata.Req
25. IL-Unitdata.Ind
26. IL-Status.Ind

Figure 2: Network Layer Architecture

Due to page limitations, we are unable to include actual full Estelle specifications in this paper. For the more detailed description of the semantics of Estelle specification components (communication channels, interactions, etc), the interested reader may consult our paper on Datalink Layer specification [2], or visit our www site at *http : //www.cis.udel.edu/ ~ amer/CECOM*. We instead present a brief overview of Network Layer architecture with a focus on the Topology Update function of the Intranet layer.

Figures 1 and 2 show the interface and general architecture of the Network Layer, which consists of Internet (IP) Layer, Subnetwork Dependent Convergence Func-

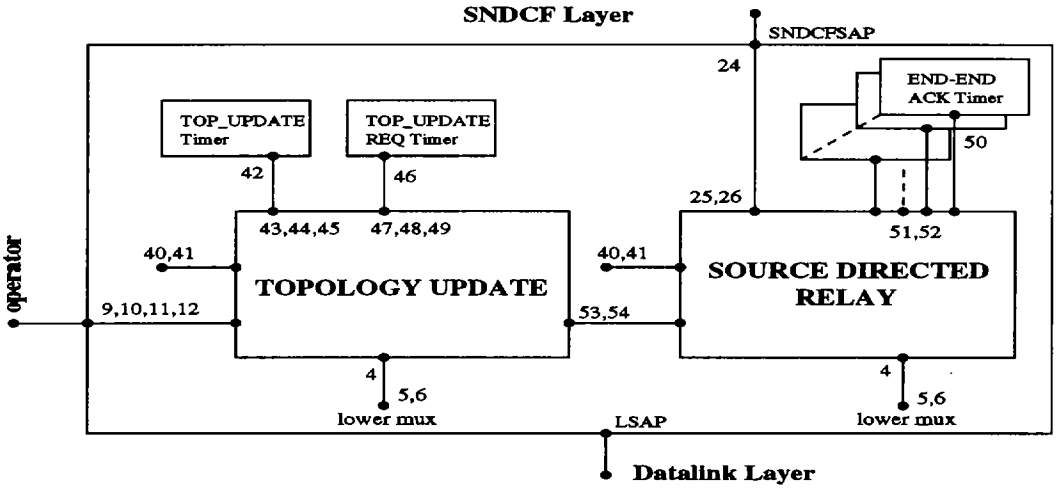


Figure 3: Intranet Layer Architecture

tion (SNDCF), and Intranet Layer. This represents the protocol stack at a single station, as well as an interface with "operator module" which can interact with several different layers in the stack. The operator module abstracts the link layer's interactions with both a human operator and a system management process.[1]

3.1 Intranet Layer Architecture

Figure 3 shows the internal structure of the Intranet Layer. The two main Intranet Layer functionalities, Source Directed Relay (SDR) and Topology Update exchange (TU), were encapsulated in separate component modules of the Intranet Layer module. This simplifies the design of the FSMs that model the entire layer, and also allows for generating test cases for each functionality separately.

The SDR module receives *IL_Unitdata_Req* messages through *SNDCFSAP* interaction point. It starts/stops a varying number of *END_END_ACK* timers, one for each IP packet that has been sent but not yet acknowledged. The TU module interacts with the SDR module by notifying it of any topology changes that take place dynamically. The TU module communicates with two timers: *Topology_Update Timer* and *Topology_Update_Request Timer*. The former is started after a topology update message is sent by the station. According to 188-220A, a station is not allowed to send another topology update message until the timer expires. The latter performs the same role for topology update request messages.

[1]Note that the numbers in Figures 1 through 3 refer to interactions, and are consistent throughout the figures (e.g., number 12 refers to *OP-min-update-per* in all three figures).

Both SDR and TU modules can send and receive messages from the datalink layer through their *lower_mux* interaction points - the messages from the two modules are multiplexed by the parent Intranet Layer module.

3.2 State Diagram and Transition Table for Topology Update

Each module in an Estelle specification is modeled as a set of communicating EFSMs (Extended Finite State Machine) after careful analysis of its behavior. Since in Section 4 we apply our test generation techniques to the TU module, in this paper we restrict ourselves to describing the corresponding EFSM for this module (Figure 4).

There are five states in the EFSM. The four active states are defined based on the status of the *Topology_Update Timer* and *Topology_Update_Request Timer*. Each timer may or may not be running at a given point in time - this gives four possible configurations. The timers' states determines the I/O behavior of the TU module, because a running timer prevents it from sending certain interactions. 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.

The final EFSM for the TU module consists of 5 states and 86 transitions.

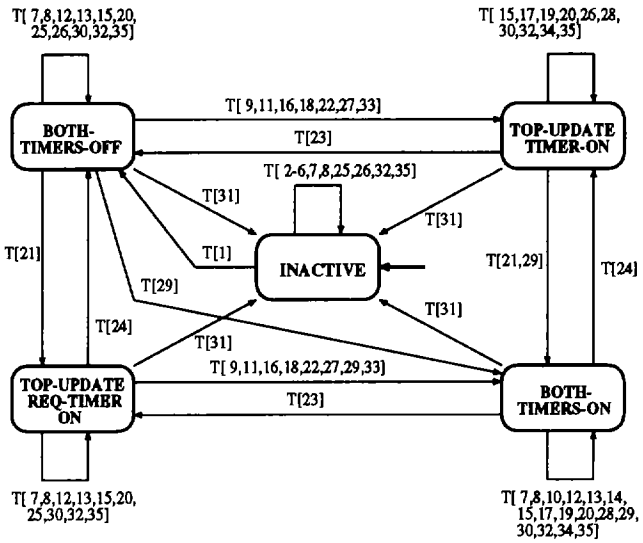


Figure 4: Extended FSM for Topology Update

4 MIL-STD 188-220A Test Case Generation

The Army Communications-Electronics Command (CECOM) Digital Integrated Laboratory (DIL) is responsible for certifying that all systems that were participating in Task Force XXI are interoperable. To perform this responsibility and future testing/certification for systems communicating over CNR, the DIL requires automated 188-220A protocol test tools. The CECOM Software Engineering Center is developing a Conformance Tester that automatically evaluates a 188-220A implementation identifying where it differs from the standard. This information can be used as a first step in the DIL's certification process, or to objectively categorize a 188-220A implementation to guide future implementations and standard evolution. The Conformance Tester capability will be used for the Army and the Army's joint requirements for years to come.

In support of this task, the UD's Protocol Engineering Lab is developing test scripts to be used by the 188-220A Conformance Tester. The test scripts specify a logical sequence of test steps that must be performed by the Conformance Tester to individually test the Data Link Layer (Types 1, 2 and 4 procedures) and Intranet Layer.

The test scripts will be used as input to the Conformance Tester which in turn will stimulate an Implementation Under Test (IUT), and assess responses to determine if the implementation has correctly implemented the protocols. Since it is known to be theoretically impossible to exhaustively test an implementation, checks should be made on those events that affect state/transition, boundary conditions, and stress points. The test scripts are to be structured as independent modular components to facilitate modifying and adding to the scripts in response to the continuing evolution of 188-220.

4.1 Black Box Testing

In conformance testing, the implementation under test (IUT) is viewed as a *black box* whose behavior is characterized by a set of observable actions, called *outputs*. Outputs are generated by applying a set of externally controllable inputs. A black-box model for a protocol is represented as a pure finite-state machine (Pure FSM) [1] [6] [4].

4.2 Transition types and testing order

For testing purposes, transitions in EFSM are divided into three groups:

- *valid transitions:* defined for the "normal" (or expected) behavior of the tested entity;
- *inopportune transitions:* inputs are semantically and syntactically correct, but arrive unexpectedly (or out of sequence) in a given state;
- *illegal input transitions:* inputs do not meet the syntax requirements and signal a faulty behavior from the far-end entity or from the transmission channel, e.g., invalid version number in intranet header.

We want to test each class of transitions separately. In *valid transitions* testing, in any given state each distinct input to an IUT corresponds to a state transition in the corresponding EFSM model. *Inopportune transitions* testing is typically performed separately from valid transitions testing. Since the number of possible inopportune transitions may be too large to be handled in practice, transitions in our specification may model only a subset of inopportune transitions.

Similarly, *illegal input transitions* testing is typically performed separately. Since there are infinitely many illegal inputs, we test the subset that includes only the most likely and/or the most important ones, based on guidance provided by 188-220A experts.

To test a single transition $s_i \longrightarrow s_j$, we must take the following steps:

- put an IUT into the state s_i;
- apply required input and compare the output(s) generated with those defined in the specification;
- verify that the new state of the EFSM is s_j (if practically possible).

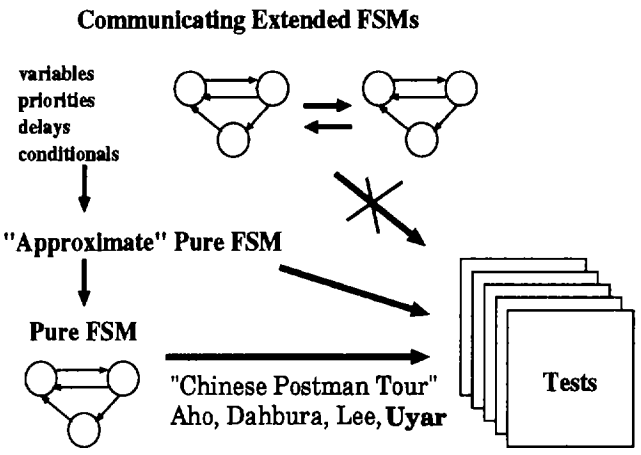


Figure 5: Test generation from Extended FSM

The entire methodology proposed for obtaining test cases consists of four phases.

4.3 Phase 1 - EFSM expansion

To apply existing test generation techniques to the system modeled as an EFSM, the first step in our approach is conversion of the Estelle specifications (which are EFSMs) into pure FSMs (Figure 5).

Suppose the EFSM contains a set of variables $VAR = \{v_1, v_2, ..., v_n\}$. Each variable has a corresponding set of values. The set of pairs $(variable, value)$ with one entry for each variable is called a *configuration.*

Given the set VAR, each state s from the original EFSM is replaced by a set of pairs $(s, configuration)$ for all possible configurations. Similarly, each original transition is replaced by a set of transitions according to the way an original transition changed/retained the value of each variable.

Ideally, the original Topology Update of EFSM of 5 states and 86 transitions should be converted to a pure FSM by using all possible configurations. This brute-force approach often leads to the well-known state and transition explosion problem [4]. The number of states and transitions in the expanded pure FSM may be infeasibly large. Therefore, based on our knowledge of the protocol, we compromise and consider only variables that are directly involved in Estelle *PROVIDED* clauses and that influence the output generated by the IUT.

This procedure converted the original EFSM to a FSM with 17 states and 612 transitions. This FSM, which was obtained from an EFSM with limited finite variable domains, is a closer approximation of an ideal pure FSM with all possible configurations. And fortunately this closer approximation does allow for some test case generation.

With complete testing theoretically not possible, initial practical concerns defined the test criterion that we need to satisfy as *"test each transition in the expanded FSM exactly once."*

4.4 Phase 2 - Valid transitions testing

The transitions in the expanded FSM were divided into 525 valid and 87 inopportune transitions. Each transition has a corresponding cost. Transitions that involve heavy computations and take longer to realize are assigned higher cost. Examples of costly transitions may be a timer's timeout and a reset transition.

The test method that we then use is a *Chinese Postman Tour* that includes all valid transitions [1] [6]. The method was developed at AT&T Bell Labs and applied successfully to testing various other communications protocols including ISDN switches, PBXs and terminals. The method produces a minimum-cost transition tour that starts and ends in the initial state, and that includes each valid transition at least once. For the graph representing an FSM to have a tour that covers each transition exactly once, the graph must be symmetric, which typically is not the case. Therefore, in the first step of the Chinese Postman Tour method, certain transitions are duplicated to make the graph symmetric at a minimum cost (such transitions are included in a tour more than once) [1] [6].

The tour length for our TU module is 815 transitions without state verification, and is estimated at 2,000 with state verification.

4.5 Phase 3 - Inopportune transition testing

All inopportune transitions are assumed to be *self loops* (i.e., start and end state are the same). To test each of them, we need to build a tour that visits each state at least once and tests a number of inopportune transitions while the IUT is in this state.

One of the problems that may occur during transition tour traversal, and that we are currently researching, is the inability to execute all self-loop transitions of a given state during one visit to this state. Since the internal structure of an IUT is unknown to the testers, the EFSM model cannot possibly capture all implementation details such as internal timers. Therefore, during testing it may not be possible to stay in a given state for the time necessary to execute all self-loop transitions (e.g., due to premature timeouts). The research is in progress to design an algorithm that builds a minimum-cost transition tour that solves this problem.

The expected tour length is 100 transitions without state verification, and is approximately 200 with state verification.

4.6 Phase 4 - Illegal input testing

Our current assumption is that all infinitely many illegal messages are self loops. In reality, some implementors may choose a different approach. For example, an IUT may go to an error state after detecting an illegal input.

As with inopportune transition testing, we need to visit each state at least once and test a number of illegal transitions while an IUT is in this state. We encounter the same theoretical problem due to timeouts - since it is likely that we will test many illegal messages in a given state, we may need to visit some states more than once.

5 Ongoing research issues

5.1 Observability and controllability

Ideally, testers should be able to generate every possible input message that is defined in the FSM for an IUT. Similarly, the output messages generated by an IUT should be observable by the testers.

One of the four [5] frameworks for OSI conformance testing is known as the *coordinated method* (Figure 6). A basic assumption of the coordinated method is that exposed interface exists below the IUT, but no exposed upper interface is necessary above the IUT. The interfaces serve as *points of control and observation* (PCOs); i.e., points at which a testing system can control inputs to and observe outputs from an IUT. In practice, testers may not have direct access to the interface(s) between the lower/upper tester and an IUT.

Based on the feedback that we got from CECOM's 188-220A Conformance Tester developers, we concluded that an explicit upper tester is unlikely to exist in the testing framework. This makes transitions that are fired by inputs coming from upper layers untestable. The same issue exists in testing of 188-220A datalink layer. However, the implementation will contain both intranet and datalink layers. The datalink layer IUT EFSM contains several transitions fired by inputs coming from the intranet layer. We are currently investigating the possibility of using feedback from the intranet layer as an implicit upper tester for the datalink layer IUT. Then the transition tour for the IUT will have transitions that generate outputs to the intranet layer. As response to these outputs the intranet layer generates inputs to the IUT (thereby simulating an upper tester).

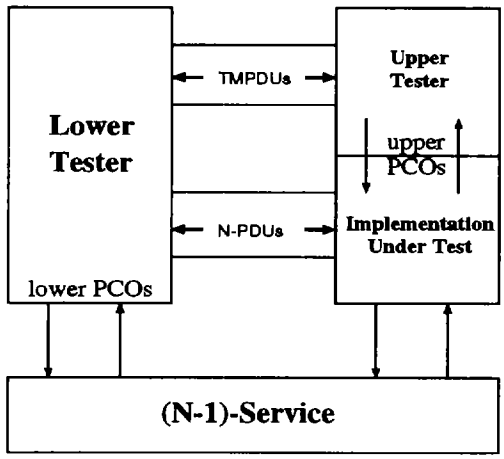


Figure 6: Coordinated method

The coordinated method seems to be the best approach for testing 188-220A under the constraints of the real testing environment. Stimuli to intranet layer serve here as inbound test management protocol data units (TMPDUs in Figure 6), which are used to coordinate upper and lower testers, with the lower tester being the master and the upper tester a slave.

An example of controllability problems may be a valid transition with the following *input/event* field in the Type 1 Service Datalink module transition table [8] [2]:

```
receive: DL-Unitdata.Req and
(intranet message type is IP packet) and
(coupled ack required) and
(multidestination packet)
```

To test this transition, the tester must be able to perform the following steps:

- generate required input - the *DL_Unitdata_Req* from the intranet layer;
- specify message type - *IP_packet*;
- specify multiple destination addresses in a message;
- specify an appropriate TOS field in a message, so that a coupled ack should be required;
- observe messages sent by IUT to physical layer;
- optionally verify if FSM changed its state in accordance with the specification.

In cases where the tester has no access to PCOs between the Intranet and Datalink layers, or cannot generate IP packets from upper layers, the transition becomes untestable, unless the Intranet layer is used as an implicit upper tester.

5.2 Limiting subtour length

A **subtour** is a sequence of transitions from a full transition tour that starts and ends in the initial state.

For practical testing reasons, it is important that the subtour length be limited. Because of limited controllability of the IUT, testers may be unable to execute an arbitrarily long transition sequence without resetting the IUT to the initial state. On the other hand, because setting up a new test is costly, neither do we want subtours that are as short as one or two transitions.

Currently, the *Chinese Postman Tour* is generated without addressing this problem. Therefore, some subtours are long (over 200 transitions), whereas several others are only one or two transitions long. We are investigating this problem with a view to generating a transition tour whose subtour lengths are more uniformly distributed.

5.3 Timing constraints during testing

Although existing test generation methods concentrate on optimizing the test sequence length, these methods place no restrictions on the order in which the tests can be applied to an IUT. As explained in Section 4.5, the most common restriction stems from an IUT's timers that, during actual testing, may limit the duration that the IUT can remain in a particular state. A methodology to generate test sequence under timing constraints was developed at UD's Protocol Engineering Lab. The results are available in [7].

6 Conclusion

In the effort to develop MIL-STD 188-220A, it is essential that the implementations of the protocol conform to its specification to achieve interoperability. In this paper we have described a technique for deriving test cases for conformance testing from the Estelle specification of the protocol. We used a powerful and efficient method of *Chinese postman tour* that allows automatic generation of minimum-cost test sequences.

We have also identified several research issues that are being addressed with FY97 support. These issues arise from the limited controllability and observability of an IUT and practical restrictions on the transition tour length and form.

Initially, the test generation method was applied to the Intranet Layer of MIL-STD 188-220A. Our immediate goal, which is consistent with suggestions by CECOM's Conformance Tester developers, is to produce test cases for the Data Link Layer Types 1-4.

The views and conclusions contained in this document are those of the authors and should not be interpreted as representing the official policies, either expressed or implied, of the Army Research Laboratory or the U.S. Government.

References

[1] A.V. Aho, A.T. Dahbura, D. Lee, M.U. Uyar. *An Optimization Technique for Protocol Conformance Test Generation Based on UIO Sequences and Rural Chinese Postman Tours.* IEEE Transactions on Communications, 39(11), Nov 1991.

[2] P. Amer, G. Burch, A. Sethi, D. Zhu, T. Dzik, R. Menell, M. McMahon. *Estelle Specification of MIL-STD 188-220A DLL.* Proc MILCOM 96, Oct 1996.

[3] S. Budkowski, P. Dembinski. *An Introduction to Estelle: A Specification Language for Distributed Systems.* Computer Networks and ISDN Systems, 14(1), 1987, 3-24.

[4] D. Lee, M. Yannakakis. *Principles and Methods of Testing Finite State Machines.* Proc. IEEE, Aug 1994, 1090-1123.

[5] R.J. Linn. *Conformance Testing for OSI protocols.* Computer Networks and ISDN Systems, 18(3), 1989/1990, 203-219.

[6] M.U. Uyar, A.T. Dahbura. *Optimal Test Sequence Generation for Protocols: The Chinese Postman Algorithm Applied to Q.931.* Proc. IEEE Global Comm. Conf., 1986, 68-72.

[7] M.U. Uyar, M.A. Fecko, A.S. Sethi, P.D. Amer. *Minimum-Cost Solutions for Testing Protocols with Timers.* Technical Report No. 97-17, CIS Dept., University of Delaware, Newark, DE, 1997. Submitted for publication.

[8] *Military Standard - Interoperability Standard for Digital Message Device Subsystems (MIL-STD 188-220A), 27Jul95.*

[9] *Military Standard - Interoperability Standard for Digital Message Device Subsystems (MIL-STD 188-220), 7May93.*

[10] *ISO International Standard 9074: Estelle - A Formal Description Technique Based on an Extended State Transition Model.* Information Processing Systems - Open System Interconnection, 1989.