

이기종 컴퓨터간 정보 교환용 한글 VT Protocol 개발

연구기관 : 한국소프트웨어개발연구조합

국립중앙도서관
91.9.18

과학기술처

제 출 문

과 학 기 술 처 장 관 귀하

본 보고서를 "이기종컴퓨터간 정보 교환용 한글 VT Protocol개발"과제의 최종완료
보고서로 별첨과 같이 제출합니다.

1991년 7월

주관연구기간명 : 한국 소프트웨어 개발 연구조합

총괄연구책임자 : 강 태 현

책 임 연 구 원 : 김 종 진 (한국컴퓨터통신)

선 임 연 구 원 : 최 흥 식 (한국컴퓨터통신)

외 8명

요 약 문

1. 제목

이기종 컴퓨터간 정보교환용 한글 VT Protocol 개발

2. 연구 개발의 목적 및 중요성

TELNET의 한글화 개발을 통해 TCP/IP Network 환경에서 한글을 지원하는 PC Host Application이 가능하도록 하며, PC를 네트워크에 연결하여 사무자동화와 같은 Application이 기존의 네트워크 관련 소프트웨어의 수정없이 가능하도록 한다.

3. 연구개발의 내용 및 범위

VT Protocol 분석은 ISO의 VT Protocol, TELNET VT Protocol, SUPDUP Virtual Protocol, CCITT의 VT Protocol이 그 대상이 되었다. VT Software 분석은 PC/TCP 및 호스트용 VT Software에 대하여 실시하였다. TELNET 한글화 작업은 PC/TCP에서 제공되는 library routine들을 이용하여 TCP/IP Network를 이용하는 모든 서비스를 제공받고 Communication Channel을 통하여 교환되는 8-bit transparent stream에 한글코드 변환 및 terminal emulation 기능을 수행할 수 있는 형태로 수정·보완하였다.

4. 연구 개발의 결과 및 활용에 대한 건의

현재 개발된 시제품과 유사한 형태의 개발사업이 국내의 여러기관에서 제각기 독자적으로 전개되어져 왔다. 그러므로, 본 연구결과의 공유와 확장 그리고 다양한 분야로의 활용을 위해서 지속적으로 여러 분야에서의 상호 의견 교환이 요구된다.

목 차

1. 서 론	1
2. Virtual Terminal 기능	3
2.1 Virtual Terminal 환경	3
2.2 Virtual Terminal 수행과정	4
2.3 Virtual Terminal Model	6
3. Virtual Terminal 사례분석	9
3.1 TELNET	9
3.2 Super Duper(SUPDUT)프로토콜	10
3.3 CCITT X.3, X.28, X.29, X.25	12
3.3.1 Asynchnonous terminal을 위한 protocol	12
3.3.1.1 Command Mode/Data Transfer	14
3.3.1.2 PAD parameters(X.3)	15
3.3.1.3 X.28 Commands	17
3.3.1.4 X.29 Protocol	17
3.4 ISO/OSI Virtual Terminal	19
3.4.1 Model	19
3.4.2 ISO/OSI Virtual Terminal Environment	20
3.4.3 OSI Virtual Terminal Protocol	27
3.5 X.Window	29
3.5.1 Model	30
4. PC/TCP 한글화	32
4.1 배 경	32
4.2 P/TCP 개요	34
4.2.1 PC/TCP의 기능	35
4.3 한글화 작업 개요	37

4.4 한글화 Module 설명	49
4.4.1 관련 structure	49
4.4.2 관련된 function	51
5. 결론	64

1. 서론

초기의 Terminal은 local host에 물리계층을 통하여 연결되는 형태로 사용되어 왔다. 그러나 컴퓨터 통신 기술의 점점 발달하고 터미널의 사용자가 자기가 연결된 host computer의 사용을 넘어서 원거리 computer에 접근하여 정보를 처리하고자 하는 욕구가 증대되면서 virtual terminal에 관련된 기술이 발전되기 시작했다.

Virtual Terminal 기술은 computer vendor등이 컴퓨터 hardware, Operation System을 제작하는 기술을 토대로하여 발전된 propriety virtual terminal software 및 hardware 형태로 시작되었다. 이러한 형태의 virtual terminal은 remote host에 login하는 기능을 제공하는 외에 특별히 고려되어야 할 사항이 없었다. 즉 이러한 형태의 virtual terminal에서는 동종의 computer terminal이 연결되는 형태이기 때문에 상호 기종 및 통신방식의 차이에서 생기는 어려움이 배제되었다.

그러나 공중 데이터망, local area network(LAN)등을 이용하여 이기종의 컴퓨터들이 접속되고 여러 형태의 terminal이 여러 형태의 컴퓨터에 연결됨으로서 terminal과 computer간에 terminal이 갖는 기능을 수용하기 위한 문제가 발생되었다. 이와 같은 상황에서는 터미널에 대한 단순한 제어기능(transmission speed, parity, flow control 등)을 가지고 모든 터미널을 제어하는 것이 불가능하다는 문제점이 발생되었다. 즉 host computer에서 개개의 terminal에 의존하는 제어를 하지 않고, terminal이 모든 종류의 host computer에 같은 형태의 host computer에 접근하는 것과 같이 할 필요가 생겼다. Virtual terminal은 이와 같은 목적을 달성하기 위한 수단으로 제안되었다.

virtual terminal의 기본개념은 가상적인 terminal을 설계해 두고, 실제 네트워크 환경에서 한 terminal이 상대 host를 접근하고자 할때, 양측의 computer(terminal이 부착

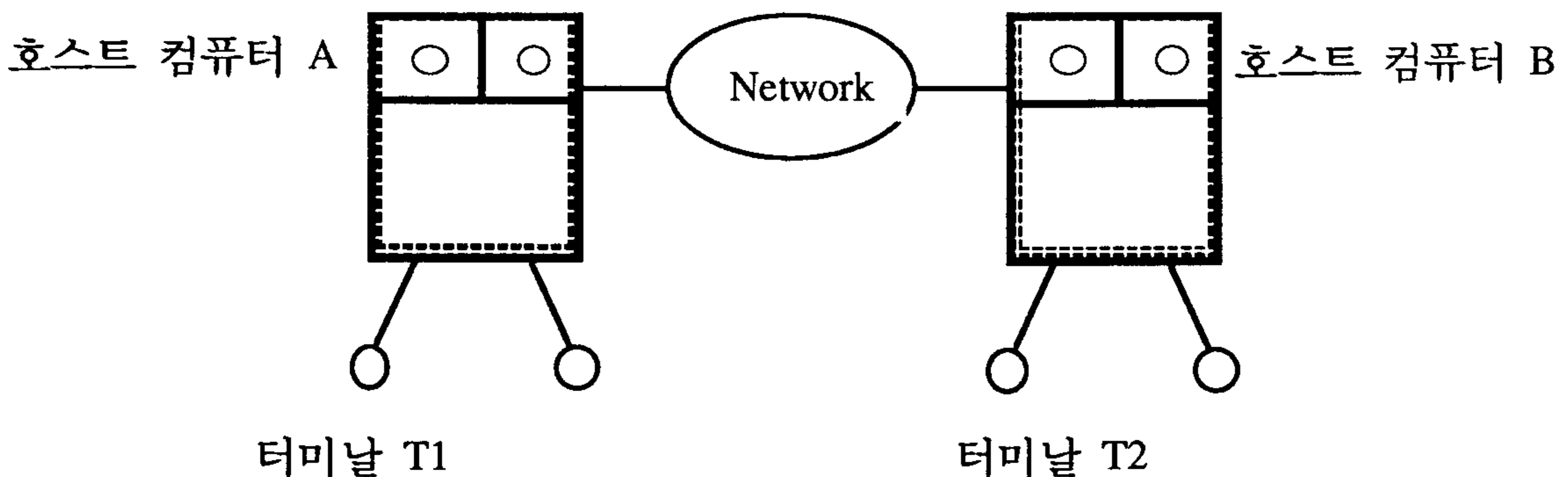
된 computer 및 접근하고자 하는 컴퓨터)가 자기가 표현하고자 하는 terminal의 기능(scroll up, scroll down, home, clear등)을 가상의 computer에 mapping하고 상대쪽은 virtual terminal에서부터 실제 terminal로 다시 mapping하는 형태로 변환하는 것이다.

virtual terminal은 ARPANET에서 사용된 TELNET protocol을 비롯하여 SUPDUP등의 많은 virtual terminal이 제안되었으며 현재 사용중에 있다. ISO(International Standard Organization)의 TC97에서도 이기종 간의 virtual terminal standard를 제정하였으며, 선진국의 경우 이러한 표준을 따르는 software가 시험제작되는 단계에 있다.

2. Virtual Terminal 기능

2.1 Virtual Terminal 환경

Virtual Terminal은 아래의 그림 <2.1>과 같은 네트워크 환경에서 동작된다. 그림 <2.1>에서 호스트 컴퓨터 A에 접속된 터미날 T1은 호스트 A를 이용하기 위한 전용 단말기로 정의할 수 있다. 네트워크 환경에 의해서 호스트 컴퓨터 A와 B가 연결된후에 터미날 T1은 자기가 접속된 호스트 컴퓨터 A뿐만 아니라 네트워크 application program(remote login 또는 Virtual Terminal Application)을 통하여 마치 호스트 컴퓨터 B에 연결된 터미날처럼 호스트 컴퓨터 B의 사용이 가능하다.

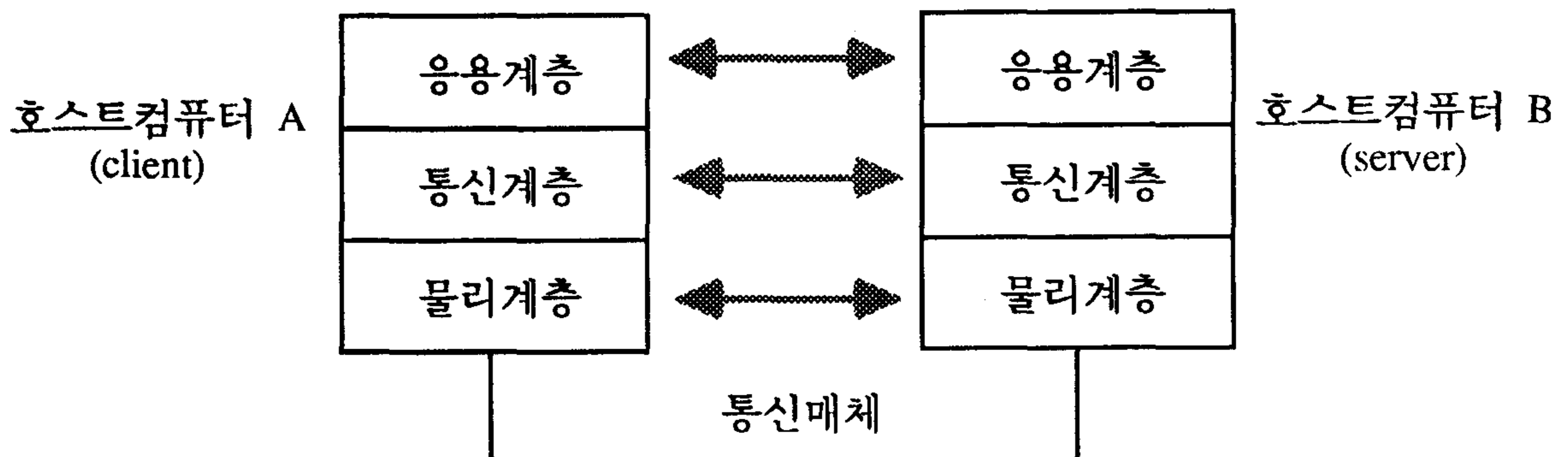


<그림 2.1>

그림 <2.1>에서 호스트 컴퓨터는 터미날 T1에 관련된 정보 (scroll기능, cursor up, down 기능)를 기록한 database를 이용하여 터미날 T1이 마치 자기의 local terminal(터미날 T2)인것처럼 제어하게 된다. 초보적인 Virtual Terminal에서는 제어 기능이 대부분 터미날의 physical한 전송특성(parity, speed, echo enable/disable, flow control등)에 의존한 제어를 하는 것이 대부분이었으나, 점차 터미날의 종류가 다양해지고 터미날이 갖는 기능이 복잡해짐에 따라서 이러한 기능들을 충분히 이용할수 있는 형태의 Virtual Terminal 기능을 필요로 하게 되었다.

2.2 Virtual Terminal 수행 과정

일반적으로 Virtual Terminal 기능을 수행하기 위해서는 그림 <2.1>과 같은 환경에서 다음과 같은 계층적 소프트웨어가 필요하다.



호스트 컴퓨터 A와 B는 Virtual Terminal 서비스를 수행하기 위하여 상호 통신하여야 한다. 즉, A와 B는 양방향의 error-free, flow-controlled, transparent 통신 서비스를 각각의 응용계층인 Virtual Terminal server 및 client에게 제공한다. 실제 예로서 DoD에서는 이러한 통신 서비스로 TCP(Transmission Control Protocol) 및 IP(Internet Protocol)을 제공하며, rlogin, telnet등의 Virtual Terminal 응용은 TCP/IP service 위에서 동작하게 된다. CCITT(International Telegraph and Telephony Consultative Committee)에서도 X.29라는 프로토콜(PAD-to-PAD 프로토콜)을 제정하였는바 이는 터미날을 제어하고 양방향의 안전한 (flow control등의 문제점이 없는) 통신 서비스로 X.25 프로토콜을 사용하는 구조로 되어 있다. 즉 X.29 프로토콜은 X.25 service 위에서 동작하는 형태로 모델이 설정되어 있다. 이 밖에도 이와 비슷한 구조의 가상 터미날 서비스를 제공하기 위한 프로토콜 및 소프트웨어는 현재 수종이 있으며 주로 operating system에 내장되어 있거나 computer maker가 제작하는 형태로 보급되고 있다.

통신계층의 기본적인 기능을 물리계층을 통하여 전달되는 digital data stream을 기본으로 하여 에러가 없고, 순서적인 데이터를 virtual terminal을 위해서 동작되는 응용 프로세스(server/client)에게 제공하는 기능을 수행한다. OSI/RM(Open Systems Interconnection / Reference Model)에 따르면 OSI Session계층이 제공하는 서비스가 이에 해당될수 있다.

호스트 컴퓨터 A(client)의 virtual terminal응용 프로세스는 일반적으로 아래의 기능을 수행한다.

Local Terminal Control

Remote Host의 virtual terminal 응용과 통신

호스트 컴퓨터 B(server)는 일반적으로 아래의 기능을 수행한다.

Virtual Terminal처리기능(logon등)

Remote Host의 Virtual Terminal응용과 통신 기능

즉 호스트 컴퓨터 A는 local terminal T1에서 입력되는 모든 data stream을 Virtual Terminal응용 프로그램을 통하여 호스트 B로 전송하게되며, 호스트B의 Virtual Terminal응용 프로그램은 통신 서비스를 통하여 전달되는 data stream이 마치 local Terminal T2에서 전달된 것처럼 취급하여 처리하게 한다. 호스트 B에서 처리된 결과를 호스트 A에 연결된 터미날 T1으로 보내기 위해서, 호스트 B는 터미날로 전송될 data stream을 호스트B의 Virtual Terminal 응용 프로그램, 통신서비스를 이용하여 호스트 A의 Virtual Terminal 응용 프로그램으로 전송한다. 이때 호스트 A의 Virtual Terminal 응용은 호스트 B로부터 전송받은 data stream을 local terminal T1의 특성을 고려하여 (flow control, echo 상태등) 터미날로 전달하게 된다.

일반적으로 위와 같은 기능을 도식화하면 아래의 그림 <2.3>과 같은 기능을 각 호스트에서 수행하게 된다.

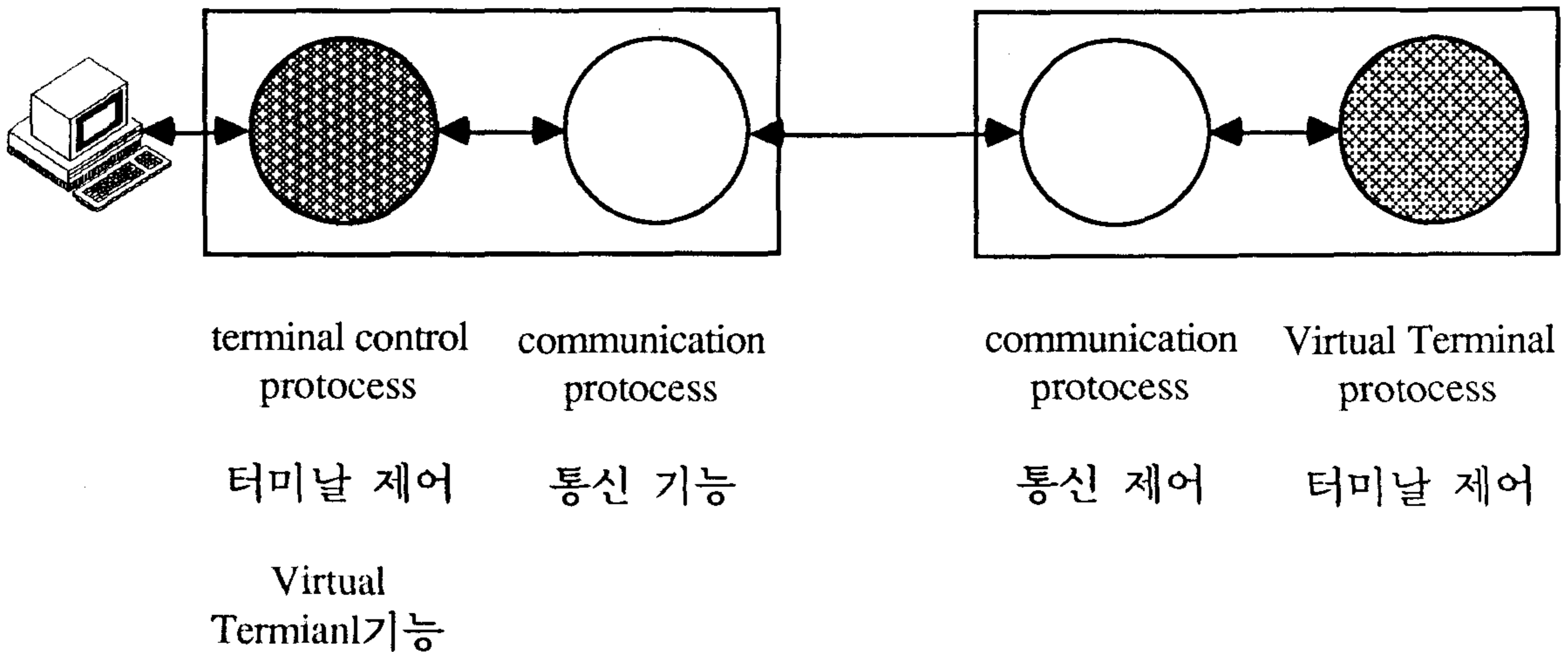


그림 <2.3>

현재 사용중이거나 재정중인 Virtual Terminal protocol/software들은 위의 그림 <2.3>에서 각각의 communication process가 갖는 기능 및 protocol 각각의 Virtual Terminal process가 갖는 기능이 각기 고유한 통신 서비스의 형태에 알맞는 형태로 구성되어 있다. 즉 DoD는 TCP/IP와 telnet, rlogin, CCITT는 X.25와 X.29등의 자기 고유한 통신 서비스가 반영될수 있도록 설계되어 있다.

2.3 Virtual Terminal Model

Virtual Terminal을 수행하기 위한 모델은 터미날이 갖는 기능, 통신서비스 모델등을 통합하여 결정된다. 즉 현재 다량의 이기종의 graphic terminal 및 graphic workstation등이 (IBM Graphic Cards, Graphic Terminal(TEK, SUN, MAC등) 다량 보급됨에 따라서 실제 Virtual Terminal기능의 수행시 어떠한 기능을 가진 단말이 사용되고 있는지가 중요한 요소가 등장된다. 즉 각각의 graphic terminal 또는 station

또는 dummy terminal들이 가지고 있는 기능이 각각 다르고 이들을 제어하기 위한 제어 시퀀스/명령등이 각기 다르기 때문이다. 따라서 호스트 컴퓨터가 이들 각각의 워크스테이션을 이용하여 접근될때 N-N의 mapping문제가 발생되며 이를 해결하기 위하여 보통의 경우 가상의 model을 설정하는 것이 일반적인 방법으로 인식된다.

즉 terminal이 dummy terminal(단순한 scrolling기능, cursor position, display기능등)의 기능을 갖느냐, 또는 복잡한 도형/문서등을 표현하는 기능을 갖느냐에 따른 계층적 모형으로 분류하고 이들 계층에 대한 가상의 terminal을 정의하는 방법이 현재 사용되고 있는 Virtual Terminal 모형이다.

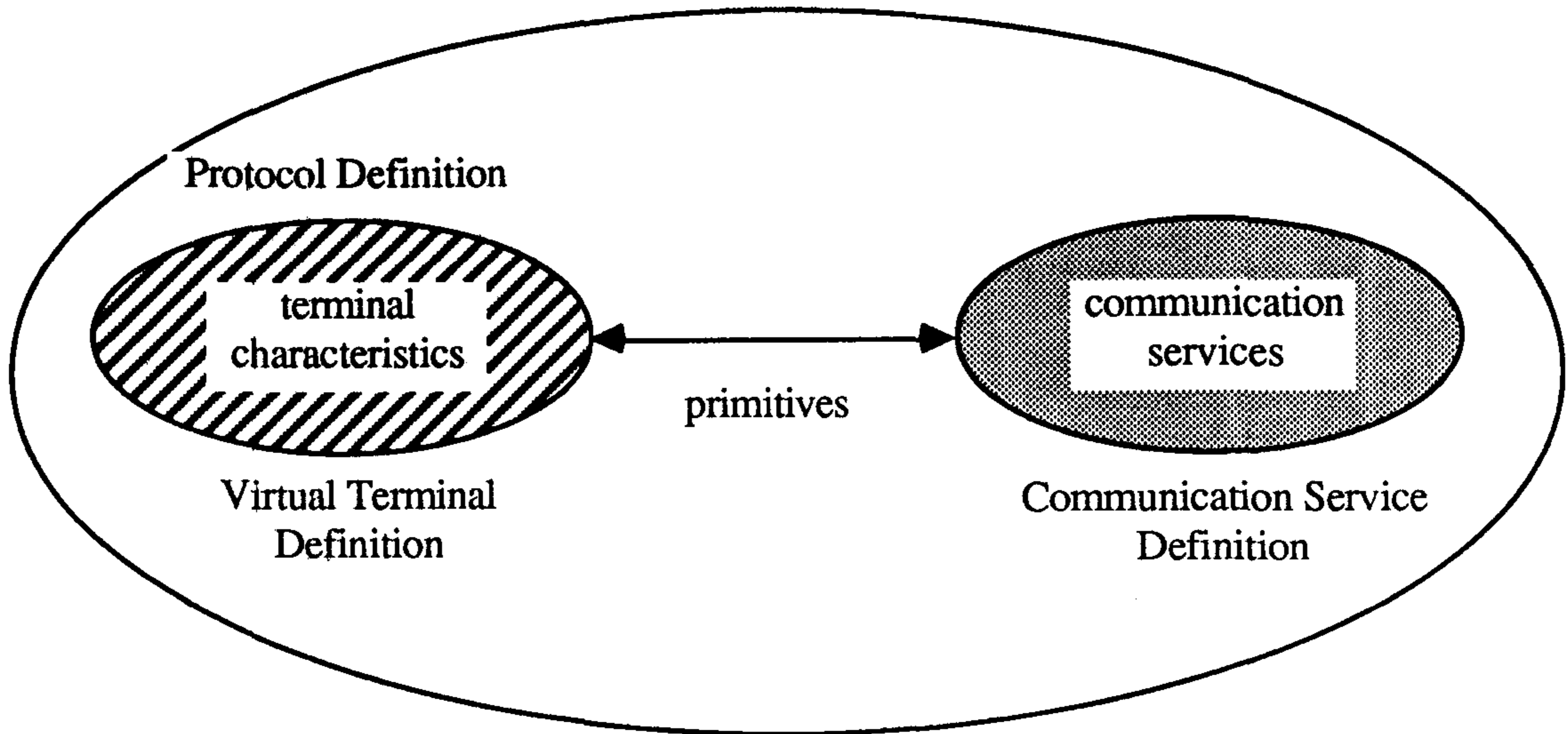
ISO에서 Virtual Terminal은 shared data area에 터미널 특성을 고려하여 데이터의 수정을 가하는 entity(producer)와 이 shared data를 이용하는 entity(consumer)로 modeling된다. 여기에서 shared data area는 conceptual communication area(CCA)로 불리우며 실제로 각 entity는 각각의 local CCA를 가지고 있으며 상대 entity로부터 이것이 변화된다는 정보를 입수하여 이의 수정을 가하게 된다.

ISO에서는 이러한 Virtual Terminal 응용이 사용되는 터미널을

- ◆ Basic Class
- ◆ Form Class
- ◆ Image Class
- ◆ Graphic Class

의 4가지 class로 분류하여 정의하고 있으며 현재 basic class에 대한 표준이 완성 단계에 있다. Basic class terminal은 scroll mode 및 line mode terminal로 구성되며 이러한 기능을 갖는 터미널을 제어하기 위한 서비스/프로토콜의 표준을 제정하였다.

Virtual Terminal의 수행을 위한 모델은 각각의 터미날의 갖는 기능을 고려하여 이들 터미날을 제어하기 위한 제어기능을 정의하는 것으로 이루어지는 데 일반적으로 아래의 그림과 같은 형태로 정의된다.



Virtual Terminal Model

즉 Virtual Terminal은 서비스될 terminal의 특성 및 통신 서비스 이들 entity간의 Interaction 방법등으로 모델링되며 각각 고유한 프로토콜 구조 및 대상 터미날등을 고려하여 정립된다.

3. Virtual Terminal Protocol 사례 분석

3.1 TELNET

ARPANET에서 사용된 TELNET Protocol은 터미널과 Process를 동일시하며 negotiation을 하기위한 primitive를 가지는 Virtual Terminal protocol이다. 아래의 그림<3.1>에서 네트워크 Virtual Terminal의 keyboard는 네트워크를 통하여 연결된 리모트 호스트의 스크린에 연결되며 사용자로부터의 입력을 상대방 호스트의 application에게 전달하도록 되어 있다. 여기에서 프로세스와 터미널은 네트워크 환경하에서 동일한 관점에 놓이게 되며 이러한 환경하에서 사용자가 자기의 Virtual Terminal에만 치중하는 처리가 가능하게 된다.

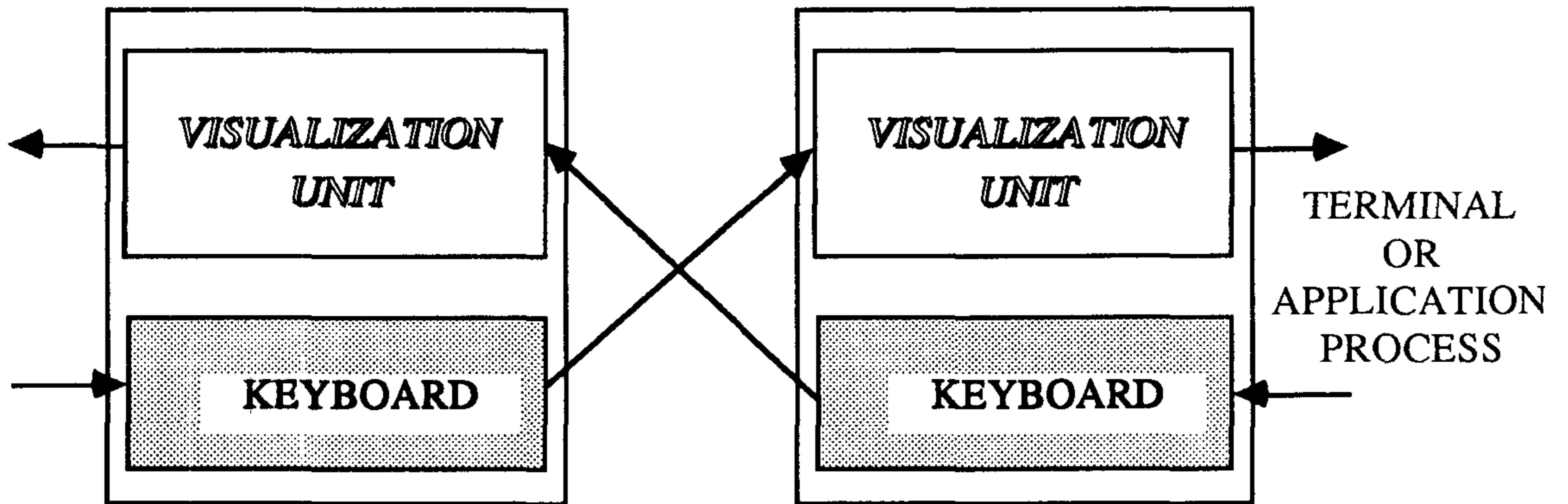


그림 <3.1>

TELNET Protocol에서의 negotiation은 복잡한 기능을 제공하기 위하여 다음의 4가지 방법을 사용된다.

- ◆ DO<option>
- ◆ DONT <option>
- ◆ WILL<option>
- ◆ WONT<option>

DO는 상대방에서 option을 처리하게 하며, DONT는 상대방에게 option을 수행하지 못하게 하며, WILL은 자신이 수행하도록 하고, WONT는 자기가 option을 수행하겠다는 것이다.

3.2 Super Duper(SUPDUP) 프로토콜

SUPDUP프로토콜은 TELNET Protocol의 super의 의미를 나타낸다. TELNET protocol이 scroll mode에 적합한 반면 SUPDUP protocol은 graphic terminal에 대한 operation을 수행할수 있는 기능이 있다. 또한 이 protocol은 remote text editor를 위한 local support 기능도 가지고 있어서 이러한 유형의 application에 유용하다. 즉 TELNET protocol이 network Virtual Terminal 이라는 것을 teletype로 간주하는 반면 SUPDUP에서는 그림 및 text를 처리하는 display를 그 기본 단말로 간주한다는데서 TELNET과 차이점이 있다. SUPDUP는 양방향의 8-bit stream의 data stream pair를 이용해서 동작되며 ITS(The Incompatible Time Sharing System)의 Input/Output Buffer Format에 기초하여 시작되었다.

SUPDUP가 동작하는 환경은 아래의 그림과 같다.

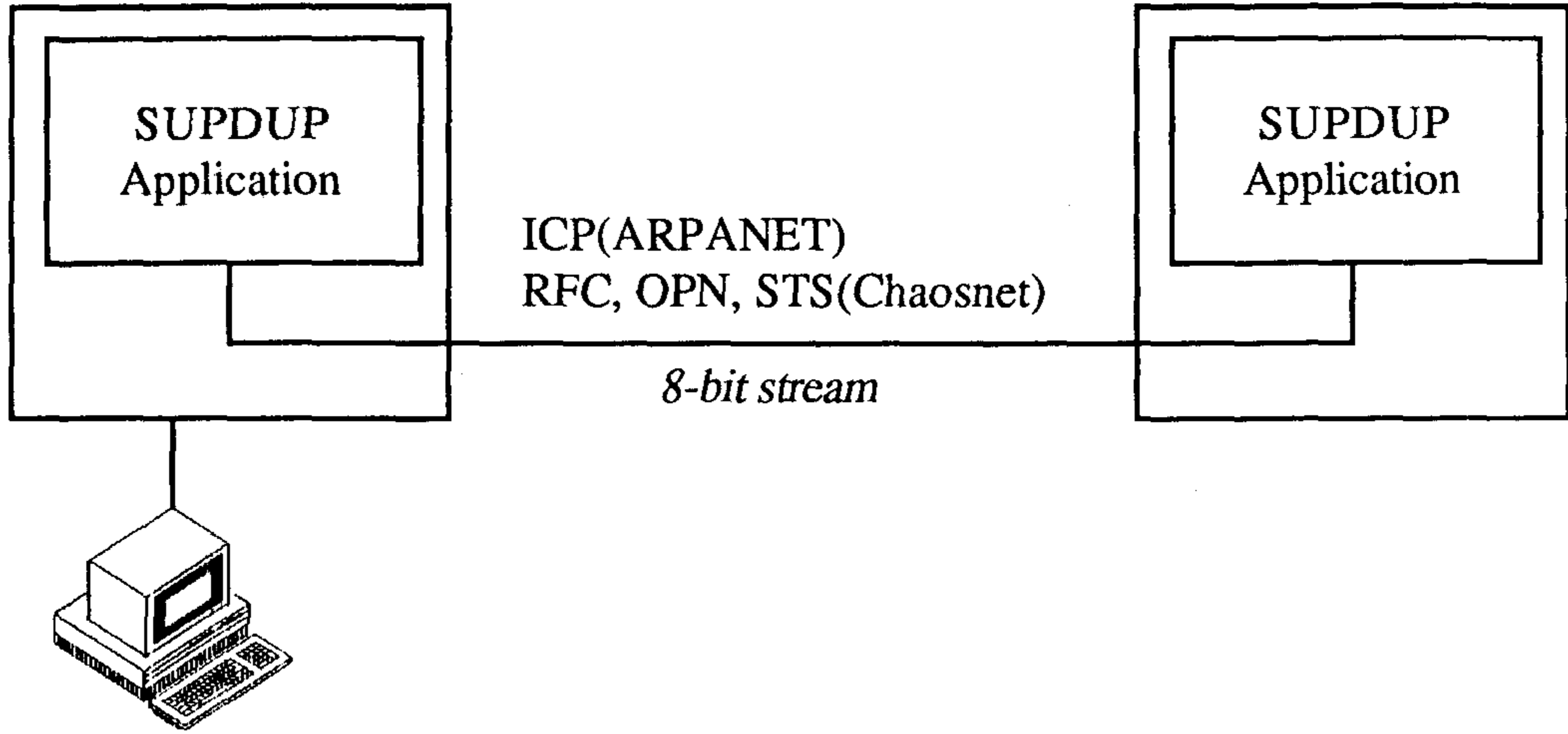


그림 <3.2>

SUPDUP는 ASCII text로 구성된 local greeting message와 함께 terminal characteristic을 remote system으로 보내는 것으로 시작된다. Terminal characteristic에는

- ◆ Convert lower case to upper case
- ◆ Terminal Can do ERASE
- ◆ Terminal Can move cursor BACK
- ◆ Terminal Can OVERPRINT
- ◆ Terminal Can Insert/Delete
- ◆ Local system supports SUPDUP rectangle

등의 30여개의 terminal characteristic이 전달된다. 이러한 터미날의 기능은 상대측 SUPDUP application이 어떠한 Terminal control primitive (예, 사각형, line-up등)을 이용하여 상대측 터미날을 제어할것인가를 결정하기 위한 기초 정보가 되며, 이러한 정보를 상대측 SUPDUP server가 네트워크를 통하여 연결된 터미날에 원하는

형태의 출력을 하기위한 기본 자료로 이용된다.

SUPDUP프로토콜은 MIT(Massachusetts Institute of Technology)의 TTS의 모든 terminal에서 가능함은 물론 8-bit code체계를 이용하는 모든 operation system에서의 응용이 가능하다. SUPDUP protocol은 위에서 열거된 기능들을 이용할수 있는 많은 display명령어를 이용하며 (bit 형태로 encoding된) 이러한 primitive를 수행하기 위한 message 형식을 위한다. 현재 TTS에서는 8-bit와 36-bit code를 이용하며, 실제 SUPDUP protocol은 터미날의 종류와 무관하게 사용하는 것이 가능하다.

3.3 CCITT X.3, X.28, X.29, X.25

CCITT에서는 Synchronous packet switched network(X.25)에 Asynchronous terminal들을 접속하기 위하여 PAD(Packet Assembler/Deassembler)라고하는 protocol을 정의하였다. PAD는 Public network (X.25망)에 접속된 어떠한 HOST와 X.25 connection을 통하여 사용자에게 Virtual Terminal 기능을 제공하도록 정의하였다.

CCITT권고안 X.3, X.28, X.29는 Network에서 이용되는 Unintelligent start-stop(일명 asynch terminal) terminal을 어떻게 동작하게 하느냐 하는 방법을 정의하고 있다.

3.3.1 Asynchronous Terminal을 위한 Protocol

Asynchronous terminal은 Dummy형태의 terminal로서 X.25망에 접속하여 X.25망에 접속하는 장치는 X.25 packet을 구성할 수 있는 기능을 갖는 장치가 되어져야 한다. 일반적으로 Asynchronous 터미날은 Unintelligent 형태이기 때문에 X.25 packet을 처리할 수 있는 기능이 없다. 또한 이 터미날들의 사양들도 다양한 형태로 존재한다. CCITT에서는 X.25망에 Asynchronous 터미날을 접속할수 있도록

정의하고 있다. 그림<3.3>

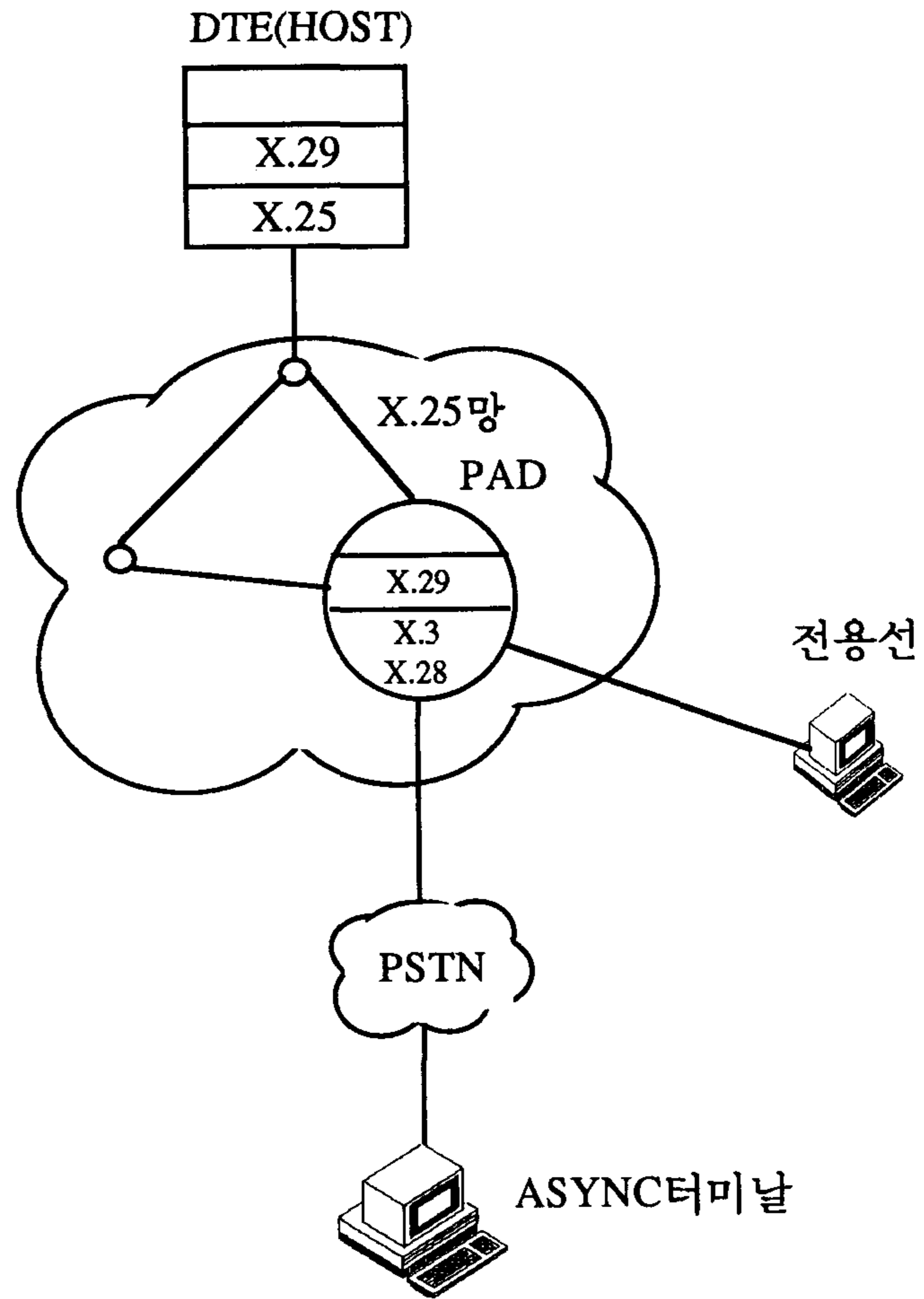


그림 <3.3>

3.3.1.1 Command Mode와 Data Transfer

Asynchronous terminal이 PAD에 연결되어 있을때 PAD는 command mode 또는 data transfer mode로 동작을 한다.

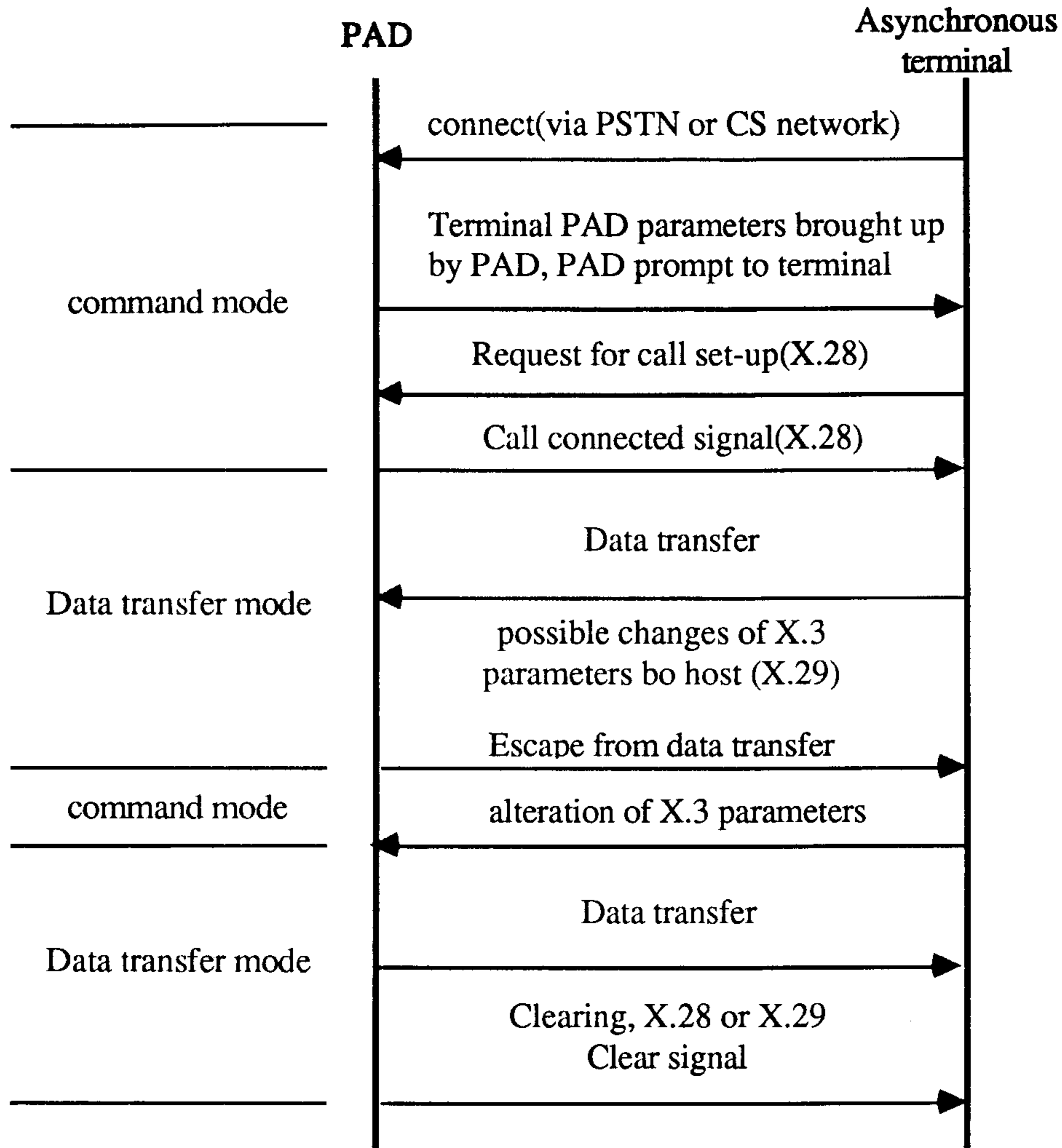
Command Mode

사용자가 PAD에 접속되었을 때 사용자는 command mode에 있게 되고 터미널 screen에 쓰여지는 모든것을 PAD에서는 command로 번역되고 이는 remote X.25 HOST에는 전달되지 않는다.

Data Transfer

터미널이 Data Transfer Mode에 있을때 터미널에서 PAD로 전송되는 모든 character 들 PAD에서 Packet으로 모으고 한 packet이다. 채워진 경우 또는 carriage return 과 같은 Spcial Character가 PAD로 전송되었을 때 이 Packet는 Data로서 remote X.25 HOST에 전달된다

Command Mode and Data Transfer



3.3.1.2 PAD Parameters(X.3)

X.3는 PAD에서 terminal을 control하기 위한 parameter들을 정의하고 있다. parameter value들은 PAD에 미리 setting 할수 있으며 터미날 사용자 또는 X.25 HOST에 의하여 변경할수도 있다.

X.3 parameter는 22개를 정의한다.

Parameter number :

- 1: Escape from Data Transfer
- 2: Echo
- 3: Data Forwarding Signal
- 4: Idle timer
- 5: Ancillary Device Control
- 6: PAD Service Signals
- 7: Procedure on Break
- 8: Discard Output
- 9: Carriage Return Padding
- 10: Line Folding
- 11: Terminal Speed
- 12: Flow Control of the PAD by the terminal
- 13: Linefeed Insertion
- 14: Linefeed Padding
- 15: Editing
- 16: Character Delete
- 17: Line Delete
- 18: Line Display
- 19: Editing of User Data
- 20: Echo mask
- 21: Parity Treatment
- 22: Page Wait

3.3.1.3 X.28 Commands

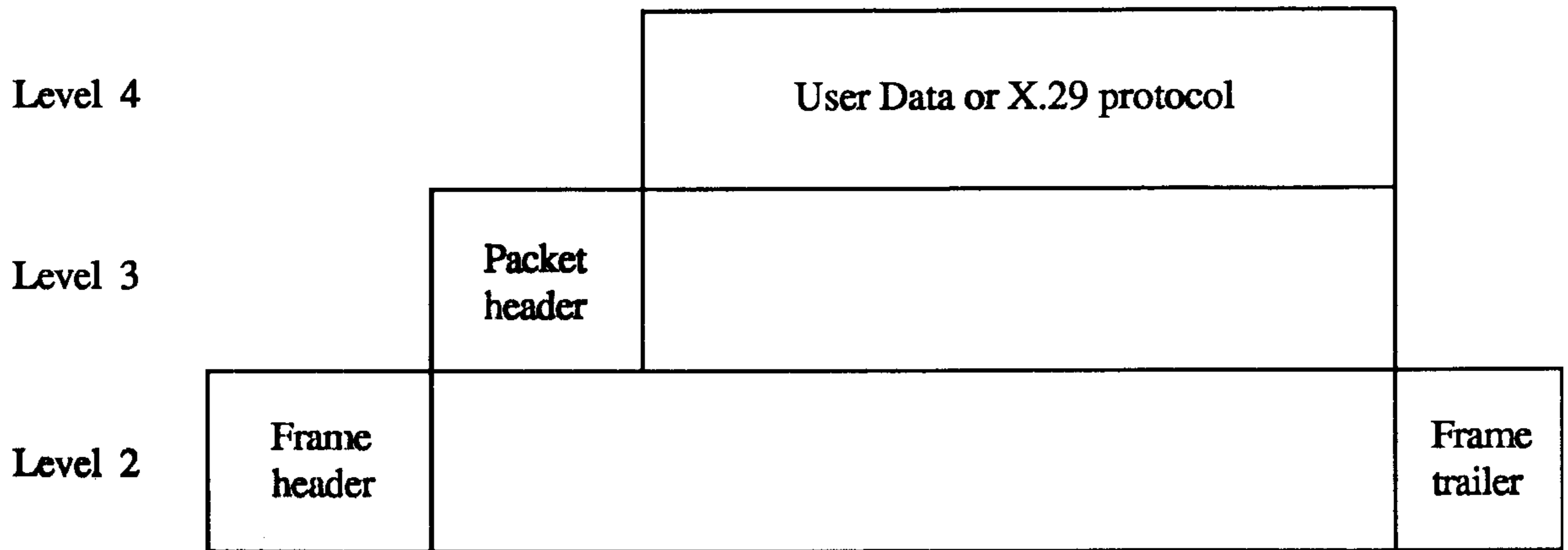
X.28은 Asynchronous 터미널과 PAD간의 인터페이스를 정의한다. X.28안에서 Command들은 아래와 같은 기능을 수행한다.

- Setup/clear virtual call
- 터미널에 적당한 표준 profile을 선택한다.
- PAD parameter들의 Value를 변경한다.
- Interrupt packet을 전송한다.
- Virtual call의 status에 관한 정보를 요청한다.
- Virtual call의 Reset을 요청한다.

3.3.1.4 X.29 protocol

X.29 protocol은 PAD와 X.25 HOST사이에 보내지는 control message들을 정의하고 있다. X.29는 network protocol에 속하여 있으면서 실제로 접속이 이루어진후 data 전송에는 기능을 담당하질 않는다. X.29 packet이 일반 X.25 data packet과 구별하기 위하여 Data packet안에 Q bit field에 '1'로 setting되어진다.

Protocol levels



Data Qualified packet

Q

1	D	0	1	LCGN
LCN				
P(R)	M	(P)S	0	
X.29 protocol				

3.4 ISO/OSI Virtual Terminal

3.4.1 Mode

ISO/OSI가 권고하는 국제표준형 Virtual Terminal에서는 Conceptual Communication Area(CCA)라는 가상의 기억장소를 이용하며 Conceptual Data Store(CDS), Control Signalling and Status(CSS), Access Control Store(ACS), Data Structure Definition(DSI)등으로 구성된다.

Conceptual Data Store는 양 Virtual Terminal 서비스 사용자로부터 데이터를 지정하기 위한 장소로서 이들 데이터는 display에 출력되거나 또는 text의 편집에 이용되고 Virtual Terminal 사용자와의 통신에 이용된다. CSD에는 데이터를 저장하기 위한 "display object"이 존재하며, Control signalling and status는 Virtual Terminal을 제어하고 상태를 점검하기 위한 상태로 이용된다[3]. Access Control Store는 half-duplex mode terminal인 경우 Conceptual Data Store에 접근할수 있는 관리를 갖도록 하거나 또는 CDS에 접근한 사용자에 대한 정보를 갖는다. DSD는 CDS와 CSS이 parameter값을 가지며 이들의 관계는 아래의 그림<3.4>와 같다.

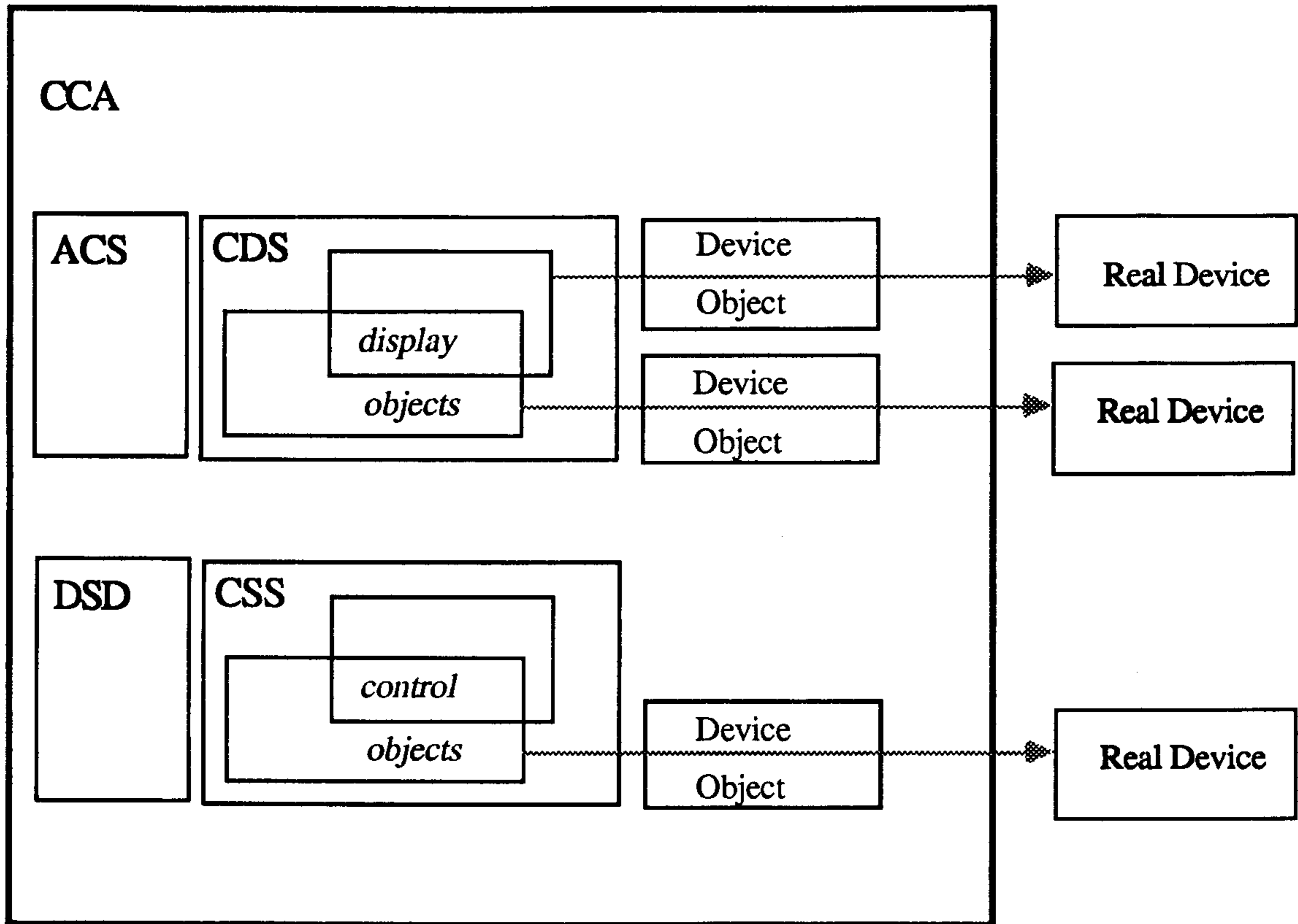


그림 <3.4>

3.4.2. ISO/OSI Virtual Terminal Environment

OSI Virtual Terminal은 아래의 그림 3.5와 같은 계층적 통신환경에서 동작하게 된다. 즉 OSI Virtual Terminal은 OSI presentation service를 기본으로 하여 동작되며 presentation계층은 Session등의 계층적 서비스 환경에서 정의되었다.

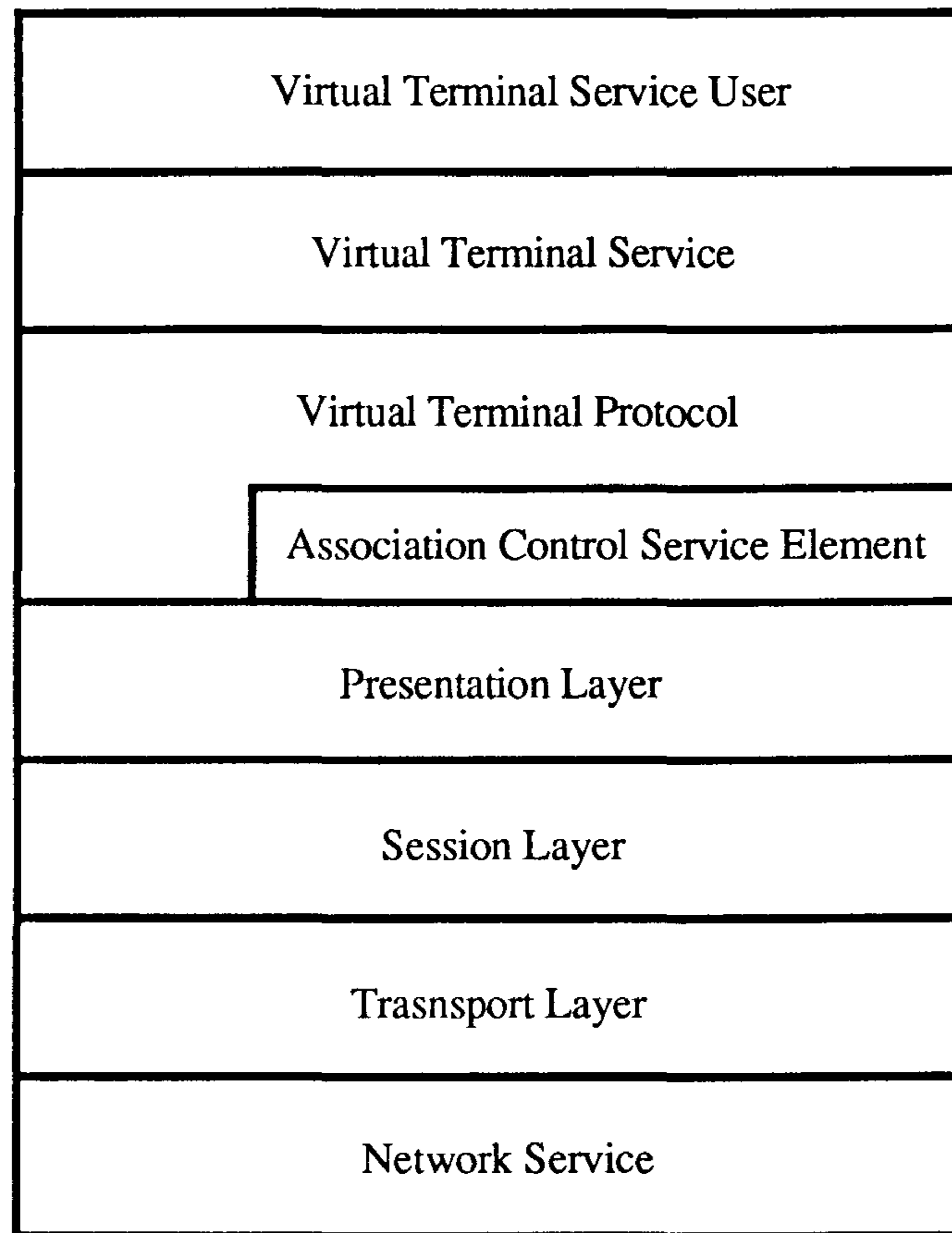


그림 <3.5>

위의 그림 3.5에서와 같이 OSI Virtual Terminal이 구성되는 모델은 OSI/RM(Open System Reference Model)로서 CCITT의 X.25등과 근본적인 차이점이 있다. 즉 OSI Virtual Terminal은 Physical communication media와 independent하게 OSI presentation service 위에서 동작되게 되며 실제 presentation service가 제공하는 encoding/decoding 등의 기능을 이용하게 된다.

PRESENTATION

ISO/OSI Presentation 계층은 복잡한 application(Virtual Terminal, file transfer, transaction

processing등)에서 사용되는 복잡한 data structure를 상호 시스템의 내부 표현방식 (예, 1's complement, 2's complement, ASCII, EBCDIC, structure등의 내부 표현방식등)과 무관하게 상호 약정된 정보형태(abstract syntax)를 전송형태(transfer syntax)로 표현하는 기능을 수행한다. 현재 abstract syntax를 표현하기 위한 도구로는 Abstract Syntax Notation One(ASN.1)이 사용되고 있으며 transfer syntax를 표현하기 위한 encoding rule로서 "Basic Encoding Rule"이 사용되고 있다.

Presentation계층이 제공하는 서비스를 이용하여 응용계층은 응용계층이 상호 교환하여야하는 데이터를(예, 터미널제어를 위한 데이터)를 상호 내부 표현방식과 무관하게 전송이 가능하다. 또한 Presentation계층은 application에서 사용하는 abstract syntax와 transfer syntax의 조합으로 구성되는 "presentation context"를 변환시킬수 있는 기능을 제공하는데 이것은 복잡한 응용이 presentation context를 변환하여 여러 형태의 응용과 동시에 통신할 때 매우 유용한 특징이다.

즉 Presentation layer의 주요 기능은 presentation context management라고 볼수 있으며 이외에도 presentation계층은 session계층에서 제공하는 Dialogue Control등의 기능을 이용하여 application계층에게 session이 제공하는 서비스를 제공하기 위한 primitive 들도 역시 제공한다.

Presentation layer protocol은 이러한 일련의 서비스를 제공하기 위한 기능들을 모여서 몇개의 functional unit으로 분류하며 이 functional unit에는 presentation의 기본 서비스인 kernel functional unit 및 context management function unit등으로 분류된다.

SESSION

Session layer는 OSI transport service를 지원받아 session user에게 하나의 통신

session을 control하기 위해 필요한 기능을 제공한다. session이 제공하는 주요한 기능은 Dialogue Control, Synchronize, Activity Management 기능으로 볼수 있다. 즉 Session 계층은 두개의 응용이 상호 통신할때 Physical 접속 형태와 무관하게 응용 계층이 양방향의 통신이 가능한 mode(full duplex), 단방향만의 통신이 가능하도록 하는 half-duplex mode등을 제공한다. 또한 상호 동기를 맞추고, 이것을 바탕으로 application 이 recovery 기능을 수행하기 위한 도구를 제공한다.

Session계층은 또한 Session의 사용자가 transport connection과 무관하게 session service를 받을수 있도록 transport service 및 transport connection을 control하며 이러한 기능을 수행하기 위하여 transport connection의 재사용등의 기능을 포함하고 있다. 종합적으로 session계층은 application 하나의 communication session을 열어서 닫을때 까지 필요한 session control 기능을 제공하며 session user에게 데이터 path를 만들어 주는 기능을 수행한다. Session이 이러한 기능들을 제공하기 위한 기능군인 functional unit은 다음과 같다.

Functional Unit	SPDU code	SPDU Name
Kernel	CN	Connect
	OA	Overflow Accept
	AC	ACCEPT
	RF	Refuse
	FN	Finish
	DN	Disconnect
	AB	Abort
	AA	Abort Accept
	DT	Data Transfer
Negotiated	NF	Not Finished

release	GT	Give Token
	PT	Please Tokens
Half-duplex	GT	Give Tokens
	PT	Please Tokens
Duplex		No Additional SPDUs
Expedited data	EX	Expedited Data
Type Data	TD	Type Data
Capability data	CD	Capability Data
exchange	CDA	Capability Data Accept
Minor	MIP	Minor sync Point
Synchronize	MIA	Minor sync Ack
	GT	Give Tokens
	PT	Please Tokens
Major Synchronize	MAP	Major Sync Point
	MAA	Major Sync Ack
	PR	Prepare
	GT	Give Tokens
	PT	Please Tokens
Resynchronize	RS	Resynchronize
	RA	Resynchronize Ack
	PR	Prepare
Exceptions	ER	Exception Report
	ED	Exception Data
Activity management	AS	Activity Start
	AR	Activity Resume

AI	Activity Interrupt
AIA	Activity Interrupt Ack
AD	Activity Discard
ADA	Activity Discard Ack
AE	Activity End
AEA	Activity End Ack
PR	Prepare
GT	Give Tokens
PT	Please Tokens
GTC	Give Token Confirm
CTA	Give Token Ack

TRANSPORT/NETWORK

OSI Transport 및 Network 계층을 Physical Media를 통하여 전달되는 데이터를 처리하여 사용자(응용 계층)가 low control multiplexing, splitting 등의 데이터 전송에 직접 관련된 부분에 대한 서비스를 제공받을 수 있도록 한다. 네트워크 계층은 통신하고자 하는 응용 entity의 physical location에 관련 없이 상호 데이터의 전송이 가능하도록 하는 기능을 수행한다. Network layer가 수행하는 주요 기능은 아래와 같다.

- ◆ Addressing
- ◆ Relaying
- ◆ routing

이러한 기능을 이용하여 transport 계층을 session layer등과 같은 transport 사용자가 network layer에 무관하게 안전한 접속 path를 만드는 한편 상대측 transport사용자와 local transport 사용자 사이에 양방향의 전송 경로를 만들어 주는 기능을 한다. Transport layer는 그 기능 및 하위계층 네트워크 서비스 제어기능등에 의하여 Class0부터 Class4까지 다양한 기능을 제공한다. 즉 transport의 사용자는 필요한 transport service 및 제어해야 할 네트워크 계층 및 네트워크 계층이 제공하는 서비스등을 고려하여 transport class를 정하게 된다. Transport layer가 제공하는 기능은 아래와 같다.

- ◆ Network Connection Assignment
- ◆ Transport PDU transfer
- ◆ Segmenting and Reassembling
- ◆ Concatenation and seperation
- ◆ Connection Estabilishment
- ◆ Connection Refusal
- ◆ Normal Release
- ◆ Error Release
- ◆ Association of TPDU's with Transport Connection(TC)
- ◆ Data TPDU numbering
- ◆ Expedited Data Transfer
- ◆ Reassignment after failure(network)
- ◆ Resynchronization
- ◆ Retention Until acknowledgement of TPDU's
- ◆ Multiplxing and Demultiplexing
- ◆ Explicit flow control
- ◆ Checksum

- ◆ Frozen Reference(transport connection)
- ◆ Retransmission on Timeout
- ◆ Resequencing
- ◆ Inactivity Control(network connection)
- ◆ Treatment of Protocol Errors
- ◆ Splitting and recombining

위의 transport layer기능들은 각 class에 따라서 적용되는 기능과 적용되지 않는 기능으로 분류될수 있다. 이러한 기능들의 사용 여부를 상호 확인 하기 위해서 transport layer는 connection을 형성할때 이러한 기능의 사용여부에 대하여 negotiation 할수 있도록 되어 있다. 일반적으로 선택되는 class에 따라서 기능이 정해지며 negotiation은 보통 negotiated-down으로 이루어진다.

3.4.3 OSI Virtual Terminal Protocol

OSI Virtual Terminal protocol은 터미날을 이용하는 사용자에게 Virtual Terminal service를 제공하기 위한 수단으로 간주될수 있다. 즉 Virtual Terminal 응용 프로세스는 Virtual Terminal service를 수행하기 위한 server로 간주된다. 즉 Virtual Terminal 응용 프로세스를 이용하여 어떤 호스트를 이용할때 그 호스트에서는 Virtual Terminal client응용 프로세스가 상대측 remote host에는 Virtual Terminal server가 가동되는 것인 일반적인 Virtual Terminal mode1이다. 여기에서 Virtual Terminal 응용 프로세스가 가지는 기능은 TELNET등의 경우에서와 같이 사용자측의 터미날이 어떠한 형태의 터미날인가에 의해서 많이 좌우된다.

ISO에서는 이러한 사용자 터미날의 기능에 의하여 basic class, form class, image class, graphic class로 나누어서 각기 해당되는 표준 프로토콜을 제정하고 있다. 현

재 많이 진행되어 있는 것은 "basic class"로서 scrolling등의 기능이 있는 기본형 터미날이 표준 모델로 정의된다.

Basic Class에서 수행되는 Operation은 S-mode 및 A-mode로 분류되며 S-mode는 twoway alternate dialogue를 이용하여 어느 한 순간에 한 Virtual Terminal user(access right가 있는)만이 display object를 제어할수 있도록 하고, A-mode에서는 두 Virtual Terminal 사용자가 임의로 display object를 임의로 제어할수 있는 mode를 말한다.

Basic class에는 optional capability에 따라서 제공되는 몇개의 기능군으로 나누어지는데 이것은 아래와 같다.

- ◆ Kernel
- ◆ Switch Profile Negotiation
- ◆ Multiple Interaction Negotiation
- ◆ Negotiated Release
- ◆ Urgent Data
- ◆ Break

ISO Virtual Terminal은 display object에 대한 가능정의 및 이들의 display object에 대한 control object으로 구성되며 실제의 device가 display object에 device object을 통하여 mapping되는 형태로 구성되어 있다. Display object은

- ◆ display-object-name
- ◆ DO-access
- ◆ Dimensions
- ◆ erase-capability

- ◆ foreground-color-capability
- ◆ background-color capability

등의 primary parameter 및 secondary parameter로 구성된다.

Secondary parameter에는 bound, windw, address등의 parameter가 적용되며 이들 parameter가 update등의 control 기능을 수행하기 위해서 이용된다.

이들이 parameter를 이용하여 Virtual Terminal service가 실행될때 최초로 connection을 형성하는 과정을 거치게 된다. 즉 OSI presentaion 및 ACSE를 이용하여 Virtual Terminal 사용자간에 connection이 형성되며 이 과정에서 negotiation이 이루어지게 된다. Connection estabilishment시 local host의 특성을 remote 호스트의 Virtual Terminal server에게 보냄으로서 Virtual Terminal을 정의할 수 있는데 이러한 방법이 가장 초기 단계인 VT-A등급으로 분류된다. 또한 Single-Interaction-Negotiation을 수행할수 있는 시스템을 VT-B등급, Multiple-Negotiation을 수행할수 있는 시스템을 VT-C등급으로 분류하고 있다.

VT-B등급은 Profile이름과 profile parameter값을 한번만의 negotiation으로 바꾸도록 하는 것이다. 즉 새로운 Virtual Terminal을 정의가 가능하며 이러한 기능은 더욱 동적인 Virtual Terminal의 정의가 양쪽 Virtual Terminal에 관련된 프로세스 사이에 일어나는 것이 가능하도록 만들어준다.

3.5 X.WINDOW

X.WINDOW System은 graphical user interface를 개발하는 프로그래머들을 위한 Industry standard software system이다. X.WINDOW의 가장 중요한 특징중에 하나는 단일한 device-independent architerture를 갖는 것이다. X protocol은 이를 제공하

는 어떠한 hardware위에 graphic와 text를 포함하는 windows들을 display할수 있다. X-based applicaiton은 mainframe, workstation, personal computer를 포함하는 이기종환경 하에 수행할수 있도록 관련 기능들을 제공한다.

3.5.1. Architecture

X.WINDOW System architecture는 client-server model을 기반으로 하고 있다. server는 screen위의 window들을 생성/처리하고, xt/graphic들을 처리하며, keyboard와 mouse와 같은 input device들을 조정한다. [그림 <3.6> 참조]

X.server가 제공하는 기능을 이용하는 application을 client라 말한다. client는 x.server와 asynchronous byte stream protocol을 이용 network connection을 경유하여 통신을 한다. X.protocol을 TCP/IP, DECnet을 포함하는 많은 network protocol들을 제공한다. 다수의 client들은 한 server에 동시에 접속될수도 있고 한 client가 다수의 server와도 가능하다.

X.architecture는 server에 구현된 device-dependent한 부분은 사용자에게transparent하도록 되어 있다.

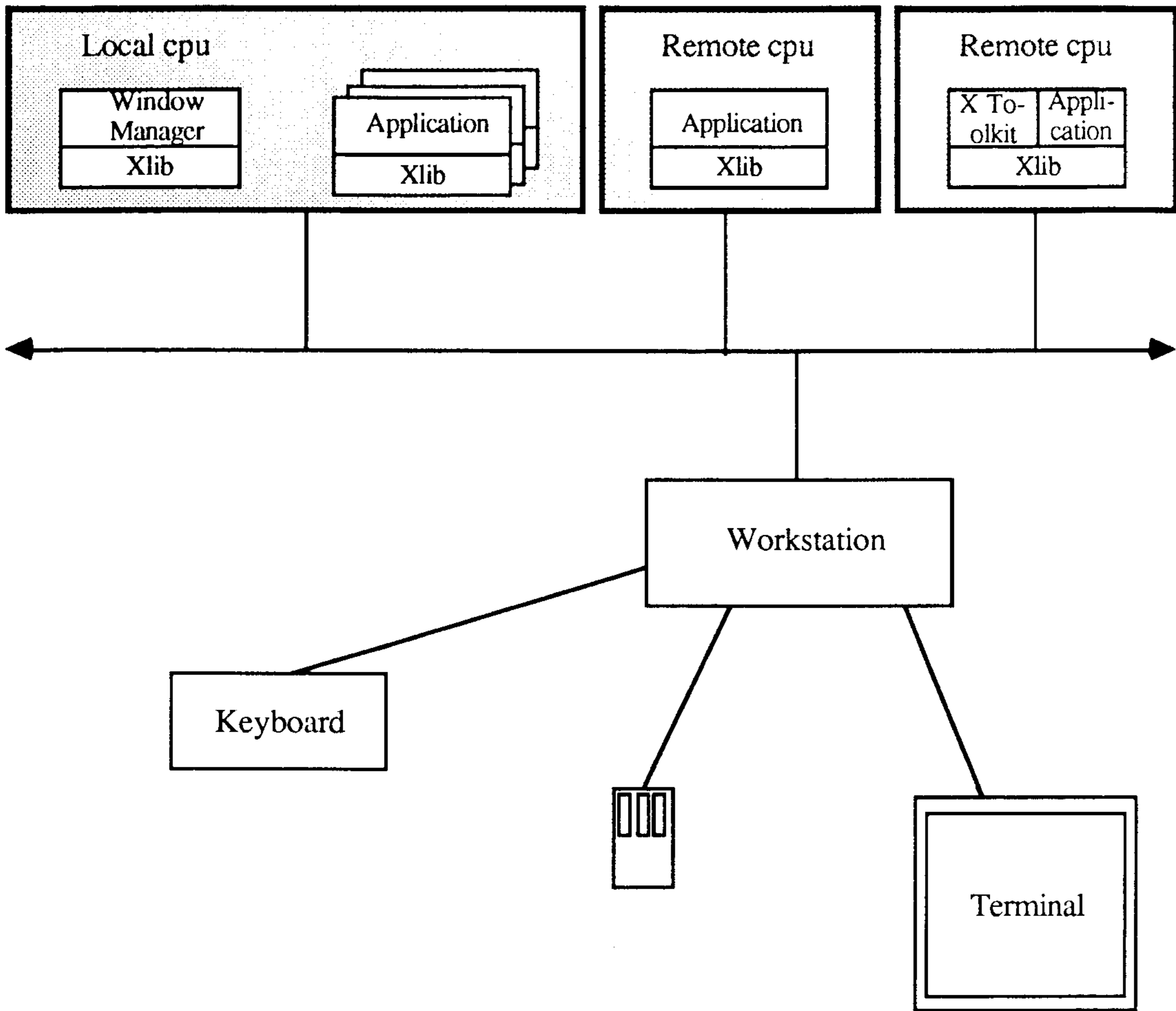


그림 <3.6>

4. PC/TCP 한글화

4.1 배경

TELNET, X.29등의 Virtual Terminal이 wide area-network을 그 동작 모델로 설정한 반면 PC/TCP는 personal computer를 대상으로 Virtual Terminal을 구현한 것이라고 볼 수 있다. 즉 PC의 보급 확장과 PC를 단순한 word processor의 개념을 넘어서 network상의 하나의 desk top processor로 사용하려는 추세가 확장됨에 따라서 PC에 network 기능이 점차 부각되기 시작했다.

PC/TCP는 근거리 LAN등에서 다수의 PC와 호스트 컴퓨터 사이에 고가의 주변정치를 연결하여 사무실, 빌딩, 공장, 사무 및 공장 자동화의 실현을 위한 네트워크 기술이다. 이것은 이제까지 각 개인별 업무처리에 한정적으로 사용되어 왔던 PC를 네트워크로 연결하여 네트워크 내의 호스트 접근등의 기능을 수행한다.

PC/TCP의 기능은 아래와 같다.

- ◆ Network 제어 기능
- ◆ file transfer
- ◆ mail transfer
- ◆ Virtual Terminal

PC/TCP는 이러한 기능을 수행하기 위한 hardware 및 software로 구성되며 hardware에는 PC-LAN을 이루기 위한 PC communication board, transiver, cable등이 있고 software는 communication controller board와 operation system(MS-DOS)와의 interface 기타 위의 기능을 제공하기 위한 software module들이 존재한다.

PC/TCP가 동작되는 환경은 IBM 호환 XT, AT 모두가 가능하며 DOS version 2.0이

상이 필요하다. 기존의 PC/TCP는 이러한 hardware 및 operating system environment가 구비된 시스템에서 사용이 가능하나 궁극적으로 한글이 포함된 data의 전송시 (file transfer, telnet) 서로 다른 한글 code를 사용함으로써 한글 호환성 문제가 발생하게 되었다. 즉 현재 사용되는 한글 code는 KSC-N, KSC-16, KSC5601 등 여러가지 형태가 있고 각 PC maker가 서로 다른 한글 체계를 수용하기 때문에 호환성 문제가 발생되었다.

따라서 한글 code를 행정망 한글 code(KSC5601)을 중심으로 각서가 conversion과정을 수행하여야 하며 이러한 문제는 PC만의 문제가 아니라 LAN상에서 사용되는 computer system(micro, mini, main_frame)에서도 역시 상호 문제점으로 나타나고 있다. 이러한 환경은 아래의 그림과 같다.

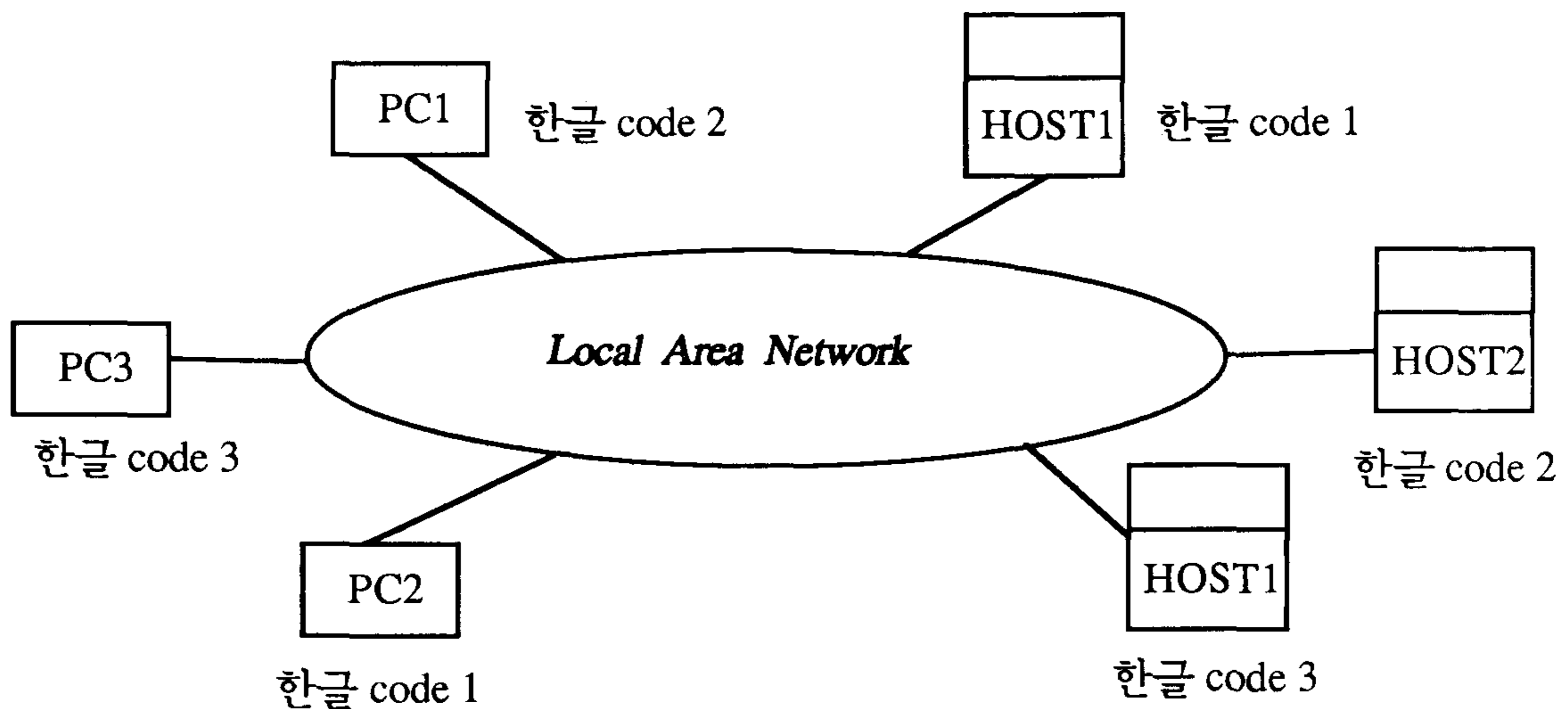


그림 <4.1>

위와 같은 환경에서 한글 code가 서로 같은 시스템들 사이에는 한.영 모드의 모든 application이 가능하나, PC1과 Host3또는PC2와는 한글 호환성이 없어서 문제점이 될수 있다. 즉 위의 네트워크 환경에서 모든 PC에 PC/TCP를 설치 운영하는 것은 가능하나 실제 많은 응용이 한글로 되어 있는 업무등에는 이러한 한글 코드

변환의 어려움이 따른다.

4.2. PC/TCP개요

TCP/IP(Transmission Control Protocol/Internet Protocol)은 미국방성(DoD, Department of Defense)에서 이기종간 통신을 통한 자원 및 정보의 공유를 촉진하기 위해서 미국방성 표준으로 제정한 통신 프로토콜중의 일부에 속한다. Internet Protocol은 Network layer에 해당되는 프로토콜로서 relay, routing기능이 그 중요한 기능이다. 이것은 두개 이상의 network이 연결될때 TCP로 하여금 서로 다른 네트워크에 있는 통신 entity끼리 자유로운 통신을 수행할수 있도록 relay 기능을 수행하여 준다. 즉 IP는 network information, routing information등을 이용하여 IP의 사용자가 실제 network간의 연관관계등을 모르고 마치 단일 네트워크에 연결된 통신 entity처럼 상호 접속을 가능하게 한다.

이러한 Network Independent한 서비스를 IP로부터 받아서 TCP는 multipling, end-to-end communication control등의 transmission에 관련된 서비스를 제공한다. 즉 TCP의 서비스를 제공받는 응용은 network independent, transmission error-free한 서비스 위에서 application(ftp, rlogin등)에만 치중하는 프로세스를 동작시키는 것이 가능하다.

실제로 TCP/IP는 미국방성 표준으로 만들어진 이래 DEC, Applo, SUN, HP, Ptime등 대다수의 mini와 micor computer에 구현되어 현재 널리 사용중에 있다. TCP/IP의 개요는 다음의 그림과 같다.

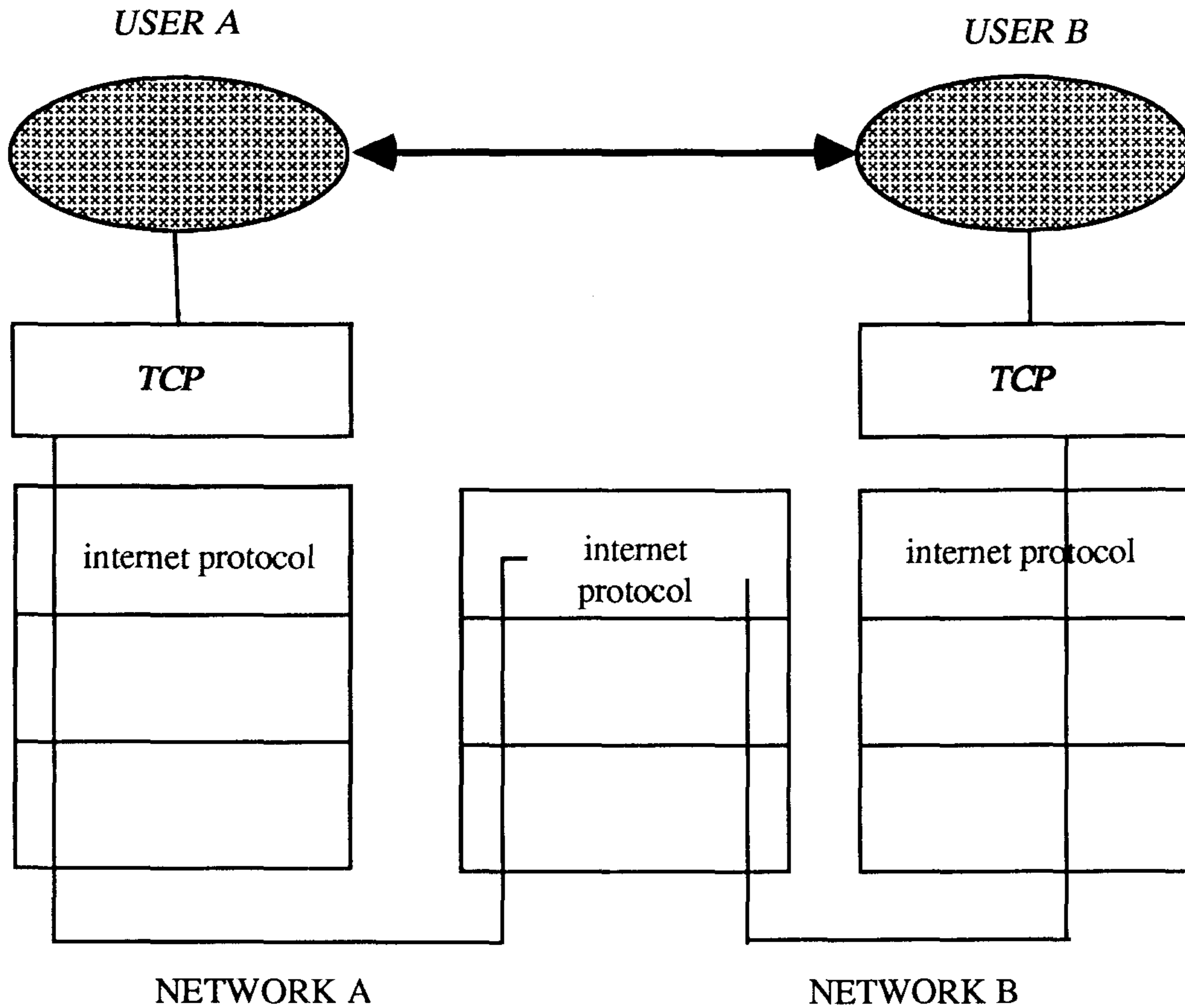


그림 <4.2>

PC/TCP는 ETHERNET, IBM Token Ring, ProNet, StarLAN등을 통하여 TCP/IP를 탑재한 컴퓨터와도 통신 가능하며 이기종간의 file transfer, mail transfer, remote command execution등의 기능을 통하여 네트워크내의 자원 공유를 가능하게 하는 기능을 제공한다.

4.2.1 PC/TCP의 기능

PC/TCP가 제공하는 기능들은 아래와 같다.

- ◆ 파일 전송 기능
- ◆ remote login
- ◆ 전자우편
- ◆ 인쇄출력
- ◆ remote-backup
- ◆ remote command execution
- ◆ network testing
- ◆ user information
- ◆ 한글지원
- ◆ 분산처리 데이터 베이스 제공

파일 전송은, local/remote의 호스트 file을 access할수 있으며 DoD file transfer protocol을 이용하는 어떠한 컴퓨터외도 통신이 가능하다. file transfer에는 file transfer기능 이외에도 remote directory의 listing, remote file의 delete등의 약간의 file management 기능도 제공한다. 이것은 현재 ISO standard에 정의된 FTAM protocol[ISO8571]에 정의되는 파일 관리 기능의 subset정도에 해당된다.

Remote login은, 네트워크상에 접속된 컴퓨터에 접속이 가능하며 teletype mode Virtual Terminal 처럼 동작한다. 즉 TCP로 부터 제공받는 transparent data stream과 pseudo device driver등을 이용하여 네트워크 사용자에게 네트워크 서비스를 통한 호스트 컴퓨터의 액세스를 가능하게 한다. Remote login 기능을 수행할때 VT100, VT52, HPIP, BIM3270등의 터미널 에뮬레이션 기능이 가능하다.

전자우편 기능은, DoD military standard인 RFC822 message format에 근거하며 네트워크상에 있는 호스트로 mail을 전송하기 위하여 역시 DoD standard인 SMTP(Simple Mail Transfer Protocol)을 사용한다. SMTP는 local host의 greeting

message로부터 시작하여(HELO), sender(MAIL FROM), Recipient(RCPT TO), mail content(DATA), 종결(QUIT)등의 command로 구성되어 있다. PC/TCP에는 SMTP server 기능도 가지고 있어서 SMTP server로서의 역할도 가능하다.

인쇄출력 기능은, remote spooling 기능을 이용하여 네트워크내의 다른 컴퓨터에 접속되어 있는 프린터를 이용할수 있는 기능을 제공한다. 이러한 spooling 기능을 이용하면 네트워크내의 여러 컴퓨터가 하나의 프린터를 마치 자기의 프린터처럼 사용할수 있는 기능을 제공하여 준다.

Remote backup 기능은, remote command execution에 근거해서 local file들의 backup을 remote tape driver에 backup 할수 있는 기능을 제공한다. 이러한 remote backup 기능은 network-wide resource sharing이라는 측면에서 매우 유용한 기능이다. 즉 네트워크(LAN)상에 연결된 tape driver가 부착되어 있지 않은 workstation등에 매우 중요한 정보를 포함하여 backup등의 기능이 필요로 할때, 하나의 tape driver를 여러개의 computer들이 공동으로 사용할수 있다는 점에서 또한 경제적이라 하겠다.

이 외에도 네트워크를 통하여 네트워크에 연결되어 있는 리모트 호스트에 등록된 사용자에 대한 정보를 알수 있는 finger daemon등을 이용하여 네트워크 환경에서 네트워크 사용자 정보를 알수 있는 기능을 제공하는 반면, 분산 정보처리를 위한 분산 DB와 같은 시스템의 구성시에도 응용이 가능하다. 또한 PC/TCP 한글화 작업의 결과로서 국내에 보급된 모든 IBM호환 기종의 한글을 완벽하게 처리하는 기능을 첨가함으로써 네트워크를 이용하여 한글 정보를 교환할때 발생하는 문제점을 제거하였다.

4.3 한글화 작업 개요

PC/TCP 에서의 Virtual Terminal은 리모트 호스트로 로그인 할수 있는 기능을 사용자에게 제공하는 것을 말한다. 여기서 대상 터미날은 teletype 형태의 dummy terminal이며, 7-bit또는 8-bit data stream을 그 기본 통신 channel로 이용한다. 즉 TCP프로세스의 서비스로 제공되는 data stream(8-bit)을 호스트의 터미날 사이의 가상의 통신회선으로 간주하고 여기에 호스트에서는 Virtual Terminal server를 PC에서는 Virtual Terminal client 각각 구동시키고 호스트 컴퓨터에서는 Virtual Terminal server의 Virtual Terminal driver를 mapping시키고, Virtual Terminal client인 PC에서는 terminal emulation process에 Virtual Terminal client를 mapping 시킴으로서 Virtual Terminal service가 제공될 수 있다.

Virtual Terminal server와 client 사이에는 TCP에서 제공되는 8-bit transparent한 communication channel이 형성되고 이 communication channel을 server와 client가 각각 자기의 application process에서 사용함으로서 Virtual Terminal service가 제공된다.

여기에서 전송되는 transparent communication channel에는 명령어를 위한 데이터 및 상호 교환하고자 하는 응용 데이터가 동시에 교환되므로 대부분의 terminal emulation program에서는 이러한 escape code 및 control code를 어떻게 처리하느냐가 중요한 관건이 된다.

여기에서는 2-byte 완성형 한글 code인 KSC5601을 host code로 정하고 terminal이 형태는 VT100을 대상으로 하였다. Control code는 ASCII code 체계에서 0 - 255 사이의 값중 0 - 31 사이의 값에 해당되며 escape code는 27로 시작되는string으로 되어 있다. Control code, escape code 이외의 data code 가운데에서 127보다 큰 값을 갖는 것은 일단 한글 code로 인식하여 처리한다. Control code, escape code는 다음과 같다.

7-bit control codes

MNEMONIC	HEX	FUNCTION
NUL	00	무시한다.
ENQ	05	Answer Back Message를 내보낸다.
BEL	07	Bell을 울린다.
HT	09	커서를 탭다음의 위치로 옮긴다.
LF	0A	커서가 1줄 밑으로 내려간다.
VT	0B	LF와 같다.
FF	0C	LF와 같다
CR	0D	cursor를 line의 맨처음 column으로 이동
SO	0E	G1 character set을 호출
SI	0F	G0 character set을 호출
DC1	11	X-ON
DC3	13	X-OFF
CAN	18	ESC도중 이 code가 들어오면 무시
SUB	1A	Substitute
ESC	1B	ESC명령어의 시작
DEL	17	수신시 무시한다.

8-bit control codes

MNEMONIC	HEX	FUNCTION
IND	84	cursor down
NEL	85	cursor를 다음 line의 1번째 column으로 이동
HTS	88	Tab Set
RI	8D	cursor up
DCS	90	Device Control String
CSI	9B	ESC명령어의 시작
ST	9C	Device string의 마지막 delimiter

예)

CSI = ESC1

SS3 = ESC0

ESCAPE CODES(ANSI MODE)

CODE	FUNCTION
Ctrl E	Answer back
Ctrl G	Sound Bell
Ctrl H	Back space
Ctrl I	cursor를 다음 tab의 위치로 이동
Ctrl J	cursor를 다음 line의 같은 column으로 이동
Ctrl K	Ctrl J와 같은 기능
Ctrl L	Ctrl J와 같은 기능
Ctrl M	cursor가 현재 line의 1 column으로 이동
Ctrl N	select G1 character set
Ctrl O	select G0 character set
Ctrl Q	X-ON
Ctrl S	X-OFF
Ctrl X	ESC Sequence를 받는 중에 이것을 받으면 ESC sequence를 무시
Ctrl Z	substitute
ESC SP Pc	Define Cursor Control
ESC # Pc	Define line attribute
ESC \$ * Pc	Define KSC2C
ESC (Pc	Define G0 set
ESC) PC	Define G1 character set
ESC * Pc	Define 한글 mode
ESC 7	Save cursor
ESC 8	ESC 7의 반대
ESC =	Enter Application Key pad
ESC >	Enter Numeric Key pad

ESC D	INdex, cursor 1 line 아래로 이동한다.
ESC E	Newline
ESC G	Cursor가 Right Natgin으로 이동한다.
ESC H	Cursor가 위치한 column을 tap position으로 이동
ESC I	back tap
ESC M	reverse linefeed
ESC F p Gn	Graphic 1 Character
ESC F f Gn Dn	수평선 display
ESC F g Gn Dn	수직선 display
ESC Fb Gn Dn	Bow display
ESC F V	Lower Case
ESC F W	Upper Case
ESC F K	한글/영문mode
ESC F I	영문전용 mode
ESC Z	Terminal ID 전송
ESC c	Hard Reset
ESC F r	Recovery how display
ESC F V	Save set up parameters
ESC F R	Recall set up parameters
ESC F S Pt	KSC2C mode에서 행을 elect
ESC [Pn @	Insert Character(Pn만 수행)
ESC [Pn A	Cursor up
ESC [Pn B	Cursor Down
ESC [Pn C	Cursor forward
ESC [Pn D	Cursor Backward
ESC [Pn ; Pm G	Box display

ESC [Pn ; Pc H(f)	Set cursor position
ESC [Pn J	erase in display
ESC [? Pn J	Erase in display to erasable character
ESC [Pn K	Erase in line
ESC [? Pn K	Erase in line to erasable character
ESC [Pn L	line insert를 n번 수행
ESC [Pn M	line delete를 Pn 번 수행
ESC [Pn N	set line per page or back/next page
ESC [Pm ; Pe 0	수평선 또는 수직선 그리기
ESC [Pn P	delete character를 Pn번 수행
ESC [Pn ; Pa S	25 the line control
ESC [Pd ; Ps ; Pp T	set main port parameter
ESC [In U	display downloadable character
	In : image number
ESC [P1 ; Pe ;P1;Pc W	Window forward scroll
ESC [P1;Pc;P1;Pc'V	Window reverse control
ESC [Pn X	erase character를 Pn번 행한다.
ESC [Pn c	request device attribute report
ESC [Pn g	tab clear
ESC [Pn h	set terminal mode 1
	Pn = 2 : keyboard lock
	4 : insert mode on
	12 : local echo off
	20 : New line mode on
	30 : 2:1 display(KOR:ENG)
	31 : character mode

- 32 : format mode on
- 33 : transmit block terminator
- 34 : transmit field separator
- 35 : user defined ESC code 수행
- 36 : local display 수행
- 37 : LFD control
- 38 : key input mode enable(ctrl function)
- 39 : clear left or right

ESC [Pn I reset terminal mode

ESC [? Pn h set terminal mode 2

Pn = 0 : Key click on

1 : application Key mode

2 : ANSI mode

3 : 132 column

4 : smooth scroll on

5 : reverse background

6 : origin mode

7 : autowrap on

8 : auto repeat on

18 : print 후 FF(ctrl-l)을 전송한다.

19 : print full screen

25 : cursor on

85 : select 한글 mode(KSC2)

ESC [? Pn l reset terminal mode 2

ESC [Ps ; Ps . . m set character attribute

Ps = 0 : all reset

1 : hold
 2 : dim
 3 : blank
 4 : underline
 5 : blind
 6 : non erasable
 7 : reverse
 21 : blank off
 26 : non erasable character off
 27 : reverse off

ESC [Pn n device status report
 ESC ? 25 n request user defined Keys
 ESC ? 28 n request keyboard type
 ESC [Ps "q Ps = 0 : set all erasable character
 1 : set non erasable character
 2 : set erasable character mode
 ESC [Pt ; Pb r set scrolling region
 ESC [Pk t set key input mode
 ESC [Pe V define ESC code
 ESC [Pm;Pn X define 한글 또는 영문 start
 ESC [Pn y define brightness
 ESC [Pn z define cursor type
 ESC Pc^In;i.... download image
 CSI ? 5 i auto print mode set
 CSI ? 4i auto print mode reset
 CSI 5 i controller print mode set

CSI 4 i controller printer mode reset
 CSI ? 1 i cursor가 있는 line이 data를 printer로 보냄
 CSI i or 0 i screen copy

Dummy terminal은 host에서 전달된 명령에 따라 Keyboard와 monitor와 cursor move로 대부분의 일이 처리된다. 여기서 실제로 터미널 emulation 기능이 진행되는 과정은 다음과 같다.

TFTP TIME NAME Service Resolution	TELENET FTP SMTP LPR
UDP	TCP
IP	

Protocol Family

Protocol family에서 telnet상에서 VT100을 구현하기 위하여는 hostname을 결정하는 nm_res_name, host와 connection을 이루기 위한 tn_open, 하나의 character를 얻기 위한 tn_getc, 하나의 character를 넘겨주기 위한 tn_putc, connection을 단절하는 tn_close등이 주요한 기능으로 기능으로 진행된다.

```
main part
{
...
    host = nm_res_name("server", NULL, 0);
    tn = tn_open(host, 0, "vt100")
...
}
```

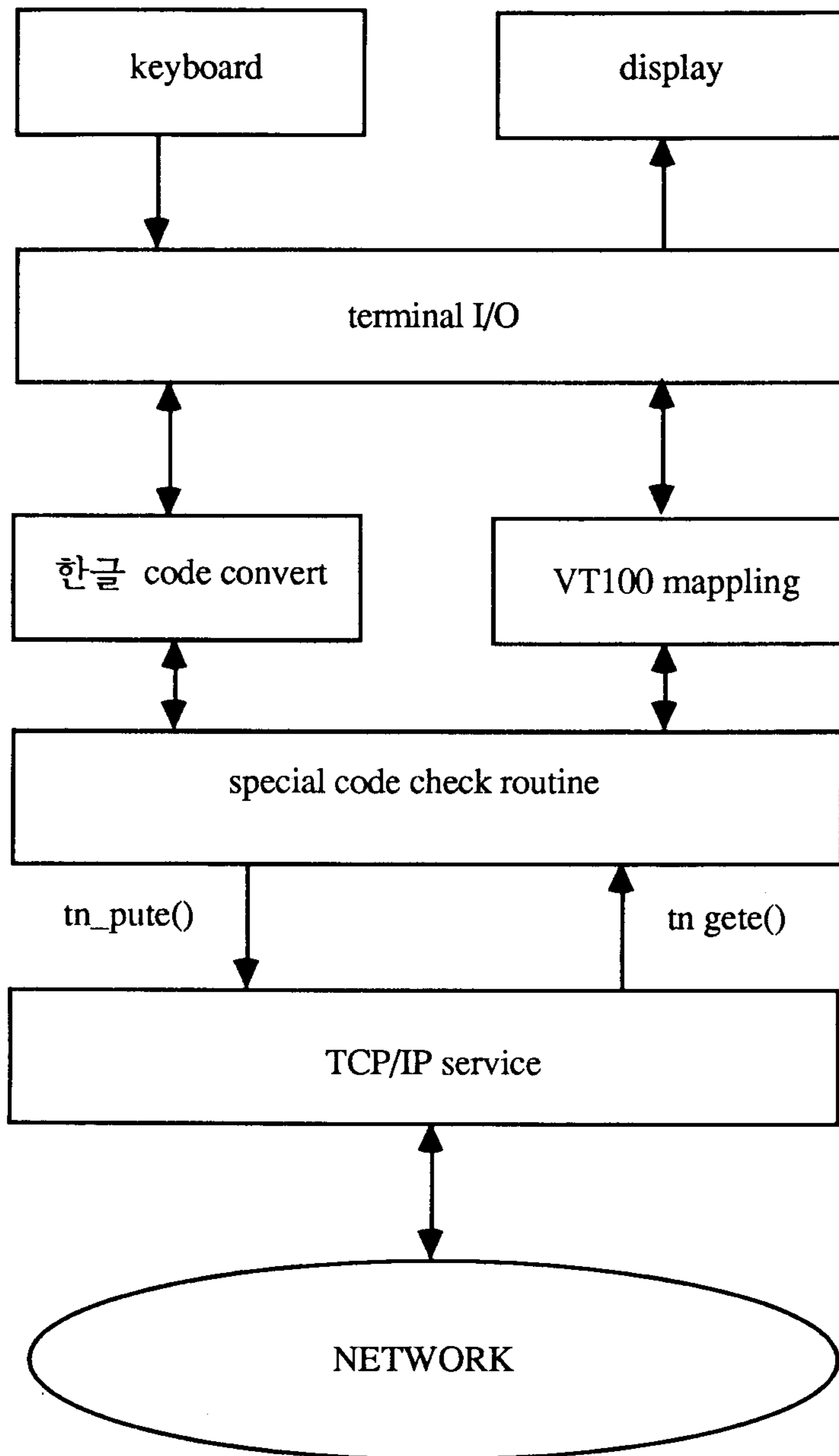
```

while(quit)
{
    if(key()){
        ch = inkey();
        ...
        tn_pute(tn, ch);
        if(ch >= 128) && (logout != 0x44){
            ch = inkey();
            tn_pute(tn, ch);
        }
    }
    ...
}

```

위와 같이 TCP/IP와 터미널 emulation program의 기본 기능인 input/out의 handling을 할수 있는 프로그램을 기본으로 하고 여기에 한글 code가 입력되었을 때 한글 conversion을 수행하기 위한 기능 및 VT100 terminal emulation 기능을 수행하기 위한 기능 모듈이 동작하게 된다. VT100 emulation을 위하여는 앞에서 정의된 기능을 제공하기 위한 기능 모듈로 구성된다.

이러한 기능을 제공하기 위한 기능 모듈은 다음과 같다.



위 그림과 같이 본 작업은 한글 code변환, VT100 emulation, TCP/IP interface의 세가지 부분으로 나뉘어서 진행되었으며, 호스트의 한글 code는 KSC5601 PC쪽의 한글 code는 KSC5601을 포함하는 한글 code 변환 모듈에 의해서 수행된다. 현재 TELNET등의 terminal emulation은 물론 ftp등의 한글 file transfer도 현재 가능하다.

4.4 한글화 Module 설명

TCP/IP INTERFACE

TCP/IP와 통신하는 부분으로 관련 parameter 및 function는 다음과 같다.

4.4.1 관련 structure

```
typedef struct telnet {  
    int tn_tcp;        /* the telnet's TCP connection */  
    int tn_echom;     /* Echo mode */  
                    /* LOCAL - local echo */  
                    /* REMODE - remote echo */  
    int tn_echongo;  /* Echo negotiation request outstanding */  
                    /* NORMALMODE */  
                    /* LECHOREQ - IAC DONT ECHO was sent */  
                    /* RECHORE1 - IAC DO ECHO was sent */  
    char tn_bin_mode[2];  
                    /* binang mode states : [0] remote, [1] local */  
                    /* OFF (0) */  
                    /* ON (1) */
```

```

        /* reqtosend */
        /* reqtostop */
int tn_wmode; /* write mode */
int tn_chars; /* # of valid characters in buffer */
int tn_next; /* next character in buffer */
unsigned char tn_buf[512]; /* buffer */
int tn_wchars; /* Numver of characters in vobuf */
int tn_wnext; /* Next char to write in wbuf */
unsigned char tn_wbuf[28]; /* Output buffer */
char *tn_cname; /* canonical name of rem host */
int tn_term_index; /* indicates tern type negotiated */
char *(*send_type) (); /* func ptr --> rets type to send */

```

} telnet

- ◆ REMOTE ECHO MODE : 키보드에서 입력된 데이터를 PC상에 디스플레이하지 않고 호스트로 전송하는 모드
- ◆ LOCAL ECHO MODE : 키보드에서 입력된 데이터를 PC상의 화면에 직접 디스플레이 한다.
- ◆ NORMALMODE : REMOTE, LOCAL ECHO MODE를 요구하지 않은 상태
- ◆ LECHOREQ : Local echo mode을 요청하는 상태
- ◆ RECHOREQ : remote echo mode을 요청하는 상태
- ◆ binary mode status : binary data 전송 가능 상태

◆ 사용된 code Define

```
#define BLOCK 1
#define NOBLCK 2
#define URGENTM 1
#define EXERYC 1
#define LOCAL 1
#define REMOTE 2
#define NORMALMODE 1
#define LECHOREQ 2
#define RECHOREQ 3
#define OFF 0
#define ON 1
#define reqtosend 2
#define reqtostop 3
```

4.4.2 관련된 function

1) connect Form

(1) Application Proggam에서 Connection을 할때에 host name이 결정된 다음에 호출하면 된다.

(2) call format

```
telnet *tn_open(host, port, send_term_func)
in_name host; /*host to connect to */
unsigned port; /* TCP socket to connect to */
```

```
Char>(*send_term_func);    /* terminal type string */
```

(3) Return Value

Telnet descriptor을 만들지 못하면 NULL을 return하고 Telnet descriptor가 만들어지면 Telnet connection descriptor에 대한 pointer가 return된다.

2) Disconnect Form

(1) Application Program에서 telnet connection을 단절하고자 할때 이 function을 호출한다.

(2) Call Form

```
void tn_close(tn)
```

```
telnet *tn;          /* connection to close */
```

(3) Return Value

연결 절단이 이루어지면 TRUE, 연결 절단이 이루어지지 않으면 FALSE을 되돌려준다.

3) Get Form

(1) Telnet connection tn상에서 유용한 data가 있으면 Data를 되돌려 주고 유용한 Data가 없으면 없다는 것을 알려준다.

(2) Call Form

```
int tn_getc(tn)
```

```
telnet *tn;          /* the Telnet connection */
```


(3) Return Value

유용한 data가 있으면 data를 되돌려 주고 유용한 data가 없으면 -1를 되돌려 준다.

4) Put Form

(1) Telnet 연결 tn상에 data을 보낸다.

(2) Call Form

```
int tn_putc(tn,c)
```

```
telnet *tn;
```

```
char c;
```

(3) Return Value

전송 성공이면 TRUE, 전송실패이면 FALSE를 되돌려 준다.

5) Local mode set

(1) Telnet 연결 tn을 local echo mode로 설정한다.

(2) Call Form

```
int tn_local(tn)
```

```
telnet *tn;
```

(3) Return Value

Local echo mode로 request를 할수 있으면 TRUE, 그렇지 않으면 FALSE를 되돌려 준다.

6) Remote mode set

(1) Telnet연결을 remote echo mode로 설정함으로 호스트가 data를 echo하도록 해준다.

(2) Call Form

```
int tn_remote(tn)
```

```
telnet *tn; /* the Telnet connection */
```

(3) Return Value

호스트에게 remote echo을 요구할 수 있으면 TRUE, 요구할 수 없으면 FALSE을 되돌려 준다.

7) Binary mode set

(1) Telnet 연결 tn상에 WILL, DO을 보냄으로 Telnet binary mode에 대한 협상을 한다.

(2) Call Form

```
int tn_rqst_bin(tn)
```

```
telnet *tn; /* the telnet connection */
```

(3) Return Value

WILL/DO을 보내는데 성공하면 TRUE 실패하면 FALSE을 보낸다.

8) Binary mode 확인

(1) Telnet binary option상태를 시험한다.

(2) Call Form

```
int tn_is_bin(tn)
```

```
telnet    *tn;          /* the telnet connection */
```

(3) Return Value

양방향 전송모드가 binary mode이면 TRUE을 되돌려주고 binary mode가 아니면 FALSE을 되돌려 준다.

9) Break

(1) Telnet control sequence IAC INTP(interrupt process)을 보낸다.

(2) Call Form

```
int tn_break(tn)
```

```
telnet *tn; /*the Telnet connection */
```

(3) Return Value

Interrupt process가 이루어지면 TRUE, 이루어지지 않으면 FALSE을 되돌려준다.

10) Abort

(1) Telnet 연결 tn을 abort한다. 연결상의 모든 상태가 없어지고 descriptor가 없어진다.

(2) Call Form

```
Void tn_abort(tn)
```

```
telnet    *tn;          /* connection to abort */
```

(3) Return Value

없음

11) Server 확인

(1) 연결상 Telnet server에게 "Are you there" query를 보내고 응답이 있는지를 확인한다.

(2) Call Form

```
int tn_ayt(tn)
```

```
telnet *tn;
```

(3) Return Value

Query를 보낼 수 있으면 TRUE, 보낼 수 없으면 FALSE를 보낸다.

화면 Control

TCP/IP INTERFACE의 function을 사용하여 호스트로부터 들어오는 Control Character 및 Escape Sequence에 대한 처리로 사용되는 function은 다음과 같다.

4.4.3.1 관련된 function

1) Cursor Control Sequences

1.1) Cursor position

(1) 커서를 이동시킨다.

(2) Call Form

Void Cursor Position (S)

char *s;

(3) Return Value

없음

1.2) Cursor up

(1) 커서를 좌우 위치 변화 없이 한 타인 또는 (지정된 타이)위로 이동시키며 커서의 위치 top line이면 무시한다.

2) Call Form

Void Cursorup(s)

char *s;

(3) Return Value

없음

1.3) Cursor Down

(1) 커서를 좌우 위치 변동없이 한라인 또는(지정된라인) 아래로 이동시킨다. 이때에 마지막 라인이면 무시한다.

(2) Call Form

Void CursorDown(s)

Char *s;

(3) Return Value

없음

1.4) Cursor Forward

1) 커서를 한 컬럼(지정된 컬럼) 좌로 이동 시킨다. 이때 첫번째 컬럼이면 무시한다.

(2) Call Form

Void CursorForward(s)

Char *S;

(3) Return Value

없음

1.5) Cursor Backward

(1) 상하위치 변동없이 커서를 한 컬럼(지정된 컬럼) 우로 이동시킨다. 이때 마지막 컬럼이면 무시한다.

(2) Call Form

Void CursorBack(s)

Char *s;

(3) Return Value

없음

1.6) Horizontal and Vertical Postion

(1) 지정된 위치로 커서를 이동한다.
위치 지정이 없으면 Home으로 간다.

(2) Call form

Void Hupostion(s)

Char *S;

(3) Return Value

없음

1.7) Save Cursor Position

(1) 현재의 cursor 위치를 저장한다.

(2) Call Form

Void savecursor(xposition, gposition)

int xposition;

int yposition;

(3) Return Value

없음

1.8) Restore Cursor Position

(1) 저장된 Cursor 위치로 커서로 이동시킨다

(2) Call Form

Void RestoreCursor(s)

Char *s;

(3) Return Value

없음

2) Erasing

화면에서 text를 지우는데 사용

2.1) Erase in Display

(1) 화면전체를 지우고 커서는 home position으로 이동한다

(2) Call Form

Void EraseDisplay(s)

Char *s;

(3) Return Value

없음

2.2) Erase in Line

(1) 현재 커서의 위치부터 라인의 끝까지 지운다.

(2) Call Form

Void Eraseinlime(s)

Char *s;

(3) Return Value

없음

2.3) scroll up

(1) 지정된 scroll 영역을 지정 라인 위로 이동시킨다.

(2) Call Form

Void scrollup(lines, x1, y1, x2, y2)

int line ; /* 이동시킬 라인수 */

int x1 ; /* 시작 좌우 위치 */

```
int y2; /* 시작 상하 위치 */
int y1; /* 마지막 좌우 위치 */
int y2; /* 마지막 상하 위치 */
```

(3) Return Value

없음

2.4) Scroll Down

(1) 지정된 scroll 영역을 지정 라인 아래로 이동시킨다.

(2) Call Form

```
Void scrolldown(lines, x1, y1, x2, y2)
```

```
int lines;
```

```
int x1, y1, x2, y2;
```

(3) Return Value

없음

한글 Code Conversion

서로 다른 한글을 동일한 한글(호스트한글)로 변환하고자 할때 사용된다

4.4.3.2 관련된 functions

1) Code Information

나의 한글 코드 및 호스트 한글 code에 대한 정보를 얻는다.

(1) 한글코드에 대한 정보를 화일에서 읽어 낸다.

(2) Call Form

```
int      GetHan_Inf(mycode, hostcode)
int      mycode;
int      hostcode'
```

(3) Return Value

Information을 얻을 수 있을때는 TRUE, 얻을 수 없을 때 FALSE을 되돌려준다.

2) make hangule code

한글 Code Table를 만든다.

(1) 한글 code table를 만들어 변환가능 상태로 만든다.

(2) Call Form

```
make hancode(**S, **T)
Char **S; /* Source Code Table */
Char **T; /* Target Code Table */
```

(3) Return Value

없음

3) Code 변환

(1) 나의 한글코드를 호스트 한글코드로 호스트 한글코드를 나의 한글코드로 변화하는 부분

(2) Call Form

```
Void Cenversion(stype, sch, type, tch)
```



```
int stype ;      /* 한글코드의 종류 */
unsigned int sch; /* first byte for hangule code */
int type;       /* 한글 코드의 종류 */
unsigned int tch; /* second byte for hangule code */
```

(3) Return Value

해당 한글 코드가 없을 때는 blank가 seh, tch에 들어간다

5. 결론

OSI Virtual Terminal protocol의 한글화를 목표로 하고 있는 본과제를 수행하기 위해서 현재 국제적인 Virtual Terminal protocol 및 국제 표준의 Virtual Terminal 기능을 조사 분석하였다. 조사된 바에 의하면 Virtual Terminal 기능은 대상 터미널이 갖는 기능과 네트워크 아키텍처등에 관련되어 Virtual Terminal protocol 및 기능이 정의되고 있다.

OSI Virtual Terminal의 한글화 작업을 위한 일환으로서 본 연구팀에서는 현재 널리 사용중에 있는 PC/TCP를 한글의 이용이 가능한 형태로 수정작업을 하였다. 본 작업의 결과는 향후 작업의 rapid prototyping 형태로 본과제에서 응용되며, 실제적으로 적용가능한 protocol/network인 telnet, TCP/IP를 대상으로 하였기 때문에 실제 응용성을 고려한 작업으로 사려된다.

현재 PC/TCP의 한글화 작업이 완료되어 사용중에 있으며, Virtual Terminal 뿐만 아니라 file transfer등의 utility에도 역시 한글화 작업이 완료되어 있다. 본 작업의 결과로 현재 많이 사용되고 있는 TCP/IP network 환경에서 한글을 포함하는 PC Host Application이 가능해 졌으며 이러한 작업의 결과로 PC를 네트워크에 연결하여 사무자동화등의 Application이 기존의 네트워크 관련 소프트웨어의 수정없이 가능하게 되었다.