Research on Protocol Conformance Testing Method Based on Petri net

SHU De Qin[a], FAN Hao[b] and Zhang Liang[c]
College of Information Science and Engineering, Shandong Agricultural University, Tai'an, China
[a]shudeqin@126.com,[b]fanhao@sdau.edu.cn, [c]zliang@sdau.edu.cn
Corresponding author: Fan Hao, fanhao@sdau.edu.cn.

Keywords: protocol conformance testing, Petri nets, formal method, testing set

Abstract. Network protocol conformance testing is a functional black box testing. It tests whether that if the implementation of a protocol consistent with consists with the protocol standards according to the protocol text description. A method of network protocol conformance testing with Petri net is presented. Using this method, not only the protocol is described by Petri net but also testing set, testing operation, testing execution, testing judgment are all described by Petri net. This makes Petri net, a formal tool, to run the whole process through protocol conformance testing. This is easy to understand, even though the whole test process can be visualized by graphical method and made the whole process strictly accurate. In order to illustrate the correctness of the method, the Petri net model of SIP protocol and conformance testing process of SIP protocol are given.

Introduction

Protocol testing is a kind of black box testing, which tests that if the implementation of a protocol consists with the protocol standards according to the protocol text description [1]. It is the basis of several other tests. With Petri net the formalized method of protocol conformance testing, now has become one of research hot spots. Petri nets graphical representation methods and strict mathematical definition, makes the conformance testing work more accurate and have a theoretical basis.

In paper [2], it explores Input-Output Conformance (IOCO) test generation with Colored Petri Nets (CPN) to propose a novel test purpose model driven IOCO test selection approach. Although this method is a kind of software conformance testing method also has great reference value for network protocol testing. In paper[3], a novel incentive mechanism with detailed algorithm for its practical application is proposed. It constructs colored Petri Nets (CPN) based formal model, then validates the accuracy of this model. In addition, there are a lot of literature to use Petri net to the protocol conformance testing research [4,5].

A lot of research and practice of network protocol conformance testing have been done in-depth discussion. But now, there is no protocol conformance testing method that makes the Petri net the formalized tools through protocol testing work always. This paper puts forward a network protocol conformance testing method based on Petri nets, using Petri nets to formalized description of protocol, the protocol conformance testing stages: test preparation, test operation, test execution and test judgment, each portion of test can be integrated in a formalized (Petri net) environment.

Petri net basis concepts and definition

Definition 1[6] A triples $\Sigma=(P,T;F)$ satisfies the conditions described below is called a net, where

(1) $P \cap T = \phi$

(2) $P \cup T \neq \phi$

(3) $dom(F)=\{x \in P \cup T \mid \exists y \in P \cup T:(x,y)\in F\}$

$cod(F)=\{x \in P \cup T \mid \exists y \in P \cup T:(y,x)\in F\}$

$dom(F) \cup cod(F) = P \cup T$

(4) $F \subseteq ((T \times P) \cup (P \times T))$

Let $N=(P,T;F)$ be a net. Mapping $M:P\to\{0,1,2\cdots\}$ is called a marking of N . Two-tuples (N,M) is called a Petri net. *P* is the set of Place and *T* is the set of transition, there are no intersections of them. *F* is the function to show flow relationship between *P and T*. $dom(F)$ is called former domain and $cod(F)$ is called the after domain.

Definition 2[6] Petri net system is a net with marking which is a set of quadruple $\Sigma=(P, T; F, M)$ and have the following fire rules.

(1) $t\in T$,if $\forall p\in {}^{\bullet}t$: $M(p)\geq 1$, said *M* makes *t* enabled , which is expressed $M[t>$ (when ${}^{\bullet}t=\phi$, *t* is said to be enabled under any marking). Where ${}^{\bullet}t=\{s\mid s\in S\wedge(s,t)\in F\}$, $M(p)$ is the token number in the place p .

(2) If marking *M* makes transition *t* enabled, then *t* can be fired. Transition *t* fire under marking *M* to reach the new marking M' (which is expressed $M[t>M'$), for $\forall p\in P$:

$$M'(p)=\begin{cases}M(p)-1 & \text{when} \quad p\in {}^{\bullet}t-t^{\bullet}\\ M(p)+1 & \text{when} \quad p\in t^{\bullet}-{}^{\bullet}t\\ M(p) & \text{others}\end{cases}$$

Definition 3[6] Let be $N=(P,T;G,M_0,\Sigma,h,F)$ a labeled Petri net, where $(P,T;G,M_0)$ is a Petri net , Σ is a finite alphabet, $h:T-\Sigma$ is the labeled function, F is the termination state set, then $L(N)=\{\alpha\in\Sigma^*\mid\exists\sigma\in T^*\wedge M_0[\sigma>M\wedge M\in F\wedge h(\sigma)=\alpha\}$ is defined as the language of Petri net.

Definition 4[6] Let $\Sigma=(N_1,M_0)$ be a Petri net, $N_2=(P,\mathrm{T},\mathrm{G})$ be an occurrence net. If $\phi:N_2\to N_1$ satisfies the following conditions:

(1) $\forall t_1,t_2\in T:(t_1\neq t_2):\phi(t_1)=\phi(t_2)\to t_1^{\bullet}\neq t_2^{\bullet}\wedge {}^{\bullet}t_1\neq {}^{\bullet}t_2$

(2) $\forall p\in P:|\{t\mid\phi(t)=p\wedge {}^{\bullet}t=\phi\}|\leq M_0(p);$

Then (N_2,ϕ) is the process of Σ .

We give the concept of Petri net related to this paper, others theorems can be referred to [6].

Conformance testing method based on Petri net

This section will detail the steps of network protocol conformance testing method.

1. The scheme of network protocol conformance testing

The working process of the protocol conformance testing through is shown in fig.1.

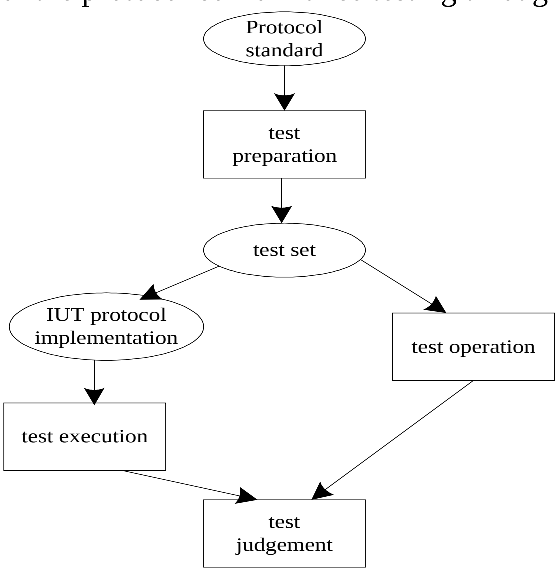

Fig.1 The overall scheme of protocol testing consistency

Test preparation: mainly aims at the research framework of protocol testing, which focus on how to get from the description of the protocol standard of protocol testing required test set.

Test set: is the set of protocol testing scripts and provide for each test case of protocol testing, to find a suitable language or a formula, which can express agreement in the form of simple, which hopes to use fewer test cases to cover as much as possible protocol operation situations.

Test operation（to get the expected response sequence of test cases）:Mainly complete for a given test case analysis of the expected response sequence.

Test execution (to get the expected response sequence of test cases): is used to do that test set makes a incentive/response test of IUT, incentive implementation under IUT response protocol data unit（PDU）, generate test log files.

Test judgment: Test records were analyzed, and compared with expected response sequence of test cases, and then test results are given.

IUT protocol implementation: IUT is the implementation under test that is the protocol ready to be tested.

2. The architecture of protocol conformance testing method

The architecture of protocol conformance testing is as shown in fig.2.

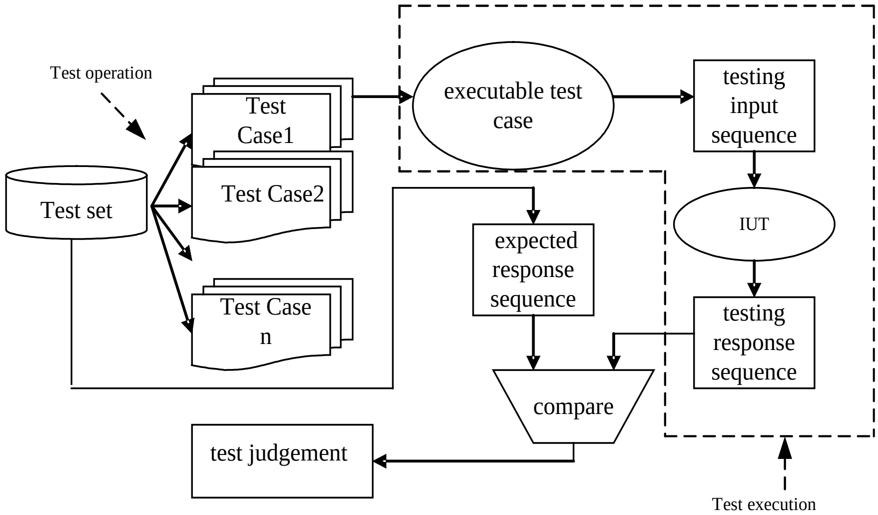

Fig.2 The Petri methods of test operation, execution and judgment

By the test preparation, the text description discusses in preparation for the test, in view of the protocol to be tested, test set is given. More than one test case is generally required to test the a functional test or a part of the test of protocol, in order to cover all possible operating conditions of the protocol. Test set is expressed by Protocol Petri net model, test case network system is expressed by the Petri net process.

After test preparation stage, the test set and the test case are obtained, test work is divided into two tied for the steps. One step is to generation executable test cases. With each executable test case the initial conditions are given (such as parameter values, the initial value of timer counter, etc.). Each executable test case is given to the IUT (the implementation under test) to execute. This can make IUT respond. IUT response result is the PDU (protocol data unit). PDU records the behavior of IUT when executable test case execute. PDU can also form test log files. The second step is to get the expected response sequence of protocol. This process is to analyze the process of the Petri net. The process of Petri net has the actual physical meaning. We can set up a corresponding relationship between the process of Petri net and actual behavior of IUT. Finally, by testing the decision, the actual response sequence of protocol is compared with the expected response sequence to determine whether the protocol entity’s behavior conformance with the protocol text description.

3. Protocol modeling method of Petri net

We have proposed three kinds of modeling method of the protocol. One of the first is the user for their understanding of the protocol the manual modeling. Second kind of methods corresponding paper [8], is the Petri net modeling method base on Entity behavior Description Language

(Commutation Protocol Entity Behavior Specification and Description Language) of PMA. Third kind of methods corresponding paper [7], is the Petri net automatic modeling method based on system behavior sequence. In this paper, we will use the third kind of methods. How to use the method to generate protocol's Petri net model will be detailed descript in this paper.

4. Detailed method of protocol conformance testing

It is as shown in fig.2. The method mainly uses the following ideas: a functional description of the protocol Petri net model can be considered as an abstract test set ATS, which simulates all the possible operating modes. If given an initial marking, the Petri net language of the model is given, which is equivalent to knowing the expected sequence and running state of all the protocols. Any string that is accepted by the Petri net can reflect the behavior of the protocol. The string can be used as an expected response sequence procedures by a test case. According to the initial marking of Petri net and the label of place, we can generate the executable test cases. Executable test cases will be executed by the IUT. IUT response result is the PDU (protocol data unit)which record the results of its operation. The running results can be transformed into the Petri net transition string σ. So any accepted σ by Petri net can be expressed as string which reflect the protocol a certain behavior. String σ can be as a test case expected running process. The initial identification based on Petri net and annotation of the libraries can generate executable test cases. Run the executable test cases to the IUT (implementation under test), documenting the results of its operation. Then we transform running results string σ into Petri net transition sequences .If σ can be accepted by the Petri net of protocol and after $M_0[\sigma > M'$, M' to match with the final desired state of protocol, then the test is successful. Otherwise, if M' is not to match the final desired state of protocol, let Petri net model be the initial marking M_0, fire the transitions sequence σ, $M_0[\sigma > M''$. Obviously, M'' is an error running state of the protocol. Finally, according to the test results, the test report is generated.

Example of conformance testing

Here we give an example to illustrate the conformance testing method in previous part.

1. Petri Net model of Protocol SIP

In order to illustrate the section of the modeling method, we give the following example. SIP (Initiation Protocol Session) is IETF (Engineer Task Force Internet) proposed. It works in the application layer, the full name session initialization protocol. Using the modeling method of sections (*3 Protocol modeling method of Petri net*), we give the Petri net model of SIP, see in Fig.3. Table 1 shows the explanations of places and transitions in SIP model. Applying Petri net model, SIP protocol's conformance can be tested in a variety of cases of SIP entities. This example is also the further advance of the SIP protocol test specification.

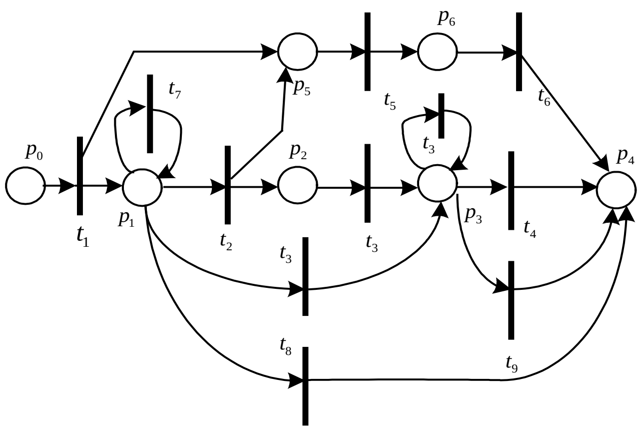

Fig.3 The Petri net model of SIP protocol

Table 1 the explanations of places and transitions

Place	explanation	transition	explanation
P0	User initialization state(Pre-calling)	t1	Send out invite
P1	After send invitation, enter calling state	t2	Receive data frame 1
P2	Receive data frame 1,enter Proceeding state	t3	Receive connect data ,send ACK
P3	send out ACK, enter completed state	t4	Timer D overtime
P4	Termination state STOP	t5	Receive data frame 2
P5	ANY state, can receive data	t6	Send ACK success
P6	Receive data frame 2, ready to send ACK	t7	Timer A overtime, retransmits invite
		t8	Timer B overtime or transmission error
		t9	transmission error

For client terminal, there are four states, Pre-calling, calling, proceeding, completed. User initialization state is Pre-calling, when user request a new transaction, the client terminal first enters the calling state. At the same time, the client transaction is transmitted to the transport layer for network transmission. If the transaction is in the case of a calling state, timer B overtime, transactions should be notified. The transaction layer can not generate ACK. If the transaction receives a response in the calling state, the status is moved to the proceeding state. No matter the transaction is in the calling state or proceeding state, when the response is received, the transaction's state is migrated to the completed state. At the same time, the corresponding ACK must be generated and sent to the transport layer. For the accurate description of the system, two special states are introduced here. During the SIP running process, the customer may occur as follows. Received responding of data frame 2, according to the above needs, an increase of ANY is added. Add a system to the stop state STOP. At the beginning of the SIP protocol startup, the place p_0 has a token, the rest of places have no token. So the initial marking of Petri net model is $M_0(P_0)=1, M_0(P_1)=M_0(P_2)=M_0(P_3)=M_0(P_4)=M_0(P_5)=M_0(P_6)=1$. The timer overtime value is set to 200ms.

2. Generate Test set and Test Cases of SIP Protocol

According to the Petri net model of SIP in fig.3 and the method as mentioned in part (*4. Detailed method of protocol conformance testing*), we can get test set and test cases. Because the fig.3 represents the running process of SIP, so the Petri net model in fig.3 can be regarded as a test set.

SIP protocol has 3 kinds of running ways, corresponding to three test cases. The test cases in net system are given in fig.4. $Case1, Case2, Case3$ are the test cases which express all the possible behaviors branch of SIP Protocol. $Case1, Case2, Case3$ are all expressed by Petri net process. The meaning of each test case is as follows:

$Case1$: It represents the following test procedure. SIP protocol enters the trying state when a new transaction instance is generated by client terminal. At this time, if the timer timeout retransmission request. If client terminal receive a ACK response, SIP move to the proceeding state and notify the transport layer. If the final response is received by client terminal, the SIP status is moved to the completed state.

$Case2$: Client terminal can send more than one response to the server side transactions. In the proceeding state, the transaction sends these responses to the transport layer for network transmission, but does not affect the state. If the retransmission request is received in this state, the transaction will retransmit the most recent response ACK. If the data frame 2 response is received in this state(proceeding state), the transaction instance sends the data frame 2 to the transport layer for transmission, but is not responsible for retransmission.

*Case*3: For non-invitation and ACK request, the initial state of SIP protocol is in frying state. Once enter frying state, any retransmission request received will be discarded. In this state, if the response is received from the upper level, the server will enter proceeding state. If the client terminal response is received by server terminal in this state, the SIP is moved to completed state. The response is sent to the transport layer for transmission at the same time.

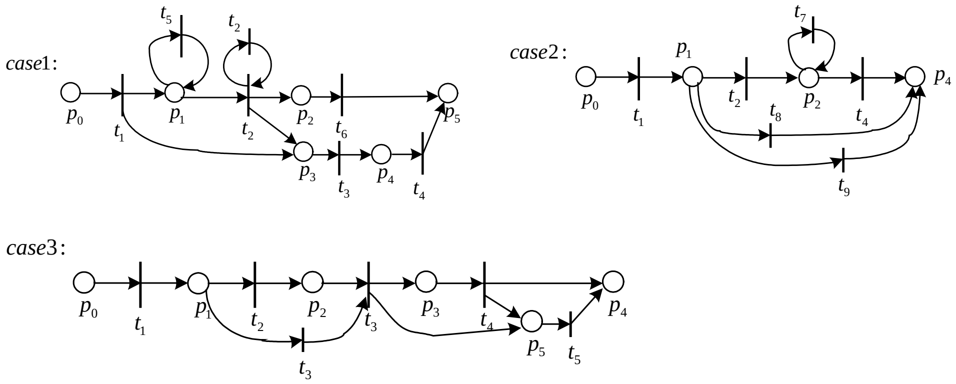

Fig.4 The test cases of protocol SIP

3. Conformance testing of protocol SIP

There are a lot of test paths of SIP protocol. We give one of the most representative testing processes because of limited length of this paper.

Test execution and test operation:

We give a complete test procedure through a combination of test cases. *Case*2 and *Case*3 can be grouped into the following Petri net process, see in fig.5. In the course of the session, the server terminal accept communications request. If the server think the response is correct, it process the request by the processing steps of the correct received request. If communications request does not match any of the session, server terminal generates 'transaction doesn't exists' answer. If the answer is incorrect, the server returns an initial state. The transition firing sequence $t_1t_2t_3t_8$ represents the process of the server receiving the correct response. The transition firing sequence $t_1t_9t_{10}t_{11}$ represents that the communications request does not match any of the session and the server returns an initial state.

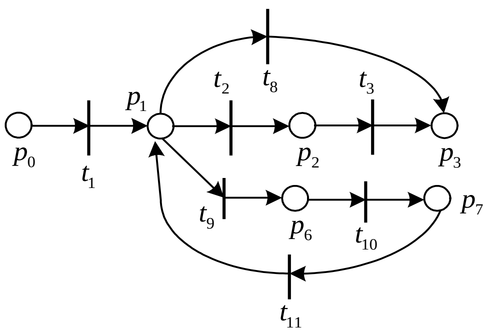

Fig.5 One of the test paths of SIP protocol

Test judgment:

We set up the initial conditions $M_0(P_0)=1$ (see in part *4 Petri Net model of Protocol SIP*) .Let SIP protocol practical operation, then we can get protocol running sequence. Comparing the actual response sequence and expected response sequence in fig.5, we can get whether the implementation of a protocol conforms to its specification.

Conclusions

In this paper, we give a general solution of network protocol conformance testing method based on Petri network. The overall scheme of network protocol conformance testing method, protocol test set and test case generation method, Petri net method of test operation, test execution, test execution, test method, are given in detail.

It can be seen that the method make Petri net this formalization model throughout the whole process of protocol conformance testing, and abstract test set, test case, test sequence are all expressed by Petri net. This makes the whole testing process can use the graphical method of visual representation, easy to understand, and make the whole process strictly accurate.

As a practical example, we give the process of protocol model establishment. And the process of conformance testing of protocol SIP. The correctness and effectiveness of the method are illustrated with the example.

Conformance testing, of course, is a complex problem, there are some problem haven't in-depth research, such as how to deal the test case to get measured implementation executable test cases, how to observe to get the actual response sequence of protocol agreement and so on. These problems remain to be further research.

Aknowledgement

This work is supported by Science and technology project of colleges and universities in Shandong Province（J11LG26） and Youth Science and Technology Innovation Fund Project of Shandong Agricultural University (23659).

References

[1]ISO/IEC JTC1/SC21 WG7. Information retrieval, transfer and management for OSI, framework: formal methods in conformance testing. Committee Draft CD13245-1, 1995.

[2]LIU Jing, LI Ru, et al. PN4TS: Test Selection Approach with IOCO Based on Colored Petri Nets, Chinese Journal of Computers, 2014, 37(12):2451-2462.

[3]JIA Lei, LIU Jing,et al. Design and Analysis of Incentive Mechanism in BitTorrent Protocol, Journal of Frontiers of Computer Science and Technology, 2014,8(3):275-287.

[4]HuaLi, Xin-ming,Ye, et al. Modeling Interactive Property of MIPv6 with Pet Net for interoperability Testing, The Second International Conference on Information and Computing Serene (ICICZO09), Manchester, England, 2009: 313-316.

[5]Yu Xue, Yi Xing, Hua Li, Xin-ming Ye. Research on the Interactive Property Testing Based on Petri Net, 2012 International Conference on Systems and Informatics (ICSAI2012),2012, 2466-2470.

[6]Wu Zhe-hui, The Introduction of Petri Net, Beijing: China Machine Press, 2006.

[7]FAN Hao, SHU De-qin, Zhang Liang, Petri Net Automatic Modeling Method Based on System Behavior Sequence, Applied Mechanics and Materials, 2014, vol 668-669:499-507.

[8]Fan Hao, Wu Zhe-hui, Communication Protocol Entity Behavioral Specification and Description Language， Journal of Computer Research and Development. 2007,44(11).1839-1848.