

# LAN을 이용한 적합성 시험 시스템의 구현과 No.7 TCAP에의 적용에 관한 연구

## (A Study on the Implementation of Protocol Conformance Test System on LAN and the Conformance Test for CCITT No. 7 TCAP)

尹 載 日,\* 盧 承 煥\*\*, 趙 顯 俊\*\*, 金 惠 鎮\*\*  
(Jae Il Yoon, Soong Hwan Ro, Hyun Joon Cho, and Duck Jin Kim)

### 要 約

본 논문에서는 이 기종 간의 LAN을 이용하여 구현된 적합성 시험 시스템을 이용하여 CCITT No. 7 TCAP 프로토콜<sup>[1]</sup>의 적합성 시험을 수행하였다. 본 실험에서 사용된 상태천이도와 상태 천이표는 CCITT 권고안 Blue Book Q.774의 문답처리기능 응용부의 분석을 통해 이루어졌고 전체적인 테스트 시퀀스는 복수개의 UIO(unique input output)에 의한 테스트 시퀀스 산출 알고리즘<sup>[2]</sup>을 이용하여 산출되었다. 다시 이들 전체 테스트 시퀀스는 Q.713권고안에 따라 파라미터가 확립된 후 각 테스트 시퀀스는 실제 적용 가능한 테스트 시퀀스를 이룬다. 이러한 테스트 시퀀스는 LAN상에서 CSMA/CD 방식을 따르는 Ethernet을 경유하여 테스트 서버와 함께 수행되고 있는 IUT(implementation under test)에 전달되는 방식으로 전체 테스트의 수행이 이루어졌다. 테스트 시스템의 UT(upper tester)는 트랜잭션 처리부의 상위에 위치하는 컴포넌트 처리부의 기능을 담당하고 LT(lower tester)는 트랜잭션 처리부의 하위에 위치하는 신호 연결 제어부의 기능을 담당하여 에 물레이션에 의한 수행이 이루어졌다.<sup>[3]</sup>

### Abstract

Given a protocol specification, the task of testing whether an implementation conforms to the specification is called conformance testing. In this paper, the design and implementation of the protocol conformance test system using LAN on heterogeneous systems is described. The test sequence used in this test system is generated using an algorithm on the basis of multiple UIO. The usability of this test system as an application was confirmed by performing the conformance test on the CCITT No. 7 TCAP(transaction capabilities application part), implemented by ISDN Laboratory, Korea University in 1990.

### I. 서 론

현대의 통신 시스템은 다양한 사용자의 요구에 부응하고자 전문화와 더불어 기능의 신뢰성이 크게 요구되었다.

이에 부응하여 통신 프로토콜의 신뢰성, 효율성 그리고 유지보수의 용이성이 점차 중요한 문제로 대두되었고 이와 관련하여 FDT(formal description technique)를 이용하여 프로토콜을 정확하게 기술하고자 하는 프로토콜 명세(protocol specification)분야, 정의된 프로토콜 규격의 에러를 검출하고자 하는 프로토콜 검증 분야, 실제 프로토콜의 구현분야, 구현된 통신 시스템의 기능에

\* 學生會員, \*\* 正會員, 高麗大學校 電子工學科  
(Dept. of Elec. Eng., Korea Univ.)

接受日字: 1992年 4月 28日

대한 프로토콜 테스트 분야에 대한 연구가 수행되고 있다.

프로토콜 명세분야에서는 프로토콜의 디자인 단계에서 프로토콜의 내부구조를 고려하여 규격화된 방법으로 프로토콜을 정확하게 기술하고자 하며 그때 이용되는 FDT로서는 CCITT의 SDL과 ISO의 ESTELLE, LOTOS등이 가장 일반화되어 있다. 프로토콜이 실제로 구현되기 전 정의된 서비스를 제공할 수 있는지를 검증하고자 하는 프로토콜 검증 분야는 검증하고자 하는 프로토콜이 갖추어야 할 고유의 기능을 검사하는 verification 분야와 프로토콜 고유의 기능과는 관계 없이 프로토콜 규격이 지켜야 할 일반적인 성질을 검사하는 validation 분야로 구분되고 있다. 효율성과 신뢰성을 갖춘 통신 시스템의 구현을 목적으로 하는 프로토콜 구현분야에서는 현재 ADA, CHILL, OCCAM, MODULA-2, C, PASCAL과 같은 언어로 이미 검증된 프로토콜의 구현을 수행하며 앞에서 언급된 FDT 전용의 컴파일러, 디버거, 에디터등을 이용하여 구현되기도 한다. 마지막으로 프로토콜 테스트분야에서는 구현된 통신 시스템을 통신 선로를 통해 TM (test manager)과 연결하고 TM에서 준비된 테스트 시퀀스에 의해 시스템의 기능을 하나씩 시험하게 된다. 이러한 테스트는 IUT의 테스트 목적에 따라 conformance 테스트, performance 테스트, robustness 테스트, reliability 테스트의 4가지로 구분된다. Conformance 테스트는 IUT가 표준 프로토콜 규격에 일치하여 동작하는지의 여부를 검사하며 performance 테스트는 동일 프로토콜을 기반으로 구현된 다른 시스템과의 성능 비교를 수행하며 robustness 테스트는 다양한 에러 환경에서 어느 정도의 에러 회복 능력을 수행하는지를 검사하게 된다. 또한 모든 프로토콜 테스트의 수행을 위해서는 신뢰성 있는 테스트 시스템의 설계와 테스트 시퀀스의 정확한 기술 (description)과 효율적인 테스트 시퀀스의 산출을 위한 테스트 시퀀스 생성기가 필요하게 된다.<sup>[3,4,5,6]</sup>

## II. 프로토콜 적합성 검증 시험

### 1. 프로토콜 적합성 시험

OSI 프로토콜 적합성 시험을 위해 두가지의 적합성 제한 조건을 제시하고 있는데 static conformance requirements와 dynamic conformance requirements가 그것이다. static conformance requirements는 통신망에서의 상호 동작을 용이하게 하기 위해 허용되는 최소한의 기능(capability)을 정의하고 있으며 dynamic conformance requirements는 관찰 가능한 통신 시스템의 임의의 행동(behaviour)이 관련된 OSI 표준 프로토콜에 의해 허용되는지를 결정하기 위한 조건들이다.<sup>[3]</sup>

OSI 적합성 시험을 위해 시스템 구현자가 제시해야 하는 정보로서는 PICS와 PIXIT가 있다. PICS는 구현된 시스템의 기능과 선택사항들에 관련된 것으로 구현 단계에서 생략된 시스템의 특징들을 포함한다. PIXIT는 그 외의 IUT에 관련된 정보로서 테스트 환경구축에 필요한 정보를 갖게 된다.<sup>[3]</sup>

ISO가 제안한 OSI 적합성 시험방법은 시험대상이 되는 엔터티로부터의 관찰 가능한 출력과 제어 가능한 입력의 견지에서 기술되고 있으며 또한 제어와 관찰이 행해지는 엔터티의 SAP(service access point)를 규정함으로써 나타내어지고 이들 시험방법에 대해 시험대상이 되는 IUT는 하나의 계층일 수도 있고 여러개의 계층(multi layer)으로 구성될 수도 있다.

적합성 시험은 ISO에 의해 권고된 OSI 기준모델로 Local, Remote, distributed, Coordinated 테스트 방법의 4가지가 있다. 이들은 테스트 방법의 특이성으로 인해 고유의 장점들을 지니지만 또한 피할 수 없는 문제점들도 내포하게 되었다. 먼저 Local 테스트 방법은 가장 우수한 에러 검출 능력을 가지고 있지만 IUT와 테스트를 수행하는 테스트 시스템이 하나의 머신에 공존해야 한다는 문제점으로 인해 실질적인 테스트 수행에는 부적합 한것으로 나타났다. 둘째로 Remote 테스트 방법은 SUT(system under test)내에 테스트 수행을 위한 어떤 소프트웨어도 존재하지 않음으로 인해 전체 테스트 시스템의 구조를 간결화 하지만 결과적으로 테스트 능력이 크게 제한되고 말았다. Distributed, Coordinated 테스트 방법은 가장 보편화 되어 있고 그 구조적 형태도 적합성 시험에 알맞는 것으로 되어 있다. 그러나 이 두 방법 역시 SUT내에 테스트 수행을 위해 존재하게 되는 테스트 구성요소로서 UT의 구현을 요하게 되어 전체 테스트 시스템의 구조를 복잡하게 하며 SUT내에서 IUT의 상위계층을 담당하는 UT와 테스트 시스템의 LT와의 분리로서 다른 머신에 존재하는 LT와 UT의 동기화(synchronization)를 어렵게 하고 말았다. 따라서 SUT내의 UT와 테스트 시스템의 LT를 같은 머신에 통합하여 운용하고자 하는 시도가 이루어졌고 이것은 바로 SUT내에서 테스트 수행을 위해 요구되는 테스트 소프트웨어의 오버헤드(overhead)를 감소시키며 LT와 UT사이의 제어를 간결히 이룰 수 있으므로 전체 테스트 시스템의 간결화를 이루게 되었다. 본 논문에서는 SUT내에 존재하는 테스트 소프트웨어의 양을 줄임과 동시에 LT와 UT사이의 동기화를 강화하고 간결화하는 Ferry Principle<sup>[7]</sup>의 개념을 도입하여 프로토콜 적합성 검증 시스템을 구현하고자 하며 실질적인 데이터 측 테스트 시퀀스의 전송은 Ethernet과 TCP/IP 프로토콜을 기반으로 하는 이기종간의 LAN 상에서 이루어지도록 하였다.<sup>[3,4,5,6]</sup>

본 논문에서 구현하고자 하는 프로토콜 검증 시스템은 클라이언트, 서버 개념을 도입하여 구현하고자 하며 LT와 UT, TM이 존재하게 되는 테스트 시스템측을 테스트 클라이언트로 IUT가 존재하는 SUT내에 존재하여 테스트 기능을 수행하는 최소한의 기능 모듈을 테스트 서버로 명칭한다. 테스트 시스템의 전체적인 제어는 사용자 인터페이스를 담당하는 TM이 다루게 된다.

그림 1은 ISO와 CCITT에서 사용하는 대표적인 적합성 시험 절차를 나타낸 것이다. 첫번째 단계는 표준 프로토콜 규격과 관련된 static conformance requirements와의 일치 여부를 검사하기 위한 PICS의 분석과정이다. 두번째 단계는 테스트 선택과정으로 프로토콜 규격으로부터 추출된 적합성 테스트 시퀀스로부터 적절한 테스트 시퀀스를 선택하기 위해 PICS와 PIXIT가 이용되며 이 과정에서 추상적인 테스트 시퀀스가 실행가능한 테스트 시퀀스로 변환된다. 세번째 단계에서 실질적인 테스트 수행이 이루어지며 IUT의 기본적인 연결능력 테스트와 IUT의 관찰 가능한 기능들과 관련하여 PICS의 확립을 위한 capability 테스트 그리고 behaviour 테스트등이 이루어지게 된다. behaviour 테스트는 IUT의 올바른 행동패턴을 검사하는 과정으로 테스트의 핵심이 된다. 네번째 과정에서는 결과의 분석이 이루어지고 다섯번째 단계에서 최종적인 적합성 검토가 이루어지게 된다.

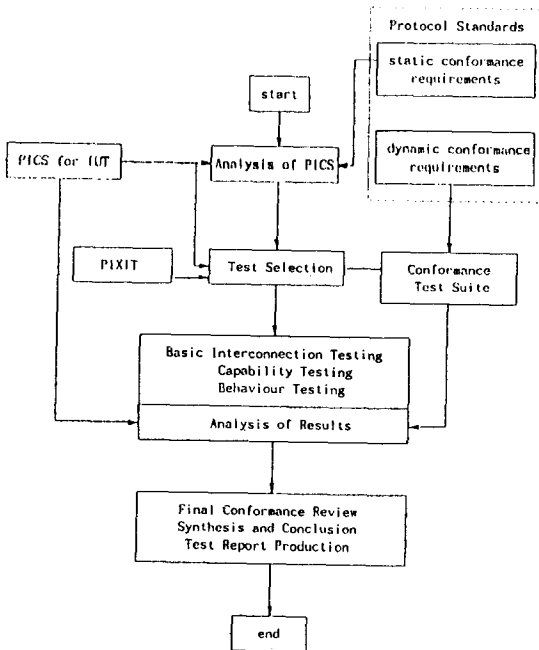


그림 1. 테스트 절차<sup>[3]</sup>  
Fig. 1. Test procedure.

### III. LAN을 이용한 프로토콜 적합성 검증 시스템의 구현

#### 1. 시스템 개요

구현된 프로토콜 적합성 검증 시스템의 전체 구조 및 검증하고자 하는 IUT, SUT 그리고 각 구성 요소간의 서비스의 관계를 그림 3에 나타내었다. 테스트 수행의 모든 제어는 TM이 담당하며 실제 테스트 수행에 필요한 테스트 시퀀스는 복수개의 UIO를 이용해 얻어지는 최적화된 테스트 시퀀스를 이용한다. 테스트 시스템에서 테스트 클라이언트는 테스트 서버의 상태를 요구하는 제어패킷을 전송하여 테스트 수행 중 테스트 서버의 상태를 파악할 수 있으며 테스트 서버가 갖고 있는 루프 백(loop back) 모드를 이용하여 Ethernet으로 연결된 LAN 상에서의 테스트 시퀀스의 전송 시간을 고려하게 된다. 또한 채널(channel)의 다중화(multiplexing)로 여러 계층으로 구성되는 IUT의 테스트 수행은 물론 서로 다른 계층의 동시적이면서도 독립된 테스트 수행도 이를 수 있도록 하였다. 본 연구에서 개발된 프로토콜 적합성 검증 시스템의 각각의 기능을 설명하면 다음과 같다.

OpenTask( )는 테스트 서버와의 채널을 확립하고자 하는 기능을 담당하며 사용프로토콜과 접속 상대 지정은 선택적으로 이루어진다. OpenTask( )에서는 connection ID를 이용해 최대 4개의 접속 기능을 가지게 된다. LoadTask( )는 입력 화일로 주어지는 테스트 시퀀스를 시스템으로 장착하는 기능을 수행하며 ModeTask( )는 테스트 클라이언트의 수행 모드를 결정하는데 앞서 설명한 대로 각각의 테스트 시퀀스의 적용과 그에 따른 IUT의 반응 결과의 추적을 요하는 경우에는 대화형(interactive) 모드로 그리고 모든 테스트 시퀀스의 TM에 의한 관리를 원하는 경우에는 자동화(automatic) 모드로 수행된다. CloseTask( )는 테스트 수행을 끝마치고자 할 때 또는 테스트 수행이 에러 발생등의 이유로 무의미 해졌을 때 전체 테스트 시스템의 수행 종결을 위해 모든 테스트 절차를 마무리 짓고 접속을 해제한다. SystemTask( )는 테스트 수행을 실질적으로 수행하는 모듈로 OpenTask( )에서 접속 설정한 상대와 connection ID를 이용해 LoadTask( )에 의해서 초기화된 테스트 시퀀스를 이용해 ModeTask( )에서 결정한 테스트 시스템 모드로 구동된다. 각각의 에러 상황은 경고음으로 확인된다.

이상은 테스트 클라이언트측의 모듈들을 설명한 것이고 테스트 서버측은 상대적으로 간단하다. 이것은 전체 테스트 시스템의 간결화를 의미한다. 테스트 서버측은 일련의 초기화 과정과 테스트 클라이언트측의 OpenTask( )에서 요구하는 접속 설정과 CloseTask( )에서 요구하는 접속 해제 기능에 응답하는 의무를 가지며

SystemTask( )의 제어에 응답한다. 이 외에 테스트 서버는 IUT와의 인터페이스를 담당한다. 또한 테스트 서버측 테스트 수행을 위해 필요로 되는 SUT내의 최소한의 기능 모듈은 테스트 클라이언트로부터의 제어 패킷에 응답하여 자신의 상황을 알리고 현재의 에러 상황을 보고하고 테스트 시퀀스의 전송 시간등을 파악하고자 하는 테스트 클라이언트의 요구에 응답해야 하는 의무를 가진다. 테스트 서버와 IUT 사이의 인터페이스에는 필연적으로 I/O 멀티플렉싱(multiplexing) 기능이 필요하며 경우에 따라서는 IUT 사이의 인터페이스를 변화 조정 할 필요가 있다.<sup>[8]</sup>

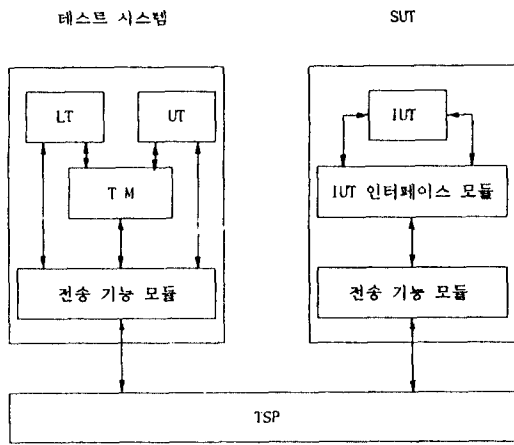


그림 2. 테스트 시스템의 각 구성 모듈과 그 서비스의 관계  
Fig. 2. Block diagram of conformance test system.

2. 시스템의 각 구성 요소간의 관계와 서비스의 정의  
전체 테스트 시스템은 구현 단계에서 계층 구조를 갖게 되어 각각의 기능 모듈은 스스로 하나의 계층의 역할을 담당하여 자신의 사용자에게 서비스를 제공하게 된다. 적합성 검증 시스템의 데이터 서비스는 테스트 클라이언트측의 LT와 UT 그리고 테스트 서버측의 IUT 인터페이스 모듈이 테스트 시퀀스와 그에 따른 IUT의 반응 결과를 송, 수신 할 수 있도록 하는 서비스로 정의된다. 사용자의 지시에 따라 TM이 테스트 클라이언트측의 전송 기능 모듈과 SUT내의 전송 기능 요구를 받아들일 수 있게 된다. 그림 2에 보이는 것처럼 이 서비스의 사용자는 테스트 시스템의 LT와 UT 그리고 SUT내의 IUT 인터페이스 모듈이며 서비스 제공자는 테스트 시스템과 SUT내에 각각 존재하는 전송 기능 모듈이다. 전송 기능 모듈은 TSP(transport service provider)의 도움을 받아 이

서비스를 수행한다. 적합성 검증 시스템의 관리 서비스는 전송 기능 모듈을 제어하기 위해 제공되는 서비스로 사용자는 TM이며 서비스 제공자는 테스트 시스템측의 전송 기능 모듈이다. TM은 이 서비스를 이용하여 전체 테스트 시스템을 제어 하면서 사용자의 요구에 맞춰 테스트를 수행하게 된다. 각각의 기능을 설명하면 먼저 접속 설정과 접속 해제 서비스는 사용자가 테스트 클라이언트와 테스트 서버간의 접속 채널을 확립하고자 하는 요구에 해당하는 서비스로 이들 역시 하위 계층에 해당하는 전달 서비스의 지원을 받는다. 제어 서비스는 LAN 상에서 패킷 데이터로 전송되는 테스트 시퀀스의 전송 시간을 고려하고자 할 때와 테스트 클라이언트가 테스트 서버의 상태를 확인하고자 할 때 사용되는 서비스로 전달 서비스로 전달 서비스의 구체적인 기능을 필요로 한다. 적합성 전송을 위해 전송기능 모듈에게 제공되는 서비스이다. 이들 서비스는 ISO 기준 모델 트랜스포트 계층의 서비스로 매핑되며 각각의 서비스들은 전송 기능 모듈이 자신의 사용자들에게 서비스를 제공 하기위해 요구되는 기능들을 담당한다.

3. 테스트 시스템의 제어와 테스트 시퀀스의 전달을 위한 패킷의 정의

테스트 클라이언트로부터 테스트 서버로 전송되는 패킷의 종류는 제어 패킷과 데이터 패킷의 둘로 구분된다. 제어 패킷은 테스트 서버의 상태를 파악하고 그에 따른 수행 절차와 에러 상황 등에 대한 테스트 클라이언트의 대처 그리고 LAN 상에서의 패킷의 전송 시간등에 대한 감시 기능을 위한 것이며 데이터 패킷은 IUT의 상위 계층과 하위 계층의 SAP로의 입력을 위한 테스트 시퀀스의 실질적인 전송을 담당하게 된다. 패킷의 구조는 다음 그림 3과 같다. Q는 데이터와 제어패킷의 구별을 나타

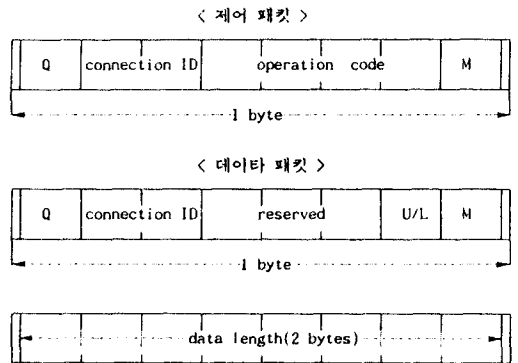


그림 3. 패킷의 구조  
Fig. 3. The structure of packet.

내고 U/L은 상, 하위계층의 구별을 위한것이고 M은 데이터가 계속 됨을 의미한다.

### IV. 실험 및 검토

#### 1. 접속 설정, 접속 해서 그리고 일련의 테스트 시퀀스 와 그 결과의 전송

다음 그림 4은 접속 설정에 따른 절차를 구현된 테스트 시스템의 함수들과 그 인터페이스의 관점에서 전송 기능 모듈을 중심으로 나타낸 것이다. 사용자로부터의 접속 설정 요구는 전송 기능 모듈이 TM에게 제공하는 Connect ( ) 함수에 의해 수행된다. 결과적으로 TM이 Connect ( )를 이용해 지정된 상대와의 접속 설정을 요구하게 된다. 다시 전송 기능 모듈은 TSP가 제공하는 ToLAN ( ), FromLAN ( ) 인터페이스 함수들을 이용해 아래 그림에서 보이는 것처럼 순서대로 ConnectReq; ConnectInd, ConnectRsp, ConnectCnf를 송, 수신하게 된다. 그 후 설정된 connection ID를 TM에 알린다.

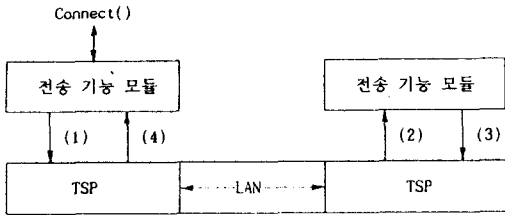


그림 4. 접속 설정 수행  
Fig. 4. Connection procedure.

다음 그림 5는 접속 해제에 따른 절차를 역시 전송 기능 모듈을 중심으로 나타낸 것이다. 유저의 접속 해제 요구는 전송 기능 모듈의 Disconnect ( ) 인터페이스 함수에 의해 ToLAN ( )을 거쳐 테스트 서버측의 전송 기능 모듈에게 FromLAN ( ) 인터페이스 함수에 의해 전달되며 Disconnect ( )의 호출은 테스트 서버측의 응답을 기다리지 않는다. 이것은 DisconnectReq, DisconnectInd의 송, 수신을 의미하며 바로 테스트 클라이언트의 TM에게 접속 해제 요구가 완료되었음을 알린다.

다음 그림 6은 전송 기능 모듈을 중심으로 한 테스트 시퀀스와 그 결과의 전송 절차를 보인다. 테스트 서버측의 전송 기능 모듈로 전송된 테스트 시퀀스는 IUT 인터페이스 모듈로 전송되어 IUT의 상 하위 계층으로 각각 전송되고 다시 IUT의 결과를 역시 전송 기능 모듈을 통

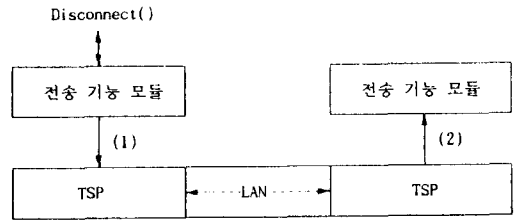


그림 5. 접속 해제 절차  
Fig. 5. Disconnection procedure.

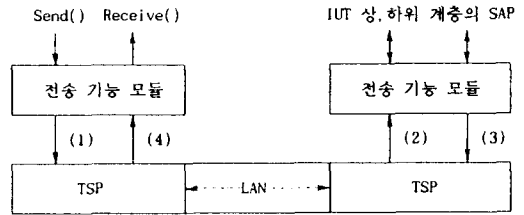


그림 6. 데이터의 송 수신  
Fig. 6. Data transmission / reception.

해 수신된다. 테스트 클라이언트측에서는 Receive ( ) 호출시 타이머의 동작이 시작되고 timeout 전에 IUT의 반응이 없는 경우에는 예상 출력이 없는 것으로 가정한다.

#### 2. UIO를 이용한 테스트 시퀀스의 산출<sup>[2]</sup>

UIO시퀀스는 유한 상태 머신(finite state machine)을 기반으로 테스트 시퀀스를 구하는 4가지 방법중의 하나로 시스템의 상태에 따른 특성을 반영한다. CCITT No. 7 TCAP 프로토콜의 적합성 검증 시험에는 이러한 UIO 테스트 시퀀스가 이용되었으며 그림 7에 트랜잭션 처리부에 대한 상태천이도를 나타내었고 그림 8은 그림 7의 상태 천이도에 따른 실제 테스트 시퀀스이다.

본 실험에서 사용된 상태천이도와 상태 천이표는 CCITT 권고안 Blue Book Q.774의 문답처리기능 응용부의 분석을 통해 이루어졌고 다시 이들 전체 테스트 시퀀스는 Q.713권고안에 따라 파라미터가 확립된 후 각 테스트 시퀀스는 실제 적용 가능한 테스트 시퀀스를 이룬게 된다.<sup>[9,10]</sup>

#### 3. CCITT No. 7 TCAP의 적합성 검증 시험<sup>[1]</sup>

CCITT에 의해 권고된 No. 7 공통신 신호방식은 디지털 통신망내에서 독립된 신호채널을 통해 다수의 사용자 정보 채널에 관련된 신호 정보 및 망 운용, 관리에 필요한 정보를 처리하는 국제적인 통신 프로토콜로서 음성 및 미

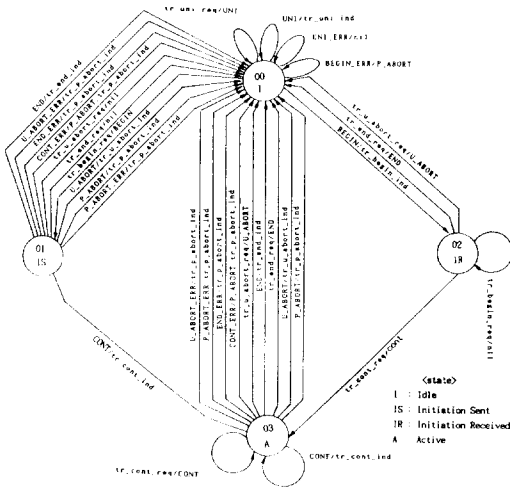


그림 7. 트랜잭션 처리부의 상태천이도  
 Fig. 7. State Transition diagram of transaction sublayer.

- [1] idle→is (tr\_begin\_req/BEGIN)
- [2] is→active (CONT/tr\_cont\_ind)
- [3] active→idle (P\_ABORT\_ERR/tr\_p\_abort\_ind)
- [4] idle→idle (BEGIN\_ERR/P\_ABORT)
- [5] idle→ir (BEGIN/tr\_begin\_ind)
- [6] ir→ir (tr\_begin\_req/nil)
- [7] ir→ir (tr\_begin\_req/nil)
- [8] ir→active (tr\_cont\_req/CONT)
- [9] active→active (CONT/tr\_cont\_ind)
- [10] active→idle (END\_ERR/tr\_p\_abort\_ind)
- [11] idle→idle (tr\_uni\_req/UNI)
- [12] idle→is (tr\_begin\_req/BEGIN)
- [13] is→active (CONT/tr\_cont\_ind)
- [14] active→idle (P\_ABORT/tr\_p\_abort\_ind)
- [15] idle→is (tr\_begin\_req/BEGIN)
- [16] is→idle (END/tr\_end\_ind)
- [17] idle→idle (UNI\_ERR/nil)
- [18] idle→is (tr\_begin\_req/BEGIN)
- [19] is→idle (END\_ERR/tr\_p\_abort\_ind)
- [20] idle→idle (BEGIN\_ERR/P\_ABORT)
- [21] idle→is (tr\_begin\_req/BEGIN)
- [22] is→active (CONT/tr\_cont\_ind)
- [23] active→active (CONT/tr\_cont\_ind)
- [24] active→idle (END\_ERR/tr\_p\_abort\_ind)
- [25] idle→is (tr\_begin\_req/BEGIN)
- [26] is→idle (U\_ABORT\_ERR/tr\_p\_abort\_ind)
- [27] idle→idle (tr\_uni\_req/UNI)
- [28] idle→idle (tr\_uni\_req/UNI)
- [29] idle→is (tr\_begin\_req/BEGIN)
- [30] is→active (CONT/tr\_cont\_ind)
- [31] active→idle (U\_ABORT\_ERR/tr\_p\_abort\_ind)
- [32] idle→idle (UNI\_ERR/nil)
- [33] idle→idle (UNI/tr\_uni\_ind)
- [34] idle→is (tr\_begin\_req/BEGIN)
- [35] is→active (CONT/tr\_cont\_ind)
- [36] active→idle (U\_ABORT/tr\_u\_abort\_ind)
- [37] idle→idle (tr\_uni\_req/UNI)
- [38] idle→is (tr\_begin\_req/BEGIN)
- [39] is→idle (tr\_u\_abort\_req/nil)
- [40] idle→is (tr\_begin\_req/BEGIN)
- [41] is→active (CONT/tr\_cont\_ind)
- [42] active→active (CONT/tr\_cont\_ind)
- [43] active→active (CONT/tr\_cont\_ind)
- [44] active→idle (END\_ERR/tr\_p\_abort\_ind)
- [45] idle→ir (BEGIN/tr\_begin\_ind)
- [46] ir→idle (tr\_end\_req/END)
- [47] idle→idle (UNI/tr\_uni\_ind)
- [48] idle→is (tr\_begin\_req/BEGIN)
- [49] is→idle (P\_ABORT\_ERR/tr\_p\_abort\_ind)
- [50] idle→is (tr\_begin\_req/BEGIN)
- [51] is→idle (U\_ABORT/tr\_u\_abort\_ind)
- [52] idle→idle (BEGIN\_ERR/P\_ABORT)
- [53] idle→is (tr\_begin\_req/BEGIN)
- [54] is→idle (CONT\_ERR/P\_ABORT, tr\_p\_abort\_ind)
- [55] idle→idle (UNI\_ERR/nil)
- [56] idle→is (tr\_begin\_req/BEGIN)
- [57] is→idle (P\_ABORT/tr\_p\_abort\_ind)
- [58] idle→idle (tr\_uni\_req/UNI)
- [59] idle→is (tr\_begin\_req/BEGIN)
- [60] is→active (CONT/tr\_cont\_ind)
- [61] active→idle (CONT\_ERR/P\_ABORT, tr\_p\_abort\_ind)
- [62] idle→idle (UNI/tr\_uni\_ind)
- [63] idle→is (tr\_begin\_req/BEGIN)
- [64] is→active (CONT/tr\_cont\_ind)
- [65] active→idle (tr\_u\_abort\_req/U\_ABORT)
- [66] idle→idle (UNI\_ERR/nil)
- [67] idle→is (tr\_begin\_req/BEGIN)
- [68] is→active (CONT/tr\_cont\_ind)
- [69] active→active (tr\_cont\_req/CONT)
- [70] active→active (CONT/tr\_cont\_ind)
- [71] active→idle (END\_ERR/tr\_p\_abort\_ind)
- [72] idle→idle (UNI/tr\_uni\_ind)
- [73] idle→is (tr\_begin\_req/BEGIN)
- [74] is→active (CONT/tr\_cont\_ind)
- [75] active→idle (tr\_end\_req/END)
- [76] idle→ir (BEGIN/tr\_begin\_ind)
- [77] ir→idle (tr\_u\_abort\_req/U\_ABORT)
- [78] idle→ir (BEGIN/tr\_begin\_ind)
- [79] ir→active (tr\_cont\_req/CONT)
- [80] active→idle (END/tr\_end\_ind)
- [81] idle→idle (BEGIN\_ERR/P\_ABORT)
- [82] idle→is (tr\_begin\_req/BEGIN)
- [83] is→idle (tr\_end\_req/nil)
- [84] idle→idle (UNI/tr\_uni\_ind)

그림 8. 트랜잭션 처리부에 대한 테스트 시퀀스  
 Fig. 8. Test sequence for transaction sublayer.

음성 교환 서비스를 위해 교환기간 신호 기능을 수행할 뿐만 아니라 망을 지능화하면서 동시에 효율적으로 망을 운용, 관리하기 위해 교환기와 망내 특수 설비(데아타베이스, 운용유지보수 센터 등)간에 사용되는 정보 전송을 제어한다.

No. 7 신호 시스템은 크게 하위계층 프로토콜(계층 3)인 망서비스부(network service part)와 상위계층 프로토콜(계층 4)인 사용자부(user part)로 나누어진다. 본 논문의 시험 대상인 문답처리기능 응용부는 착신과금 서비스나 신용통화 서비스와 같은 지능망 서비스를 제공하기 위해 1988년 CCITT에 의해 Q. 771~Q. 775에 권고된 프로토콜로서 No. 7 신호방식의 계층 4에 속한다. 문답처리기능 응용부는 크게 트랜잭션 처리부와 컴포넌트 처리부로 구성된다. 본 논문에서는 1990년 한국전자통신연구소의 위탁과제로 고려대학교에서 구현한 문답처리기능 응용부 시스템의 트랜잭션 처리부에 대해 실험을 수행하였다.

테스트 시퀀스는 LAN상에서 CSMA/CD 방식을 따르는 Ethernet을 경유하여 테스트 서버와 함께 수행되고 있는 IUT에 전달되고 타임 아웃등에 의한 비정상적인 경

우가 아니라면 IUT의 응답을 전달하는 방식으로 전체 테스트의 수행이 이루어졌다. 테스트 시스템의 UT는 트랜잭션 처리부의 상위에 위치하는 컴포넌트 처리부의 기능을 담당하고 LT는 트랜잭션 처리부의 하위에 위치하는 신호 연결 제어부의 기능을 담당하여 에뮬레이션에 의한 수행이 이루어졌다. 이와 같이 구성된 테스트 시스템을 이용하여 적합성 시험을 수행한 결과 IUT의 타이머측에 의한 오류가 확인되었고 이의 수정 후 트랜잭션 처리부는 권고안의 규격과 일치하여 동작됨이 확인되었다. 복수개의 UIO를 이용한 테스트 시퀀스에 의한 전체적인 IUT의 적합성 시험외에 테스트되었던 세부 항목들은 다음과 같다.

- (1) 비구조형 다이얼로그에 대한 테스트
- (2) 구조형 다이얼로그를 시작하고 확립된 다이얼로그를 종료한 경우에 대한 테스트
- (3) 구조형 다이얼로그를 시작하고 확립된 다이얼로그를 중단한 경우에 대한 테스트
- (4) 비정상 상황 처리에 대한 테스트
- (5) 확립된 트랜잭션에 대한 무응답에 대한 테스트

## V. 결 론

현대의 통신망이 필연적으로 신뢰성과 정확성을 갖춘 통신 프로토콜을 요구하게 됨에 따라 통신 프로토콜의 규격화와 체계화가 일반화 되었고 이러한 통신 프로토콜을 바탕으로 구현된 시스템은 상용화 이전에 표준 프로토콜 규격과 부합되게 구현되어 동작하는지의 여부를 검사하는 적합성 시험이 이루어져야 한다. 또한 이러한 적합성 시험은 최적화된 환경에서 효율적인 테스트 시퀀스에 의해서 이루어 질때 그 의미를 갖게 된다.

따라서 본 논문에서는 신뢰성 있는 전송 프로토콜, 낮은 에러 발생률, 높은 에러 회복 능력, 고속 전송 매체등으로 특징 지워지는 LAN을 이용함으로써 신뢰도 높은 프로토콜 적합성 테스트 시스템(protocol conformance test system)을 구현하여 시스템의 유용성과 신뢰성을 확인하고자 하였다. 이 기종간의 LAN을 구성한 이유는 시스템이 통합화되는 경향에 부응하고자 하는 목적외에 시스템 사용자의 유저 인터페이스를 쉽게 강화할 수 있다는 장점때문이었다. 이외에 시스템의 확장성과 간결성의 문제도 선택에 큰 비중이 되었다. 본 논문에서 구현된 한

응용 프로그램으로서의 유용성은 CCITT No. 7 TCAP 프로토콜에 적용해 어느정도는 확인이 되었으나 다단 접속(multiple connection)의 기능이나 다층(multi layer) 구조의 프로토콜 적합성 시험은 이루어지지 못했다.

본 논문을 통해 LAN의 한 응용 프로그램(application program)으로서 구현된 프로토콜 적합성 테스트 시스템은 통신 시스템의 기능 구현시 시스템 개발의 효율성을 높일 수 있을 것으로 기대된다.

## 參 考 文 獻

- [1] 유명식 "CCOTT No.7 신호 시스템의 문답처리 기능 응용부(TCAP) 기능구현," 석사 학위논문, 1990. 고려대학교.
- [2] 기장근 "프로토콜 적합성 검증을 위한 최적 시험절차 추출 자동화에 관한 연구" 박사 학위논문 1991. 고려대학교.
- [3] D. Rayner, "OSI conformance testing," Computer Networks and ISDN systems 14, pp. 79-98, 1987.
- [4] Gregorr Bochmann, "Protocol specification for OSI", Computer Networks and ISDN Systems 18, pp. 167-184, 1990.
- [5] Richard J. Limn. Jr, "Conformance testing for OSI protocols", Computer Networks and ISDN systems 18, pp. 203-213, 1990.
- [6] "Conformance testing : architecture and test sequence," Computer Networks and ISDN Systems 18, pp. 203-213, 1990.
- [7] H.X. Zeng & S.T. Chanson, B. R. smith, "On ferry clip approaches in protocol testing," Computer Networks and ISDN Systems 17, pp. 77-88, 1989.
- [8] W. Richard Stevens, "UNIX network programming", Prentice-Hall, 1990.
- [9] CCITT Blue Book recommendation Q.771-Q.775, 1988.
- [10] CCITT Blue Book recommendation Q.700-Q.716, 1988.
- [11] John Henshall, "OSI explained : End to end computer communication standards", Halsted Press, 1988.

著 者 紹 介



尹 載 日 (學生會員)

1967년 11월 19일생. 1989년 8월 고려대학교 전자전산공학과 졸업. 1992년 8월~현재 고려대학교 전자공학과 대학원졸업(공학석사) 주 관심분야 : Computer Network, Operating System, Parallel

Processing 등임.



盧 承 煥 (正會員)

1962년 8월 19일생. 1987년 8월 고려대학교 공과대학 전자공학과 졸업(공학사). 1989년 8월 고려대학교 전자공학과 대학원 졸업(공학석사). 1989년 9월~현재 동대학원 박사과정 재학중. 주관심분야 : 정보통신, CAT(Computer Aided Test) 등임.

CAT(Computer Aided Test) 등임.

金 惠 鎮 (正會員)

1933년 12월 12일생. 1957년 3월 서울대학교 공과대학 전자공학과 졸업(공학사). 1962년 1월 미국 일리노이 공대 대학원 전자공학과 졸업(공학석사). 1972년 2월 고려대학교 대학원 전자공학과 졸업(공학박사). 1962년 6월~1967년 9월 한국 원자력 연구소. 1967년 9월~1971년 2월 서울대학교 전임강사 및 조교수. 1971년 3월~현재 고려대학교 부교수 및 교수. 1984년 9월~1987년 9월 고려대학교 생산기술연구소 소장. 1985년 대한전자공학회 회장. 1985년 3월~1988년 한국전기통신공사 이사. 1985년 6월~1991년 12월 한국전자통신연구소 부이사장. 1990년 3월~현재 고려대학교부설 정보통신기술공동연구소 소장. 주관심분야 : 정보통신, 컴퓨터구조, 문자인식 등. 28권 B편 8호 참조.

趙 顯 俊 (正會員)



1963년 1월 28일생. 1985년 2월 고려대학교 전자공학과 졸업(공학사). 1987년 8월 고려대학교 대학원 전자공학과 석사졸업(공학석사). 1991년 9월~현재 고려대학교 대학원 박사과정. 1989년 1월~1991년 8월 (주)

쌍용컴퓨터 시스템연구소 연구원. 주관심분야 : Computer Network, ISDN, 지능망, 컴퓨터구조 등임.