

# 대덕 지역정보망에의 OSI 상위계층 프로토콜 접속 연구

최선환<sup>o</sup>, 박승철, 채의근, 한기수, 최양희, 이영희

한국전자통신연구소

## An OSI IMPLEMENTATION ON DAEDUK SCIENCE COMPLEX NETWORK.

Sunwan Choi, Seungechul Park, Yigeum Chae, Kisoo Han, Yanghee Choi and Younghee Lee

Electronics and Telecommunications Research Institute

### 요 약

대덕망에 OSI 프로토콜을 구현하기 위한 일련의 작업들은 ETRI의 OSI Group에 의해서 1985년경부터 시작이 되었다. 현재 OSI의 FTAM(File Transfer, Access and Management) 프로토콜과 ACSE(Association Control Service Element), presentation 계층, session 계층이 상위계층 프로토콜로 구현되었으며, 상위계층에서 필요한 통신 서비스를 제공하기 위해서 transport 계층 및 대덕 MAN의 통신 서비스를 이용하기 위한 interface들이 구현되었다. 이러한 interface는 순차적으로 dual ring optical cable을 control 하여 대덕 MAN에 위치한 node들 사이에 OSI 상위계층 프로토콜의 서비스를 제공하게 된다.

### 1. 서 론

대덕 MAN은 대덕 지역의 교육기관 연구기관을 연결하는 고속도의 통신망으로서 현재 망구성을 위해서 필요한 연구 작업들이 진행중에 있으며, 1988년말경부터 시험 운용할 예정에 있다. 초기의 작업은 3개의 node 사이에서 광 케이블을 이용한 기본적인 데이터 통신 서비스와 video 서비스등을 제공하기 위한 작업이 수행되었다.

이러한 일련의 작업들을 효율적으로 수행하기 위해서 대덕 MAN 구성팀은 4개의 그룹으로 세분화되었으며 각각 망구성, 망접속장치 개발, 응용계층 지원, video 서비스 구현등의 연구 개발을 위한 작업들을 수행하였다. 첫번째의 망구성 그룹은 대덕 연구단지내에 광케이블의 포설 및 광케이블 접속장치(FIM: Fiber Interface Module)의 개발을 위한 연구를 수행하였고, 두번째의 망접속장치 개발 그룹은 망구성 그룹에서 개발한 FIM의 서비스를 이용하여 host computer나 terminal에게 망접근 기능을 부여할수 있는 망접속장치(RIU: Ring Interface Unit)의 개발을 위한 연구를 수행하였다. 세번째의 응용계층 지원 그룹은 망접속장치를 이용하여 MAN에 접속된 컴퓨터들 사이에서 OSI FTAM 프로토콜을 이용한 파일 전송 서비스를 지원하기 위한 연구를 수행하였다. 여기에서는 OSI 참조 모델에 입각한 파일전송 서비스를 제공하기 위하여 FTAM 프로토콜 및 FTAM이 필요로하는 제반 프로토콜의 구현이 고려되었다. 즉 FTAM에서 필요로 하는 서비스를 제공하기 위하여, ACSE, presentation, session등이 상위계층 프로토콜로 고려되었으며 FTAM에서 필요한 데이터의 표현 전송을 위하여는 ASN.1 (Abstract Syntax Notation One)이 고려되었다. 하부계층 프로토콜은 session 프로토콜을 지원하기 위한 transport 계층 및 MAN 서비스를 이용하기 위한 Ring Interface Unit의 host interface가 고려되었다.

상위계층 프로토콜의 구현을 위한 작업은 specification, implementation과 testing의 phase로 나뉘어서 수행되었으며 FTAM을 위한 specification을 만드는 작업은 NBS, SPAG등의 기관에서 만들어진 profile과 호환성을 갖도록 하였으며, 이렇게 작성된 specification에는 primitive, parameter 기타 사항들이 포함되게 된다.

상위계층 프로토콜의 구현을 효율적으로 수행하기 위해서 ISO에서 개발한 formal description technique(FDT)인 ESTELLE를 적용하였으며, ESTELLE로 작성된 specification에서부터 C-code를 생성하는 자동 생성 기법을 이용하였다. ESTELLE specification을 C-code로 변환하기 위해서 NBS에서 개발된 ESTELLE-C compiler가 이용되었으며, ESTELLE-C compiler를 이용해서 프로토콜 상태 변화등의 기본적인 프로토콜 machine을 생성하고, 상.하 interface등을 직접 C-Language를 이용하여 coding 하는 방법을 사용하였다. 여기서 ESTELLE-C compiler가 생성하는 프로토콜 machine은 전체의 약 40% 정도가 되며 나머지 60% 정도는 상하위 interface 및 프로토콜 machine을 제외한 부분이며 이러한 부분은 생성된 code에 부가해서 구현을 하였다.

향후에는 대덕망에 FTAM 뿐만 아니라, Message Handling System, Office Documentation Architecture, Transaction Processing등의 프로토콜들도 적용되리라 예상되며 프로토콜의 성격에 따라서 LOTOS등의 FDT 적용도 고려되고 있다.

### II. 구현을 위한 specification

구현을 위한 specification이라함은 functional unit, service primitive, parameter, service 정도, 초기치등 구현시 필요한 모든 부분을 고려하여 작성된 specification을 말하며 본 장에서는 여기에서 고려한 각 프로토콜들에 대한 기본적인 specification을 기술하였다.

#### FTAM

기존의 file system들이 제공하는 서비스의 차이점들을 보완하기 위한 파일전송 서비스의 표준화를 말한다. 즉 기존의 파일 시스템들이 제공하는 attribute, 접근방식, 구조등의 차이점을 해결하기 위한 모든 작업을 위하여 공통의 가상 파일 저장소가(VFS: virtual file store) 정의되고 FTAM 프로토콜은 VFS의 특징을 고려하여 정의되며 실제 구현에서는 VFS와 각 시스템의 RFS(Real File System)간의 mapping을 이용하여 파일전송 관리 서비스를 수행하게 된다. 구현을 위한 FTAM specification은 ISO 8571을 참조하여 작성되었으며 SPAG의 A.111 profile과 호환성을 갖도록 작성되었다. FTAM 구현을 위한 specification은 아래와 같다.

Constraint Set: unstructured

Abstract Syntax: ISO 8571-FTAM, ISO 8571-FADU, Simple Text, Simple Binary, NBS-AS1

Document Type: FTAM-1, FTAM-3, NBS-1, NBS-2

#### ACSE

ISO 8649에 정의된 서비스를 제공한다.

ASN.1 Encoding/Decoding

FTAM-PDU, FTAM-FADU, ACSE-PDU, PRESENTATION PDU의 encoding/decoding에는 ISO8825의 ASN.1 Encoding 규칙이 적용되었다.

Presentation

Presentation kernel functional unit만을 제공한다.

Session

- o Functional Unit: kernel과 duplex functional unit
- o Segmentation
- o Basic Concatenation

Transport

Transport Class()가 제공된다.

III. Formal Protocol Specification

상위계층 프로토콜의 구현을 위해서는 ISO에서 개발된 ESTELLE FDT가 사용되었다. ESTELLE은 extended Finite State Machine에 근거한 모델로서 ISO pascal programming language와 몇개의 상호 연관적인 modular 구조로 구성된다. ESTELLE로 작성된 specification은 몇개의 상호 관련성을 갖는 개체들의 집합으로 구성되며, 이러한 모듈들은 양방향의 채널을 이용하여 상호 작용하게 된다. 모듈은 ESTELLE specification의 기본적인 구성 요소로서 specification에서 특정 개체를 표현하게 된다.

모듈의 기본적인 구성요소는 header와 body로서 header에는 body에서 사용되는 identifier 뿐만 아니라 모듈 header와 관계된 body를 나타내기 위해서 header-identifier도 포함한다. 모듈 header에는 모듈이 처음 시작될때 주어지는 모든 파라미터를 전달받기 위해서 formal 파라미터가 정의되며 다른 모듈에서 사용할 변수의 선언 및 다른 모듈과의 통신을 위한 interaction point들이 정의된다. 모듈 body에는 다음과 같은 것들이 기술된다.

- o **declaration part:** constant, variable, procedure, channel, module, state definition part
- o **initialization part:** local variable 및 interaction의 초기화
- o **state transition part:** "when", "from", "to", "priority", "provided" 등의 절이 사용되며 각 event들에 따라서 state 변환이 정의된다.

모듈은 몇 개의 부모모듈로의 구성이 가능하며 부모모듈 역시 계속적으로 부모모듈로의 구성이 가능하다. 이러한 ESTELLE의 구조에서 "specification"이라는 것은 최상위 level의 모듈이 된다. 여기에서는 각각의 프로토콜들이 "specification"으로 구성되어 있으며 FTAM과 ACSE의 "specification" 구조는 다음과 같다.

FTAM, ACSE, Presentation의 Formal Specification

OSI 응용계층에서, 응용개체라는 것은 진행중인 응용 프로세스의 통신부분을 표현하는 부분을 말하는데 FTAM 응용개체는 FTAM 서비스와 ACSE로 구분된다. 하나의 FTAM 응용개체의 invocation은 다른 응용개체의 invocation과는 독립적인 통신 기능을 수행하게되며, 하나의 응용개체의 invocation 역시 하나의 응용-관계(application association)에 대응하게 된다. 이것은 여러개의 관계들이 동시에 존재할수 있다는 것을 말한다.

FTAM 응용개체는 그 역할에 따라서 initiating entity와 responding entity로 나뉘어진다. Initiating entity는 가상의 파일 저장소(VFS)에 접근하고자 하는 응용 프로세스가 필요로 하는 통신 서비스를 제공하고, responding entity는 initiating entity에 대응해서 수동적으로 동작하기 위한 서비스를 제공하며 두개의 개체(entity)들은 상호 독립적으로 수행된다.

이러한 FTAM 응용개체의 특징에 따라서 본 구현에서는 initiating entity의 기능을 수행하는 I-ENTITY를 위한 specification과 responding entity의 기능을 수행하는 R-ENTITY를 위한 specification으로 분리하였다. 각 specification에는 최상위 level을 의미하는 "specification"이라는 keyword가 사용되며 몇개의 모듈로 세분된다. <그림1>은 I-ENTITY와 presentation specification을 나타낸다.

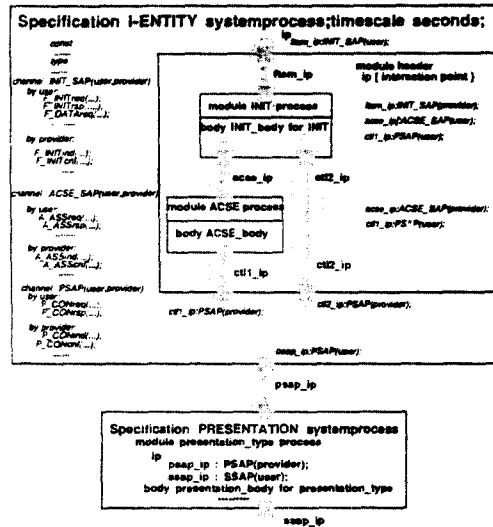


그림1 I-Entity, Presentation Formal Specification

I-ENTITY 모듈은 INIT 부모모듈과 ACSE 부모모듈로 구성되는데, 각각 initiator 역할을 수행하기 위한 FTAM 서비스 요소의 부분적인 특징과 ACSE의 특징을 나타낸다. FTAM과 같은 응용개체에게 표현 서비스를 제공하는 표현계층 개체가 각각의 응용개체와 통신하기 때문에, FTAM 개체 역시 INIT나 ACSE 모듈들에게 표현계층에서 전달되는 서비스를 전달하기 위한 function들이 필요하며 I-ENTITY specification내의 제어 function이 이러한 기능을 수행하게 된다.

R-ENTITY는 RESP와 ACSE 모듈로 구성되는데 여기서 RESP 모듈은 FTAM 개체의 수동적 역할을 나타낸다. 각 ENTITY(I와 R) specification에 대해서, 하나 또는 그 이상의 instance들이 비동기적으로 동작될수 있으며 각각의 instance들은 응용개체의 하나의 invocation을 나타낸다.

I-ENTITY내에는 INIT\_SAP, ACSE\_SAP, PSAP의 3개의 channel이 정의된다. INIT\_SAP은 FTAM 사용자 모듈과 INT 모듈 사이, ACSE\_SAP은 INIT 모듈과 ACSE 모듈 사이, PSAP은 I-ENTITY와 ACSE 모듈, presentation 모듈, INIT 모듈들 사이의 통신을 위한 channel로 정의된다.

모듈간의 통신은 모듈의 header부에 정의된 하나 또는 그 이상의 양방향의 링크는 두개의 interaction point를 연결함으로써 생성된다. 채널 정의부에서 정의되는 interaction primitive들에 대해서는 각각의 interaction primitive들이 grouping되며, 이때 grouping된 primitive들이 user로서 사용될것인지 또는 provider로서 사용될것인지가 interaction point에 정의된다. <그림1>에 4가지 예의 interaction point들이 도시되어 있으며, 여기에서 presentation 모듈은 I-ENTITY와 session 프로토콜을 위한 session 모듈과 통신을 하는 제어 모듈을 포함하는 specification으로 구성되어 있다.

## Session과 Transport의 Formal Specification

Session specification은 session 모듈과 세계의 외부 모듈(user 모듈, transport 모듈, timer 모듈)과 세계의 interaction point로 구성되는데 각각 channel 정의부에 정의된 session 서비스를 위한 *ssap\_id*, transport 서비스를 위한 *tsap\_id*, timer 서비스(timer start, cancel, response)를 위한 *timer\_ip*로 구성된다.

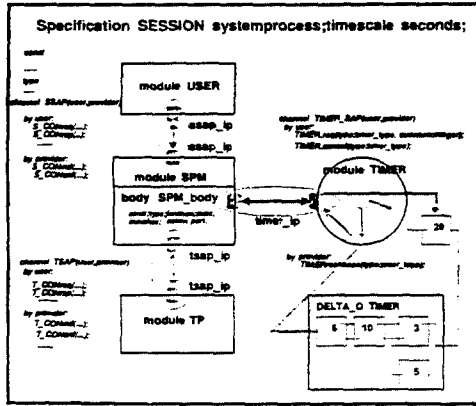


그림2 Session Formal Specification Structure

<그림2>가 session specification의 구조를 나타내며 이것은 FTAM specification의 구조와 비슷한 구조를 갖는다. 여기에서 SPM 모듈이 timer 서비스를 필요로 하며, 구체적으로는 session 프로토콜 machine(SPM)이 session abort 서비스를 요청하거나 abort를 요구하는 session protocol data unit(abort SPDU)를 받았을 때 필요하다. Transport specification 역시 session specification과 비슷한 구조로 구성되었다.

## IV. 구현

본 장에서는 ESTELLE로 기술된 formal specification과 ESTELLE-C compiler를 이용하여 상위계층 프로토콜을 구현하는 과정에 대해서 기술하였다. 여기에서 고려된 대덕 지역망의 망구조는 아래의 <그림4>와 같다.

### ESTELLE-C compiler를 이용하는 반자동생성 방식

앞장에서 기술한 바와 같이 ESTELLE-C compiler는 ESTELLE로 기술된 specification을 가지고 C-code를 생성한다. 이 compiler는 또한 각 모듈의 동작을 표현하는 Extended State Machine과 interaction들을 위한 자료구조들을 생성하게 된다. 이렇게 생성된 code는 추상적인 데이터 형태의 instance를 표현하게 되는데 여기에서 추상적인 데이터 형태란 기본적인 interaction들과 통신하는 확장된 finite state machine들에 의해서 표현되는 모듈 instance들의 동적인 체계를 나타낸다. <그림3>에 이러한 일련의 과정들이 나타나 있다.

구현과정에 걸쳐서, 모든 source code가 ESTELLE-C compiler에서 생성되는 것이 불가능하기 때문에 실제 구현자들은 specification과 독립적인 부분들(library function, data type, input/output part)과 buffer management, event processing routine과 timer library등을 제공하여야 한다.

Compiler에 의해서 생성된 code들은 FSM library들을 이용하는 형태로 생성되며, library에는 transition 수행을 위한 scheduling, 모듈 instance들 사이의 primitive interaction의 상호 전달, queuing, 모듈 instance의 initialize/release등을 수행하는 기능들을 제공한다.

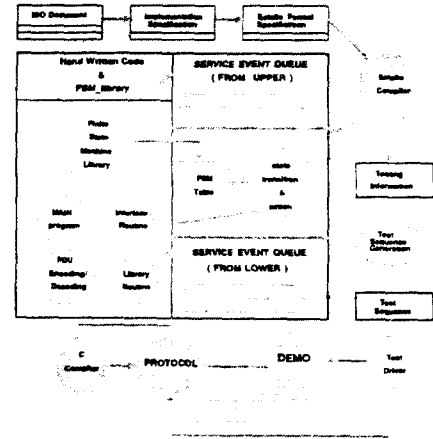


그림3 Implementation Procedure

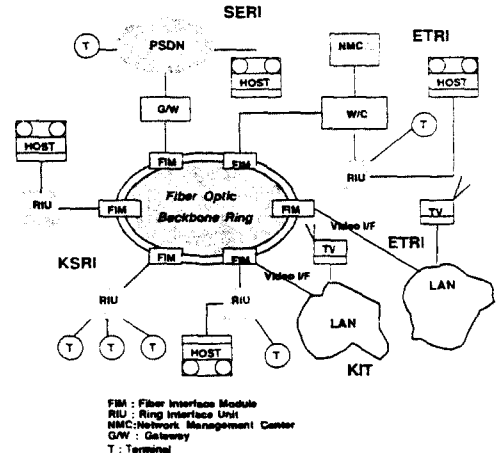


그림4 대덕망구조

## V. 결론

본 논문에서는 대덕지역망을 고려한 OSI FTAM, ACSE, Presentation, Session, Transport 계층의 구현사항과, Estelle FDT를 이용한 설계, 그리고 Estelle-C Compiler를 이용한 구현방법에 대하여 기술하였다. 이렇게 Protocol Engineering에 따른 프로토콜의 구현은 OSI protocol specification의 변경에 따른 확장과 수정의 용이함은 물론 protocol의 흐름을 파악하기 쉽고 자동생성기법을 통해 프로그램이 자동생성되므로 구현 시간을 단축할 수 있다.

본 논문에서는 대덕지역망에 OSI FTAM 프로토콜을 이용한 파일전송/관리 서비스를 대덕망의 사용자에게 제공하기 위한 프로토콜과 FTAM 프로토콜을 대덕망의 노드들 사이에서 수행하기 위해서 필요한 제반 상위계층, 하위계층 프로토콜의 구현에 대해서 기술하였다. 향후 대덕지역망의 노드들 사이에서 표준화된 프로토콜을 이용하여 사용자들에게 많은 서비스를 제공하기

위해서는 FTAM 프로토콜 뿐만 아니라 Message Handling System, Document Exchange등의 응용 프로토콜의 구현이 고려되며 이러한 프로토콜들의 구현시에는 본 연구과정에서 구현한 ACSE, presentation, session, transport 및 ASN.1 encoding library등의 응용계층을 지원하기 위한 프로토콜 소프트웨어들이 이용 또는 참조될수 있다.

#### REFERENCES

- [1] ISO 7498, Information Processing Systems - Open Systems Interconnection - Basic Reference Model
- [2] ISO 9074, Information Processing Systems - Open Systems Interconnection - Estelle - A Formal Description Technique Based on an Extended State Transition Model
- [3] S. Park, S. Choi, K. Han, and Y. Choi. "A Systematic Approach to Implement OSI FTAM Application Entity," Winter Computer Communications Workshop, Feb. 1988.
- [4] Y. Choi, S. Choi, K. Han, S. Park, Fundamental Study on ISO Upper Layer Protocols: Final Report, May 1988.
- [5] S. Chung, H. Park, and etc., A Study on Construction of Metropolitan Area Network: Final Report, May 1988.