

컴퓨터 통신 기술

安 淳 臣, 朴 仁 燮

(正 會 員)

高麗大學校 工科大学 電子電算工學科

I. 서 론

1970년대와 1980년대 초반까지의 컴퓨터 산업의 변화를 우리는 보았다. 컴퓨터와 통신이 결합하면서 네트워크를 이용한 컴퓨터 통신의 등장은 종래의 통신방식에 일대 변혁을 가져오게 되었으며 이를 컴퓨터 통신혁명(computer communication revolution)이라고 불러주는 까닭은 다음과 같다.¹⁾

- 데이터 프로세싱(computers)과 데이터 커뮤니케이션(transmission vs switching equipment)간의 근본적인 차이점을 더 이상 찾기 힘들며

- 데이터, 음성 및 화상 통신간의 근본적 차이점을 찾기 힘들며

- Single processor computer, multiprocessor computer, local network, metropolitan network, LHN(long-haul network)에 사용되는 통신선로(line)의 구분이 점차 모호해지고 있는 실정이다.

이러한 변화는 컴퓨터 생산업체와 데이터 통신 관련 업체간의 병합을 유도하여 하나의 산업으로 탈바꿈되어가고 있는 실정이며, 기존의 컴퓨터 생산업체들은 자사의 컴퓨터 기종에 네트워크 기술을 가미시켜 각기 독자적인 네트워크 체제를 개발하여 공급하고 있다. 서로 호환성이 없는 네트워크가 범람하면서 각기 다른 네트워크를 사용하는 사용자간에 불편이 따르게 되고 지역단위와 국가단위의 통일된 네트워크 구성이 어렵게 되자 네트워크 구성에 있어서 표준화 문제가 대두되었고, 국제 표준화에 대한 연구가 ISO(international standard organization)과 CCITT(consultative committee on international telegraphy and telephone)를 중심으로 진행되어 데이터 통신을 위한 표준화 모델 및 권고사항등이 제창되었다.

본 고에서는 컴퓨터 통신을 위해 필요한 계층적 구조와 각 계층내의 구조를 2,3절에 설명하였고, 통신기

능의 구현은 link layer를 실제로 들었으며, 구현단계까지의 가능한 오류를 검출하는 검증(verification)과 이러한 통신기능들과 분산 시스템간의 관계, 마지막으로 현재 개발되어 운용되고 있는 네트워크들에 대해 전체적인 소개 형식으로 기술하였다.

II. 통신의 계층적 구조

계층화는 두 시스템간의 통신에 필요한 기능을 layering하여 각 계층을 수직적으로 배열하고, 가장 아래의 계층으로부터 상위의 계층으로 올라 갈수록 제공하는 기능의 범위 및 성격이 고도화 되도록 한 것이다.²⁾

예를 들어 설명하면, 아프리카 케냐의 원주민과 인도 사람이 국제간 통신선로를 이용하여 전화통화를 한다고 가정해 보자. 이를 서로 다른 O/S를 가지는 두 컴퓨터 시스템과 대응하여 생각해 보면 더욱 이해가 쉬울 것이다. 이때 물론, 케냐 원주민의 말을 인도사람이 알아들을 수 없기 때문에 두 사람간의 원활한 의사소통을 위해서는 중간단계에 통역관 및 전화 시스템이 필요할 것이다. 이의 예를 그림 1에 도식하였다.

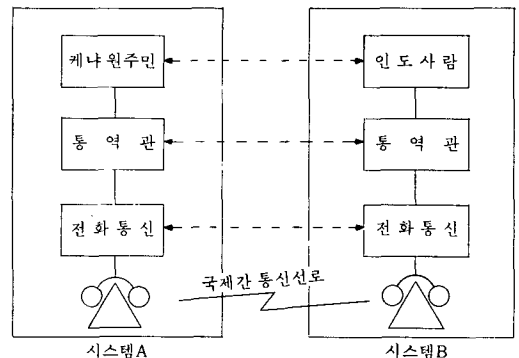


그림 1. 계층화의 예

이때 각 시스템은 서로 다른 3개의 기능으로 구분하여 볼 수 있으며, 이 각각의 기능을 계층이라고 하며, 상위 계층간의 통신을 위해서는 하위 계층의 서비스를 받아야만 대화를 할 수 있음을 알 수 있다. 이러한 계층화는 컴퓨터 네트워크 구조에 사용되는 기본 방법으로 아키텍처에서 사용되는 통신제어에 대한 분해방법을 나타낸다. 이제 네트워크에서 사용하는 계층화의 개념을 좀더 자세히 살펴보자. 계층화를 이용한 시스템의 설계는 유용한 소프트웨어 기술의 하나로써 널리 사용되어 왔다. 계층화를 통해 복잡한 시스템을 몇개의 층으로 나누어 설계 및 구현함으로써 여러 가지 잇점을 얻을 수 있다. 계층화 설계의 개념은 그림 2 처럼 나타낼 수 있다.¹⁾

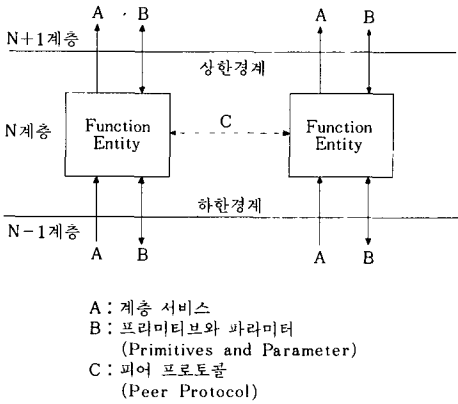


그림 2. 계층 구조의 개념

계층 구조에서 각 계층은 N+1 계층을 상위 계층으로 하고 N-1 층을 하위 계층으로 하는 독립적인 N 계층으로 생각할 수 있다. 여기서 각 계층은 하위 계층의 서비스를 받아 구현되며 상위 계층에서 필요로 하는 서비스를 제공해 준다. 이때 N 계층간의 통신(peer protocol)을 위하여 주고받는 정보들을 N protocol primitives라고 하며, N 계층의 기능 구현을 위해 하위 층으로부터 서비스를 제공받게 되는데 이때 하위계층으로부터 제공받는 정보를 service primitives라고 하며, 서비스를 제공받게 되는 point(하나 또는 둘 이상이 될 수 있음)를 SAP(service access point)라고 한다. N+1 층에서는 그림 3에서와 같이 SAP을 그 이름(name)으로 인식하게 되고, N 계층에서는 동일한 점(SAP)을 그 주소(address)로써 인식하게 된다. 따라서 모든 계층에 있어서 주소매핑(mapping)이 필요한

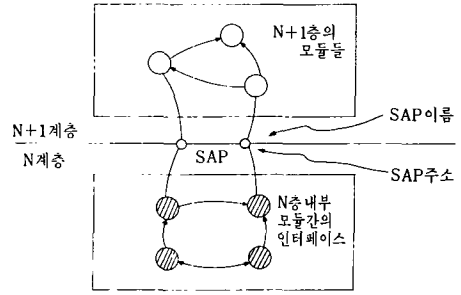


그림 3. Service Access Points

데 이것은 매핑 테이블을 이용하거나, 계층적 주소매핑(hierarchical address mapping) 방법을 이용하게 된다.

N층의 서비스는 그림 3에서 보는 것과 같이 몇개의 모듈로서 분해될 수 있으며 계층들과 같이 각 모듈들은 모듈간의 인터페이스에서 잘 정의된 서비스의 집합을 제공하며, 내부의 구현형태는 이 인터페이스의 다른편에서 볼 수 없다. 여기서 인터페이스는 두개의 실체(프로세스 혹은 모듈)간의 정보의 교환을 위한 규약들의 집합으로 정의되고 다음과 같은 세가지 요소들로 구성된다.

- 가시적인 추상적 대상(예, 프로세스)과 이에 허용된 동작들 및 관련된 파라미터들의 집합
 - 이러한 동작들의 적법한 결과를 다루는 규칙들의 집합
 - 동작들과 파라미터들을 위해서 요구되는 규약들의 코딩 및 포매팅
- 이상으로 계층화에 대한 개념을 알아 보았고 계층화를 이용함으로써 얻어질 수 있는 장점은 다음과 같다.
- 한층 내에서 사용되는 내부구조, 메카니즘, 코딩 및 알고리즘이 다른층들에게는 보이지 않는다.
 - 복잡한 시스템은 보다 이해하기 쉬운 부분들로 나뉘어질 수 있다.
 - 주어진 층을 구현하는 알고리즘과 메카니즘을 제공하는 서비스에 아무 영향을 미치지 않고 변형시킬 수 있으며 인터페이스에서 제공되는 서비스는 변형시키지 않은 채로 제공될 수 있기 때문에 시스템의 개발 및 발전이 용이하다.
 - N+1 층에서 N층의 서비스들을 공유하거나 중복함으로써 또 다른 서비스들을 제공할 수 있다.
 - 한 층의 또 다른 구현형태가 동시에 존재할 수가 있다.
 - 한 층의 서비스들 중 일부 또는 전부가 필요하지

않은 경우에는 그 층이나 부층은 간략화되거나 제거될 수 있다.

• 계층화 된 시스템의 정확한 동작에 대한 확신을 각 층을 순서대로 검사하고 분석함으로써 보다 쉽게 얻을 수 있다.

이상과 같은 계층화는 네트워크 아키텍처에서 사용되는 통신제어를 분해하는 방법으로서 요구되는 모든 통신기능을 몇 개의 군으로 나누어 이들 군을 계층적으로 정돈하는 것이다. 이와 같이 계층화 된 아키텍처에서 각 층은 할당 된 통신 기능의 군을 갖으며 어떤 층은 단지 하나의 통신 기능만 가질 수 있고 어떤 통신 기능은 단지 한층에만 할당 될 수도 있다. 한편, 어떤 통신 기능들은 몇개의 층 또는 전체의 층에 할당 될 수도 있다.

네트워크 구조를 여러 계층과 모듈들로 적정하게 분해하는 것은 모호한 일이지만 층의 수를 결정하고 각 층의 내용을 어떻게 명시할 것인가를 선택 하는데 영향을 줄 수 있는 몇몇 일반적인 관점들은 다음과 같다.^{1,4,51}

- 통신 기능들 상호간의 연관성 및 무관성
- 지역적 구조에 대한 설계
- 사용된 어드레싱에 대한 설계
- 기존의 하드웨어와 소프트웨어의 특수성
- 기존 통신 규범(특히 표준화된 규범)
- 형식과 기능적으로 무관한 부분의 존재
- 전체 시스템에 요구되는 모듈성(modularity)
- 가정한 설치에 바탕을 둔 특정한 요구
- 데이터 메시지의 분해(fragmentation) 장소
- 다중화의 장소

이상으로 네트워크 아키텍처에서 사용하는 계층화의 개념과 계층화를 이용함으로써 얻을 수 있는 장점들에 대해 알아 보았다. 다음 절에서는 이러한 계층화의 대표적인 예를 볼 수 있는 ISO의 참조모델 7 층과 컴퓨터 통신을 위해 필요한 고유한 통신 기능에 대해 살펴 보겠다.

Ⅲ. 통신 기능의 표현

먼저 국제 표준화 기구(ISO)에서 제창한 OSI(open system interconnection) 참조 모델 7 층에 대해 알아 보자.

각 계층에 대한 통신 제어 요소들의 기능 및 목적을 살펴보면 다음과 같다.^{1,4,5,6,7,8,91}

• 물리층(physical layer) : 데이터 링크체간의 비트 전송을 위한 물리적 연결을 설정, 유지, 해제하기 위

계층 7	응 용 층	Application Layer
계층 6	표 현 층	Presentation Layer
계층 5	세 셴 층	Session Layer
계층 4	트랜스포트층	Transport Layer
계층 3	네트워크 층	Network Layer
계층 2	데이터링크층	Data Link Layer
계층 1	물 리 층	Physical Layer

그림 4. OSI/RM-7 Layer

한 기계적이고, 전기적, 기능적, 순서적인 수단을 제공한다. 물리적 연결은 중간의 다른 개방 시스템을 포함할 수도 있다. 물리체(physical entity)는 물리적 매체에 의하여 서로 연결되어 있다.

• 데이터 링크층(data link layer) : 네트워크체간에 데이터 링크 연결을 설정, 유지, 해제하여 DLSU(data link service unit)를 전송하기 위한 기능적이고 순서적인 수단을 제공하여 하나 또는 그 이상의 물리적 연결에 의하여 이루어진다. 또한 물리층에서 발생하는 복구 가능한 에러를 검출하고 복구하며 네트워크 층이 물리층에서 데이터 회로의 상호 연결을 조정할 수 있도록 하는 역할을 한다.

• 네트워크층(network layer) : 개방 시스템들 사이에서 네트워크의 연결을 유지, 설정 그리고 해제하는 기능을 부여하고 트랜스포트층 사이에 NSDU(network service data unit)를 교환하는 기능을 제공한다.

• 트랜스포트 층(transport layer) : 트랜스포트 서비스는 세션을 갖고 있는 두 사용자 사이의 명확한 데이터 전송을 제공해 준다. 또한 가능한 네트워크 서비스를 최대한으로 이용하여 최소한의 경비를 갖고 각각의 세션층에서 요하는 작업을 수행해 준다. 트랜스포트 층에 정의된 규범은 end-to-end 제어 의미를 갖으며 트랜스포트 층 기능은 네트워크 서비스 특성에 의존하게 된다.

• 세션 층(session layer) : 표현층들이 함께 동작하기 위한 대화(dialog)의 형성과 동기, 그리고 데이터 교환을 다루기 위한 층이며 이를 위해 세션 층상의 양쪽 표현층간의 세션 연결을 하여 정확한 데이터의 교환을 지원하게 된다.

• 표현 층(presentation layer) : 응용체들이 통신하거나 참조할 수 있도록 정보를 표현하는 기능들을 담당한다. 표현층은 단지 데이터가 표현되는 구문에만 관계하며 응용층과 관계있는 semantics와는 무관하다.

표현층은 사용되는 공통적인 표현을 제공함으로써 응용층에서는 구문과 관계있는 형태의 정보표현이 필요 없는 구문독립성(syntax independence)를 제공한다.

• 응용 층(application layer) : 응용층은 최상위 층이며 응용 프로세서들이 OSI 여건에 접근하는 수단을 제공함으로써 응용 프로세서들이 상호간에 유용한 정보교환의 창구 역할을 담당하며 응용체의 형태로 표현된다. 다른 시스템과 정보교환을 위해서는 응용층의 규범을 사용해야 하는데 이것이 PDU(protocol data unit)이다. 응용층이 제공하는 서비스는 응용의 종류에 관계없이 개방 시스템을 연결하는 방법을 제공하며 화일 전송, 데이터 베이스 접근, 그리고 작업(job)의 전송등을 제공한다.

이상의 7 개 계층중에서 1-4 층은 두 시스템사이의 데이터 트랜스포트를 담당하며, 5-7 층은 두 시스템간의 분산처리를 위한 상호협력을 담당한다. 공중 전기 통신망의 경우 1-3 층의 기능을 수행하며 각 연결된 시스템들은 4-7 층의 기능을 담당한다. 이러한

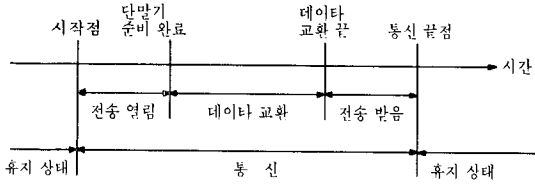
OSI reference model을 층별로 구현하는 경우 설계 및 디버깅에 잇점이 있는 반면에 많은 층간의 인터페이스를 야기 시킴으로써 이로 인한 오버헤드가 시스템의 성능 저하를 초래하므로, 이상의 일곱 계층을 몇 개의 서브 그룹으로 나누어 이를 한데 묶어서 구현하는 방식이 사용되기도 한다. 그 예로 그림 5에서와 같이 여러 컴퓨터 제조회사들은 앞에서 설명한 일곱가지 계층 전부 혹은 그 일부를 사용하고 있다. 계층 1, 2, 3은 보통 명백히 분리되지만 계층 4~7은 서로 혼합되어 사용되어 지고 있음을 알 수 있다.¹⁰⁾

이상에서 본 각각의 계층은 컴퓨터 통신에 필요한 고유한 기능들을 수행하게 된다. 즉 컴퓨터간에 네트워크를 통하여 데이터를 상호 교환하고자 할때의 통신 과정(process)은 그림 6에 같이 통신국면(phase)으로 불리우는 몇개의 부분으로 나뉘어질 수 있다.¹¹⁾

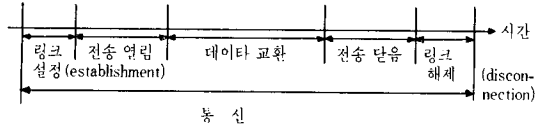
그림 6 (a)는 전용선을 이용한 단말기와 주 컴퓨터(host)와의 데이터 링크의 국면을 보여주고 그림 6 (b)는 네트워크를 이용할 경우의 스위치식 데이터 링크

Names used by the International Standards Organization	Other Terms Used in This Book	Terms Used by Telephone Companies, Following the CCITT Recommendation X. 25	Terms Used in IBM's SNA Architecture	Terms Used in DEC's DECNET Architecture	Terms Used in UNIVAC's DCA Architecture	ARPA Internet layers	UNIX 4.3BSD Implementation layers
Layer 7 Process control	Session services subsystem	—	Function management	Dialogue layer	Termination system	Process/ Applications	User Programs and libraries
Layer 6 Presentation control		—					
Layer 5 Session control		—					
Layer 4 Transport end-to-end control	Transport subsystem	Level 3	Transmission control	Network services protocol (NSP Layer)	Transport network system	Transport	Protocols
Layer 3 Network control			Path control			Internet	
Layer 2 Link control	Common network	Level 2	SDLC (Synchronous Data Link Control)	DDCMP (Digital Data Communications Management Protocol)	UDLC (Universal Data Link Control)	Network	Network interfaces
Layer 1 Physical control			Level 1				

그림 5. 서로 다른 이름을 갖는 7 레벨 계층



(a) 전용 데이터 링크의 국면



(b) 스위치식 데이터 링크의 국면

그림 6. 통신국면의 예

의 국면을 보여주고 있다. 이와 같이 통신과정(process)의 여러 국면들로의 분해는 통신제어의 범위안에서 행해지는 동작들을 여러 그룹으로 나누는 것을 의미하며 이러한 그룹의 동작들을 통신 기능이라 하고, 이러한 통신기능은 위의 예에서 각 국면(phase)에 할당되었음을 볼 수 있다. 그러므로, 링크설정 기능, 전송열음 기능, 데이터 교환기능, 전송닫음 기능, 링크해제 기능 등을 알 수 있으며 각각의 기능은 더 세부적인 기능으로 분류할 수 있으나 여기서는 이러한 다섯가지 기능에 대해서만 간략히 소개한다.

먼저 링크설정 통신 기능의 목적은 주어진 시간내에 두 규범지국간에 물리적 선로를 연결하는 것으로 handshaking 이라고도 한다. 통신기능 방법은 사용된 스위칭 네트워크에 의하여 결정된다. 그것은 수신지국의 addressing, 송신지국과 교환노드사이의 signalling, 그리고 송, 수신 지국의 시그널링과 관계가 있으며 한 스위칭 네트워크에서도 여러개의 링크 설정방법이 있을 수 있으며 그 방법은 목적에 따라 선택될 수 있다. 이 기능은 주로 호출측 지국(calling station)에 설치되어야 하고 호출당한 지국은 링크 설정과정의 마지막 단계에 활성화 되어 있어야 한다. 네트워크에 대한 데이터는 링크설정 국면동안 교환노드에서 얻어진다. 이러한 데이터는 호출측 지국에 전달되어야 한다. 링크설정 기능은 일반 사용자의 요구로 부터 시작되거나 또는 전의 시도가 실패했다는 것을 알리는 네트워크로부터의 신호로 시작된다.

다음은 전송열음 국면동안의 통신국면에 대해서 알아보자. 전송열음 국면동안 호출측 지국은 호출당한 지국이 통신을 할 수 있고, 하겠다는 것을 확인해야 할

뿐만 아니라 호출당한 지국이 호출측 지국이 원하는 지국이라는 것을 확인해야 한다. 예를 들면, 스위칭 네트워크에서 잘못되어 원하지 않은 지국이 연결될 수도 있다. 그러므로 이 국면의 기능은 전송열음 국면 그 자체와 연결된 지국의 확인으로 이루어진 두개의 국면으로 이루어짐을 알 수 있다. 이러한 두 기능은 동시에 행하여지지 않고 차례로 행하여지기 때문에 그에 해당하는 통신국면은 두개의 부 국면으로 나뉘어 질 수 있다.

이제 통신국면에서 가장 중요한 국면인 데이터 교환 국면을 알아보면, 정확성과 신속성이 이 국면의 성능 기준이다. 이 국면에 어떠한 경우가 발생할 수 있는가? 그 중의 하나는 전송에러가 생겨 전달되는 데이터의 정확성을 감소 시킨다는 것이다. 이런점을 방지하기 위하여 에러제어(error control) 기능이 필요하며 이 기능은 데이터 전송 도중의 에러를 검출할 뿐만 아니라 나아가서는 수정까지 할 수도 있다. 경우에 따라서는 수신측 지국에서 빈 버퍼가 없어 전달되는 데이터를 받을 수 없는 경우가 있다. 이것은 데이터의 전달 지연을 일으키며 데이터 블럭의 상실을 일으킨다. 이러한 경우에 대처하기 위해 또 다른 통신기능, 즉, 유통제어(flow control)기능이 부가되어야 한다.

일단 송, 수신측이 데이터를 상호 교환할 수 있는 준비가 끝났다고 가정하면 (양 측이 데이터 교환측면에 접어 들었다면) 에러제어(error control)기능과 유통제어(flow control)기능의 설계여부에 따라 통신시스템의 실용성 여부가 좌우된다. 현재 사용되고 있는 에러 제어기능의 구현 방법으로는 echo checking, ARQ(automatic repeat request), FEC(forward error control) 방법등이 실용화 되어 있으며 유통제어의 구현방법으로는 X-ON/X-OFF, window mechanism, sequence number를 활용하는 방법들이 사용되고 있다.¹¹⁾

IV. 통신 기능의 설계 및 구현

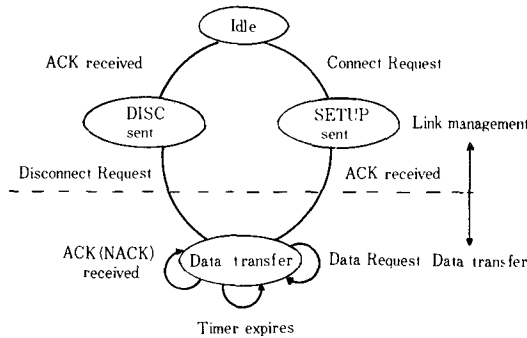
3절에서 우리는 reliable한 데이터 전송을 위해 여러 통신 기능들이 각기 제기능을 수행하여야 한다는 것을 보았다. 이 절에서는 계층화 구조 중 데이터 링크 층에서 통신 기능을 설계 및 구현하는 방법을 예로 들어 설명하겠다.

규범설계에 있어서 규범 규격(protocol specification)은 계층화 아키텍처에서 각 층의 기능 및 목적에 맞는 정확성을 표현하는 근거가 된다. 따라서 규범 규격은 해석이 모호하거나 너무 복잡하지 않아야 하며 차후

검증이나 구현 및 수정이 용이하도록 기술될 수 있어야 한다. 이러한 규격은 한 층내의 통신 제어 순서 및 방법, 메세지 코딩 및 포매팅 방법, 성능평가 척도, 에러 제어, 유통 제어, 인터페이스 방법등을 포함한다. 이러한 규격을 기술하는 방법으로, 자연언어의 사용은 사용자가 쉽게 이해할 수 있지만 종종 모호성을 포함하여 완벽성과 정확성을 검증하기가 어렵다. 따라서 자연언어 대신 형식 기술방법(formal description techniques)들이 규범 기술에 사용된다. 이러한 규범 기술 방법에는 현재 여러가지 방법들이 소개되어 있으나 가장 많이 공통적으로 쓰이는 상태 천이도(state transition diagram), 상태 천이표(state transition table) 과 고수준 구조적 언어(high level structured language)를 사용하여 링크 레벨 규범을 구현하는 예를 소개하겠다.

개하겠다.

상태 천이도는 규범 기술에 사용하는 전형적인 방법이다.¹¹¹ 그림7 (a)는 링크 설정단계와 데이터 교환단계에서 link layer가 해야 할 기능들을 보여주는 상태천이도이다. 송, 수신측이 가질 수 있는 가능한 상태(state)가 상태 명(name)과 함께 원(circle) 안에 기술되고 화살표로서 어떠한 사건(event) 발생에 따라 상태천이를 위해 해야 할 action들을 도식한다. 이렇게 상태 천이도에 표현한 것이 그림 7 (b)에 있는 상태 천이표이다. 상태 천이표는 상태 천이도에 나타나는 모든 정보를 테이블의 형태로 표현한다. 현재의 상태(상태 천이도에서 원안에 표시된 state)와 발생 가능한 사건(event)에 따라 규범이 구현해야 할 action을 기술하는 것이다.



(a) 상태 천이도

Present state \ Event	Connect Request	Data Request	Disconnect Request	ACK received	Timer expires	Action New state
	Idle (0)	Setup 1				
Setup sent (1)				Proc Ack 2	Retx Frame 1	Action New state
Data transfer (2)		TxFram 2	Disc 3	Proc Ack 2	Retx Frame 2	Action New state
Disc sent (3)				Proc Ack 0	Retx Frame 3	Action New state

Messages from user interface

Frames received from link

Internal event

(b) 상태 천이표

그림 7. 통신규범 규격

그림 7 (b)를 예로 보면 휴지 상태 (idle state)에 있는 시스템은 connect request를 받으면 링크 설정을 위하여 다음 단계인 setup sent 단계로 옮겨 간다. Setup sent 단계에서 긍정응답(ACK)을 받게 되면 응답에 따른 처리를 한 후 데이터 전송 state로 옮겨지게 되고 긍정응답을 못 받고 예정된 시간이 경과하게 되면 다시 수신측에 setup request를 하게 되며, 데이터 전송국면 동안에는 발생가능한 사건(여기서는 data request, disconnect request, ack received, timer expired)에 대한 적절한 action을 취하게 된다. 이와 같이하여 한 층내의 규범들이 해야 할 기능들을 기술한 후에는 실제 구현을 위해 그림 8에서 보는 것과 같이 고수준 구조적 언어를 사용하여 pseudo code를 작성하게 되고 이것이 완성된 후, 구현하고자 하는 시스템 환경에 적합한 언어를 사용하여 coding 및 testing, simulation, verification의 절차를 거침으로서 통신 기능을 구현시킬 수 있는 것이다.

```

program ContinuousRQ_Primary; {Modifications to incorporate link
management}
type Events = (ConnectRequest, DataRequest, DisconnectRequest,
ACKReceived, Timeout);
States = (Idle, SetupSent, DataTransfer, DiscSent);
Actions = (NA, Setup, Disc, TxFrame, ProcAck, RetzFrame);
IncomingMessage = (Connect, Data, Disconnect); {List of user
messages}
var
.
.
.
{List of Action procedures}
.
.
.
begin {Start of main body}
case IncomingMessage of
Connect: EventType = ConnectRequest;
Data: EventType = DataRequest;
Disconnect: EventType = DisconnectRequest;
end; {Case}
with Event State Table[PresentState, EventType] do
begin case Action of {List of action procedure calls}
NA :
Setup:
TxFrame:
Disc:
ProcAck:
RetzFrame:
end; {Case}
PresentState = NewState;
end;
end.
    
```

그림 8. Pseudo Code로 표현된 규범 규격

V. 통신 기능의 검증 (Verification)

넓은 의미에서의 시스템 validation의 목적은 시스템 설계 단계에서의 specification을 만족시켜야 하고 또한 사용자의 요구가 만족되었는가를 확인하는 것이다. 그러므로, validation 활동은 통신 기능의 설계 단계에서 가장 중요하며 시스템 구현 단계에서의 testing, simulation study, 해석적 성능평가 및 검증(verification)을 포함하여야 한다. 통신 기능의 검증은 시스템 specification을 근거로 하여 논리적인 타당성(logical reasoning)을 포함한다. 그러므로 검증은 가능한 설계 단계의 오류를 피하기 위하여 시스템 구현이 있기 전에 모든 설계단계에서 사용되기도 한다. Testing과 simulation은 어느 특정한 상황하에서 validation에 유용한 반면 verification은 시스템이 실제 작동시 발생 가능한 모든 상황들을 고려할 수가 있다.¹²¹

이러한 verification의 근본적인 목표는 프로토콜 운용상의 가능한 에러를 검출해 내는 것이다. 가능한 프로토콜 에러의 종류는 몇개의 그룹으로 분류해 볼 수 있으며, 그림 9는 통신 규범에서 나타나는 에러나 결함에 대한 전체적인 것을 보았다.¹⁴¹

번호	규범 에러 및 불완전성	잘못된 설계	에러의 불완전한 집합	규범 성질
1	부적절한 연속		●	
2	부적절한 초기화		●	
3	부적절한 종결		●	
4	존속성 부족	○	●	○
5	규범실패	○	●	○
6	불완전성	●		
7	데드-로그(deadlock)	●	○	○
8	루핑(looping)	●		○
9	일시적인 블럭킹	●	○	○
10	레이스(race)조건	●	○	○
11	타임-아웃 에러	○	●	
12	연속 번호의 모호성	○	●	
13	상 실		●	●
14	이중화		●	●
15	왜곡		●	●
16	순서가 어긋나게 수신함		●	●

(주) ○ : 가능한 발생 경우, ● : 전형적인 발생 경우

그림 9. 검증에 의하여 조사되는 에러의 개요

그림 9에서 처음 두개의 세로난은 잘못된 통신규범 설계, 그리고 언어적 명세의 부적합성 및 에러와 통신 실패를 의미하는 집합의 완전하지 못함에 의한 통신

규범 에러의 원인을 나타내고 있고, 맨 오른쪽 난은 통신 규범의 결함으로 생각할 것인가, 그렇지 않을 것인가에 대한 언급이다. 즉, 생각지 않았던 에러나 통신실패 때문에 더 이상 통신 규범의 성질로 존재할 수 없는 경우가 있음을 나타낸다. 그리고 검은 원(●)은 흔히 있는 에러이거나 또는 에러를 유발하는 주된 원인이 되는 것이다. 그리고 그렇지 않은 원(○)은 거의 일어나지 않거나 에러를 유발하는데 극히 영향을 미치지 않는 요소이다. 그림에서 보듯이 대부분의 통신 에러나 불완전성은 생각하지 않았던 통신 매체나 통신 규범 지국의 에러나 통신 실패에 의한 것이다. 또한 그림 9는 통신규범 에러나 불완전성의 종류를 나누어서 구분하고 있다. 즉 4개의 에러군과 3개의 단독 에러로 나누어 놓고 있다

첫번째 에러군(1, 2, 3)은 통신의 여러 국면에 있어서 동기가 깨진 결과에 의하여 생기는 것이며, 두번째 에러군(4, 5)는 통신을 수행할 수 없는 상당히 큰 실패가 일어났을 경우이다. 이러한 에러는 통신 규범이 커다란 실패에 대하여 대처할 수 있는 능력이 없으므로 생기는 것이다. 세번째 에러군(6, 7, 8, 9)는 실제 일어나는 에러는 통신 실패가 예측했던 것과 같은 경우에도 일어나는 것이다. 이것은 불완전한 통신 규범 설계 또는 언어적 명세를 형식적 명세로 바꾸는 과정이 올바르게 않으므로 인하여 생기는 것이며, 네번째 에러군(13, 14, 15, 16)은 통신규범 자체 성질과 일정한 관계가 있는 에러를 포함한다. 동시 입력하에서의 불확실성(10)은 통신 규범의 성질을 반영한다. 또한 이러한 불확실성은 통신실패에 의존하는 것이 아니라 통신규범 지국의 내부구조에 기인함을 나타낸다. 부적절한 타임아웃(11)에 의한 에러가 바로 시간에 의한 에러의 예가 될 수 있다. 마지막으로 연속번호의 모호성에 의한 에러(12)인데, 이것은 너무 적은 넘버링 모듈을 이용하므로 일어난다. 여기서 넘버링 모듈은 통신 규범의 파라미터 값인데 파라미터를 적절히 취하지 못함으로 인해서 일어나는 에러이다. 이상의 각 통신 규범 에러에 대한 상세한 내용은 reference[4]를 참조하면 된다.

통신 규범을 검증하는 방법으로는 통신 규범 형식모델이 아주 좋은 수단이 된다. 형식 모델은 기본적으로 두가지 방법이 이용되고 있다. 통신 규범의 언어적 명세에 대한 검증을 한다면 먼저 어떤 형식 모델이 적합할 것인가를 생각하고 그것에 따라 검증 알고리즘을 적용하여야 한다. 그리고, 통신 규범의 형식화 명세를 검증할 경우에는 적절한 검증 알고리즘을 이용하거나 또 또는 검증하는 형식화 명세에 더 적합한 모델을 찾아

볼 수도 있다. 보통 통신 규범의 검증은 실제보다 매우 간략한 시스템에 적용되고 있다. 예를 들면, 통신 규범 지국의 통신 실패를 무시한다거나, 타임아웃 시간이 미치는 영향을 무시한다거나, 또는 통신매체를 기껏해야 한 블록의 데이터를 전송할 수 있는 간단한 포인트-포인트 선로로 생각한다거나 하는등의 예를 들 수가 있다. 이러한 제약조건들은 검증의 대상이 간단한 통신 규범에서 복잡한 통신 규범으로 발전시켜 나갈 수 있다는 것을 보여준다. 즉, 통신 규범을 좀더 완벽하게 구성함으로써 이러한 제약 조건들을 제거할 수 있다는 것을 의미한다.

VI. 통신 기능과 분산 시스템과의 관계

분산 시스템이란 용어에 대해서는 현재까지도 완전히 일반화 된 정의를 내리지 못하고 있다. 어휘 자체의 의미로 본다면 컴퓨터 시스템의 구성 요소 및 H/W, data, 처리기능, 제어기능 중 어느 하나만 이라도 분산되어 있으면 분산 시스템이라고 할 수 있지만 이런 관점의 해석은 모호하다. 최근에는 분산 시스템에 대한 정의를 multiprocessor system이나 network operating system과 구분해서 새로운 정의를 내리고 있다. 즉 기본적인 처리능력(처리장치, 기억장치, 입출력 장치)를 가진 autonomous machine들을 통신 기술로 연결하여 보다 종합적인 컴퓨터 시스템을 구성하는 것을 말한다.¹³⁾¹⁴⁾¹⁵⁾

이러한 의미의 분산 시스템은 다음과 같은 몇 가지 특성을 만족하는 컴퓨터의 집합이라고 할 수 있다.

- 사용자에게 다이나믹하게 할당할 수 있는 일반적인 용도의 resource가 각 노드에 분산되어 있고 또한 중복되어 있다. (multiplicity)
- 분산된 요소들은 네트워크를 이용해 메시지를 주고 받음으로서 정보를 교환한다. (message passing)
- 시스템의 구성 요소들은 자기 어느 정도의 자율성을 갖는다. (local autonomy)
- 사용자는 어떤 resource의 물리적 위치를 알지 못하고도 그 resource를 사용할 수 있어야 한다. (system transparency)
- 각 노드의 자율성을 어느정도 보장하면서 시스템 전체의 policy를 결정하는 어떤 통합적인 제어기능이 있어야 한다. (unified control)

이러한 특징 혹은 조건을 만족하는 컴퓨터 시스템을 분산시스템이라고 정의하면 분산시스템의 범위는 크게 축소되며, 분산 시스템의 대표적인 장점으로서 성능향상, 시스템의 안정성 및 신뢰도의 향상, 자원공유의 용

이함 및 확장성의 용이등을 들 수 있다. 이러한 분산 시스템을 분류하는데는 앞서 언급한 다섯가지 특징 중 어느 것이 얼마나 강조되었는가에 따라 여러가지 분류 방법이 있을 수 있다.^{13,16)}

즉 노드들을 연결하는 네트워크가 LAN인가 LHN인가에 따라 나누는 방법, 노드간의 결합도(coupling 정도)에 따르는 방법등이 있을 수 있으나 여기서는 이러한 방법들을 종합적으로 설명할 수 있다고 생각되는 분산 hierarchy에 따른 분류 방법을 알아본다. 일반적으로 컴퓨터 시스템들이 계층구조를 가지고 있고 분산 시스템은 이러한 컴퓨터들이 통신 네트워크로 연결되어 상호 협력하여 수행하는 하나의 시스템으로 볼때, 통신기능이 어느 계층에 존재하는가는 중요한 의미를 가진다. 왜냐하면 통신 기능이 존재하는 계층을 포함한 그 하위계층에 있는 resource는 시스템내의 모든 노드들이 공유할 수가 있기 때문이다. 또한 통신 기능의 위치는 시스템의 특성 즉, 신뢰도, transparency, 자율성 정도에 큰 영향을 미친다. 원칙적으로 이러한 통신 기능은 어느 계층에라도 존재할 수 있으나 시스템 설계시의 요구조건 또는 제약조건들에 의해 적절한 위치가 결정된다.

기존의 컴퓨터들이 모두 명확하고 동일한 구조를 가진다고 볼 수는 없지만 근래의 컴퓨터들은 거의 모두 상당히 계층화된 구조로 설계되고 있으며 일반적으로 그림10과 같은 구조를 가진다.^{17,18)}

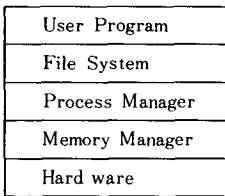


그림10. 컴퓨터 시스템의 구조

원칙적으로 통신 기능은 어느 계층에나 있을 수 있으며 모든 계층에 통신 기능이 존재해도 무방하다. 여기서 각 계층이 하는 일을 알아보고 그 계층의 manager들간에 통신 기능이 존재할 때 시스템은 어떠한 특성을 가지는 지를 알아보면,

• Hardware level

통신 기능이 이 계층에 존재할 경우 분산 시스템이라 할 수 없다. 왜냐하면 이러한 시스템은 흔히 공유 메모리를 가지는 multiprocessor 시스템으로서 mess-

age passing 방식의 통신을 하지 않기 때문에 단일 시스템으로 볼 수 있다.

• Memory manger level

Memory manager는 시스템의 메모리를 관리하고 필요에 따라 프로세스들에게 할당해 주는 일을 한다. 이러한 momory manager들간에 통신 기능이 가능하다는 것은 시스템 전체 메모리가 global하므로 한 노드의 프로세스는 다른 노드의 메모리 공간을 할당 받을 수 있다는 것을 뜻한다. 그러나 한 프로세스가 여러 노드의 메모리 공간을 이용한다는 것은 특별한 하드웨어 메카니즘을 필요로 하게 되고 신뢰도 문제와 통신 네트워크의 성질을 고려해 볼때 실용적이 못된다.

• Process manager level

이 계층은 프로세스의 생성, 종료 및 프로세스 간의 context switching뿐 아니라 프로세스간의 동기를 위한 기본적인 기능(예를 들면 semaphore)를 제공한다. 이러한 점을 고려해 보면 process manager들간의 통신 기능은 분산처리, 노드간의 load balancing, 다른 노드에 존재하는 프로세스와의 동기 문제등을 쉽게 해결해 줄 수 있다. 따라서 분산 운영체제(distributed operating system)는 이 계층에 통신 기능을 가지고 있어야 한다. 그러나 실제 분산처리 운영체제를 구성하는 데는 각 노드의 하드웨어 및 소프트웨어 환경의 이질성이 여러가지 문제점을 일으킨다. 보통 이 계층에서 통신 기능을 가지게 하려면 각 노드로 homogeneous한 컴퓨터를 이용한다.

• File system level

파일은 데이터를 영구히 저장해 둘 수 있는 기억장소를 말한다. 사용자측에서 컴퓨터 시스템을 볼때 직접 접하게 되는 permanent object로써 분산의 대상이 되는 가장 중요한 요소이다. 이 계층 manager들간의 통신 기능은 시스템 전체에 대한 directory hierarchy를 이룰 수 있게 하며, 사용자가 어떤 파일이 실제 존재하는 위치를 모르더라도 그 파일을 이용할 수 있게 해주고, 파일의 중복 및 분산을 transparent하게 해 줄 수 있다. 사실 분산 시스템의 가장 큰 장점이라고 할 수 있는 자원의 공유, 신뢰도 향상등은 사용자 object인 파일의 분산 및 중복을 통해 얻어지므로 이 계층에서의 통신 기능은 분산 시스템에서 반드시 필요하다고 할 수 있다.

• User program level

이 계층에만 존재하는 통신 기능은 사용자로 하여금 다른 노드에 access할 수는 있게 하지만 이때는 분산 시스템의 장점을 충분히 발휘할 수 없다. 시스템내에

는 global한 resource가 없으며, 따라서 사용자는 원하는 resource의 위치를 정확히 알아야만 그 resource를 이용할 수 있게 된다. (분산 데이터 베이스는 운영체제위에 올려진 또 하나의 운영체제이므로 예외가 된다.) 그러나 이러한 경우 각 노드의 자율성은 최대로 보장된다.

이상에서 볼 때 분산 시스템에서의 통신 기능은 process manager level 또는 file system level에 위치하

는 것이 적합한 것을 알 수가 있다. 앞서 언급하였듯이 분산 시스템을 interconnection technology와 computer system이 결합된 것으로 볼 때, 이의 기술개발에 앞서 컴퓨터 네트워크 기술개발, 분산제어 이론에 관한 연구, resource의 분산 및 중복방법, 분산 시스템의 계층 구조에 대한 철저한 연구가 뒷받침 되어야 할 것이다.

Characteristics of Some Notable Networks(see Table 11a for an explanation of the symbols used here)

Name	Center	Extent	Hosts	Users	Layers	Services	Quality
<i>Research networks</i>							
ARPA Internet	USA	3, 8	2, 050	?	i T	lfmno	56, 000, m, 9
ARPANET	USA	1, 1	150	?	a T	lfmo	56, 000, m, 9
MILNET	USA	2, 3	400	?	a T	lfmo	56, 000, m, 9
MINET	Europe	1, 4	?	?	x T	lfmo	9600, m, 8
CSNET	USA	4, 10	170	?	i C	m	1200, h, 8
Phonenet	USA	4	128	?	d M	m	1200, h, 7
X25NET	USA	1	18	?	x T	lfm	9600, m, 8
ARPANET (CSNET hosts)	USA	1, 1	25	?	l T	lfmo	56, 000, m, 9
Cypress	USA	1	6	?	p T	lfmo	9600, m, 9
NSFNET	USA	1, 1	65	?	i T	lfmo	T1, m, 9
MFENET	USA	1, 1	120	?	i D	fm	56, 000, ?, ?
SPAN	USA	1, 1	100 +	?	px D	fm	56, 000, m, 8
MAILNET	USA	2, 3?	28	1, 800	dx M	m	1200, h, 9
JANET	UK	1, 1	915	?	x B	lfmo	4800, ?, ?
EAN networks	Europe	3, 12	33	?	x X	m	2400, m, 9
CDNnet	Canada	1, 1	32	?	x X	lmno	2400, m, 9
COSAC	France	1, 2	27	?	x X	fmn	1200, d, 7
DFN	Germany	1, 1	6	?	x X	fm	9600, ?, ?
<i>Company networks</i>							
Xerox Internet	USA	3, 4	?	12, 000	i N	lfmo	56, 000, h, 9
Xerox RIN	USA	1	?	4, 000	i NPT	lfmo	56, 000, h, 9
Xerox CIN	USA	3	?	8, 000	i N	lfmo	56, 000, h, 9
DEC's Easynet	USA	4, 20?	10, 000+	60, 000?	d D	lfmno	56, 000, m, 9
IBM's VNET	USA	4, ?	2, 200	?	p R	fmo	9600, h, 9
<i>Cooperative networks</i>							
BITNET	USA	3, 21	1, 306	?	p R	fmo	9600, h, 8
BITNET	USA	1, 2	845	?	p R	fmo	9600, h, 8
NETNORTH	Canada	1, 1	91	?	p R	fmo	9600, h, 8
EARN	Europe	1, 17	363	?	px R	fmo	9600, h, 8
Asianet	Japan	1, 1	7	?	p R	fmo	9600, h, 8
FidoNet	USA	2, 4?	500	?	d F	m	1200, d, 4
ACSNET	Australia	1, 1	300	?	depX A	fmno	1200, h, 9
UUCP mail	North America	4, 5	7, 000+	200, 000?	dx U	m	1200, d, 5
USENET news	North America	4, 4	2, 500+	50, 000?	dx UATR	n	1200, -, 7
EUnet	Europe	1, 13	896	?	dpX UAT	lfmno	1200, h, 8
SDN	Korea	1, 1	100	?	epX TU	lfmno	2400, d, 8
JUNET	Japan	1, 1	160	?	x U	mn	2400, d, 8

Legend for Table II

Extent: A pair of the numbers of continents and nations reached

Layers: Network layer (left column) and internet or transport layers (right column)

Protocols for the network layer

- a: ARPANET-style PSN (BBN 1822) communications subnet
- d: Dial-up telephone
- e: An ethernet
- i: An internet over various network layers
- p: Leased telephone line
- x: 25 (usually over leased telephone line)

Protocols for the internet and transport layers

- A: ACSNET's SUN-III
- B: JANET's Coloured Book
- C: CSNET; TCP/IP (ARPANET); TCP/IP on X.25 (X25NET); MDMF2 (Phonenet)
- D: DEC's DECNET
- F: FidoNet
- N: Xerox Network Services protocol suite
- P: Xerox PARC Universal Packet (PUP) protocol
- T: ARPAs TCP/IP protocol suite
- U: AT & T's UUCP
- R: IBM's RSCS
- X: CCITT/ISO X.400 and related protocols

Services: l: remote login; f: file transfer; m: mail; n: news; o: other

Quality: A triple of speed, delivery, and reliability

Speed: Most typical speed of long-haul links bits per second (bps):
300, 1,200, 2,400, 9,600, 19,200, 56,000, T1 microwave

Delivery: Average delivery time for mail messages:
m: minutes; h: an hour or more; d: a day or more

Reliability: On a subjective scale from 1 (lowest) to 10 (highest)

그림 11. Characteristics of Some Notable Networks

VII. 컴퓨터 네트워크 소개

현재 연구가 진행 중이거나 이미 개발된 컴퓨터 네트워크들을 살펴보면 ARPANET, BITNET (because it's time net), CSNET (the computer science research network), NSFNET (a network for access to super computers), UUCPNET, MAILNET 등이 있다. 컴퓨터 네트워크의 사용자는 host 컴퓨터에서 job을 수행하며 그 host 컴퓨터들은 LAN (ex, Ethernet) 으로 연결되며, 그 LAN들은 다시 LHN (ex, BITNET, CSNET, ARPANET) 으로 연결된다. LAN에서는 특별한 목적의 local resource (ex, fileserver, printer 등)에 대해서 available 하며 나아가 LHN에서는 슈퍼 컴퓨터나 전문 데이터 베이스 같은 remote resource에 대한 access를 제공하여 준다. 각 컴퓨터 네트워크는 gateway라는 것으로 연결되는데 이는 네트워크들 간의 데이터를 교환시켜 주며 네트워크에 걸친 routing information을 제공하여 준다. Gateway의 복잡도는 기존에 연결된 컴퓨터 네트워크에 사용된 프로토콜의 유사성에 의존하고 있다.^[18, 20, 21]

몇몇 international academic network로서는 JANET

(joint academic network: United Kingdom), EARN (european academic research network), EUNET (european unix network), SDN (system development network: Republic of Korea), HEANET (higher education authority network: Ireland), JUNET (japanese unix network), ACWNET (Australia), CDN-NET (Canada) 등이 있다. 이들 중 우리나라의 SDN에 대하여 간단히 설명하면 현재 21개 대학과 연구소에 있는 60대 이상의 host가 전송속도 1.2kbps-9.6kbps인 전용선으로 연결되어 있다. 이 네트워크는 UUCP와 TCP/IP based protocol을 제공하고 있으며 public X.25 네트워크를 경유하여 미국의 UUCPNET 및 CSNET와 연결되어 있으며 태평양 연안의 몇몇 국가들이 SDN과 연결되어 PACNET을 구성하고 있다. 이상에 소개한 학술 연구용 네트워크들은 연결되어 있는 노드들의 규모나 사용하고 있는 장비만 차이를 보일 뿐 그 근본 목적은 동일하다. 이 외에도 여러 네트워크들이 개발 운용되고 있으며 각 네트워크 시스템의 특성들은 그림 11을 참조하고 더 자세한 내용을 원하는 독자는 reference^[20, 21]을 참고하기 바란다.

일반적으로 네트워크를 사용하는 사용자는 통신 subnet의 특성이 서비스의 availability에 영향을 줄 때를 제외하고는 네트워크의 프로토콜, 통신수단, subnet와 internet의 연결에 대해서는 의식을 할 필요가 없다. 컴퓨터 네트워크가 갖는 장점으로는 collaboration의 증대, resource공유, report의 빠른 전송, 정보에 대한 손쉬운 접근등이 있다. 또한 네트워크가 제공하는 대표적인 서비스 기능으로는 electronic mail, file transfer, remote system access, database access, remote job entry, remote spooling, 분산 editor, remote procedure call 등이 있으며, 앞으로 가능하게 될 서비스로는 multi-media data transfer 즉, voice, text, graph가 한 메시지에 혼재하여 전달되는 서비스를 비롯하여 높은 대역폭의 네트워크 capability에 의한 그래픽 이미지의 실시간 전송, computer based conference 등이 머지 않아 실현될 것이다.

VII. 결 론

본 고에서는 컴퓨터 통신에 관한 전반적인 사항을 기술하였다. 통신의 계층적 구조, ISO 참조 모델 7 층의 기능소개, 통신 규범 구현에 관한 내용을 예로 들어 설명하였고 검증의 역할, 이때 발견되는 에러의 종류와 통신기능과 분산 시스템의 관계, 현재 운용되고 있는 학술연구용 네트워크 등에 대해지면 관계상 소개형식으로 서술하였다. 이에 대해 특히 관심이 있는 독자들은 참고문헌의 참조를 바란다.

參 考 文 獻

[1] William Stallings, *Data and Computer communications*. Macmillan Publishing Company, 1985.

[2] 최양희, "Computer Communication Protocols," 대한전자공학회 교환연구회, 전자교환기술, 제 2 권, 제 1 호, 1986. 6.

[3] 나근식, 이광휘, 안순신, "Ring Type LAN의 설계 및 구현," 고려대학교 전자전산공학과 Computer Network Lab. 보고서 No. 87CN0430, 1987. 4.

[4] 안순신, 데이터 통신기념, 정익사, 1985.

[5] 김준, 안순신, "Computer Network Architecture," 한국 정보과학회, 정보과학회지 제4권 제 4 호, 1986. 12.

[6] *Proc. IEEE*, "Open system interconnection (OSI) standard architecture and protocols" Dec. 1983.

[7] A. Bowers, et al., "A checklist of communications protocol functions organized using the OSI seven layer Ref. Model", *Proc. COMPCONF '83*, pp.479-487, 1983.

[8] Tannenbaum, *Computer Network*. Prentice Hall. 1981.

[9] International Organization for Standardization, "Reference model of open systems architecture", *TC97/SC16/N227*, June 1979.

[10] 이균하, 유황빈, 데이터 통신, 정익사, 1986.

[11] Fred Halsall, *Introduction to Data Communications and Computer Networks*, Addison-Wesley Publishing Company Inc., 1985.

[12] G. Bochman and C. Sunshine, "Formal methods in communication protocol design", *IEEE Transactions on Communications*, vol. COM-28, no.4, pp.624-631, April 1980.

[13] 전길남, 정철, "Distributed Computer Systems," 대한전자공학회 전자계산연구회, 컴퓨터기술 제 2 권 제 1 호, 1985. 6.

[14] 이영희, "Distributed Computer System," 대한전자공학회 전자계산연구회, 컴퓨터기술 제 3 권 제 1 호, 1986. 6.

[15] 나근식, 안순신, "분산 처리 시스템," 고려대학교 전자전산공학과, Computer Network Lab. Internal Note.

[16] D. Keefe and G.M. Tomlinson et al., *PULSE:an Ada based Distributed Operating System*, Academic Press Inc., 1985.

[17] M.J. Bach, *the Design of the Unix Operating System*, Prentice-Hall, 1986.

[18] D. Comer, *Operating System Design: The XINU Approach*, Bell Lab., 1984.

[19] 손용락, 안순신, "컴퓨터 네트워크 및 분산 시스템," 고려대학교 생산기술연구소, 논문집 제 23 호, 1987. 3.

[20] Lawrence H. Landweber, Dennis M. Jennings, Ira Fachs, "Research computer networks and their interconnection", *IEEE Communications Magazine* vol.24, no.6, June, 1986.

[21] John S. Quarterman and Josiah C. Hoskins, "Notable computer networks", *Communications of the ACM*, vol.29, no.10, October 1986. *