

병렬 TTCN을 이용한 멀티 파티 프로토콜 적합성 시험

正會員 손 홍 세*, 이 병 각**, 양 대 헌**, 송 주 석**

Multi-Party Protocol Conformance Testing using Concurrent TTCN

Hong-Se Son*, Byung-kak Lee**, Dae-Hun Nyang**, Joo-Seok Song** *Regular Members*

요 약

프로토콜 적합성 시험의 ATS(Abstract Test Suite)를 기술하기 위한 기법으로 Single TTCN(Tree and Tabular Combined Notation)이 널리 사용되어 왔으나, 불확정적 요소를 갖는 멀티 파티 프로토콜의 시험을 위한 ATS를 Single TTCN으로 작성하는 것은 많은 양의 시험 표현(test notation)을 필요로 할 뿐만 아니라 이해도를 저해시킨다. 따라서 이러한 문제점을 해결하기 위하여 병렬 TTCN이 제안되었으며 이에 대한 연구가 진행되고 있다.

본 논문에서는 기존의 singel TTCN과 병렬 TTCN의 특성을 비교 분석하였다. 그리고 B-ISDN 호/연결 제어 프로토콜인 Q.2971의 ATS 개발을 위하여 멀티 파티 프로토콜 적합성 시험 모델을 정의하고, 이 모델에 따른 프로토콜의 병렬 요소를 추출하여 생성된 결과를 분석함으로써 병렬 TTCN의 유용성을 검증하였다.

ABSTRACT

The Single TTCN has been broadly used to describe the ATS(Abstract Test Suite) for the protocol conformance testing. But if the single TTCN is used to test the multi-party protocol with the non-deterministic factors, then huge amount of test notations are needed and also it is hard to understand the ATS. Therefore, the concurrent TTCN was proposed to solve these problems and the study in progress.

In this paper, we did investigate and compare the characteristics of existing single TTCN with those of the concurrent TTCN. In order to develop the ATS of the Q.2971 which is B-ISDN call/connection control protocol, the testing model was defined and the concurrent factors were extracted according to that model. As a result, we could verify the usefulness of the concurrent TTCN.

*한국전자통신연구소
ETRI

**연세대학교 컴퓨터과학과
Computer Science of Yonsei Univ.
論文番號:96378-1127
接受日字:1996年 11月 27日

I. 서 론

최근 컴퓨터 통신의 급속한 보급에 따라서 많은 통신 프로토콜이 구현되고 있고 하나의 표준 프로토콜에 대하여 여러 제작자에 의하여 다양한 구현 제품들이 만들어지고 있다. 표준 FDT(Formal Description Technique)가 제시되었음에도 불구하고 실제 구현 제품들은 서로 호환성이 없는 경우가 많다. 따라서 정보 통신 표준의 제정, 구현뿐만 아니라 구현된 제품의 상호 운용성(interoperability)을 증진시키고 사용자에게 신뢰성 있는 서비스를 제공하기 위하여 적합성 시험(Conformance Testing)의 필요성이 강조되어 왔으며 다양한 프로토콜을 대상으로 적합성 시험서비스가 제공되고 있다. ISO(International Standard Organization)에서는 적합성 시험을 위한 시험 스위트(Test Suite)를 명세하기 위하여 표기 기법인 TTCN(Tree and Tabular Combined Notation)을 제시하여 표준화하였고 이것은 시험 절차(test sequence)를 읽기 좋게 명세하며 시험 결과에 대한 판정(Verdict)도 구체적이고 용이하게 얻을 수 있다. 그러나 기존의 TTCN으로는 불확정성(non-determinism) 요소가 포함된 프로토콜을 시험하기 용이하지 않아서 병렬 TTCN에 관한 연구가 이루어지게 되었다. 병렬 TTCN은 불확정성을 효과적으로 표현할 수 있고, 이에 따라 시험 절차(test sequence)의 수를 많이 줄일 수 있게 되었다.

본 논문에서는 멀티 파티 호/연결 제어 프로토콜인 ITU-T 권고(안) Q.2971을 Single TTCN으로 작성한 시험 스위트와 병렬 TTCN을 이용해 작성한 시험 스위트를 비교 분석한다[1]. 이를 위하여 멀티 파티 프로토콜의 시험 모델을 정의하고, 이를 기준으로 멀티 파티 프로토콜인 Q.2971의 동작 상의 병렬적인 요소를 추출해내고 이것을 이용해 ATS(Abstract Test Suite)를 만든다.

II. 적합성 시험

통신 프로토콜은 통신하는 대상간의 정보를 교환하는 규칙들을 나타내는 것으로 여러 국내의 표준화 기구에서 표준으로 정의하고 있다. 그러나 이를 구현하는 구현자 나름대로 표준을 해석하여 구현하게 되

므로 많은 해석상의 오류 및 구현 절차상의 오류로 인하여 실제 구현물은 규격대로 구현되지 않을 수 있다. 그래서 구현물이 표준에 따라서 정확하게 구현되었는지를 검증하는 절차를 수행하게 되는데 이를 적합성 시험(Conformance Testing)이라고 한다.

적합성 시험 방법 및 체계를 기술한 표준안으로는 ISO/IEC 9646(ITU X.290 시리즈)이 있다. 제1부는 ISO/IEC 9646의 전반적 개요에 대한 소개이며, 제2부는 시험에 필요한 요구 사항 및 개념, 시험 방법을 기술하고 있다[2][3]. 제3부는 표준 시험 언어인 TTCN을 정의하고, 제4부는 특정 프로토콜을 시험하기 위한 수단에 관해 기술되어 있다[4]. 제5부는 적합성 평가(conformance assessment)를 위한 시험소(test laboratory)와 의뢰자(client)에 대한 요구 사항을 기술한 것이다. 제6부는 프로토콜 프로파일 시험 규격을 다루고 있으며 마지막으로 제7부는 구현 적합성 명세(ICS: Implementation Conformance Statement)를 기술하고 있다. 또한 제1부를 비롯하여 제5부까지는 프로토콜 프로파일 시험 방법과 다자간(multi-party) 시험 방법을 기술하고 있는 개정안 1과 2를 각각 포함하고 있다.

III. 병렬 TTCN

3.1 시험 구성요소(Test Components)

병렬 TTCN은 병렬적으로(Concurrently) 실행되는 시험 구성 요소들의 명세를 나타낼 수 있다. 이러한 형식과 메카니즘은 병렬성을 사용하지 않는 ATS에서는 사용될 수 없다. 병렬 TTCN에서 시험기(Tester)는 하나의 MTC(Master Test Component)와 0개 이상의 PTC(Parallel Test Component)들로 구성되어 있는데 병렬성이 없는 Standard TTCN에서는 단지 하나의 TC(Test Component)만 있고 이것이 기본적으로 TMC이므로 따로 MTC를 선언할 필요가 없다.

[그림 1]은 병렬 TTCN의 시험 구성요소의 일반적인 배치 방법을 보여준다[5].

TC은 Test Component Declaration 테이블에서 선언되는데, 한 TC는 하나 이상의 PCO를 통해서 ITU와 통신하며 TC들은 CP(Coordination Points)를 통해서 CM(Coordination Messages)을 교환해 서로 통신한다. PTC들은 전역 결과 변수를 이용해 내부적으로

MTC와 통신할 수 있고 MTC는 하나 이상의 PTC가 실행을 종료했는지의 여부를 확인할 수 있다. Test Component Configuration Declaration 테이블은 TC의 설정을 나타내기 위해 사용된다. 이러한 선언들은 PCO와 CP들이 사용되는 방법을 나타낸다. CM (Co-ordination Message)은 ASP(Abstract Service Primitives)를 명세하기 위해 사용되는 방법과 유사하게 명세되며 ASN.1이 CM 명세를 위해 사용될 수도 있다. 또한 CM constraint는 ASP constraint와 유사하다. CM 타입의 정의와 CM constraint의 선언을 위해 특별한 형식이 제공되고 CM들은 일반적인 TTCN의 SEND와 RECEIVE 문장을 사용하여 주고 받는다.

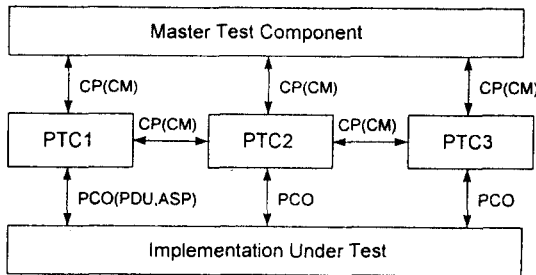


그림 1. 일반적인 병렬 시험 구성 요소 배치

3.2 TTCN의 병렬성

TTCN에서 병렬성을 지원하는 것은 불확정적인 (Non-deterministic) 프로토콜 행위와 여러 PCO를 포함하는 시험 경우(Test Case)를 작성하기 쉽고 테스트 아키텍처의 구성 요소들간의 상호작용을 외부적으로 묘사하는 언어적인 방법을 제공해 주기 때문이다. 반면에, Single TTCN에서는 큐 메커니즘에서 자동적으로 해결하도록 하거나, 구현자에게 해결하도록 하여, Single TTCN으로도 병렬성과 불확정성을 표현할 수 있으나 매우 복잡하고 각각의 경우를 모두 표현해야 하므로 표현(notation)의 수가 많아지며 명세 자체를 관독하기 힘들어진다. 따라서 병렬 TTCN은 Transaction Processing Protocols, ISDN, Network Management Protocols 등과 같은 멀티 파티 프로토콜을 시험하는데 적합하다. 병렬 TTCN은 외부적으로 불확정성을 표현하기 위해 몇가지 메커니즘을 가지고 있다. 이런 메커니즘은 여러개의 PCO들이 독립적으로 그리고 병렬적으로(concurrently) 수행할 수 있도록 해주며,

이렇게 하므로써 시험 스위트의 가독성(Readability)을 높일 수 있다.

한개 이상의 시험기들의 독립적인 수행을 위해서 이들을 조정해 주는 부분이 필요하며, [4]에서는 이런 핸드셰이크가 필요함을 명시하고 있다. 시험기들의 동기화를 위해 핸드셰이크를 구체적으로 구현할 방법이 필요하며, 본 논문에서는 SYNC 이벤트로 병렬적으로 동작하는 여러개의 시험기들을 동기화했다. SYNC 이벤트를 위한 behavior 트리는 다음과 같다.

CP! sync

CP? sync_ack

CP? OTHERWISE FAIL

위의 behavior 트리에서 OTHERWISE 문장은 sync_ack 메시지를 받기 전에 큐에 기다리고 있는 다른 메시지가 없음을 확인 시켜준다. 이렇게 하므로써, 여러개의 프로세스가 동시에 수행되는 환경에서 발생할 수 있는 데드락(deadlock)을 방지할 수 있다. 만약 동기화만을 목적으로 한다면 OTHERWISE 문장을 생략할 수 있다.

3.3 시험 요소 관리

Single TTCN에서는 최종 판정이 결정되거나 시험 경우 트리(Test Case Tree)의 리프 노드가 수행되면 시험의 수행이 종료된다. 병렬 TTCN의 경우 시험의 수행 종료에 대한 다른 기준의 필요하게 된다. 즉 하나의 시험 요소(Test Component)가 리프에 도달하고, 다른 시험 요소가 수행중일 때 어떻게 해야 할지를 결정해야 한다. 또한 라우팅 시험 요소 같은 어떤 시험 요소는 스스로 종료하지 못하는 경우도 있으므로, 시험의 종료 기준을 마련해야 한다. 병렬 TTCN에서는 다음의 규칙을 따른다.

- 시험의 수행은 하나의 시험 요소가 최종 판정값을 갖게 되면 종료된다. 이것은 모든 예비(preliminary) 판정이 이 순간 보류되는 것을 의미하므로, 최종 판정값의 할당에는 주의가 필요하다. 이런 최종 판정값을 갖는 시험 요소는 주로 PTC들의 예비 판정값으로 최종 판정을 내리는 MTC 등이 된다.

- 하나의 시험 요소는 리프에 다르면 수행을 종료한다. 하지만, 다른 시험 요소는 이것과는 무관하게 수행을 계속한다.
- 모든 시험 요소가 수행을 종료한다면 전체 시험의 수행도 종료된다.

판정값을 결정하기 위해서 병렬 TTCN에서는 다음의 메카니즘들을 따른다. 하나의 시험 요소에 대한 시험 수행 후의 리포트 판정은 NO VERDICT이며 각 시험 요소의 판정값은 시험이 종료되고 난 후 수집되어서 최종 판정값의 결정에 사용된다. 만약 어떤 시험 요소의 최종 판정값이나 예비 판정값이 FAIL이라면, 시험의 최종 판정값은 FAIL이 된다. 모든 시험 요소의 최종 판정 값이나 0개 이상의 PASS와 한개 이상의 INCONCLUSIVE라면, 그 시험 경우의 최종 판정값은 INCONCLUSIVE가 된다. 그 외의 모든 경우에서 최종 판정값은 PASS가 된다[4].

[그림 2]는 CCSS7(Common Channel Signaling System No.7-Q.764에서 정의)망에서 호를 제하는데 있어 불확정 요소가 존재하며 이를 병렬 TTCN으로 작용한 예를 보인다.

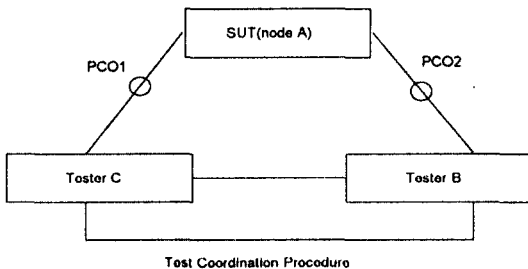


그림 2. CCSS7 Call 해제를 위한 시험 배치

CCSS7에서 [그림 2]와 같은 구성도를 가질 때 호 해제를 위한 프로토콜 시나리오는 다음과 같다. CCSS7 네트워크에서 메시지 교환을 모델링하는 시험기 B, C 두개가 있다. C로부터 B로 SUT(System Under Test)를 통한 연결이 존재한다고 가정하고 시험기 C가 연결을 해제하기 원할 때, C는 SUT에 PCO1를 거쳐 해제 메시지 "rel"을 보낸다. SUT는 받은 해제 메시지에 대하여 "rlc"로 응답하고 C가 연결을 해제하기 원한다는 것을 "rel" 메시지를 전송하여 B에게 알

려준다. B는 이 해제 메시지를 확인하고 "rlc" 메시지로 응답한다. C가 해제 과정을 초기화한 후 두 시험기는 그 과정을 독립적으로 계속 처리한다.

[그림 3]은 Single TTCN으로 위의 호 해제 절차를 시험하기 위한 ATS를 기술하였다. [그림 3]의 레벨 20에서 주어진 대체 문장은 SUT의 불확정적인 행위를 표현한다. 즉 PCO1으로의 "rlc" 메시지와 PCO2로의 "rel" 메시지의 순서가 불확정적이다. 이처럼 concurrency를 지원하지 않는 TTCN은 여러 문제점들을 가지고 있다. 두 이벤트들의 순서가 불확정적인 경우 각 이벤트들의 수행 후에 행해지는 서브 표현(notation)들은 두배의 크기가 된다. 즉, 표현의 수는 매우 큰 폭으로 증가하게 된다. x를 불확정적인 요소의 개수라고 하고 y를 모두 표현하는데 필요한 경우의 수라고 하면 $y=x!$ 의 식이 성립한다.

이것을 해결하기 위해 병렬 TTCN을 이용한다. [그림 4]는 병렬 TTCN으로 이 프로토콜을 시험하는 트리를 보여준다. 병렬 TTCN을 사용하게 되면 불확정적인 각각의 경우에 대해서 모두 기술할 필요없이 각각의 시험기로 분리해 병렬적으로 작동하게 할 수 있다. 그러므로 $y=x$ 의 표현의 수만 있으면 가능하다.

두 PCO를 독립적, 병렬적으로 조정하기 위해 동기화 메카니즘이 필요하며 이 예에서는 SYNC 키워드로 표현된다. 이 SYNC 키워드는 CM(Coordination message)로 구현될 수 있고 CP(Coordination Point)를

Level	Behaviour	Verdict
10	PCO1?OTHERWISE	FAIL
20	PCO2?OTHERWISE	FAIL

Default tree

Level	Behaviour	Verdict
10	PCO1!rel	PASS
20	PCO1?rlc	
30	PCO2?rel	
40	PCO2!rlc	
20	PCO2?rel	PASS
30	PCO1?rlc	
40	PCO2!rlc	

그림 3. CCSS7 Call 해제 예제 ATS

Level	Behavior	Verdict
10	Pcol!rel	PASS FAIL
20	SYNC(testerB)	
30	PCO1?rlc	
30	PCO1?OTHERWISE	

Tree for tester C

Level	Behavior	Verdict
10	SYNC(testerC)	PASS FAIL
20	PCO2?rel	
30	PCO2?rlc	
20	PCO2?OTHERWISE	

Tree for tester B

그림 4. 두 병렬 트리로 기술된 호 해제

통해 전달 된다. 그리고, 이 메시지는 시험기(tester)간의 조정을 담당하게 된다. 아래예의 SYNC(파라미터)에서 파라미터는 실제로 CP를 의미하며 SYNC 프로시저는 CP에 적절한 CM을 전송하게 된다.

IV. Q.2971 분석 및 시험 시스템의 서

4.1 Q.2971 분석[1][11]

ITU-T 권고안 Q.2971은 DSS 2(Digital Subscriber Signalling System No. 2) 점 대 다중점 호/연결제어를 위한 사용자 망 접면 계층 3 규격이다. Q.2971은 점 대 점(point-to-point) 가상 채널 호/연결에 관련된 신호 처리 절차를 규정한 Q.2931 권고 안을 포함한 점 대 다중점 가상 채널 호/연결(하나의 루트와 여러개의 리프 사이의 연결)의 신호 처리 절차를 규정하므로, Q.2971 프로토콜을 지원하는 가입자 단말 뿐만 아니라 Q.2931 가입자 단말도 리프 노드로 설정하여 사용할 수 있다.

Q.2971에서 정의하고 있는 사용자쪽이나 네트워크쪽에 존재하는 파티의 상태는 다음과 같다.

- (a) Null: 파티가 더 이상 존재하지 않는 상태로, endpoint reference값이 할당되지 않는다.
- (b) Add Party Initited: SETUP이나 ADD PARTY 메시지가 인터페이스의 반대편으로 전송된 상

태를 나타낸다.

- (c) Add Party Received: SETUP이나 ADD PARTY 메시지가 인터페이스의 반대편에서 수신된 상태를 나타낸다.
- (d) Party Alerting Delivered: ALERTING이나 PARTY ALERTING 메시지가 인터페이스의 반대편으로 전송된 상태를 나타낸다.
- (e) Party Alerting Received: ALERTING이나 PARTY ALERTING 메시지나 인터페이스의 반대편으로 수신된 상태를 나타낸다.
- (f) Drop Party Initiated: DROP PARTY 메시지가 전송된 상태를 나타낸다.
- (g) Drop Party Received: DROP PARTY 메시지가 수신된 상태를 나타낸다.
- (h) Active: UNI의 사용자쪽에서, 사용자가 CONNECT, CONNECT ACKNOWLEDGE, 또는 ADD PARTY ACKNOWLEDGE 메시지를 받거나 ADD PARTY ACKNOWLEDGE 메시지를 전송한 상태를 나타낸다. UNI의 네트워크에서는, 네트워크가 CONNECT, CONNECT ACKNOWLEDGE, 또는 ADD PARTY ACKNOWLEDGE 메시지를 보내거나 ADD PARTY ACKNOWLEDGE 메시지를 수신한 상태를 나타낸다.

Q.2971에서의 연결 방식은 루트에서 리프로 한쪽 방향 만으로의 전송만을 지원한다. 여기서 연결(call)에 참여하고 있는 각 파티들은 연결이 설정된 상태에서 더 첨가되거나 제거될 수 있다. 연결은 루트와 하나의 리프 사이에 연결을 요청함으로써 설정된다. 이 설정이 이루어져서 Alerting이나 Active 상태가 되는 경우, 다른 리프들도 루트로부터의 Add Party 요청에 의해 이 연결에 첨가될 수 있게 된다. 호가 활성화되어 있는 상태에서 리프는 첨가되거나 제거될 수 있으며 새로운 리프는 Add Party 요청을 통해서 루트에 의해 첨가될 수 있다. 리프를 제거하는 것은 루트나 리프에 의해서 수행될 수 있다.

[그림 5]는 Q.2971의 호 설정 절차를 나타낸다. 호를 요청하는 쪽에서 SETUP 메시지 또는 호가 이미 설정 되어 있는 경우에는 ADD-PARTY 메시지를 전송하므로써 호 설정 요청을 보낸다. 네트워크에서는

호 설정 요청시 CALL-PROC 메시지로써 요청하는 쪽에 요구가 받아들여졌음을 알린다. 호 설정 요청을 받은 쪽에서는 호 설정이 가능한 경우 CONNECT 메시지로 호 설정을 알린다.

루프가 다음 Add Party 요청을 하기 위해 먼저 한 Add Party 요청에 대한 응답을 기다릴 필요는 없다. 즉, 동시에 여러개의 Add Party 요청이 가능하다. 마찬가지로 다음 Drop Party 요청을 위해서 먼저 한 Drop Party 요청에 대한 응답을 기다릴 필요가 없다. 즉, 동시에 여러개의 Drop Party 요청도 가능하다. Drop Party를 수행한 결과로 리프가 하나도 남아있지 않다면, 전체 호는 해제된다[1].

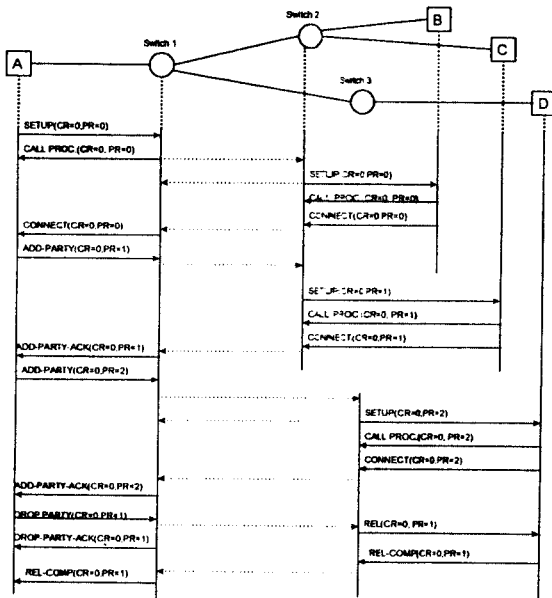


그림 5. Q.2971의 호 설정 절차

4.2 시험 시스템

이 절에서는 Q.2971 프로토콜 적합성 시험을 위한 ATS의 일부분을 소개한다. 이것은 Q.2971의 권고안을 따르는 것으로 사용자 사이의 단방향 데이터 전송을 위한 점 대 다중점 호/연결 설정 및 해제에 관련된 신호방식을 기술한다.

본 논문에서는 ISO 9646 part II의 원격 단일 계층 방법(Remote Single Layer Test Method)을 이용하여 시험 경우를 작성한다. 이것은 SUT와 떨어진(remote)

시스템에서 조정하고 관찰하는 시험 방법이다.[그림 6]은 RS-Method를 나타낸다.

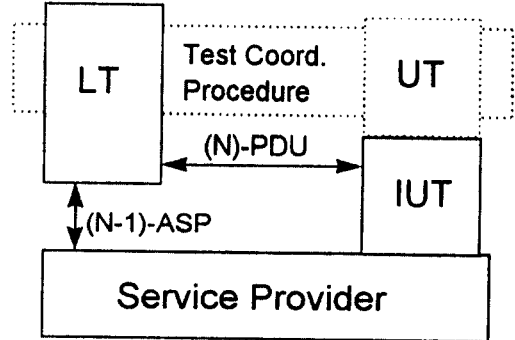


그림 6. RS 시험 방법

본 논문에서 작성한 ATS는 Q.2971 네트워크 망 측 프로토콜을 가지고 있는 시험 대상 시스템(IUT), 호 연결 제어 절차를 시작하는 루트 시스템(LT1), 첫번째 파티(first party)로 호/연결 제어 절차를 수행할 리프 노드(LT2), 첫번째 파티의 호 설정 후 연결 가능한 여러 파티들의 동작을 대표하는 두번째 파티를 위한 리프 노드(LT3)로 구성된다. IUT는 ITU-T I.413에서 정의된 B-NT1에 해당되는 기능을 수행하고 LT1, LT2, LT3는 B-NT1의 동작을 시험하기 위한 절차들을 수행하게 된다. 네트워크 측 프로토콜의 상태는 호출하는 쪽(Calling Side)과 호출되는 쪽(Called Side)의 시험 가능한 복합 상태를 추출하고 이에 맞게 시험 경우를 작성하였다.

4.2.1 Single TTCN에 의한 시험 절차

[그림 7]은 Single TTCN으로 Q.2971의 시험 경우를 위한 모델을 보여준다. 이 모델에서 실제로 시험기는 하나로 이루어지며 CP나 CM은 사용되지 않는다. [그

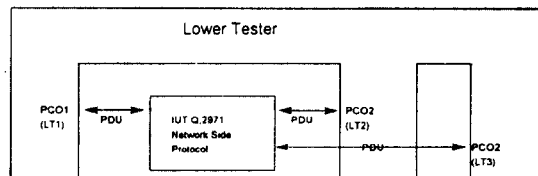


그림 7. Single TTCN 시험 구성도

림 8)은 루트에 의하여 두번째 파티의 파티 연결 해제를 나타내는 절차로써, LT1으로부터 DROP PARTY 메시지를 받은 후에 IUT가 LT1에 DROP PARTY

ACKNOWLEDGE와 STATUS 메시지를 LT1에 보내고 LT3에 RELEASE 메시지를 보내는 과정을 나타내는 시험 경우의 Dynamic behaviour이다.

Purpose : Verify that the IUT sends a DROP PARTY ACKNOWLEDGE message and a STATUS message to LT1 and RELEASE message to LT3 after receiving a DROP PARTY message with unrecognized information element from the LT1. The final IUT states of the calling side and called side interface are expected to be the Null party state(PN0) and the Drop Party Initiated party states(PN5) respectively.		
1	*연결이 이루어진 상태로 만드는 preamble	
2	LT1?DROP P START T398MAX, START T_NORESP1	
3	LB1 : LT1?DROP_P_ACK CANCEL T398MAX	(pass)
4	LB2 : LT1?STATUS CANCEL T_NORESP1	(pass)
5	LB3 : LT3?REL CANCEL T_NORESP1	(pass)
6	+LT1, LT3의 상태가 (P_NULL,DROP_P_INIT)인지를 검사.	
7	+Null party state로 변환시키는 postamble	
8	+REL가 아닌 다른 유효한 메시지를 받은 경우	
9	GOTO LB3	
		
14	LT3?REL CANCEL T_NORESP1	(pass)
15	LB4 : LT1?STATUS CANCEL T_NORESP1	(pass)
16	+LT1, LT3의 상태가 (P_NULL,DROP_P_INIT)인지를 검사	
17	+Null party state로 변환시키는 postamble	
18	+STATUS가 아닌 다른 유효한 메시지를 받은 경우	
19	GOTO LB4	
		
32	LT1?STATUS	(pass)
33	LB5 : LT1?DROP_P_ACK CANCEL T398MAX	(pass)
34	LB6 : LT3?REL CANCEL T_NORESP1	(pass)
35	+LT1, LT3의 상태가 (P_NULL,DROP_P_INIT)인지를 검사.	
36	+Null party state로 변환시키는 postamble	
37	+REL가 아닌 다른 유효한 메시지를 받은 경우	
38	GOTO LB6	
		
	LT3?REL CANCEL T_NORESP1	(pass)
		(pass)
44	LB7 : LT1?DROP_P_ACK CANCEL T398MAX	
45	+LT1, LT3의 상태가 (P_NULL,DROP_P_INIT)인지를 검사	
46	+Null party state로 변환시키는 postamble	
47	+DROP_P_ACK가 아닌 다른 유효한 메시지를 받은 경우	
48	GOTO LB7	
		
61	LT3?REL CANCEL T_NORESP1	(pass)
62	LB8 : LT1?DROP_P_ACK CANCEL T398MAX	(pass)
63	LB9 : LT1?STATUS CANCEL T_NORESP1	(pass)
64	+LT1, LT3의 상태가 (P_NULL,DROP_P_INIT)인지를 검사.	
65	+Null party state로 변환시키는 postamble	

66	+STATUS가 아닌 다른 유효한 메시지를 받은 경우	
67	GOTO LB9	
		
72	LT1?STATUS CANCEL T_NORESP1	(pass)
73	LB10 : LT1?DROP_P_ACK CANCEL T398MAX	(pass)
74	+LT1, LT3의 상태가 (P_NULL,DROP_P_INIT)인지를 검사	
75	+Null party state로 변환시키는 postamble	
76	+DROP_P_ACK가 아닌 다른 유효한 메시지를 받은 경우	
77	GOTO LB10	
		
92	+DROP_P_ACK,STATUS,REL가 아닌 다른 유효한 메시지를 받은 경우	
93	GOTO LB1	
94	?TIMEOUT T_NORESP1	(fail)
95	+Null party state로 변환시키는 postamble	
96	?TIMEOUT T398MAX	(fail)
97	+Null party state로 변환시키는 postamble	
98	LT1?OTHERWISE	(fail)
99	+Null party state로 변환시키는 postamble	
100	LT3?OTHERWISE	(fail)
101	+Null party state로 변환시키는 postamble	

그림 8. Single TTCN ATS

4.2.2 병렬 TTCN에 의한 시험 절차

병렬 TTCN을 이용한 ATS는 독립적으로 동작하는 세 개의 시험기인 MTC, PTC1, PTC2로 구성된다. 주 시험기인 MTC는 PTC1과 PTC2를 동기화 시키고, 이들로부터의 예비 판정값을 이용하여 최종 판정값을 결정한다. 또한 MTC는 1:2 모델에서 루트의 역할을 수행하며, PTC는 리프 노드의 역할을 한다.

[그림 9]는 병렬 TTCN으로 만들기 위한 시험 구성도이다. Single TTCN의 구성도와 비교해서 보면 여러개의 TC들이 있고 TC간에 통신할 수 있는 CP가 있다. 이 CP를 통해서 동기화 메시지를 보내게 된다. [그림 10]은 [그림 8]과 같은 시험 경우를 병렬 TTCN으로 만든 ATS이다.

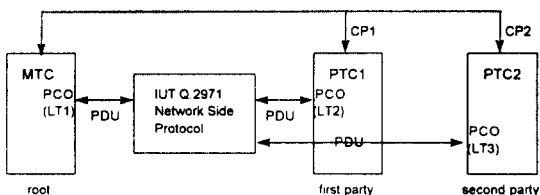


그림 9. 병렬 TTCN 시험 구성도

첫번째 시험 경우는 LT1에 DROP PARTY ACKNOWLEDGE 메시지를 보내는 것이고 두번째는 LT1에 STATUS 메시지, 세번째는 LT3에 RELEASE 메시지를 보내는 시험 경우이다. Single TTCN에서 이러한 메시지들을 하나의 시험 경우로 나타내는데 반해 세개의 시험 경우로 표현되지만 적은 수의 표현만으로 모든 이벤트들을 표현할 수 있다.

세개의 시험 요소들을 조정(coordination)하기 위해 동기화 메카니즘이 필요하며 여기에서는 SYNC 키워드로 표현했다. 이 SYNC 키워드는 CM(Coordination message)으로 구현되어 있으며 CP(Coordination Point)를 통해 전달된다. 다음은 [그림 10]에서 사용되는 병렬 TTCN의 SYNC 시험 과정을 나타낸다.

- MTC의 동기화 방법
 - CP1! sync
 - CP1? sync_ack
 - CP1? OTHERWISE FAIL
- PTC의 동기화 방법
 - CP1? sync
 - START begin_proc

Purpose : Verify that the IUT sends a DROP PARTY ACKNOWLEDGE message to LT1 after receiving the DROP PART message from LT1. The final IUT states of the calling side and called side interface are expected to be the Null party state(PN0) and the Drop Party Initiated party states(PN5) respectively.

<pre> +연결이 이루어진 상태로 만드는 preamble LT1!DROP_P START T398MAX, START T_NORESP1 LB1 : LT1?DROP_P_ACK CANCEL T398MAX +SYNC(LT2) +SYNC(LT3) +DROP_P ACK가 아닌 다른 유효한 메시지를 받은 경우 GOTO LB1 ?TIMEOUT T_NORESP1 +Null party state로 변환시키는 postamble ?TIMEOUT T398MAX +Null party state로 변환시키는 postamble LT1?OTHERWISE +Null party state로 변환시키는 postamble </pre>	(pass)
<pre> +DROP_P ACK가 아닌 다른 유효한 메시지를 받은 경우 GOTO LB1 ?TIMEOUT T_NORESP1 </pre>	(fail)
<pre> +Null party state로 변환시키는 postamble ?TIMEOUT T398MAX </pre>	(fail)
<pre> +Null party state로 변환시키는 postamble LT1?OTHERWISE +Null party state로 변환시키는 postamble </pre>	(fail)

Purpose : Verify that the IUT sends a STATUS message to LT1. The final IUT states of the calling side and called side interface are expected to be the Null party state(PN0) and the Drop Party Initiated party states(PN5) respectively.

<pre> +SYNC(LT1) LB1 : LT1?STATUS CANCEL T_NORESP1 +LT1, LT3의 상태가 (P_NULL,DROP_P_INIT)인지를 검사 +Null party state로 변환시키는 postamble +STATUS가 아닌 다른 유효한 메시지를 받은 경우 GOTO LB1 ?TIMEOUT T_NORESP1 +Null party state로 변환시키는 postamble LT1?OTHERWISE +Null party state로 변환시키는 postamble </pre>	(pass)
<pre> +STATUS가 아닌 다른 유효한 메시지를 받은 경우 GOTO LB1 </pre>	(fail)
<pre> ?TIMEOUT T_NORESP1 +Null party state로 변환시키는 postamble LT1?OTHERWISE +Null party state로 변환시키는 postamble </pre>	(fail)

Purpose : Verify that the IUT sends RELEASE message to LT3 after receiving a DROP PARTY message with unrecognized information element from the LT1. The final IUT states of the calling side and called side interface are expected to be the Null party state(PN0) and the Drop Party Initiated party states(PN5) respectively.

<pre> +SYNC(LT1) LB1 : LT3?REL CANCEL T_NORESP1 +LT1, LT3의 상태가 (P_NULL,DROP_P_INIT)인지를 검사. +Null party state로 변환시키는 postamble +REL가 아닌 다른 유효한 메시지를 받은 경우 GOTO LB1 ?TIMEOUT T_NORESP1 +Null party state로 변환시키는 postamble LT3?OTHERWISE +Null party state로 변환시키는 postamble </pre>	(pass)
<pre> +REL가 아닌 다른 유효한 메시지를 받은 경우 GOTO LB1 </pre>	(fail)
<pre> ?TIMEOUT T_NORESP1 +Null party state로 변환시키는 postamble LT3?OTHERWISE +Null party state로 변환시키는 postamble </pre>	(fail)

그림 10. 병렬 TTCN ATS

CPI? OTHERWISE FAIL

MTC는 CPI(CP2)을 통해서 sync 메시지를 PTC1(PTC2)에게 보내고 sync_ack 메시지를 기다린다. sync 메시지를 받은 PTC1(PTC2)는 자신의 시험을 시작한다. 각 PTC의 수행이 끝나고 결과를 MTC로 넘겨 주게 된다. 이 SYNC 메시지를 통해 다른 메카니즘 없이 여러개의 시험요소간의 동기화 문제를 해결할 수 있다.

Single TTCN을 위한 구성도에서는 많은 PCO들을 하나의 시험 경우에서 고려해야 하므로 시험 경우의 작성이 매우 복잡해진다. LT1에 DROP PARTY ACKNOWLEDGE와 STATUS를 내는 이벤트와 LT3에 RELEASE 메시지를 보내는 이벤트 중 어떤 것이 먼저 전달되는지 알 수 없는 불확정적인 경우가 생기므로 Single TTCN으로 구현하였을 경우 3!가지 경우의 수를 기술해야 하며 반복되는 부분이 많이 나오게 된다. 병렬 TTCN 구성도에서는 위의 3가지 이벤트를 독립된 모듈로 작성할 수 있으므로 시험 경우의 작성이 쉬워진다. 이렇게 구성했을 경우 3!개의 이벤트 대신 3가지의 이벤트 만으로 표현 가능하다. 또한 병렬 TTCN으로 시험 경우를 작성하면 각 시험 요소들을 모듈화된 단위로 표현할 수 있으므로, 여러개의 시험 요소들이 하나의 시험 경우에 혼합되는 Single TTCN 보다 가독성이 높아진다.

V. 결 론

본 논문에서는 멀티 파티를 지원하는 프로토콜인 Q.2971에 대한 시험 스위트를 Single TTCN으로 작성하는데 있어서의 문제점을 도출하고, 이를 병렬 TTCN으로 기술하는 방법을 제시했다. Q.2971을 시험하는 모델은 마스터 시험 요소 하나와 복수개의 슬레이브 시험 요소로 구성되어 있다. 마스터 시험 요소는 PTC(Parallel Test Component)의 조정(coordination)을 담당하고, 슬레이브 시험 요소는 멀티 파티에서 호의 연결 요청 및 요청에 대한 응답을 위한 일을 담당한다. 병렬 TTCN으로 기술한 시험 스위트는 불확정성(non-determinism)을 쉽게 구현할 수 있고, 적은 수의 시험 단계로 Q.2971을 시험할 수 있다.

TTCN으로 시험 구성 요소간의 동기화를 표현하는

것은 매우 어려운 작업이다. 예를 들어, 하위 시험기와 상위 시험기 사이의 동기화를 위한 잘 기술된 의미 모델이 없다. 따라서 어떤 경우에 이런 동기화는 몇개의 PCO를 이용해서 해결하거나 구현자가 적절히 조절하도록 한다. 또한 간접 전송(Implicit send)이 동기화를 위해서 사용되기도 한다. 하지만 이런 방법들은 더 복잡한 환경에서는 이용할 수 없으므로, 좀더 일반적인 방법이 필요하다. 슬레이브 시험 요소간 또는 마스터 시험 요소와 슬레이브 시험 요소간의 동기화 메커니즘에 대해서 연구해야 하며 또한 동적인 시험 요소의 생성, 제거, 제어 등에 대해 더 많은 연구가 필요하다.

참 고 문 헌

1. ITU-T Recommendation Q.2971- "B-ISDN DSS 2 UNI Layer 3 Specification for Point-to-Multipoint Call/Connection Control, ITU-Telecommunication Standardization Sector, June 1995.
2. ISO/IEC IS 9646-1, OSI Conformance Testing Methodology and Framework Part 1:General Concept
3. ISO/IEC IS 9646-2, OSI Conformance Testing Methodology and Framework Part 2:Abstract Test Suite Specification
4. ISO/IEC DIS 9646-3, OSI Conformance Testing Methodology and Framework Part 3:Tree and Tabular Combined Notation
5. Anthony Wiles, The Tree and Tabular Combined Notation-A Tutorial, Swedish Telecom, 1991.
6. R.L. Probert, O. Monkewich, TTCN:The international notation for specifying tests of communications systems, Computer Networks and ISDN Systems, Vol. 23, 1992, pp. 417-438.
7. Rudolf J. Heijink, The Introduction of Concurrency in TTCN, PTT Research Nether Laboratories.
8. Rudolf J. Heijink, 병렬 TTCN, PTT Research Nether Laboratories.
9. D. Rayner, OSI Conformance Testing, Computer Networks and ISDN Systems, Vol. 14, 1987, pp. 79-98.

0. Testing, ATM Forum Technical Meeting, Feb 6-Feb 10, 1995.
11. ITU-T Recommendation Q.2931-B-ISDN DSS2 UNI Layer 3 Specification for Basic Call/Connection Control, ITU-Telecommunication Standardization Sector, Oct 1994.



손 홍 세(Hong-Se Son) 정회원
1964년생
1986년 2월: 경북대학교 전자공학과(공학사)
1990년 2월: 경북대학교 컴퓨터공학과 대학원(공학석사)
1990년 2월~현재: 한국전자통신

연구원 선임연구원

※주관심분야: LAN 및 위성통신망 프로토콜 설계, 프로토콜 적합성 시험, ATM/B-ISDN



이 병 각(Byung-Kak Lee) 정회원
1971년생
1995년 2월: 연세대학교 컴퓨터과학과 졸업(이학사)
1995년 9월~현재: 연세대학교 컴퓨터과학과 석사과정 재학중

※주관심분야: 프로토콜 엔지니어링, 정보 보안



양 대 현(Dae-Hun Nyang) 정회원
1970년생
1994년 2월: 한국과학기술대학 전기 및 전자공학과 졸업(공학사)
1996년 2월: 연세대학교 컴퓨터과학과 졸업(이학석사)

1996년 3월~현재: 연세대학교 컴퓨터과학과 박사과정 재학중

※주관심분야: 프로토콜 엔지니어링, 정보 보안, ATM Congestion control

송 주 석(Joo-Seok Song) 정회원
1953년생

1976년 2월: 서울대학교 전기공학과 졸업(공학사)
1979년 2월: 한국과학원 전기 및 전자공학과 졸업(공학석사)

1979년 2월~1982년 2월: 한국전자통신 연구소 전임 연구원

1988년 8월: Univ. of California at Berkeley 전산과학과 졸업(박사)

1988년 9월~1989년 2월: Naval Postgraduate School Information System Department 조교수

1989년 3월~현재: 연세대학교 컴퓨터과학과 교수