

Packet Switching에 의한 공중 Computer 통신망 개발 연구  
- 제 4 부 : KORNET NNP의 PAD Protocol 및 Network  
Management Software의 구현

(Development of a Packet-Switched Public Computer Network  
- PART 4: PAD Protocol and Network Management  
Software of the KORNET NNP)

金尚龍\*, 琴 聖\*, 金濟佑\*, 吳敬愛\*, 殷鍾官\*, 李鐘樂\*, 徐仁洙\*\*, 趙東浩\*, 崔竣均\*

(Sang Ryong Kim, Seong Geum, Je Woo Kim, Chong Kwan Un, Kyong Ae Oh, Jong Rak Lee,  
In Soo Seo, Dong Ho Cho and Jun Kyun Choi)

要 約

이 논문은 packet 교환 방식에 의한 공중 통신망 개발에 관한 4 편의 논문중 마지막인 제 4 부 논문이다. 본 제 4 부 논문에서는 KORNET의 NNP에 내장되어 있는 asynchronous channel service를 위한 packet assembly/disassembly (PAD) protocol과 NNP의 운영을 위한 통신망 운영 software의 구현과정에 대하여 기술한다.

비동기형 service를 위한 LAPB system에는 PAD protocol과 X.25와의 접속을 위한 packet mode DTE/DCE 기능 및 start/stop mode terminal과의 접속을 위한 asynchronous receiver/transmitter (ART) 기능들이 구현되어 있는데 여기서는 이들에 대한 protocol 설계와 구현 방법에 대하여 설명한다. 통신망 운영 software를 구현한 MCPM system에는 각 user channel에 대한 service를 관장하는 virtual circuit management (VCM)와 NMC와 연결되어 network 및 node의 운영을 담당하는 high level protocol이 있다. 본 논문에서는 이들 각 module 들을 서비스 특성에 알맞게 구현하는 과정을 기술한다.

Abstract

This is the last part of the four-part paper describing the development of a packet-switched computer communication network named the KORNET. In this paper we describe the design and implementation of the packet assembler/dissassembler (PAD) protocol for the asynchronous channel service, and of the network management softwares. The line processing module-B (LPMB) system supporting the asynchronous line includes a PAD protocol, a packet mode DTE/DCE protocol converting to the X.25 protocol, and the asynchronous receiver/transmitter (ART) software. The network management software is operated in master central processing module (MCPM) which includes virtual circuit management (VCM) managing the user channel, the routing management and the high level protocol for communication between the network management center (NMC) and the network node processor (NNP). In this paper, the design, implementation and operation of the softwares for the above service functions will be described in detail.

\*正會員, 韓國科學技術院 通信工學研究室  
(Communications Research Laboratory, KAIST)

\*\*正會員, 韓國데이터통신(株) 派遣  
(Data Communications Corp. of Korea)  
接受日字: 1985年 6月 10日

I. 序 論

본 논문은 한국 최초로 개발된 packet 교환 방식에 의한 computer 통신망 개발에 관한 제 4 부 논문으로

서, 제 1 부의 KORNET의 개요와 NMC개발<sup>[1]</sup>, 제2부의 KORNET의 설계 및 NNP개발<sup>[2]</sup>, 제 3 부의 NNP의 X. 25 software 설계 및 구현에 관한 논문<sup>[3]</sup>에 이어 NNP의 PAD 및 network management software 개발에 관하여 중점적으로 다루고 있다.

본 논문에서는 KORNET의 NNP에서 asynchronous channel을 서비스하는 line processing module-B(LPMB) system과 NNP 운영을 위한 master central processing module(MCPM)에 대하여 설명한다.

먼저 LPMB system은 start/stop mode terminal을 network에 접속하기 위해 PAD protocol을 구현하고 있으며 X.25 protocol로 protocol conversion을 하기 위해 packet DTE/DCE(PTC) protocol을 구현하고 있다. 이러한 protocol은 X.25 protocol과 유사한 state machine design 기법을 사용하고 있으며 여러가지의 terminal 특성을 support하기 위해 parameter table을 이용하고 있다. 또한 asynchronous line 과의 접속을 위한 asynchronous receiver/transmitter(ART)를 구현하여 real time service에 맞게 운영하고 있다.

NNP의 운영 및 감독을 위하여 MCPM에는 channel 경로의 선택과 resource의 할당 등을 담당하는 virtual circuit management와 routing management가 있으며, network management center(NMC)와 연결되어 network 및 node를 운영하기 위한 high level protocol로서 primary/secondary root, session layer, primary/secondary(P/S) function 등이 있다.

본 논문에서는 LAPB와 MCPM system에서의 각 layer 별로 protocol design 및 service 방법을 설명하고 각 system의 software structure와 service module의 구동 방법을 기술한다.

제 I 장의 서론에 이어 제 II 장에서는 PAD service protocol에 대하여 기술하고, 제 III 장에서는 통신망 운영 software의 구현에 대하여 설명한다. 마지막으로 제 IV 장에서 결론을 맺는다.

## II. PAD Service Protocol

본 장에서는 KORNET에서 asynchronous start/stop(S/S) mode DTE를 연결하기 위해 구현한 service protocol과 software implementation에 관하여 설명한다. 이의 주된 기능은 packet assembly/disassembly(PAD) 기능인데 KORNET에서는 asynchronous channel을 서비스하는 hardware board인 line processing module-B(LPMB)에 구현된다.

LPMB system은 그림 1에서 나타낸 바와 같이 4개의 task로 구동되는데 physical layer와 접속되어 데

이타 전송을 담당하는 ART driver, PAD의 중앙 처리 protocol과의 접속을 위한 packet DTE/DCE(P TC) 및 다른 hardware board와 데이터교환을 위한 interstation communication protocol(ISCP)로 구성된다.

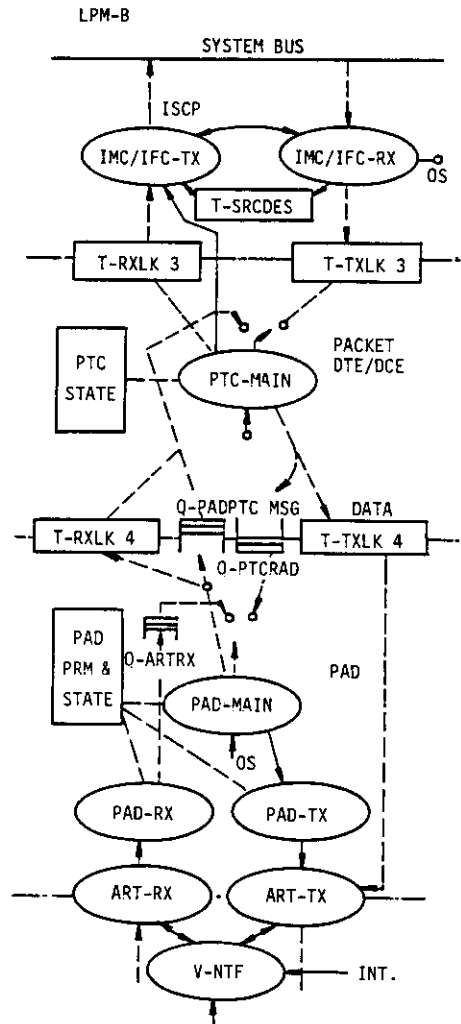


그림 1. LPMB의 전체 software 구조

Fig. 1. Overall software structure of LPMB.

### 1. PAD Protocol 및 Software구조

PAD는 start/stop(S/S) DTE로 부터 들어오는 character들을 packet으로 만들어 network으로 전송하거나 network을 통해 전송되어 온 data packet을 character로 바꾸어 S/S DTE로 보내는 일을 하는데 CCITT 권고사항 X.3에는 PAD facility에 관해, X.28에는 S/S DTE가 PAD를 이용하는 절차에 관해, X.29에는 PAD가 상대 PAD 또는 PDTE와 control in-

formation이나 user data를 주고 받는 절차에 관해 규정하고 있다.<sup>4)</sup>

이러한 PAD의 기본적인 기능을 요약하면 다음과 같다.

- Character를 packet으로 assembly하는 기능.
- Packet의 user data field를 disassembly하는 기능.
- Virtual call의 개설과 단절, reset 및 interrupt packet의 전송 기능.
- PAD service signal의 전송 기능.
- Packet이 full이거나 idle time의 종료 등과 같은 packet forwarding 상황이 발생하면 모아놓은 character들을 packet으로 만들어 전송시키는 기능.
- PAD command signal을 editing하는 기능.
- S/S DTE로 부터 전송된 break signal을 handling하는 기능.

KORNET의 PAD에서는 S/S DTE의 특성에 따라 18가지의 PAD parameter중에서 적절한 값들을 선택하여 이용할 수 있다. 이처럼 PAD는 각종 S/S DTE와 연결이 가능하게 되어 있는데 구현된 PAD parameter를 살펴보면 표 1과 같다.

표 1. 구현된 PAD parameter  
Table 1. PAD parameters implemented.

Parameter reference number	Parameter description	Selectable possible values
1	PAD recall using a character	0, 1
2	Echo	0, 1
3	Selection of data forwarding signal	0, 2
4	Selection of idle timer delay	0, 20, 255
5	Ancillary device control	0, 1
6	Control of PAD service signals	0, 1, 5
7	Selection of operation of PAD on receipt of break signal from the start-stop mode DTE	0, 1, 2
8	Discard output	0, 1
9	Padding after carriage return(CR)	0, 1 - 7
10	Line folding	0, 1 - 255
11	Binary speed of start-stop mode DTE	0, 2, 3, 4, 12, 13, 14
12	Flow control of the PAD by the start-stop mode DTE	0, 1
13	Linefeed insertion after carriage return	0, 1, 4, 5, 6, 7
14	Linefeed padding	0, 1 - 7
15	Editing	0, 1
16	Character delete	127
17	Line delete	24
18	Line display	1

PAD와 S/S DTE 사이의 flow control은 X, 25의 link layer나 packet layer에서 채택하고 있는 sliding window를 이용한 방법은 불가능하고 단지 PAD와 S/S DTE 사이의 input line buffer(ILB)의 status에 따라 flow control을 할 수 있다. 예를 들어 PAD parameter 5가 1인 경우 PAD가 S/S DTE로 부터 더 이상 character들을 받을 수 없는 경우는 X-OFF(DC3)를 S/S DTE쪽으로 전송하여 data 전송 중지를 요청하고 다시 input line buffer가 사용 가능하게 되면 X-ON(DC1)을 전송하여 계속적인 data 전송을 허락한다. 반대로 PAD parameter 12가 1인 경우에는 S/S DTE가 PAD로 부터 전송되어 오는 data들을 더 이상 받을 수 없게 되면 X-OFF를 전송하며, 이때 PAD는 data 전송을 즉시 멈추게 된다. 다시 PAD가 X-ON 신호를 받게 되면 계속 data를 전송하게 된다. PAD와 S/S DTE 사이의 이와 같은 flow control을 통해 두 processor 간의 speed matching을 이루게 되어 input line buffer의 overflow에 의한 data loss를 방지 할 수 있다.

이러한 PAD의 software 구조는 PAD의 main protocol인 PAD-MAIN, character의 송신과 수신을 담당하는 PAD-RX와 PAD-TX, timer인 PAD-Timer 및 physical layer와 접속되는 ART로 구성되어 그림 2는 이를 나타낸다.

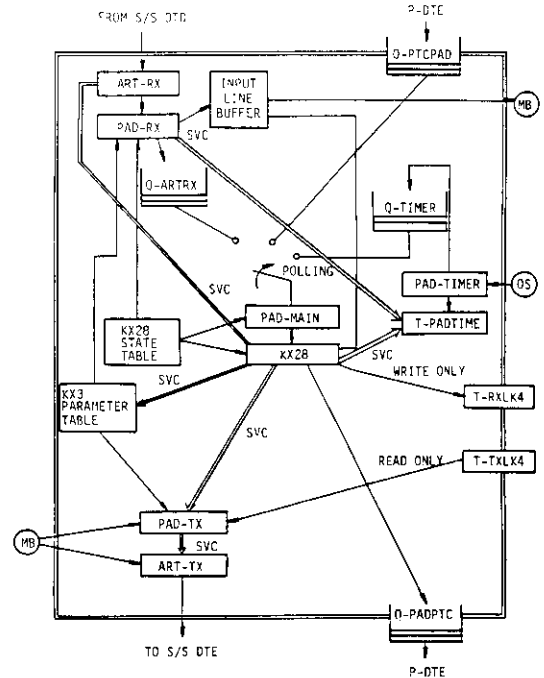


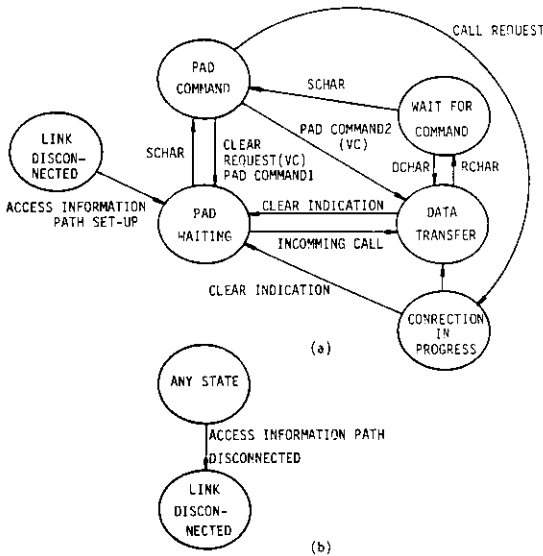
그림 2. PAD의 software 구조  
Fig. 2. Software structure of PAD.

먼저 S/S DTE로 부터 수신되는 character를 받아들이는 PAD-RX는 ART-RX로 부터 입력되는 character를 ILB에 넣고 이 사실을 Q-ARTRX를 통해 PAD-MAIN에 알려준다.

PAD-MAIN은 중심 protocol로서 input analysis, command analysis 및 KX28 main protocol 처리부로 구성된다. 이 routine에서는 S/S DTE와 PDTE 그리고 software timer에서 생긴 입력 사건들이 들어 있는 queue를 polling하여 analysis 한 후 KX28 main part를 구동시킨다. 이 KX28은 CCITT 권고사항 X. 28의 모든 protocol과 X.3의 일부를 수행하는 main task로서 KX28 parameter table과 KX3 parameter table을 사용하며 다음의 6 개의 state로 운영된다.

- LINK DISCONNECTED state
- PAD WAITING state
- PAD COMMAND state.
- WAIT FOR COMMAND state.
- CONNECTION IN PROGRESS state.
- DATA TRANSFER state.

그림 3에 KORNET에서 구현된 PAD의 state transition diagram이 나타나 있다.



(주)SCHAR: First character of string from S/S DTE  
 RCHAR: PAD recall characters from S/S DTE  
 DCHAR: Data-mode-return characters  
 PAD COMMAND1: PROF, SET, SET?, PAR?, STAT  
 PAD COMMAND2: PROF, SET, SET?, PAR?, RESET, INT  
 VC: Virtual call이 set up 되었음을 나타냄

그림 3. PAD의 state 변환도형  
 (a) Normal상태 (b) Disconnected상태

Fig. 3. State transition diagram of PAD.

(a) Normal phase (b) Disconnected phase

PAD-TX port는 KX28 main module에 의하여 구동되며 hardware controller를 구동시켜 PAD service signal과 network로 부터 들어오는 data들을 character 단위로 terminal에 내보내는 일을 한다.

PAD protocol중의 일부인 PADTIMER는 software timer로서 KX28로 부터 내려온 timer의 기동 및 정지 명령을 수행하며 time-out 발생시에는 이를 Q-timer queue를 사용하여 알린다. 이러한 동작을 위해 timer sequence table과 PAD timer table이 필요하다.

마지막으로 terminal에서 들어오는 character를 받아들이고, terminal로 내보내기 위해 ART가 있는데 전송 방향에 따라 ART-RX, ART-TX로 나뉘어진다.

먼저 ART-RX를 설명하면 다음과 같다. SIO chip에 입력된 serial data는 RX shift register에서 serial-to-parallel 변환이 된다. 이 과정에서 start bit, stop bit, parity bit등이 떨어지고 data만이 RX-buffer에 load된 후 RX character available interrupt에 의해 ART-RX가 이를 가져온다. 이 때 parity error, framing error, overrun error등의 error check가 수행되며 그림 4는 ART-RX와 PAD-RX간에 data가 입력되는 과정을 나타낸다.

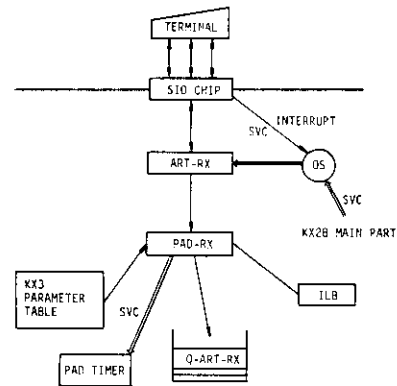


그림 4. Hardware input (ART-RX) 과 PAD input part (PAD-RX) 간의 접속 부분

Fig. 4. Interface between the hardware input part (ART-RX) and the PAD input part (PAD-RX).

한편 ART-TX는 SIO의 TX-buffer가 비게 되면 TX interrupt를 걸어 data의 전송을 요구한다. TX-buffer가 채워지면 이 data는 20-bit shift register로 들어가서 parallel-to-serial 변환을 거치고 data의 앞뒤에 start bit, parity bit 및 stop bit가 붙여진다. 이와 같이 data format이 형성되면 physical line을 통해 전송되며 그림 5는 PAD-TX로 부터 ART-TX를

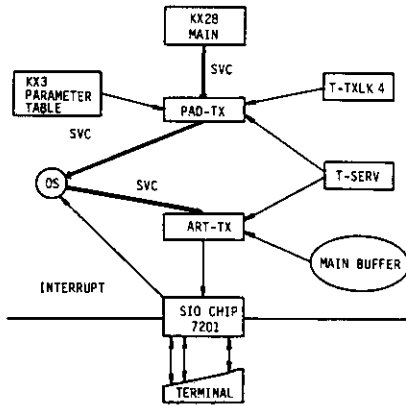


그림 5. Hardware output part (ART-TX)와 PAD output part (PAD-TX) 간의 접속부분  
 Fig. 5. Interface between the hardware output part (ART-TX) and the PAD output part (PAD-TX).

통해서 data가 전송되는 과정을 나타낸다.

이밖에 KORNET의 PAD command signal format과 time-out값은 CCITT 권고사항을 따라 구현되었다.

2. PTC의 Protocol 및 Software 구조

PTC는 PAD에 연결된 character mode data terminal equipment(CDTE)가 다른 packet mode data terminal equipment(PDTE)와 동등한 입장에서 통신을 할 수 있게 해주는 역할을 한다. 따라서 CDTE의 입장에서 보았을 때는 PTC는 DCE와 같은 역할을 해야하고, remote DTE의 입장에서 보았을 때는 PAD에 연결된 CDTE가 PDTE 처럼 보아야 하므로 DTE의 역할도 겸하고 있다. 이처럼 PTC는 NNP 내에서 PAD와 packet layer 사이의 type conversion 기능을 하고 있다.

PTC의 software 구조는 그림 6에서와 같이 크게 나누어 입력을 polling하고 analysis하는 part와 state machine 기법으로 운영되는 main part, 그리고 처리된 output을 전송하는 send part로 구성되어 있으며, 독립적으로 run되는 timer part가 있다.

먼저 polling된 입력을 분석하는 ANALPTC는 입력된 packet의 validity를 check한 뒤에 PTCMAIN으로 넘긴다. 입력의 종류로는 CDTE로 부터 오는 PAD입력과 다른 board의 packet layer로 부터의 입력 그리고 master central processing module(MCPM)으로 부터의 입력이 있다.

PTCMAIN은 PTC의 main protocol을 수행하는 부분으로 입력된 packet 별로 현재의 state를 T-PTCSTA의 state table에서 읽어와서 해당 state service

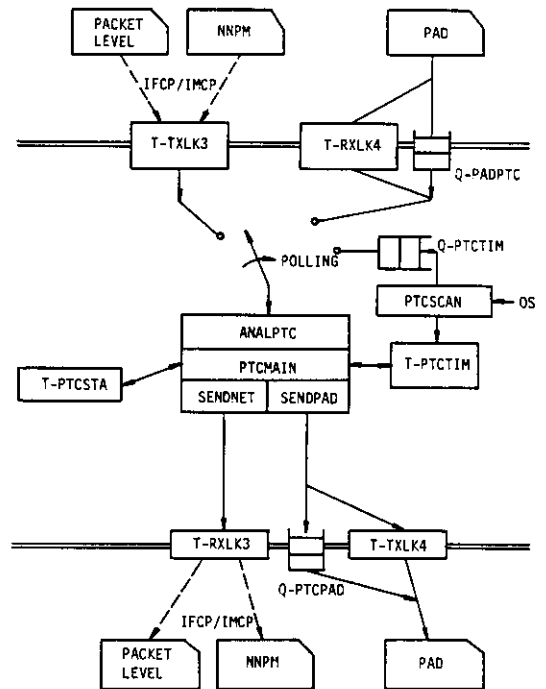


그림 6. Packet DTE/DCE (PTC)의 software 구조  
 Fig. 6. Software structure of packet DTE/DCE.

routine을 수행한다. 이들 state에는 READY state, DTE WAITING state, DCE WAITING state, CALL COLLISION state, DATA TRANSFER state의 5개의 state가 있는데 그림 7은 PTC의 state transition diagram을 나타낸다.

Timer의 start와 stop은 해당 timer의 run bit를 table T-PTCTIM에 set하거나 reset함으로써 이루어 지는데, OS에 의해 run되는 PTCSCAN가 주기적으로 검사하다가 timer expiration되면 이에 대한 message code를 Q-PTCTIM을 통해 내보낸다.

이밖에 main part에서 처리된 packet을 network쪽으로 보내기 위해 SENDNET, SENDPAD routine 등이 있다.

III. 통신망 운영 Software

KORNET에서는 NNP의 운영 및 감독을 위한 기능을 MCPM에서 구현하며 X.25 및 PAD 기능을 수행하는 LPMA 및 LPMB system에 대하여 필요한 명령을 내리고 조정을 해준다. 이를 위해 MCPM은 크게 4가지 service module로 구분되는데 각 논리 채널에 대하여 채널 특성 및 연결될 선로를 결정해주는 virtual circuit management(VCM), calling user

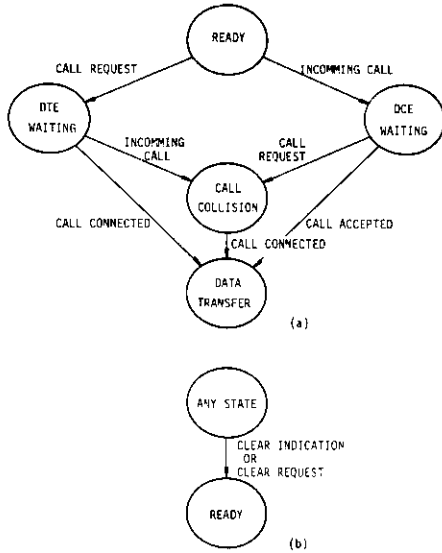


그림 7. PTC의 state 변화 요령  
 (a) 채널 개설 상태 (b) 채널 취소 상태  
 Fig. 7. State transition diagram of PTC.  
 (a) Call setup phase  
 (b) Call clearing phase

와 called user를 연결하여 선로를 결정하는 routing, network management center (NMC)와 연결되어 node 운영 및 보고기능을 수행하기 위한 high level protocol, hardware board 사이의 데이터 교환을 담당하는 interstation communication protocol (ISCP)가 있다. MCPM의 각 service 기능을 구현한 전체 software 구조를 보면 그림 8과 같다.

1. Virtual Circuit Management (VCM)

VCM의 기능은 크게 두 가지의 기능으로 대별할 수 있다. 하나의 기능은 network channel service기능의 일부로서 각 node에 연결된 사용자들에 대한 감독, 통제 및 운영을 담당한다. 또 한가지 기능은 node의 운영 및 이상 상태 발생을 처리하기 위해 primary/secondary (P/S) function으로 부터 내려온 명령을 수행한다.

이러한 기능을 수행하기 위해 VCM은 그림 9에서와 같이 VCM-ANAL과 VCM-MAIN의 2개의 module로 구성된다. 먼저 network에 가입된 사용자가 통화로의 개설 및 취소를 위한 packet을 보내오면 VCM-ANAL은 이들을 분류하여 VCM-MAIN의 해당 service routine을 불러 입력된 packet을 처리하게 한다.

VCM-MAIN은 channel service를 위한 6개의 service function과 node 운영을 위한 3개의 service function으로 분류할 수 있는데 표 2와 같다. 먼저 call

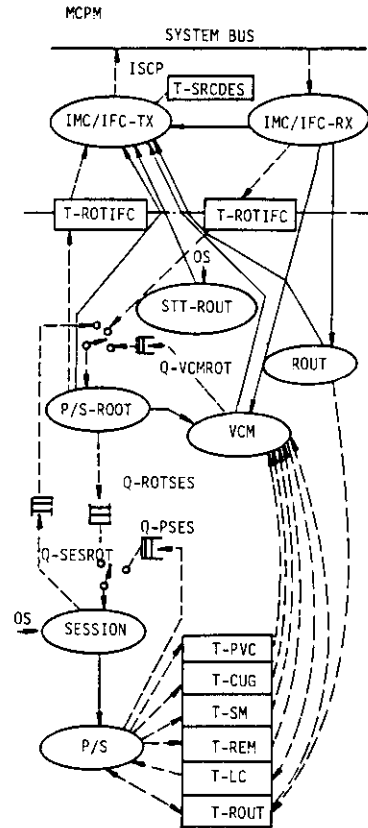


그림 8. MCPM의 전체 software 구조  
 Fig. 8. Overall software structure of MCPM.

request service function에서는 high level에서 제공한 node 운영 table을 사용하여 통화로를 신청한 사용자가 등록되어 있는 저 여부를 조사한다. 그리고 사용자가 등록한 통화로 사용 조건과 요구한 통화로 특성을 비교하여 통화로 개설 여부를 결정한다. 사용이 허락되면 node에서 현재 사용되고 있는 resource 들을 조사하여 통화로 번호와 함께 적절한 resource를 할당해 주고 그 밖에 사용자가 특별히 요청한 facility에 대하여 서비스 한다. 다음으로 연결할 통로를 얻기 위하여 routing table로부터 정보를 받아 연결해야 할 통화로를 결정하며 이 사실을 해당 LPM의 switching module로 내려보내 이용하게 한다. 그림 10은 위의 call request service에 대한 흐름도이다. VCM은 이러한 channel service를 위해 필요한 정보를 table에 기록하여 이용하는데 이러한 table들은 사용자 가입 상태를 기록하는 subscriber management table, 각 link 별 resource 사용 상태를 기록하는 resource management table, 개설된 channel의 사용 상태를 기록한 logical channel table, channel의 연결된 상황을 기

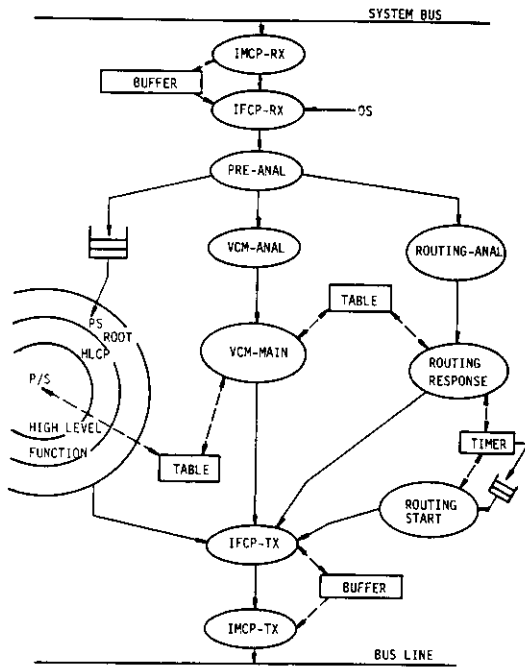


그림 9. VCM과 routing의 software 구조  
Fig. 9. Software structure of VCM and routing.

표 2. VCM의 서비스 기능.

Table 2. Service function of VCM.

Chanel service function	<ul style="list-style-type: none"> <li>• Call request service</li> <li>• Call confirmation service</li> <li>• Clear request service</li> <li>• Clear confirmation service</li> <li>• Restart request service</li> <li>• Restart confirmation service</li> </ul>
Node management function	<ul style="list-style-type: none"> <li>• Service table management</li> <li>• PVC management</li> <li>• Link management</li> </ul>

록하는 switching table, routing 정보를 제공하는 routing table 등이 있으며 그 밖에 closed user group (CUG)의 가입자를 기록한 CUG table과 permanent virtual circuit(PVC)의 사용상태를 기록한 PVC table이 있다.

Channel service를 위한 call, clear 및 restart packet의 format은 CCITT X. 25 권고사항을 따르며 call request packet내의 DTE address는 X. 121을 따랐다.<sup>1)</sup> KORNET 내부에서 사용하는 calling address 및 called address field는 기본적으로 8 digit를 사용하며 이때 calling address는 option이다. 또한 facility field나 call user data field의 사용 형식도 CCITT X. 25 권고사항을 따랐으며 구현된 facility로는 fast

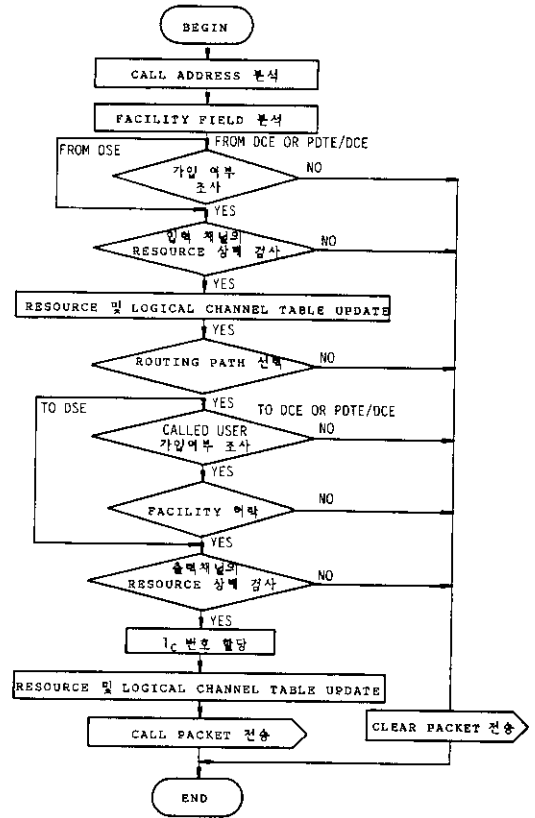


그림10. Call request의 flow diagram

Fig. 10. Flow diagram of call request service.

select, CUG, incoming call barred, outgoing call barred, one way direction, outgoing access, incoming access등이 있다.

신청한 channel에 대한 routing service는 routing table로부터 인접 node에 대한 정보를 받아 판단하며 logical channel 번호의 할당은 되도록이면 call collision의 발생을 방지하기 위해 DTE와 DCE에서 각각 높은 순서 및 낮은 순서부터 배정한다. NNP와 NNP 사이를 연결하는 data switching exchange (DSE)는 연결되는 상대 node사이에서 DTE mode 또는 DCE mode로 구분하여 할당한다.

그밖에 VCM에서는 node 운영을 위해 channel service에 사용하는 여러 table의 initialization을 행하고 information recording을 하며 P/S function으로부터 permanent virtual circuit(PVC)의 개설이나 취소 명령을 받아서 수행한다. 또한 각 LPM의 link에 대하여 status를 check하고 기동 및 정지 명령을 내린다.

## 2. Routing Management

KORNET에서 구현한 routing 방식은 ARPANET 와

유사한 분산 적응 방식이다.<sup>[6]</sup>

각 link의 통화량은 delay measure packet을 인접 node로 전송한후 이를 되돌려 받아서 계산한다. Node 간에 연결되는 경로에 관한 정보는 tree type의 topology table에 기록한다. 그리고 각 link의 통화량을 측정하여 해당하는 link의 weight 값을 topology table과 같은 tree 구조로 되어 있는 link weight table에 기록하여 사용한다. Link 내의 traffic의 측정은 주기적으로 하거나 node에 이상 상태가 발생할 때 행하며 측정 주기는 20초로 한다. 측정된 각 link의 상태에 변경이 생길 때 각 node는 변경된 link의 weight값을 flooding 방식으로 모든 node에 전송한다.<sup>[7]</sup> 여기서 flooding 방식으로 전송하는 이유는 routing table의 변경에 따른 과도기적 상태를 최소로 줄여 stability를 높이기 위함이다. 측정된 link weight를 토대로 channel을 형성하는 여러 경로 중에서 가장 적은 weight를 갖는 두 개의 경로를 선택하여 이를 routing table에 기록하여 VCM으로 하여금 이용하게 한다.

이러한 기능을 수행하기 위해 routing은 그림 9에서와 같이 routing start module과 routing response module로 구성되는데 그림 11은 이들에 대한 flow diagram을 나타낸다.

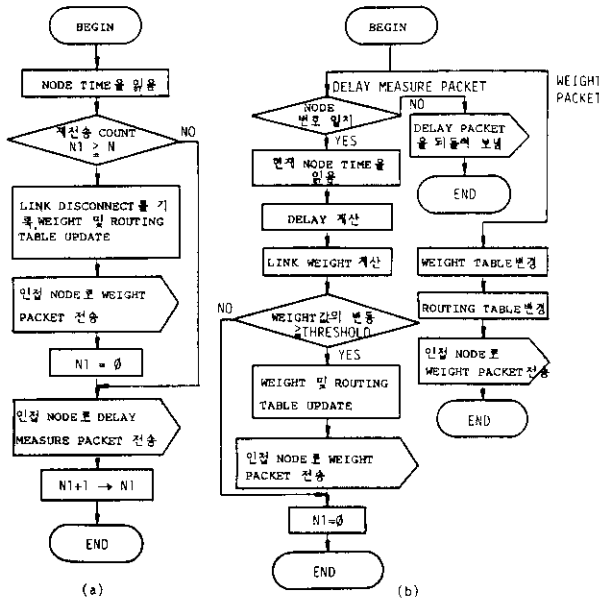
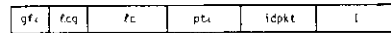


그림 11. Routing의 flow diagram  
 (a) Routing start (b) Routing response  
 Fig. 11. Flow diagram of routing service.  
 (a) Routing start routine  
 (b) Routing response routine

Routing start module은 OS에 의하여 주기적으로, 또는 VCM이나 P/S function에 의해 기동되어 현재의 node시간을 delay measure packet에 기록하여 인접 node로 전송한다. 또한 routing response module은 node로 수신된 routing packet을 받아서 delay measure packet인지 weight packet인지 구별한다. Delay measure packet이면서 자신의 node에서 발생시킨 packet일 때 round trip delay를 계산하여 link weight를 얻는다. Link weight에 변경이 생겼을 경우 weight packet을 만들어 인접 node로 전송한다. 만약 link weight packet을 받았을 경우에는 해당 link weight table을 update 시키며 이로부터 새로운 routing table을 작성한다.

그림 12는 routing packet의 format을 나타낸 것으로 여기서 idpkt는 delay measure packet과 link weight packet을 구별하며, I field에는 delay measure packet인 경우에는 node 번호와 node time이 기록되며 link weight packet인 경우에는 link 번호와 weight 값이 기록된다.



주: gfi : General format identifier  
 fcg : Logical channel group number  
 fc : Logical channel number  
 pti : Packet type identifier  
 idpkt : Routing packet identifier  
 I : Information

그림 12. Routing packet의 구조  
 Fig. 12. Format of routing packet.

### 3. High Level Protocol

NNP에서 구현한 high level protocol은 NMC가 수행하는 network management 기능과 monitoring 기능등을 돕기 위하여 primary/secondary (P/S) root, session layer 및 P/S function이 있다. 이러한 high level protocol은 CCITT의 권고사항이 아직 정립되지 않아 Bell 연구소의 BX.25를 따랐다.<sup>[8]</sup>

NMC와 NNP사이를 연결하는 protocol의 구조를 보면 그림 13과 같다. NMC에서 필요에 따라 주기적 혹은 비주기적으로 발생하는 message는 그림 13의 흐름을 따라 NNP의 P/S function으로 전달된다.

P/S root는 국제표준연맹에서 제정한 OSI의 7계층 구조에서 보면 packet layer와 transport layer의 일부를 포함하며 DTE와 DCE의 접속기능, end-user로서 virtual circuit을 관장하며 channel의 multiplexing외에 assembly 및 disassembly하는 기능, P/S function을 support하는 session layer와의 접속기능



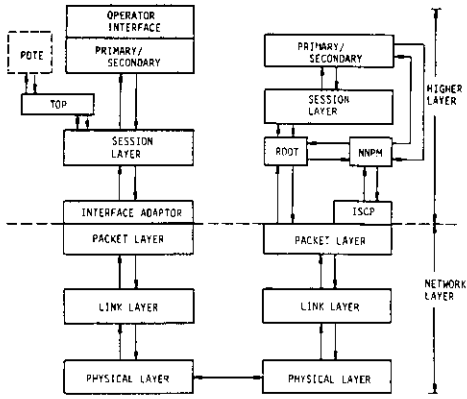


그림13. P/S function과 NMC와의 통신  
Fig. 13. Communication of P/S function and NMC.

등이 있다.

High level protocol의 software구조는 인접된 다른 module에서 들어온 packet을 polling 및 analysis하는 part와 P/S root의 main function을 처리하는 part 그리고 packet을 전송하는 part로 나뉘어지며 이러한 구조는 packet layer나 PDTE/DCE와 유사하다.

Session layer는 두개의 high level entity 즉 NMC와 NNP의 primary/secondary function사이를 연결해 준다. Session 개설시 마다 entity 간의 특성과 번호들이 정해져야 하는데, 여기에는 session을 연결하는 higher layer entity의 종류를 결정하는 origin type, one-way 또는 two-way와 같은 channel의 종류를 결정하는 session type, data code를 결정하는 data presentation type, 그리고 data의 동기화를 위한 re-synchronization option과 recovery option 등이 있다. 그 밖에 session을 구동시키는 entity는 수신 DTE에 session을 access할 수 있는 entity가 둘 이상 있을 때는 session address를 사용하여 식별하며 이때 address field는 최대로 16 octet를 넘지 않는다.

NNP의 P/S function은 NMC의 P/S function의 counter part로서 동작하며 기능 수행상 VCM과 밀접한 관계를 갖는다. NMC와 communication을 위해 message의 종류 및 내용에 대한 합의가 이루어져야 하는데 message의 종류는 크게 network management, network monitoring, subscriber management, routing monitoring 등의 4 가지로 분류된다. 이러한 4 가지 종류의 message는 header information에 의하여 구분되는데 header는 command identification, message의 전송 순서를 알리는 procedure sequence number, source와 destination terminal number 등으로 구성된다. Header 뒤에는 종류에 따라 여러 com-

mand parameter들이 붙는데 최대 20 octet까지 허용된다.<sup>11)</sup>

#### 4. Interstation Communication Protocol (ISCP)

ISCP는 모든 hardware board내에서 동일한 기능을 수행하므로 여기서 종합적으로 설명한다. 이 protocol은 서비스 기능상으로 sender와 receiver의 두 부분으로 구분된다. Protocol 기능 상으로는 packet layer에 해당하는 interfunction communication protocol (IFCP)와 link layer에 해당하는 inter-module communication protocol (IMCP)의 2 계층 구조로 구성되며 그림14에 ISCP의 software 구조가 그려져 있다.

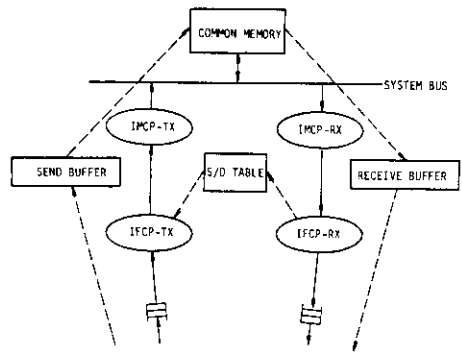
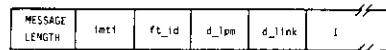


그림14. ISCP의 structure  
Fig. 14. The structure of ISCP.

먼저 IFCP-TX는 입력된 packet을 종류에 따라 구분하여 VCM에서 보내온 source-destination 연결 table을 참고하여 적절한 header를 붙인 후에 IMCP-TX에 보내 다른 board로 전송하게 한다. 이때 IMCP-TX는 header의 내용에 의해 common memory의 상대편 area로 packet을 전송한다. 다음으로 IMCP-RX는 common memory의 자신의 영역으로 부터 packet을 읽어와 IFCP-RX에게 넘겨준다. IFCP-RX는 header를 분석하여 해당되는 서비스 module로 packet을 전송한다. 이때 ISCP에서 사용하는 header format은 그림15와 같다. 여기서 imti를 이용하여 data packet, channel control packet, node control packet 등으로 packet



- imti : Inter-message type identifier
- ft-id : Function identifier
- d\_lpm : Destination line processing module
- d\_link : Destination link
- I : Information data

그림15. ISCP의 packet 구조  
Fig. 15. Packet format of ISCP.

type을 구분하고, ft-id를 이용하여 packet을 전송받을 service module을 지정한다. 그리고 d-lpm과 d-link를 이용하여 전송될 hardware board의 LPM number와 link number를 지정한다.

#### IV. 結 論

본 제 4 부 논문에서는 asynchronous channel로 character mode DTE (CDTE)를 접속할 때 쓰이는 PAD protocol과 NNP system 운영을 위한 통신망 운영 software의 구현에 대하여 기술하였다.

PAD protocol 구현에서는 service specification으로 부터 protocol 구현 및 protocol conversion 방법 등에 관해서 설명하고 또한 asynchronous line 접속을 위한 각 module의 운영방법을 기술하였다. NNP 운영을 위해 두뇌 역할을 하는 MCPM system에서는 각 channel service 방법 및 routing 선로 결정을 위한 algorithm을 제시하였으며 NMC와 연결하기 위한 high level protocol과 응용 software의 구현 방법에 대하여 설명하였다.

PAD 및 PTC protocol은 parameter table을 사용하여 서로 다른 service 특성을 갖는 여러 통신 시스템을 X.25 network에 접속하는데 쉽게 응용할 수 있다. 이 밖에 서로 독립적으로 운영되는 hardware board로 구성된 시스템의 운영 및 감독방법은 시스템이 복잡하고 service 특성이 다양한 시스템에 쉽게 적용할 수 있다.

#### 參 考 文 獻

- [1] 은종관 외, Packet Switching에 의한 공중 Computer 통신망 개발 연구-제 1부: KORNET의 개요와 Network Management Center (NMC)의 개발, 대한전자공학회지, 제22권 6號, 1985年 11月.
- [2] 은종관 외, Packet Switching에 의한 공중 Computer 통신망 개발 연구-제 2부: KORNET의 설계 및 Network Node Processor (NNP) 개발, 대한전자공학회지, 제22권, 6號, 1985年 11월.
- [3] 은종관 외, Packet Switching에 의한 공중 Computer 통신망 개발 연구-제 3부: KORNET NNP의 X.25 Software 설계 및 구현, 대한전자공학회지 제23권, 1호, 1986년 1월.
- [4] CCITT Recommendation X. 3, X. 28, X. 29, 1980.
- [5] CCITT Recommendation X. 121, 1980.
- [6] J.M. Mcquillan, I. Richer, and E.C. Rosen, "The new routing algorithm for the ARPANET," *IEEE Trans. Commun.*, vol. COM-28, May 1980.
- [7] A.S. Tanenbaum, *Computer Networks*, Prentice-Hall Inc., Englewood Cliffs, N.J., 1981.
- [8] American Telephone and Telegraph Company, BX. 25, 1982.