Application of a TTCN based conformance test environment on the Internet email protocol

Jun Bi and Jianping Wu

Dept. of Computer Science, Tsinghua Univ.,

Dahong Xiong, Caixin Wu, and Jinwen Cao

*Institute of China Electronic System Engineering,
Beijing, P.R.China*

Abstract

This paper presents the conformance testing of the Internet email protocol SMTP using an integrated test system PITS. With TTCN based test execution and flexible reference implementation, PITS could test both the OSI and Internet protocols. In this paper, we discuss two methods for testing SMTP and the design of TTCN based SMTP test suite.

Keywords

Conformance testing, protocol test system, SMTP, TTCN

1 INTRODUCTION

The protocol engineering makes it possible to apply formal methods and certain automated tools during the protocol development life cycle. Although it specifically intended for the development of OSI protocols and services, it is

possible to have a much broader scope of application for TCP/IP protocols. Today, Internet had been widely accepted as the embryo of global information infrastructure. Therefore, the reliable communication between TCP/IP products is important in the future information highway. Without conformance testing, how could we find the errors in the routers, e-mail systems, and other devices we used? That is why we should do the conformance testing as while as the interpretability testing for TCP/IP products. However, over the last decade, there is little research effort in the formal specification, validation and testing for TCP/IP protocols. We developed an integrated testing environment PITS and the TTCN based test suite of SMTP (Simple Mail Transfer Protocol). PITS is implemented using the Sun Sparc workstation and Solaris 2.4 operating system. With this system, we could test different TCP/IP protocols on the basis of their TTCN test suite and corresponding reference implementation (RI).

This paper is structured as follows, section 2 gives a brief overview of PITS. Section 3 introduces the test organization and TTCN test suite of SMTP.

2 A TTCN BASED TEST ENVIRONMENT

Many earlier test systems are designed for single protocol or single test method. Therefore, their capability is limited. By our experience, the key of a test system is the test suite (TS) and test execution (TE) mechanism. In recent years, ISO has gradually developed the test suites for their standard protocols, and these test suites are described in TTCN. The Protocol Integrated Test System (PITS), developed by Tsinghua University, aims to provide a basic platform for developing protocol testing, and at the same time provides real test system for testing network protocols. Figure 1 shows the main processing flow in PITS.

Figure 1 The processing flow in PITS.

As above, this system organizes its testing process on the basis of TTCN test cases, and uses parallel interpreting to raise test efficiency. To generate test cases, we designed a test generator. It could derive the TTCN.MP test suite from the protocol specification of EBE (External Behavior Expression), which could be obtained from the other format of protocol specifications, such as FSM, LOTOS, and Estelle [1]. For the standard TTCN.GR test suites (e.g. ISO/IEC 8882), the

TTCN editor could translate it into TTCN.MP. After test selecting on the basis of PICS and PIXIT, the test cases are interpreted and then executed by TE step by step. TE an engine which interacts with other components of PITS, and controls test process according to the content of TS and simultaneously generates all information required to produce test report. The bit stream generated by TE will be sent to the corresponding buffer and RI, at last. RI is the lower support communicating with IUT. So with the suitable test suites and RIs, PITS could test different protocols by different methods.

3 CONFORMANCE TESTING OF SMTP

3.1 The difference between OSI and TCP/IP testing

SMTP is defined in RFC 821 and RFC 822. The objective of SMTP is to send and receive email reliably and efficiently using the client/server mode. Another important feature of SMTP is its capability to relay mail across transport service environments. Comparing with the peer-to-peer OSI protocols, the client/server and relay are the important modes in the TCP/IP suites. So, when we design the test architecture, we must consider these modes. For the client/server protocols (e.g. FTP, telnet), the IUT of client or server has different protocol functions. Thus we must design different test suites for this asymmetric architecture. Although some products implement both the functions of client and server, we have to classify the two test objectives in the test suite and test them separately in practice. For the relay function in some protocols, we have to control and observe the test events from the two sides of the IUT.

3.2 Test the function of sending and receiving email

Figure 2 Test architecture of sending and receiving email.

For testing the function of sending and receiving email, we adopt the distributed test method defined in ISO 9646. Figure 2 shows this testing architecture. We implement an upper tester (UT) above IUT. For testing SMTP-receiver, UT only reports the status of IUT. When we test SMTP-sender, UT will make the IUT

send email actively and act as the SMTP client. We use two TCP connections (by UNIX socket) as test paths. TE communicates with IUT by main test path (MTP), and with UT by subsidiary test path (STP). The MTP is for “regular” test events, and STP is for “out-of-baud” information (for TE sending test control message to UT or getting response from UT). We implement the TCP-RI as a C++ class. The test events will be sent to RI from buffer according to the PCO identifiers in this class. RI and buffer could communicate by means of the following message: STARTTEST (Start a test case execution); STOPTEST (Stop a test case execution); FRAME_SEND_OUT (TE sends a ASP/PDU); FRAME_RECEIVE (TE receives a ASP/PDU); QUIT (Quit the execution of a test case).

3.3 Test the function of relaying email

Figure 3 Test architecture of relaying email.

In ISO 9646, a number of standard abstract test methods for end system have been defined. Although there are two relay system test methods in ISO 9646, the capability of YL method is too simple to be put into use in practice and YT has two test systems so the test coordination of these two testers would be a big problem. Referring to the distributed method for end system and YL for relay system in ISO 9646, we present a new test method named as “distributed loop-back”. It had been used in the SMTP relay system testing. We implement a test responder (TR) in the mail destination host send/receive the email to/from PITS via IUT. In test system, there are two PCOs for the control and observation from both sides of IUT. When TE sends a email from PCO1 to destination host, the TR will get the relayed events from IUT and return it to the test system through STP. Then this returned message could be obtained by TE from PCO2. Because the events of PCO1 and PCO2 are both from the lower interface of IUT, TR is a conceptual lower tester (LT). However, TR has not the function of test execution and it works under the control of TE. In this architecture, we use only one TE for executing the events of two PCOs, so the problem of the coordination of two LTs is solved. When test executing, the test events for different PCO could be distinguished by the buffer and be sent to the corresponding RI. It makes the test process continuously and high-efficiently. It is just the advantage of this method.

3.4 Design of TTCN based SMTP test suite

There are lots of test generation research results ([2][3][4][5][6] etc.). When we design the SMTP test suite, we use EBE as the protocol model. EBE specifies only the external behavior of a protocol in terms of the input/output sequence and their logical (function and predicate) relations. First, we refine a SMTP external state machine. For 3 protocol functions, we define 3 EBEs: SMTP-SEND-EBE, SMTP-RECV-EBE, and SMTP-RELAY-EBE. Each EBE is a four-tuple $\langle S, s_0, T, R \rangle$. Here S is the external state set, s_0 is initial external state, T is the transition set of S and R is the logic relation set of T . The transition represents most of the SMTP commands: DATA, HELO, MAIL, RCPT, RSET, SEND, SOML, SAML, VRFY, EXPN, HELP, NOOP, QUIT, and TURN. Notice that the "DATA" here is a series of lines sent from the sender to the receiver with no response expected until the last line is sent. For each command there are three possible outcomes: Success (S), Failure (F), and Error (E). The test sequence derivation method is used to identify associations between inputs and outputs through the interaction paths and their I/O subpaths. Then the generic test cases specified in TTCN.GR format can be generated from these I/O subpaths [1]. In this test suite, there are three sub-suites for testing the function of sending, receiving, and relaying. In each sub-suite, there are 8 test groups to test the procedure of each protocol state and one test group to test system parameters. There are 89 test cases in total. For example, when we test the relay function in data transformation phase, we use the following TTCN.GR test case (see table 1). Here IUT is the mail relay system and the test method is shown in figure 3. In this test case, there are two PCOs. TE will control TR to send PDUs at PCO2 then receive the IUT response PDUs from PCO2 and the relayed PDUs from PCO1 by TR. Test verdict will be gotten according to the events from both PCOs.

Table 1 A TTCN based test case of SMTP

Test Case Dynamic Behavior					
Test Case Name: IT_01					
Group : SMTP/RELAY/IT					
Purpose : To verify SMTP mail server (server@Relay) could relay a mail from <sender@Remote> to <receiver@Local> in information transfer phase.					
Default :					
Comments :					
Nr.	L.	Behavior Description	Constraint Ref.	Verdict	Comments
1		+ Pre_IT			Preamble
2		PCO2 ! Body	Body_01		(1)
3		PCO2 ? 250	250_01		250 ok
4		PCO1 ? MailFrom	MailFrom_10		(2)
5		PCO1 ! 250	250_01		250 ok
6		PCO1 ? RcptTo	RcptTo_01		(3)
7		PCO1 ! 250	250_01		250 ok
8		PCO1 ? Data	Data_01		DATA
9		PCO1 ! 354	354_01		354

10		PCO1 ? Body	Body_02		(4)
11		PCO1 ! 250	250_01	(PASS)	250 ok
12		+ CHK_IT			Check
13		PCO1 ? Otherwise		FAIL	
14		PCO1 ? Timeout T01		FAIL	
15		PCO1 ? Otherwise		FAIL	
16		PCO1 ? Timeout T01		FAIL	
17		PCO1 ? Otherwise		FAIL	
18		PCO1 ? Timeout T01		FAIL	
19		PCO1 ? Otherwise		FAIL	
20		PCO1 ? Timeout T01		FAIL	
21		PCO1 ? Otherwise		FAIL	
22		PCO1 ? Timeout T01		FAIL	
23		PCO2 ? Otherwise		FAIL	
24		PCO2 ? Timeout T01		FAIL	
Detail Comments:					
(1) <sender@Remote>		(2) MAIL FROM: <server@Relay>			
(3) RCPT to: <receiver@Local>		(4) Mail Body = stamps + Body 01			

4. CONCLUSION

In this paper, we introduce some experience with SMTP testing. The method presented in this paper is also available for another Internet mail protocol like MIME, since the mail mechanisms of MIME and SMTP are same. However, in MIME test suite, the PDU declaration part will be much different from the text mail test suite. PITS had been implemented using Sun Sparc workstation and Solaris 2.4. It had been used in practical testing activity for many OSI and TCP/IP protocols. Our further work focuses on the development of conformance testing for other Internet protocols, such as the routing protocol OSPF v2.

REFERENCES

- [1] J. Wu and S.T.Chanson, Test sequence derivation based on external behavior expression, in *2nd IFIP IWPTS*, 1989.
- [2] O.Henniger, B.Sarikaya, et al., Test Suite Generation for Application Layer Protocol from Formal Specifications in Estelle, in O.Rafiq, editors, *6th IFIP IWPTS*, 1993.
- [3] G.v.Bochmann, et al., Fault coverage of test based on finite state models, in T.Mizuno, et. al., editors, *7th IFIP IWPTS*, 1994.
- [4] S.Kang and M.Kim, Test sequence generation for adaptive interoperability testing, in A.Cavalli, et. al., editors, *8th IFIP IWPTS*, 1995.
- [5] A.Petrenko, et. al., Testing deterministic implementations from non-deterministic FSM specifications, in B.Baumgarten, et. al., editors, *9th IFIP IWTCS*, 1996.
- [6] M.C.Kim, et.al., An approach for testing asynchronous communicating systems, in B.Baumgarten, et. al., editors, *9th IFIP IWTCS*, 1996.