Defining A Test Methodology for OSI Routing Protocols

John H. Moore
IBM Corporation

corwin @ vnet.ibm.com

June 5, 1992

Abstract

As OSI routeing protocols become standardized and as implementations begin to appear, the need to ensure conformance of these implementations to the standards arises. This report proposes a methodology based on ISO 9646 for developing conformance tests for the OSI routeing protocols. It explains elements of the test methodology of importance when testing routers, and gives some examples of simple test configurations. Concern regarding a few test issues are raised, and the article concludes with a recommendation to begin work developing a test suite for one protocol.

Introduction

Scope

This report deals with the set of OSI protocols that support the forwarding of Connectionless-mode Network PDUs as defined by ISO 8473 [1]. They are the End System to Intermediate System routeing exchange protocol (ISO 9542) [2], Intermediate System to Intermediate System Intra-domain routeing information exchange protocol (ISO 10589) [3] and the Inter-domain routeing information exchange protocol (SC6 N7196) [4]. In this report, these protocols will be referred to collectively as the ISO routeing protocols. The emphasis here will be on testing the routeing information exchange protocols, ISO 10589 (a.k.a. IS-IS) and SC6 N7196 (a.k.a. IDRP).

Objective

This report is intended to serve as a framework for the development of Conformance tests for the subject OSI routing protocols. Due to the complexity of the protocols, a clear road map is required in order to efficiently proceed with test development. The author seeks to ensure that the efforts expended on each protocol are complementary, and that when taken together they form a consistent set of conformance test suites.

Interoperability testing of some of these protocols has been done successfully in North America

[5], and there is at least one effort underway to develop a conformance test system that fulfills all the requirements of the OSI routeing protocols [6]. With this interest in mind, and as implementations of these protocols become available, the need for a conformance test suite originating from ANSI X3S3.3 that uses the recognized ISO standard test methodology and notation seems clear. Whereas interoperability testing has the characteristic of providing valuable insight into how two or more different implementations of the same protocol interact under "normal" conditions, conformance testing focuses on whether or not a particular implementation follows precisely the procedures stated in the standard which, to a great extent, deal with exception conditions. Thus, the two types of testing are complementary in nature and both should be considered a necessary part of a complete test scheme. The goal, of course, is to allow routeing system vendors to produce high quality, conformant, interoperable products in a timely manner.

Test Methodology Overview

This paper proposes that conformance tests for the OSI routing protocols should be developed using the methodologies described in ISO 9646 [7]. This standard defines a general framework for developing conformance tests using the notion of an abstract test method, which is a "description of how an (implementation) is to be tested, given at the appropriate level of abstraction to make the description independent of any particular realization of a Means of Testing..". In other words, the abstract test method describes unambiguously the external behavior expected of the tester, and allows the test tool implementor the freedom to create a tester in any way he or she chooses, as long as the external behavior requirements are met.

For use in developing an Abstract Test Suite (ATS), ISO 9646 defines a notation called Tree and Tabular Combined Notation (TTCN), which provides a pseudocode-like structure for the execution of test events in a test case. Test events in TTCN unfold sequentially, and the notation provides for branching at decision points. Upon completion, each test case assigns a verdict of either Pass, Fail, or Inconclusive.

TTCN uses indentation to indicate a sequence of events. Figure 1 shows an example of the test notation. A "!" (known as a 'shriek' in testing parlance) indicates a SEND event; so "!PDU A" (line 1) means that the tester will send PDU A to the implementation under test, known as the IUT. Lines 2,3, and 4 are at the same indentation level, and are therefore *alternative* events.

	Behavior	**Verdict**
1	**!PDU_A**	
2	** ?PDU_B**	**Pass**
3	** ?PDU_C**	**Fail**
4	** OTHERWISE**	**Fail**

Figure 1.

A "?" indicates a receive event; the tester is expecting from the implementation being tested either PDU B (in which case the test verdict assigned is Pass, or PDU C, in which case the test verdict is Fail. Also, if the tester receives any other PDUs (say, a PDU A), then the "Otherwise" event is satisfied, and the verdict assigned is Fail.

One of the advantages of using TTCN is that the notation can be fairly easy to understand from a protocol perspective. The format of the send and receive events translates well to a conceptual "bows and arrows" type of representation.

Currently, extensions to TTCN (ISO 9646-3 PDAM 1) [8] are being developed to allow it to support multi-party test methods. These methods allow multiple sequential test paths (trees) to be executed simultaneously, and allow for coordination between these trees (known as Parallel Test Components, PTCs) and between PTCs and a Main Test Component (MTC). The complexity of the OSI routeing protocols is such that their test suites will require these multi-party test methods.

Basic model

Case 1

Figure 2 illustrates a basic multi-party test architecture. The Main Test Component communicates with each of two Parallel Test Components via Coordination Points 1 & 2 (CP 1 & CP 2). Both the MTC and the PTCs are simple single TTCN event trees that operate asynchronously. The CPs allow communication between these individual event trees to synchronize them together to achieve the stated test purpose.

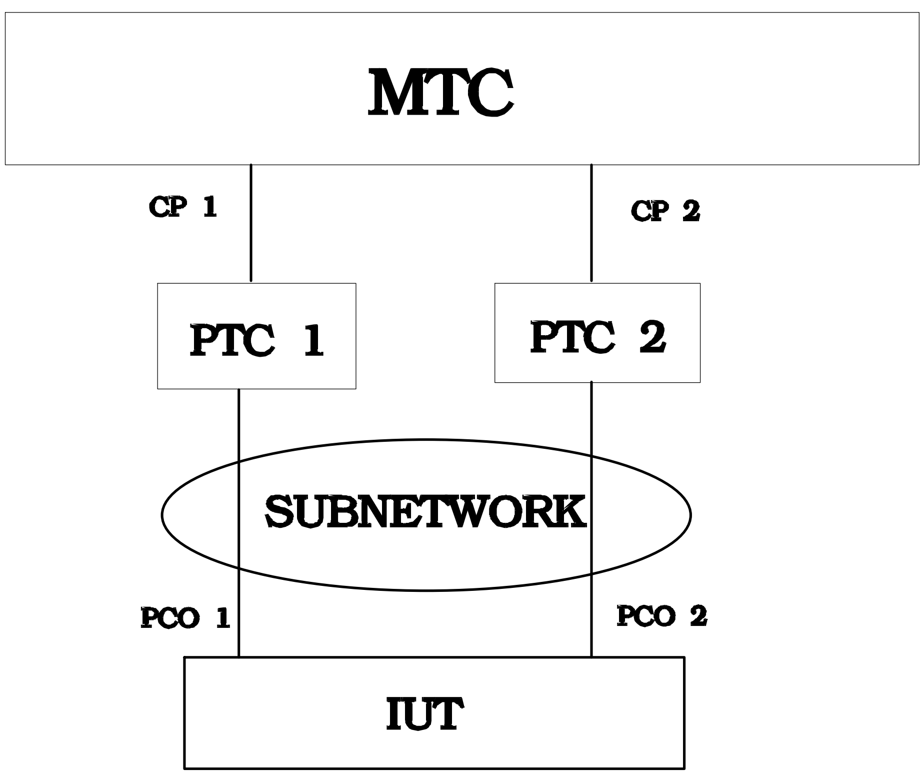

Figure 2.

The realization of these CPs is left up to the test tool implementor. If the tool is an integrated unit, the CPs would be realized with whatever inter-process communication facilities are available from the tool operating system. If the tool is distributed, with each test component residing in a separate piece of hardware, the CPs might make use of a LAN data link interface, for instance.

The messages that flow across the CPs are referred to as Coordination Messages (CM). No pre-defined CMs exist, and any message type that can be defined in TTCN can be used. Typically, CMs would implement a simple handshaking protocol designed to suspend or resume the operation of an event tree.

Figure 3 gives an example, which shows two different TTCN event trees, one for the MTC and one for the PTC. Both event trees start at the same time. The MTC sends a READY CM immediately, then waits for an OK CM from the PTC. Execution of the MTC event tree is essentially suspended until the OK is received. Similarly, the PTC event tree is initially suspended waiting for the READY CM from the MTC. Once it is received, it resumes operation and sends the OK CM.

MTC	**PTC**
!READY	**?READY**
?OK	**!OK**

Figure 3.

On the other side of the PTCs, each component communicates directly with the Implementation Under Test (IUT) with its' respective Point of Control and Observation (PCO 1 & PCO 2) through the broadcast subnetwork. The realization of these PCOs would be accomplished by using Network Service Access Points (NSAPs). The messages that would flow across these PCOs are those defined by the network layer protocol being tested.

Test cases would be crafted such that the events required to achieve the test purpose are distributed among the test components, with coordination occurring at the MTC. Each PTC would contain a valid TTCN tree of events culminating in a verdict based solely on its' own event tree, which is referred to as a preliminary verdict. These preliminary verdicts are then passed to the MTC via the respective CPs. The MTC in turn will base the final test case verdict on the preliminary verdicts.

Case 2

In the second example (Figure 4), the functions of the PCOs and CPs are similar to those in the previous example, but in this case the PTCs are on different subnetworks. From a network layer protocol perspective, PTC 1 can communicate with PTC 2 by relying on the router being tested (the IUT) to relay PDUs, since the PCOs shown are realized via NSAPs. This is useful in testing

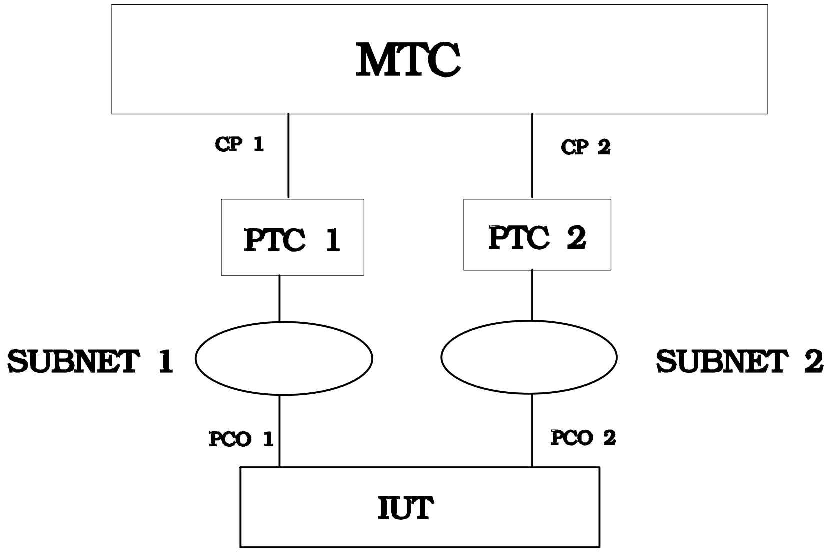

Figure 4.

the forwarding function, as described below in the test scenarios.

From a tester perspective, each PTC is still only communicating with the IUT via its' PCO, and is executing the TTCN tree based on responses from the IUT and CMs from the MTC.

Extending the Model

Figure 5 illustrates a more complex extension of the simple models presented so far. Each of the PTCs shown communicate with the MTC via CPs, and with the IUT via PCOs; not all are labeled for the sake of clarity.

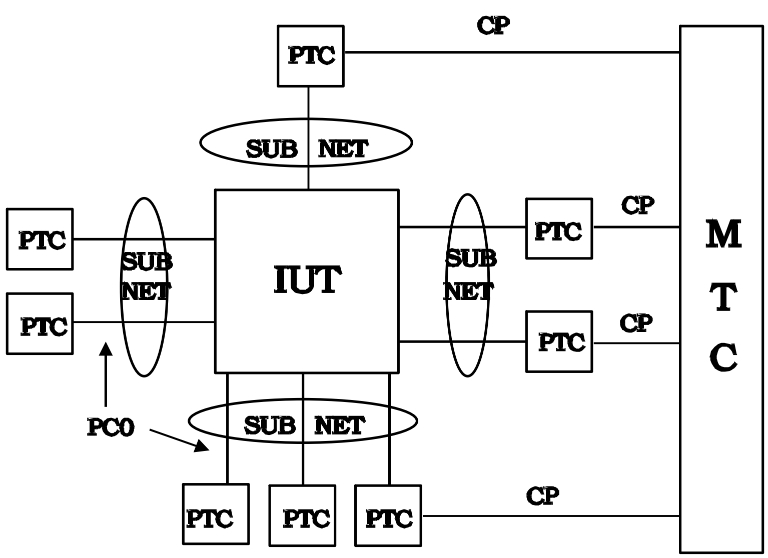

Figure 5.

A test case which used all of the PTCs shown to achieve a single test purpose would be complex indeed, and would be onerous to write. However, this type of configuration is still useful in the context of executing an entire suite. Each test case or group of test cases might require only two or three PTCs to be participating to achieve its' test purpose. A tester with automated execution capability would be able to select and drive the proper PTCs on a per test case basis. Over the course of the test campaign, all PTCs would be exercised.

Test scenarios

For illustrative purposes, let's discuss two simple test scenarios using the IS-IS protocol as an example. The two types of tests examined can be called configuration tests and forwarding tests.

Configuration/Update

A simple example is shown in Figure 6. Here, PTC 1 and PTC 2 are configured as routers on the same broadcast subnetwork as the Router under test (RUT). The MTC communicates with each of the two PTCs through CP 1 and CP 2. In the initial configuration of this test case, both PTC 1 & PTC 2 have adjacencies defined with the RUT, which has been elected as the Designated Intermediate System (DIS).

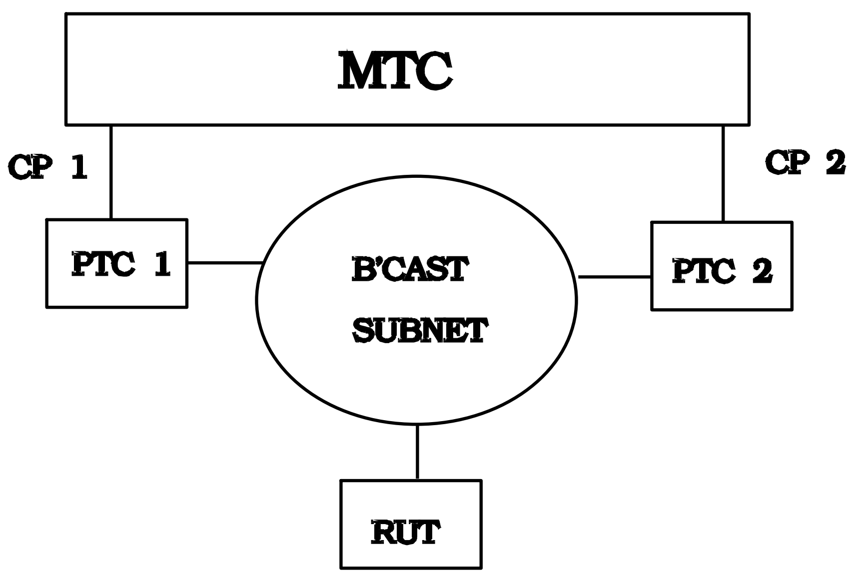

Figure 6.

The action that starts the test is that the MTC causes PTC 2 to turn off it's LAN circuit.. Possible ways in which that action might be implemented are discussed in the following sections.

After some period of time (less than **maximumLSPGenerationInterval**), the RUT will generate a Link State PDU (LSP) which will be received by PTC 1. The IUT passes this particular test if the LSP received does not contain the PTC 2 adjacency, and fails if it does or if no LSP is generated.

Forwarding

Figure 7 illustrates an example of a configuration used to test the forwarding function. This type of configuration is useful for indirectly testing the correctness of the Forwarding Information Base.

Here, the PTCs act as End Systems capable of generating CLNP PDUs. The initial configuration for the test is that the RUT is aware of each of the PTCs, and the network is stable. The subnetworks can be either broadcast or general topology. PDUs are generated at one PTC (PTC 1, for instance) with a destination address of another PTC (say, PTC 3). The test verdict is Pass if PTC 3 receives the PDU intact.

This simple test scenario can be expanded upon to include a number of these types of transactions to be performed simultaneously or in sequence. Another useful variation is to consider the RUT to be an arbitrary mesh network of routers which is subject to test. This goes beyond the bounds of normal conformance testing, but may aid in the general understanding of how implementations work together.

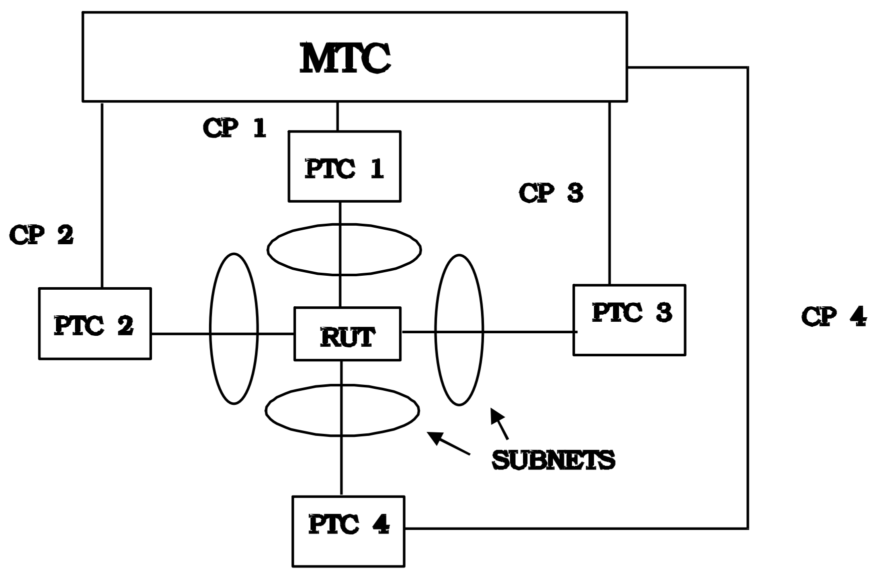

Figure 7.

Issues

Configuration Initialization

One of the more difficult tasks in writing conformance tests for the OSI routing protocols will be creating initialization routines that put the IUT and the various PTCs in a known state to begin the test. The initialization routines will require that each of the PTCs acts as a full implementation would in responding to PDUs sent from the IUT. Timers in the IUT, such as those requiring periodic receipt of a LSP PDU, will have to be satisfied by the test implementation. These requirements will likely cause the test case writer to keep the test configurations simple in order to avoid complex initialization procedures. This can be viewed as a positive thing, since these conformance tests are intended to simply verify that a product being tested has implemented the standard correctly, one function at a time.

Another related issue involves determining the appropriate "base state" for testing OSI routing protocols. Normally, a conformance test system will initialize the IUT to some known state, then initiate the proper protocol procedures to bring the IUT to the state in which the test will be executed. Here the question is, which state should be the base state? Should it be one where the router has an empty Routing Information Base, which will get built form scratch for each test case? Or should it be some defined state in which the router has a known set of adjacencies from which all tests (or more likely, a group of tests) will be executed?

Changing Parameters in the IUT

In some conformance testing scenarios, it is desirable for an action to be initiated by the IUT. This case is notated in TTCN by a construct known as an implicit send, which in essence says that the tester is expecting some action from the IUT without specifying how that action is to be initiated. The way a test tool implementor might handle this type of construct is to have the tester pause and send a message to the test operator instructing him or her to initiate the action in question. Of course, this assumes that the test operator has some means of manipulating the IUT to cause it to initiate the event.

This testing scenario could be useful in situations where the event in question is a change to an internal parameter of the IUT. An example, again using IS-IS, might be a situation where adding a Manual Adjacency to a router being tested would cause the ID of the adjacency to be included in the next LSP sent. The verdict of the test would be based on whether or not the LSP was sent correctly.

Conclusion

In summary, this paper proposes that work be started within X3S3.3 to develop conformance tests for the OSI routing protocols. These tests will complement existing and proposed interoperability tests and aid in the effort to produce conformant, interoperable OSI router implementations. The test methodology proposed is based on ISO 9646, which is being widely implemented for testing ISO and CCITT protocols. Further, testing the OSI routing protocols will require the multi-party testing extensions to 9646, currently in PDAM status within ISO.

References

[1] ISO 8473. *Information Systems - Data Communications - Protocol for providing the connectionless-mode network service.* 1988.

[2] ISO 9542. *Information processing systems - Telecommunications and information exchange between systems - End system to Intermediate system routeing exchange protocol for use in conjunction with the Protocol for providing the connectionless-mode network service*. 1988-08-15.

[3] ISO 10589. *Information technology - Telecommunications and information exchange between systems - Intermediate system to Intermediate system inra-domain routeing information exchange protocol for use in conjunction with the protocol for providing the connectionless-mode network service*. 1992-04-30.

[4] SC6 N7196. *Information Processing Systems - Telecommunications and Information Exchange between Systems - Protocol for Exchange of Inter-domain Routeing Information among Intermediate Systems to Support forwarding of ISO 8473 PDUs.* March 16, 1992.

[5] Doug Montgomery. *NIST Cooperative Laboratory for OSI Routeing Technology*, National Institute of Standards and Technology, May 20, 1992.

[6] Debra Tang, Jinsheng Liu, Jean-Philippe Favreau. *The IS-IS Multi-Party Conformance Test System,* National Institute of Standards and Technology. April 28, 1992.

[7] ISO/IEC 9646. *Information technology - Open System Interconnection - Conformance testing methodology and framework*. 1991.

[8] ISO/IEC 9646-3/PDAM 1. *Information Technology - Open systems interconnection - Conformance testing methodology and framework - Proposed Draft Amendment 1 to ISO/IEC 9646 part 3: TTCN extensions.* 1991-11-25.