Interoperability Test Suite Derivation for the ATM/B-ISDN Signaling Protocol

Jaehwi Shin and Sungwon Kang
Korea Telecom Research & Development Group
Sochogu Umyundong 17, Seoul 137-792, Korea
e-mail: {jhshin, kangsw}@sava.kotel.co.kr

Abstract

In this paper, we propose a test derivation method suitable for testing interoperability for the class of communication protocols like the ATM/B-ISDN signaling protocol. For this, we begin by defining the notion of interoperability test case. Next we select an effective interoperability test architecture by carefully comparing advantages and disadvantages of three candidate test architectures. Then we present an interoperability test suite derivation method based on the definition of interoperability test case. For that, we give an algorithm for deriving from a given pair of FSMs a composite FSM and an interoperability test suite in parallel. In addition, occurrence orderings of output messages for all interoperability test cases are analyzed and their regularity is exploited to reduce the length of interoperability test cases.

Keywords

Interoperability testing, Test architecture, Signaling protocol, Test optimization

1 INTRODUCTION

Conformance testing that checks whether an implementation is correct with respect to the relevant standards has limitation in ensuring interoperability. Thus, even a pair of two conforming implementations can fail to interoperate [RafC 90][APRS 93]. Two main causes of non-interoperation of conforming implementations are ambiguity of protocol standards and incompleteness of conformance testing. Thorough validation of standards is necessary to prevent the former cause of non-

A. Petrenko et al. (eds.), *Testing of Communicating Systems*

interoperation whereas some sort of direct testing of interoperation is necessary to overcome the latter cause of non-interoperation.

There are some research work on interoperability testing done in the past. [RafC 90] deals with interoperability test suite generation based on reachability analysis. However, it is not based on a rigorous definition of interoperability and considers only the case where lower testers exist between two Implementations Under Test (IUTs). [VerB 93] expounds experiences with interoperability testing of FTAM protocol that uses only a single tester between two IUTs and thus has limited capability. [APRS 93] derives conformance test suite and interoperability test suite separately and later manually combines them to reduce the number of conformance test cases. [KanK 97] develops methods for systematically dealing with symmetric communication protocols but uses an interoperability test architecture that does not observe the interface behavior between two IUTs.

In this paper, we propose a test derivation method suitable for interoperability testing of the class of communication protocols like ATM/B-ISDN signaling protocol. Although there is no consistent agreement on definition of interoperability and methodology of interoperability test derivation, for systematic derivation of interoperability test suite it is essential to first state what precisely is meant by interoperability testing. Thus, we first discuss the meaning of interoperability testing and the conditions for a test case to be an interoperability test case.

Next, we investigate interoperability test architectures. By carefully comparing advantages and disadvantages for three candidate interoperability test architectures, we single out among them the most effective test architecture. The selected test architecture has testers for the sides of IUTs facing the external environment and a monitor between two IUTs so that it can check the internal interface between the two IUTs.

Next, we present an interoperability test suite derivation method based on the definition of interoperability testing. For that, we give an algorithm for deriving from a given pair of Finite State Machines(FSMs) a composite FSM and an interoperability test suite in parallel. By applying the algorithm to the ATM signaling protocol as defined by ATM Forum UNI 3.1 specification[AF UNI] and ATM Forum PNNI specification[AF PNNI1][AF PNNI2], 26 interoperability test cases are derived.

Since communication protocols have multiple interfaces in general, when messages of a test case are output concurrently at interfaces, it is difficult to capture correct orderings efficiently in test case description. Thus, we analyze the occurrence orderings of output messages for all derived interoperability test cases and use the regularity to reduce the length of interoperability test cases.

This paper is organized as follows: Section 2 introduces ATM signaling protocol defined by UNI 3.1 and PNNI and describes FSMs for them. In Section 3, we state the conditions for a test case to constitute interoperability test case and classify interoperability test cases according to their message interaction patterns. In Section 4, we consider three test architectures for interoperability testing of ATM signaling protocol and select the most appropriate test architecture by comparing their merits and demerits. In Section 5, based on the definition of interoperability

test case and the selected test architecture, we describe a method for deriving interoperability test suite from specifications given in FSM model. In Section 6, we show the application results. Section 7 concludes the paper.

2 OVERVIEW OF THE ATM SIGNALING PROTOCOL

Switch as equipment implementing ATM signaling protocol can be classified into local switch that handles subscriber call and transit switch that performs call transfer function. Thus, there are three interconnection combinations for a pair of switches: i.e., (local switch, local switch), (local switch, transit switch), and (transit switch, transit switch).

Though a respective interoperability test suite can be obtained for each combination using the same method to be shown later in this paper, we only consider interoperation of two local switches.

In local switches, ATM call progresses through both User-Network Interface(UNI) and Network-Network Interface(NNI). We consider ATM Forum UNI 3.1 Specification[AF UNI] and ATM Forum PNNI Specification[AF PNNI1][AF PNNI2] for UNI and NNI specifications for the local switch. The UNI 3.1 specification provides functions and procedures for the ATM user and the ATM network to access each other. The PNNI specification provides functions and procedures to establish and clear calls through ATM networks.

Among the messages for UNI 3.1 signaling layer, SETUP, CALL PROCEEDING, CONNECT, and CONNECT ACKNOWLEDGEMENT messages are used to establish an ATM call and RELEASE and RELEASE COMPLETE messages are used to clear the ATM call. Figure 1 is the FSM model of the Signaling layer of UNI 3.1 specification.

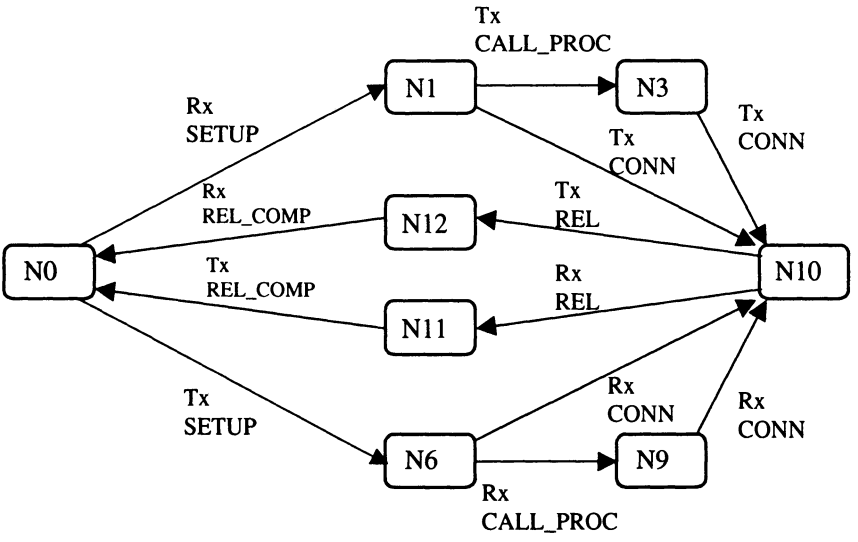


Figure 1 FSM model for UNI 3.1 network-side signaling protocol.

Among the messages for PNNI signaling layer, SETUP, CALL PROCEEDING, ALERTING and CONNECT messages are used to establish ATM calls and RELEASE and RELEASE COMPLETE messages are used to clear ATM calls. Figure 2 is the FSM model of the Signaling layer of PNNI specification.

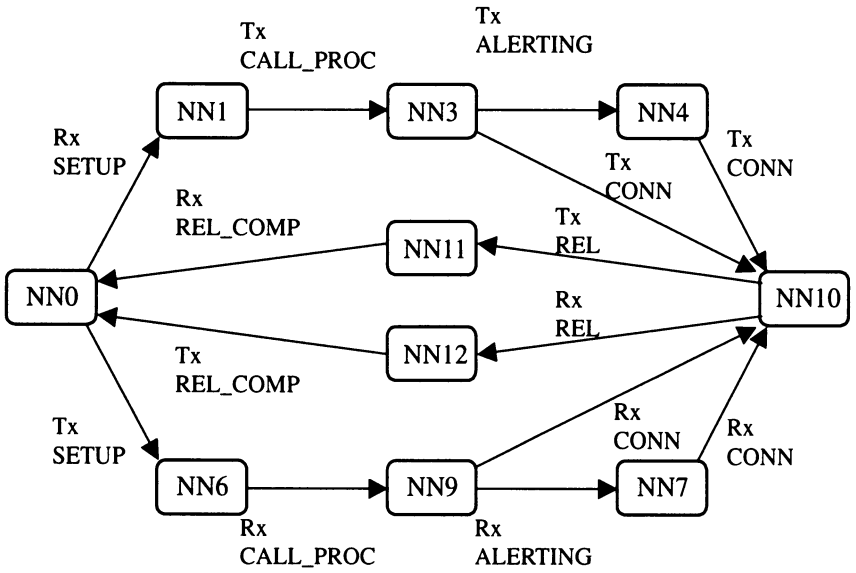


Figure 2 FSM model for PNNI signaling protocol.

3 DEFINITION OF INTEROPERABILITY TESTING AND TYPES OF INTEROPERABILITY TEST CASES

As already mentioned in Introduction, there is no established consistent framework for interoperability testing either in academy or in industry. ISO and ETSI define interoperability briefly as follows:

·ISO/IEC JTC1 DTR-10000[ISO DTR10000]: The ability of two or more systems to exchange information and to mutually use the information that has been exchanged.

·ETSI ETR 130[ETSI ETR130]: Protocol Interoperability: the ability of a *Distributed System* to interchange PDUs via the *Communication Platform.*

A common characteristic of interoperability testing that the above definitions point to is as follows: Compared to the conformance testing that checks whether an implementation comforms to its specification, *interoperability testing* checks two interconnected equipment for correct operation. That is, interoperability testing checks correct response of two IUTs when an external input is applied to the interconnected IUTs whereas conformance testing checks for correct response of IUT when an input is applied to it. This distinction can be clearly seen with Figure 3 and Figure 4. The viewpoint of regarding the response of implementations for a

single input as constituting a test case is called 'Single Stimulus Principle'[AraS 92]. It lets us have the most fine-grained test cases and has been adopted in [LuBP 94] and [KanK 97].

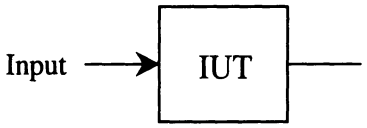
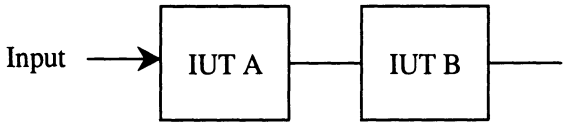
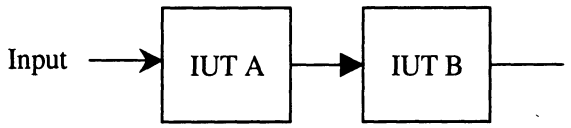


Figure 3 Conceptual diagram of conformance testing.



(a) Interconnection of two IUTs and application of an input



(b) Minimal requirement for an interoperability test case

Figure 4 Conceptual diagram of interoperability testing.

On the other hand, not all applications of an external input message lead to an interoperability test case. According to the above definition of interoperability, an *interoperability test case* should (1)check whether two IUTs exchange information and (2)use the information that has been exchanged. When an external input is applied to the IUT A as depicted in Figure 4(a), at least one message must be transferred from IUT A to IUT B in order to meet the requirements (1) and (2).[1] Therefore, in this paper, we consider a sequence of interactions as one interoperability test case if the interactions begin at a stable state and end at a stable state and satisfies Figure 4(b) when an external input is applied to a system composed of two IUTs.

There can be various patterns of message exchanges that satisfy Figure 4(b). Analyzing ATM signaling protocol composed of UNI 3.1 and PNNI, we can classify all possible patterns of such interoperability test cases as in Figure 5. This classification is used in Section 6 to minimize the length of test cases.

In Figures 3, 4 and 5, quadrangle arrows indicate external input messages and triangle arrows indicate output messages.

1 In this section, to simplify discussion, we consider only the cases where external inputs are applied to IUT A.

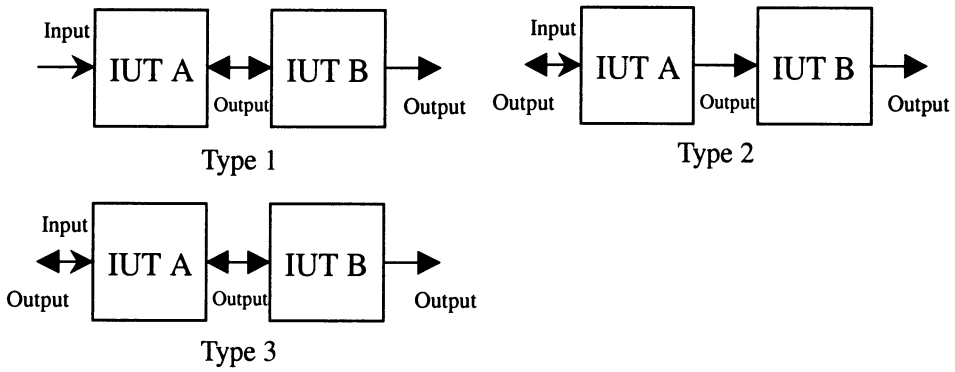


Figure 5 Type classification of interoperability test cases.

4 SELECTION OF THE INTEROPERABILITY TEST ARCHITECTURE

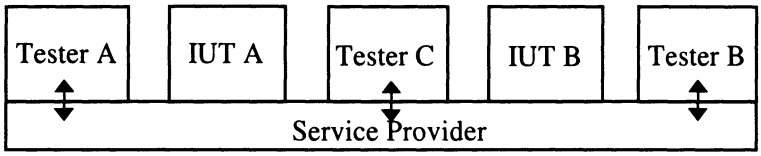
It is very important to select an appropriate test architecture since test architecture affects the effectiveness of the whole process of test suite derivation and efficiency of the resulting test suite. For this, we consider as candidates the three test architectures in Figure 6. The arrowhead in the figure indicates functionality. So a double-headed arrow represents a *Point of Control and Observation*(PCO) and indicates that the tester has both observability and controllability. A single-headed arrow as in Figure 6(b) represents a *Point of Observation* (PO) and indicates that the tester has only observability and hence is a monitor.

Now, we consider interoperability test case execution under each of the three test architectures. We select a Type 1 test case for the purpose of illustration.

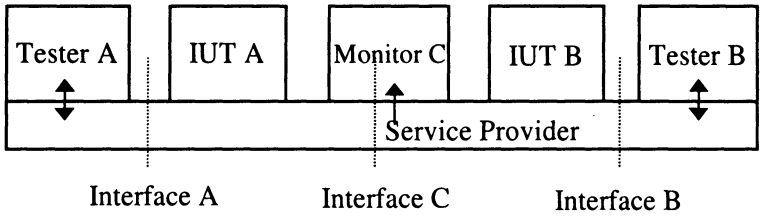
- Test Architecture I: Test starts by Tester A sending a message to IUT A. Then, IUT A sends a message to IUT B and Tester C only reads the message. As this is a Type 1 test case, IUT B sends messages to IUT A and Tester B, and these messages are read by Tester C and Tester B, respectively. After the message exchange finishes, Tester A, Tester B, and Tester C send STATUS ENQUIRY messages to verify the reached states of IUT A and IUT B. In particular, Tester C sends STATUS ENQUIRY in two directions, one to IUT A and the other to IUT B.

- Test Architecture II: Test starts by Tester A sending a message to IUT A. Then, IUT A sends a message to IUT B and Monitor C only reads the message. As this is a Type 1 test case, IUT B sends messages to IUT A and Tester B, and these messages are read by Monitor C and Tester B, respectively. After the message exchange finishes, Tester A and Tester B send STATUS ENQUIRY messages to verify the reached states of IUT A and IUT B. Unlike Test Architecture I, Monitor C does not send a STATUS ENQUIRY.

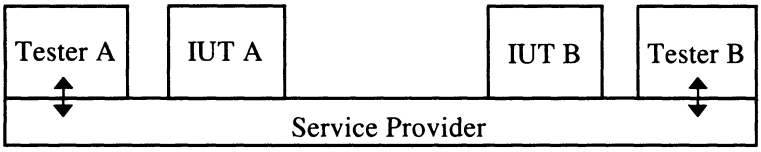
- Test Architecture III: Test starts by Tester A sending a message to IUT A. Then, IUT A sends a message to IUT B but this message is not read. IUT B sends messages to IUT A and Tester B, and only the message sent to Tester B is read. After the message exchange finishes, Tester A and Tester B send STATUS ENQUIRY messages to verify the reached states of IUT A and IUT B. In this test architecture, interface C is not examined at all.



(a) Test Architecture I



(b) Test Architecture II



(c) Test Architecture III

Figure 6 Interoperability test architectures.

Based on the above test execution outlines, comparison results for the three test architectures can be summarized as in Table 1. We consider which test architecture is most appropriate for interoperability testing of local switch ATM signaling protocol. First, Test Architecture III has the lowest analytic power and cannot directly analyze behavior at PNNI. Although Test Architecture I is most powerful, the structure of derived test cases are also most complex, which makes it difficult to adopt it for practical use. Therefore, we select Test Architecture II for interoperability test suite derivation of the ATM signaling protocol.[2]

2 Although Test Architecture III has limited error detection capability, Architecture III is very efficient when test resource(such as the number of links, the number of processors, etc) is limited or handling of the link between IUT A and IUT B is difficult.

Table 1 Comparison of interoperability test architectures

	Test Architecture I	Test Architecture II	Test Architecture III
Character-istics	• Generic test architecture • Use Tester at interface C • Detection of both message error and state transfer error at interface A, B and C	• Use Monitor instead of Tester at interface C. • Detection of both message error and state transfer error at interface A and B • Detection of message error at interface C	• Use neither Tester nor Monitor at interface C • Detection of both message error and state transfer error at interface A and B • Errors at interface C are not detected
Merits and Demerits	• Most difficult to realize • Maximum test coverage	• Simpler to realize than Test Architecture I • High test coverage	• Simplest to realize • Minimum test coverage

5 DERIVATION OF INTEROPERABILITY TEST SUITE

When two IUTs (IUT A and IUT B) implementing specifications A and B are tested with Test Architecture II, an interoperability test suite can be derived as follows.

Interoperability Test Suite Derivation Procedure

Step 1. Derivation of FSM M_A and M_B for the control flow of the specifications A and B.

Step 2. Composition of FSM M_A and M_B.

Step 3. Derivation of interoperability test case skeletons by selecting among all the transitions of the composite FSM only the transitions that satisfy the definition of interoperability test.

Step 4. Derivation of interoperability test cases by parameterization of interoperability test case skeletons. (One or more interoperability test cases can be obtained by parameterization of one skeleton.)

We define FSM model in Section 5.1 and present an Algorithm for Step 2 in Section 5.2.

5.1. FSM Model

We mentioned the necessity of representing a protocol specification with an FSM to formally describe the derivation process of composite FSM. Let M be the FSM for the protocol specification of an IUT. Then M can be represented as follows:

Definition 1 A FSM is a 5-tuple $<\mathbf{S}, S_0, \mathbf{L_{in}}, \mathbf{L_{out}}, \mathbf{Tr}>$ where:

(1) $\mathbf{S}=\{S_0, S_1, ..., S_{n-1}\}$ is a set of states of M,

(2) $S_0 \in \mathbf{S}$ is the initial state,

(3) $\mathbf{L_{in}}=\{v_1, v_2, ..., v_m\}$ is a set of input symbols,

(4) $\mathbf{L_{out}}=\{u_1, u_2, ..., u_k\}$ is a set of output symbols, and

(5) $\mathbf{Tr} \subseteq \{S—v/U \rightarrow S' \mid S,S' \in \mathbf{S} \wedge v \in \mathbf{L_{in}} \wedge U \subseteq \mathbf{L_{out}}\}$ is a set of transitions.

Bold letters in Definition 1 represent sets. " S—v/U→S' " describes a transition where S, S', v, and U represent, respectively, starting stable state, the final stable state, an input symbol and a set of output symbols.

Let Π be the composite FSM describing the combined behavior of two FSMs in Test Architecture II of Figure 6(b). Let M_A and M_B be the FSMs describing specification A and B, respectively. In communication protocols, it is usually the case that at most two messages are output in response to a stimuli from the external environment. Thus we assume that, for M_A and M_B, U in Definition 1 consists of at most two output symbols, each for different interfaces. Then Π can be defined as follows:

Definition 2 The composite FSM Π is a 5-tuple $< \mathbf{S_\Pi}, S_{\Pi,0}, \mathbf{L_{\Pi,in}}, \mathbf{L_{\Pi,out}}, \mathbf{Tr_\Pi}>$ where:

(1) $\mathbf{S_\Pi}=\{S_{\Pi,0}, S_{\Pi,1}, ..., S_{\Pi,n-1}\}$ is a set of global state, and $S_{\Pi,i}=(S_A, S_B)$, where S_A and S_B are states of M_A and M_B respectively.

(2) $S_{\Pi,0} \in \mathbf{S_\Pi}$ is the initial state.

(3) $\mathbf{L_{\Pi,in}}=\{v_{\Pi 1}, v_{\Pi 2}, ..., v_{\Pi m}\}$ is a set of input symbols. It consists of input messages for the interface A and B.

(4) $\mathbf{L_{\Pi,out}}=\{U_{\Pi 1}, U_{\Pi 2}, ..., U_{\Pi k}\}$ is a set of output symbols. Each element of $\mathbf{L_{\Pi,out}}$ is of the form [u1, u2, u3, u4], where u1, u2$\in \mathbf{L_{out,A}}$ and u3, u4$\in \mathbf{L_{out,B}}$. In more detail, u1, u2, u3, and u4 represent messages transmitted by M_A via interface A, M_A via interface C, M_B via interface C, and by M_B via interface B.

(5) $\mathbf{Tr_\Pi} \subseteq \{S_\Pi—v_\Pi/U_\Pi \rightarrow S_\Pi' \mid S_\Pi, S_\Pi' \in \mathbf{S_\Pi} \wedge v_\Pi \in \mathbf{L_{\Pi,in}} \wedge U_\Pi \in \mathbf{L_{\Pi,out}}\}$ is a set of transitions.

5.2. Interoperability Test Suite Derivation Algorithm

Given FSMs M_A and M_B describing specifications A and B, respectively, the composite FSM Π and the interoperability test suite IOPTS can be derived in parallel by the following Algorithm:

Algorithm: (Interoperability Test Suite Derivation)

Input: FSMs $M_A=<\mathbf{S_A}, S_{A,0}, \mathbf{L_{in,A}}, \mathbf{L_{out,A}}, \mathbf{Tr_A}>$ and $M_B=<\mathbf{S_B}, S_{B,0}, \mathbf{L_{in,B}}, \mathbf{L_{out,B}}, \mathbf{Tr_B}>$. Let $\mathbf{L_{in,A,E}}$ be the subset of $\mathbf{L_{in,A}}$ consisting of messages on interface A, and $\mathbf{L_{in,B,E}}$ be

the subset of $\mathbf{L}_{in,B}$ consisting of messages of interface B.

Output: FSM $\Pi=<\mathbf{S}_{\Pi}, S_{\Pi,0}, \mathbf{L}_{\Pi,in}, \mathbf{L}_{\Pi,out}, \mathbf{Tr}_{\Pi}>$ and Interoperability Test Suite **IOPTS** $\subseteq \mathbf{Tr}_{\Pi}$.

$\mathbf{L}_{\Pi,in}:=\mathbf{L}_{in,A,E} \cup \mathbf{L}_{in,B,E}$
$S_{\Pi,0}:=(S_{A,0},S_{B,0})$
$\mathbf{S}_{\Pi}:=\{S_{\Pi,0}\}$
$\mathbf{Tr}_{\Pi}:=\{\}$
IOPTS:={}
NEW:=$\mathbf{S}_{\Pi}$
while NEW $\neq \emptyset$ **do begin**
    gs_i:=choose-any(NEW)
    NEW:= NEW − {gs_i}
    Input := $\mathbf{L}_{\Pi,in}$
    while Input $\neq \emptyset$ **do begin**
        v:=choose-any(Input)
        Input :=Input − {v}
        U_out:=[, , ,]
        gs_f:=state(gs_ilv)
        U_out:=output(gs_ilv)
        $\mathbf{L}_{\Pi,out}:= \mathbf{L}_{\Pi,out} \cup \{U\_out\}$
        $\mathbf{Tr}_{\Pi}:=\mathbf{Tr}_{\Pi} \cup \{gs\_i — v/U\_out \rightarrow gs\_f\}$
        if (gs_f $\notin \mathbf{S}_{\Pi}$)
        then begin
            $\mathbf{S}_{\Pi}:=\mathbf{S}_{\Pi} \cup \{gs\_f\}$
            NEW:=NEW $\cup$ {gs_f}
        end
        if (U_out[2] $\neq$ NULL $\vee$ U_out[3] $\neq$ NULL)
        then
            IOPTS:=**IOPTS** $\cup \{gs\_i — v/U\_out \rightarrow gs\_f\}$
    end while
end while

In the algorithm, choose-any(A) is a function that chooses an arbitrary element of A, and state(gs_ilv) and output(gs_ilv) represent, respectively, the final stable state and the output messages when an input v is applied to a stable state gs_i. U_out consisting of output messages is a four dimensional vector such that the meaning of each element is the same as that of the element of $\mathbf{L}_{\Pi,out}$ in Definition 2.

The transition set $\mathbf{Tr}_{\Pi}$ obtained by the Algorithm includes the initial stable state before a transition occurs, input at that state, output by the input and the final stable state reached by the transition. However, $\mathbf{Tr}_{\Pi}$ contains conformance test cases in addition to interoperability test cases. **IOPTS** is the pure interoperability test suite without conformance test cases which have been filtered out by applying the second if-clause.

6 APPLICATION TO THE ATM SIGNALING PROTOCOL

For Step 1 of the interoperability test suite derivation in Section 5, we derived the FSM in Figure 7 from the two FSMs in Figures 1 and 2 that describes the entire behavior of ATM signaling protocol.[3] [4]

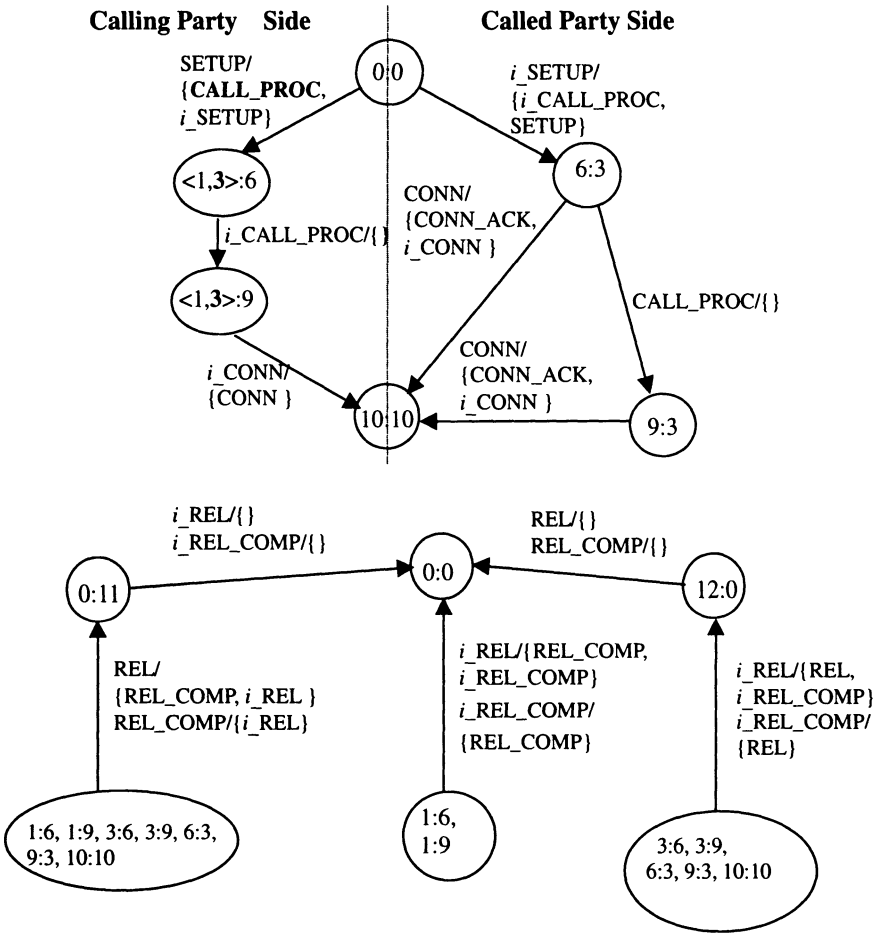


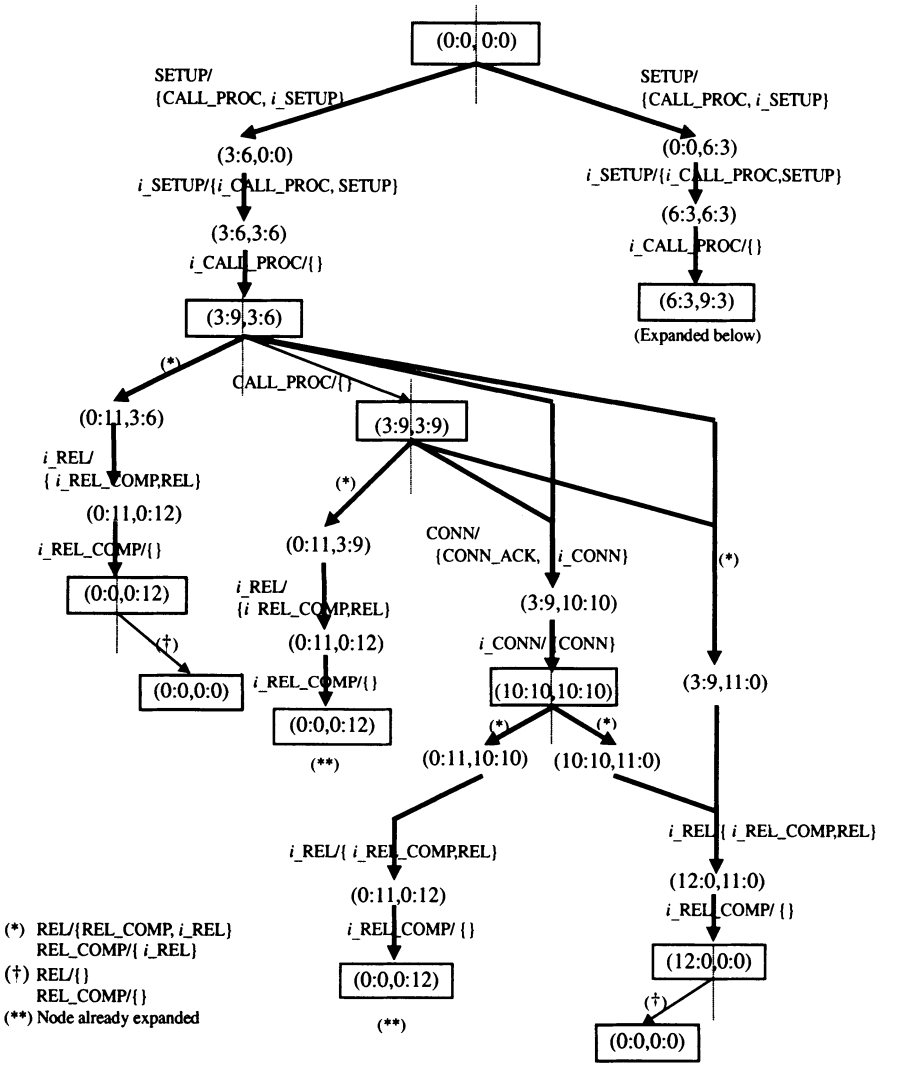
Figure 7 FSM model for the ATM signaling protocol.

Figure 8 depicts composite FSM obtained by applying the Algorithm to the FSM in Figure 7 where $M_A = M_B$. In Figure 8, a global state, for example, (3;9,3;6)

[3] Specification for the entire behavior of the signaling protocol for the ATM local switches has not been published. Thus Figure 7 need be inferred from UNI 3.1 specification and PNNI specification.

[4] Sending of CALL_PROC in response to SETUP by the UNI 3.1 network-side is an optional feature. It is assumed here that IUT A and IUT B both send CALL_PROC.

indicates that UNI state and NNI state of IUT A are N3 and NN9, respectively, and UNI state and NNI state of IUT B are N6 and NN3, respectively. Transitions in bold line in Figure 8 are interoperability test cases for the ATM signaling protocol.



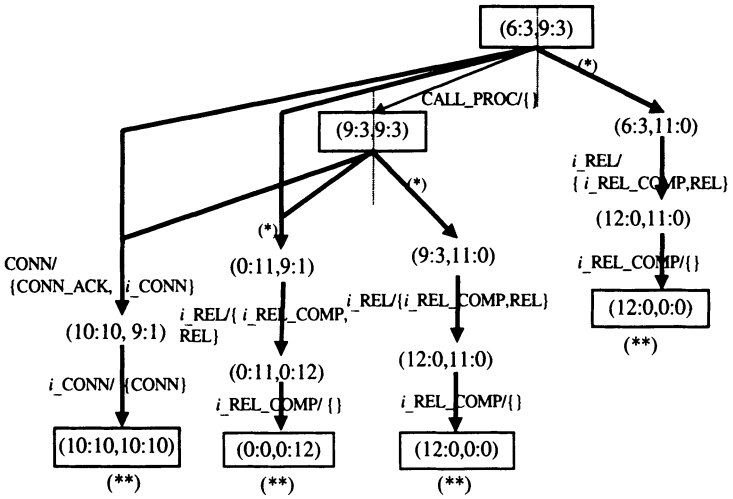


Figure 8 Composite FSM Π for the ATM signaling protocol.

6.1. Test Suite Structure

Derived test cases are grouped according to the call processing function in Figure 9. As seen from Appendix 1, both of /ESTABLISH/A_TO_B/ and /ESTABLISH/B_TO_A/ groups contain 3 test cases, and both of /CLEARING/A_TO_B/ and /CLEARING/B_TO_A/ groups contain 8 test cases. In the figure, A_TO_B and B_TO_A mean that ATM call direction is, respectively, from Tester A to Tester B and the reverse direction. Test group /CLEARING/COMMON/ has 4 test cases and is to test call clearing after a call has been established regardless of the call direction. In Appendix 1, a suffix attached to the end of an input message indicates the direction of the message. For example, REL_A represents RELEASE is transferred from Tester A to IUT A. The rightmost column of the table is the type of each test case assigned according to Figure 5.

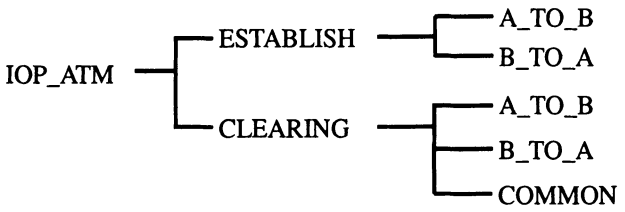


Figure 9 The interoperability test suite structure for ATM signaling protocol.

6.2. Ordering of Output Messages

The output from each test case of the derived interoperability test suite consists of 3 or 4 messages and is expressed as [α, β, γ, δ] using a vector notation. To describe

it using (sequential) TTCN, we must consider all possible orderings of expected output messages.[5] As an example, we must allow 4!=24 cases for 4 output messages if there is no constraint on the message sequence. However, output messages may have certain causality relations depending on the protocols at hand. By the causality relation, the actual number of message orderings allowed can be significantly reduced, which implies we can reduce the length of test cases when describing them using sequential TTCN.

For example, if we express the output of TC1 which is represented by (0:0,0:0) — $SETUP_A$/[CALL_PROC, *i*_SETUP, *i*_CALL_PROC, SETUP] → (3:9,3:6) by [α, β, γ, δ], then there are two causal relations of β→γ and β→δ according to the ATM signaling protocol. Thus, all possible orderings of output messages for TC1 can be represented by α→β→(γ,δ) and β→(α,γ,δ).

If we calculate orderings of output messages for all interoperability test cases in a similar way, we can see that orderings are determined by the type of an individual test case and the direction of the input message as shown in Table 2.

Table 2 Output message ordering patterns

Types	Sender of the input message	Output messages	Possible orderings of output messages
1	Tester A	[, β, γ, δ]	β→(γ,δ)
	Tester B	[δ, γ, β,]	
2	Tester A	[α, β, , δ]	α→β→δ and β→(α,δ)
	Tester B	[δ, , β, α]	
3	Tester A	[α, β, γ, δ]	α→β→(γ,δ) and
	Tester B	[δ, γ, β, α]	β→(α,γ,δ)

In Table 2, (γ,δ) implies γ and δ can occur in an arbitrary order and (α,γ,δ) implies α, γ and δ can occur in an arbitrary order. In Table 2, an empty place in vector notation indicates absence of the corresponding output message. According to Table 2, because TC2 represented by (3:9,3:6) — $CONN_B$/[CONN, , *i*_CONN, CONN_ACK] → (10:10,10:10) is Type 2 and the message input direction is from Tester B, all possible orderings of output messages can be represented as CONN_ACK → i_CONN → CONN and i_CONN → (CONN_ACK, CONN).

When two IUTs interwork, we cannot know in advance in which sequence messages will be produced. Table 2 shows compressed ordering patterns from which all the possible message orderings can be extracted when we are to realize

[5] When we use concurrent TTCN, we do not need to calculate the occurrence orderings of the output messages because each PCO (or PO) checks messages independently. But, the use of concurrent TTCN is yet in its early stage.

test cases using TTCN. Thus, we can obtain minimal length test cases by exploiting these patterns. Figure 10 shows the TTCN description for TC2 using this principle.

```
LTb! CONN
  LTb? CONN_ACK
    LTc2? i_CONN
      LTa? CONN
  LTc2? i_CONN
    LTb? CONN_ACK
      LTa? CONN
    LTa? CONN
      LTb? CONN_ACK
```

Figure 10 TTCN description for TC2.

LTa and LTb represent PCOs, respectively, at interface A and at interface B. LTc2 represents a PO at interface C that reads messages sent by IUT B. The number of orderings considered in Figure 10 is 3.

7 CONCLUSION

To ensure interoperation of communication system, interoperability testing is esential in addition to conformance testing. In this paper, we presented systematic interoperability test suite derivation method for the class of communication protocols like ATM signaling protocol.

We analyzed in detail the conditions for a test case to constitute an interoperability test case and showed that interoperability test cases for ATM signaling protocol can be classified into three types based on the patterns of message interactions. After a careful comparison of three candidate interoperability test architectures for interoperability testing of two equipment implementing ATM signaling protocol, we selected Test Architecture II as it provides a good test coverage at a low cost.

Next, we presented a method to derive interoperability test cases that satisfy interoperability test case conditions under the selected test architecture. It includes an algorithm that performs composition of FSMs and derivation of interoperability test suites in parallel. The algorithm produced 26 interoperability test cases for the ATM signaling protocol.

Furthermore, we showed that possible orderings of output messages are determined by the types of test cases and the direction of the initial external input message. We used this regularity to reduce the lengths of interoperability test cases.

In this paper, we considered only control flow and message types. If we further take into account information elements of messages, more than one test cases may be obtained from a single test case (skeleton). Currently we are investigating how such aspect can be incorporated into our method.

As an alternative method of deriving interoperability test cases, one can think of extracting a test suite that includes both control flow and data flow at once, by deriving first an extended-FSM(EFSM) for each entity and composing the two EFSMs. Although this method has one less step than one in this paper, further study is needed to see which method is superior to the other since using the extended-FSM increases the complexity of test derivation.

8 REFERENCES

[AF UNI] ATM Forum af-uni-0010.002, "ATM UNI interface specification version 3.1"

[AF PNNI1] ATM Forum af-pnni-0055.00, "Private Network-Network Interface Specification Version 1.0(PNNI v1.0)," Letter Ballot, March 1996.

[AF PNNI2] ATM Forum af-pnni-81.00, "Private Network-Network Interface (PNNI) Version 1.0 Errata and Protocol Implementation Conformance Statement (PICS) Proforma," March 1997

[APRS 93] Arakawa, N., Phalippou, M., Risser, N. and Soneoka, T., "Combination of conformance and interoperability testing," Formal Description Techniques, V (C-10) M. Diaz and R. Groz (Eds.), Elsevier Science Publishers B. V. (North-Holland), 1993.

[AraS 92] Arakawa, N. and Soneoka, T., A Test Case Generation Method for Concurrent Programs, *Protocol Test Systems, IV*, J. Kroon, R.J. Heijink and E. Brinksma (Eds.), Elsevier Science Publishers B. V. (North-Holland), 1992.

[ISO DTR10000] ISO/IEC JTC1 DTR-10000, "Information Technology - Framework and classification of international standard protocol," 1994

[ETSI ETR130] ETSI ETR(ETSI Technical Report) 130, "Methods for Testing and Specification(MTS); Interoperability and conformance testing/A classification scheme," April 1994

[KanK 97] Kang, S., and Kim, M., Interoperability Test Suite Derivation for Symmetric Communication Protocols, Forte/PSTV'97, Osaka, Japan, November 1997.

[LuBP 94] Luo, G., Bochmann, G. and Petrenko, A., Test Selection Based on Communicating Nondeterministic Finite-State Machines Using a Generalized Wp-Method, *IEEE Transactions on S.E., Vol. 20, No. 2*, Feb. 1994.

[RafC 90] Rafiq, O. and Castanet, R., "From Conformance Testing to Interoperability Testing," The 3rd International Workshop on Protocol Test Systems, 1990

[VerB 94] Vermeer, G. S., and Blik, H., Interoperability Testing: Basis for the Acceptance of Communicating Systems, Protocol Test Systems, VI(C-19), Elsevier Science Publishers B. V. (North-Holland), 1994.

9 BIOGRAPHY

Jaehwi Shin received the B.S. and M.E. degrees in electronic engineering from Hanyang University and Seoul National University in 1991 and 1993, respectively. Currently he is with the Korea Telecom Research and Development Group as a member of Protocol Engineering Team. His research interests include intelligent networks, communication protocol testing, digital signal processing and circuit theory.

Sungwon Kang received B.A. from Seoul National University in Korea in 1982 and received M.S. and Ph.D. in computer science from the University of Iowa in U.S.A in 1989 and 1992, respectively. Since 1993, he has been a senior researcher at Korea Telecom. In 1995-1996, he was a guest researcher at National Institute of Standards and Technology of U.S.A. In 1997, he was the co-chair of the 10th International Workshop on Testing of Communicating Systems. Currently he is the head of the Protocol Engineering Team at Korea Telecom R&D Group. His research interests include communication protocol testing, program optimization and programming languages.

Appendix 1. Interoperability test suite for ATM signaling protocol (Test Architecture II)

Number	Initial Global State	Input message / Output messages	Final Global State	Type
Test Group: ESTABLISH/A_TO_B/				
TC1	(0:0,0:0)	$SETUP_A$ / [CALL_PROC, i_SETUP, i_CALL_PROC, SETUP]	(3:9,3:6)	3
TC2	(3:9,3:6)	$CONN_B$ / [CONN, , i_CONN, CONN_ACK]	(10:10,10:10)	2
TC3	(3:9,3:9)	$CONN_B$ / [CONN, , i_CONN, CONN_ACK]	(10:10,10:10)	2
Test Group: ESTABLISH/B_TO_A/				
TC4	(0:0,0:0)	$SETUP_B$ / [SETUP, i_CALL_PROC, i_SETUP, CALL_PROC]	(6:3,9:3)	3
TC5	(6:3,9:3)	$CONN_A$ / [CONN_ACK, i_CONN, , CONN]	(10:10,10:10)	2
TC6	(9:3,9:3)	$CONN_A$ / [CONN_ACK, i_CONN, , CONN]	(10:10,10:10)	2
Test Group: CLEARING/A_TO_B/				
TC7	(3:9,3:6)	REL_A / [REL_COMP, i_REL, i_REL_COMP, REL]	(0:0,0:12)	3
TC8	(3:9,3:6)	REL_COMP_A / [, i_REL, i_REL_COMP, REL]	(0:0,0:12)	1

Number	Initial Global State	Input message / Output messages	Final Global State	Type
TC9	(3:9,3:9)	REL_A	(0:0,0:12)	3
		[REL_COMP, i_REL, i_REL_COMP, REL]		
TC10	(3:9,3:9)	REL_COMP_A	(0:0,0:12)	1
		[, i_REL, i_REL_COMP, REL]		
TC11	(3:9,3:6)	REL_B	(12:0,0:0)	3
		[REL, i_REL_COMP, i_REL, REL_COMP]		
TC12	(3:9,3:6)	REL_COMP_B	(12:0,0:0)	1
		[REL, i_REL_COMP, i_REL,]		
TC13	(3:9,3:9)	REL_B	(12:0,0:0)	3
		[REL, i_REL_COMP, i_REL, REL_COMP]		
TC14	(3:9,3:9)	REL_COMP_B	(12:0,0:0)	1
		[REL, i_REL_COMP, i_REL,]		
Test Group: CLEARING/B_TO_A/				
TC15	(6:3,9:3)	REL_B	(12:0,0:0)	3
		[REL, i_REL_COMP, i_REL, REL_COMP]		
TC16	(6:3,9:3)	REL_COMP_B	(12:0,0:0)	1
		[REL, i_REL_COMP, i_REL,]		
TC17	(9:3,9:3)	REL_B	(12:0,0:0)	3
		[REL, i_REL_COMP, i_REL, REL_COMP]		
TC18	(9:3,9:3)	REL_COMP_B	(12:0,0:0)	1
		[REL, i_REL_COMP, i_REL,]		
TC19	(6:3,9:3)	REL_A	(0:0,0:12)	3
		[REL_COMP, i_REL, i_REL_COMP, REL]		
TC20	(6:3,9:3)	REL_COMP_A	(0:0,0:12)	1
		[, i_REL, i_REL_COMP, REL]		
TC21	(9:3,9:3)	REL_A	(0:0,0:12)	3
		[REL_COMP, i_REL, i_REL_COMP, REL]		
TC22	(9:3,9:3)	REL_COMP_A	(0:0,0:12)	1
		[, i_REL, i_REL_COMP, REL]		
Test Group: CLEARING/COMMON/				
TC23	(10:10,10:10)	REL_A	(0:0,0:12)	3
		[REL_COMP, i_REL, i_REL_COMP, REL]		
TC24	(10:10,10:10)	REL_COMP_A	(0:0,0:12)	1
		[, i_REL, i_REL_COMP, REL]		
TC25	(10:10,10:10)	REL_B	(12:0,0:0)	3
		[REL, i_REL_COMP, i_REL, REL_COMP]		
TC26	(10:10,10:10)	REL_COMP_B	(12:0,0:0)	1
		[REL, i_REL_COMP, i_REL,]		