

공중 패킷 스위칭 네트워크의 개발에 관하여

-KORNET 개발 현황을 중심으로-

金 洛 明*, 殷 鍾 官**

金星電氣(株)*, 韓國科學技術院 教授(工博)**

I. 개 요

데이터 통신이란 일반 음성통화를 위한 전화통신과 구별되는 개념으로써, 여러 가지 정보자료들을 한 가입자와 다른 가입자사이에 상호 전달해 주는 기능을 말한다. 여기서는 팩시밀리, 비데오텍스등에 의한 영상전송이나, 테렉스등에 의한 문자전송, 컴퓨터등에서 생성되는 수치자료의 전송등, 다양한 형태의 정보교환이 포함된다. 한 걸음 더 나아가 지금 선진제국에서는 음성신호의 디지털화 기술을 응용하여, 음성통화까지 데이터 통신의 영역으로 흡수하려는 노력도 활발하다. 이상의 여러 가지 통신의 목적을 이루자면 먼저 두 가입자사이에 어떤 통화로를 마련하여야 한다. 이때 단일 공용의 통신망이 없다면 한 가입자는 그가 통화하고자 하는 대상의 수만큼 통화로를 설비하여야 한다. 연구 결과에 의하면, 실제 운용되고 있는 컴퓨터 시스템에서 가입자와 컴퓨터사이의 통화로 실점유율은 통화시간의 5%에 불과하다고 한다.^[1] 따라서 모든 통화로를 일대 일로 연결한다 할 때의 효율은 5%밖에 되지 않는다는 결론이다.

패킷 스위칭 네트워크(packet switching network)는 전술한 다양한 데이터 통신의 요구를 하나로 묶어 해결하면서 통화로 이용의 효율성을 극대화하는데 목적을 둔다. 즉, 각 가입자로부터 입력되는 메세지들을 표준화된 데이터 묶음, 즉 '패킷(packet)'으로 만들어 그 패킷에 송신측 주소, 수신측 주소, 전송될 길(route) 등에 관한 정보를 부여함으로써 여러 입력 신호들이 하나의 채널로 다중화 된다 하더라도 혼선이 없도록 설계한 통신망을 말한다. 패킷 스위칭 네트워크는 다음과 같은 장점을 지닌다.

- 효율적인 회선 이용
- 분산형 교환기능
- 융통성 있는 회선망의 구성
- 높은 신뢰도

• 항시 접속 가능

본 논고에서는 현재 과학기술원에서 국내 최초로 개발중인 컴퓨터 네트워크인 개발 현황을 중심으로 패킷 스위칭 네트워크 기술에 관하여 서술한다. 먼저 제Ⅱ장에서 통신망이 갖추어야 할 기본 구조를 검토하고 교환단국인 network node processor(NNP)의 구성을 토의한다. 패킷 스위칭 네트워크의 개발에는 하드웨어에 비해 소프트웨어의 비중이 매우 크다. 제Ⅲ장 및 제Ⅳ장에서는 통신망의 설계를 위한 소프트웨어 및 하드웨어 개발 단계를 살펴본다. 제Ⅴ장에서는 네트워크 운용에 관한 제반변수를 예측하기 위한 컴퓨터 시뮬레이션 방식을 토론했고 제Ⅵ장에서 결론을 맺는다.

II. 네트워크의 기본구조

1. 통신망의 조성 및 설계

앞에서 살펴 보았듯이 패킷 통신망은 지역적으로 떨어져 있고, 서로 다른 데이터 포맷이나 프로토콜을 가진 메세지 소스들을 적절한 방법으로 연결하여 효과적인 리소스 사용을 도모하고 있다. 일반적으로 패킷 통신망은 그림 1과 같은 구조를 이룬다.

그림 1에서 보듯이 가입자 컴퓨터나 터미널들은 직접 또는 간접적으로(이 경우 네트워크 컨센트레이터를 포함) NNP라 불리는 패킷 스위칭 단국에 연결된다. NNP는 입력 데이터나 타 NNP로부터 온 패킷을 받아 적절한 데이터 처리를 하고 그 데이터가 가야할 목적지를 살펴서 그에 맞는 길로 재송출하는 기능을 가진다. NNP에는 각종 프로토콜-송수신을 위한 규약-이 마련되어 가입자와의 오해없는 다이아로그를 이룬다. 한편 network concentrator(NC)는 여러 개의 가입자 터미널로부터 오는 데이터를 일차적으로 집적화하는 곳으로 스태티스틱알 멀티플렉싱과 간단한 데이터 프로세싱 기능을 가지고 있다. 결국 통신망의 설계는 가입자의 지정학적 위치를 고려하여 네트워크

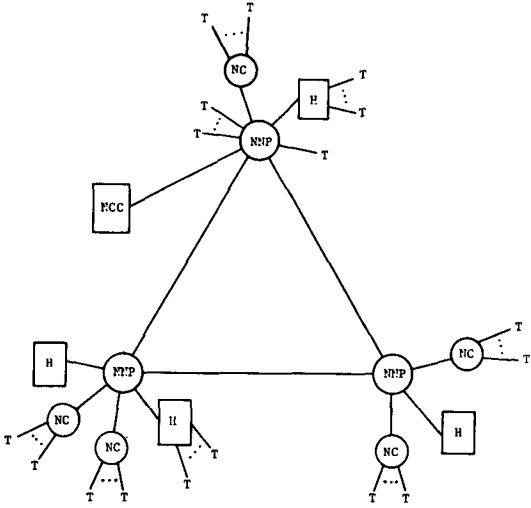


그림 1. 기본적인 패킷 통신망의 조성
(주 : NNP - Network node processor, NC - Network concentrator, NCC - Network control center, H - Host computer, T - Terminal)

의 요소(NNP, NC 등)를 적절히 배치하고 이러한 요소 상호간의 전송선 용량등을 예상되는 수요에 맞게 결정하는 일을 말한다.

이때 NNP의 배치와 상호 연결을 구상하는 것을 간선 설계(backbone design)라 하고 NC의 위치 설정을 다루는 부분을 지선 설계(local network design)라 한다.

2. 루팅 및 플로우 콘트롤 알고리즘

설계 과정을 통하여 네트워크가 정해지면, 다음의 과정으로 통신망내에서 데이터가 흘러갈 길을 지정하고 가장 효율적인 데이터의 흐름을 유지할 수 있도록 하는 알고리즘의 마련이 필요하다. 이때 전자를 루팅, 후자를 플로우 콘트롤 기능이라 부른다. 루팅

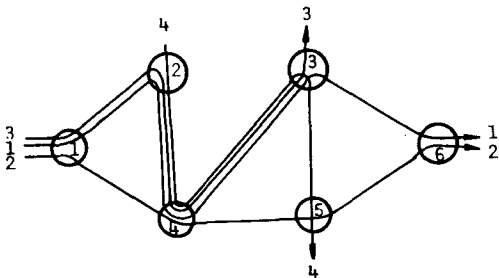


그림 2. 루팅 구현의 예

의 한 예를 보면 그림 2와 같다. 루팅 알고리즘은 대체로 최소 지연 통화로의 선정을 목표로 한다. 즉, 송신 NNP와 수신 NNP가 정해졌을 때 양자간에 만들 수 있는 모든 루트에 대하여 전송 지연을 계산해 보고 그 중에서 최소의 지연을 가지는 것을 골라내는 것이다. 루팅 알고리즘은 그 제어방식에 따라 집중제어방식과 분산제어방식의 두 가지로 대별된다. 집중제어방식은 네트워크 전체의 운영을 총괄하는 network control center(NCC)에서 네트워크의 이용상태에 대한 모든 자료를 다 수집하여 루팅 테이블을 한꺼번에 계산하고 그 결과를 각 NNP에 분배하는 방식이다. 대표적인 알고리즘으로는 Dijkstra의 알고리즘이 있다.^[5] 반면 분산제어방식은 루팅 테이블을 계산하는 기능은 각 NNP에 분산하여 부여하고 각 NNP는 인접 NNP들로부터 정기적, 혹은 비정기적으로 네트워크 오는 상태에 대한 새로운 정보를 받아 테이블을 그때 그때 업데이트하도록 마련하였다. 대표적인 예로는 초기 arpanet에서 응용된 Ford와 Fulkerson의 알고리즘이 있다.

한편, 아무리 도로망이 잘 마련되어도 어느 한쪽으로 차량이 몰리면 정체현상이 심해지듯이 일시적으로 어느 선로의 이용율이 선로의 처리 능력을 넘어서면 통화로의 포화상태가 발생한다. 이러한 포화상태 미연에 방지할 수 있는 알고리즘으로 플로우 콘트롤, 즉 통화량의 평가 및 제어방식이 마련된다.^{[7][8]} 개념적으로 보아 플로우 콘트롤이란 각 NNP에 마련된 queueing buffer의 이용에 일정한 한계를 둬으로써 여분의 버퍼 스페이스를 피크 트래픽의 흡수에 이용한다는 정책이라고 볼 수 있다.

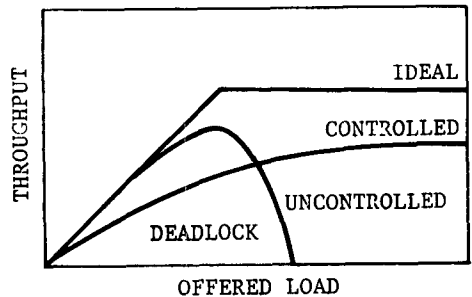


그림 3. 플로우 콘트롤의 효능

그림 3은 플로우 콘트롤의 효용을 그래프로 나타낸 것이다. 그림에서 보듯이 트래픽이 어느 정도 증가하면, 플로우 콘트롤이 없을 경우에는 급격한 성능 저하가 있는 반면 플로우 콘트롤이 운용되면 지속적인 성

능 상승 효과를 얻을 수 있다. 플로우 컨트롤의 대표적인 예로는 IBM의 SNA, CCITT 권고사항등에서 채택하고 있는 윈도우 컨트롤 알고리즘이 있다.^[9]

3. NNP의 구성

패킷 스위칭 네트워크 개발에서 가장 중요한 부분이 되는 것은 NNP의 개발이다. 전술한 사항들을 토대로 볼 때 NNP에는 그 NNP에 연결되어 있는 가입자에 대한 서어비스 기능, 他 NNP로부터 오는 데이터의 신속한 처리기능등이 릴 타임 프로세스로 마련되어야 한다. 또한 하드웨어 및 소프트웨어 기능의 많은 부분이 모듈별로 독립되고, 추후의 기능확장에도 적응성이 우수하여야 하므로 멀티 프로세스 시스템의 구조를 취하는 것이 좋다. NNP에 내장되어야 할 소프트웨어로는 가입자 서어비스를 위한 패킷 (assembly/disassembly PAD) 기능을 비롯하여 데이터 전송 프로토콜인 X.25의 제반 규정을 담은 프로그램, 루팅 알고리즘, 플로우 컨트롤을 위한 버퍼 매니지먼트 프로그램 등이 기본이 되고, 기타 시스템의 보전에 관계되는 기능으로서 링크 장애의 보고기능, 선로 이용에 관한 자료수집 프로그램, 각종 네트워크 요소의 검진 기능 등이 추가되어야 한다. 세부적인 소프트웨어 및 하드웨어 개발 기술에 관한 사항은 제Ⅲ장 및 Ⅳ장으로 미룬다. 종합하여 NNP의 구조를 개념적으로 도시하면 그림 4와 같다.

Ⅲ. 프로토콜 및 소프트웨어의 마련

1. 소프트웨어의 구조

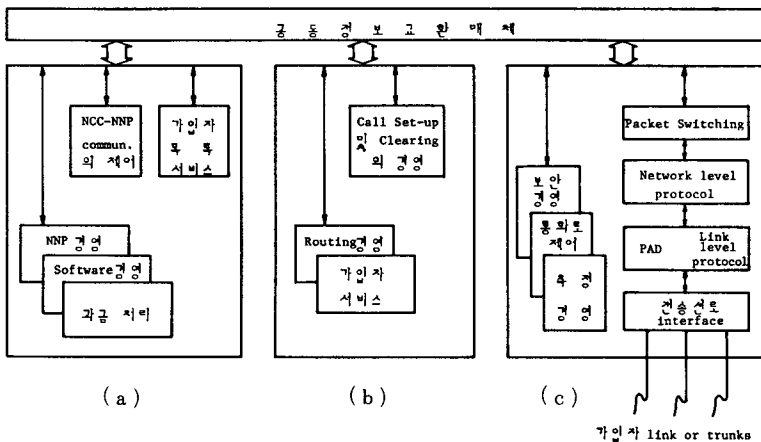


그림 4. 개념적으로 본 NNP의 구조

- (a) NNP 고유기능 및 운영기능 (b) 가입자 통화 서비스
- (c) 가입자 선로 및 NNP간 트렁크 제어(주: PAD - packed assembler / disassembler)

통신망의 궁극적인 목표는 가입자가 만드는 데이터를 일정한 전송 설비를 이용하여 목적지까지 원활하게 전달하는데 있다. 그런데 가입자의 종류나 전송할 데이터의 포맷등은 매우 다양하므로 통신망의 설계에는 호환성 (compatibility)이 가장 강조된다. 이러한 목적으로 International Standards Organization (ISO)에서는 7 레이어로 된 패킷 스위칭 네트워크의 표준 모델을 제시하였다(그림 5).

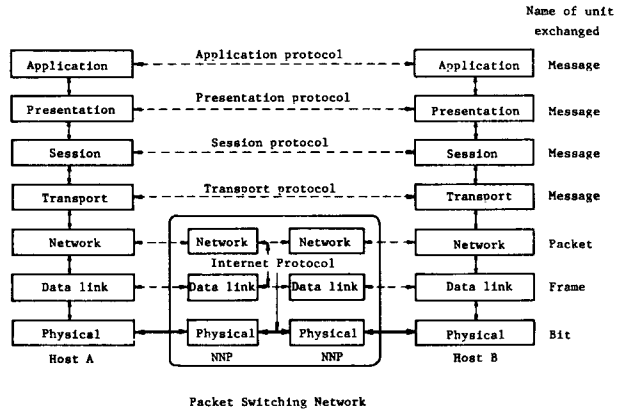


그림 5. ISO에서 제정한 오픈 시스템 인터코넥션 모델 (7 레이어 구조)

그림 5에서 상하 두 레이어사이의 신호 교환을 정의하는 것을 인터페이스라하고 동일 레이어들사이의 신호교환을 정의하는 것을 프로토콜이라 부른다. 이중 인터페이스에 관한 사항은 비교적 간단하며 표준화

되어 있지 않으나 프로토콜에 관한 규정은 여러 형태의 표준 모델이 있고, 각 레이어마다 독립적으로 정의되어 있다. 대표적인 프로토콜의 예를 몇 가지 들면 다음과 같다. 먼저 피지컬 레이어를 위한 프로토콜로서는 EIA RS-232-C를 비롯하여 RS-449, CCITT X.20, X.21 등이 있다. 또 데이터 링크 레이어용 프로토콜로서는 ANSI에서 제정한 ADCCP, ISO에서 제정한 HDLC, IBM에서 제정한 SDLC, CCITT에서 제정한 X.25 프레임 레벨 규정 등이 있다. 기타 네트워크 레이어 이상에 적용되는 규정은 CCITT에서 제정한 X.25 패킷 레벨 규정(ISO 표준 모델 중 레이어 3 및 레이어 4의 일부를 만족시킴) 외에는 제대로 표준화된 규정은 없고, 여러 통신망들이 독자적으로 제정하여 운용하고 있다. 이러한 예로는 arpanet의 file transfer protocol(FTP), transaction-oriented protocol(TOP) network voice protocol(NVP) 등이 있다.

이상과 같이 통신망에서 응용할 수 있는 프로토콜들은 다양하나 현재 세계적인 추세가 CCITT의 제반 규정을 많이 따르고 있으며 현재 과학기술원에서 개발 중인 KORNET도 CCITT 규정을 따라 설계함을 원칙으로 하고 있으므로, 본 논고에서는 CCITT 규정을 중심으로 NNP 소프트웨어 개발의 제반 단계를 설명하기로 한다.

2. NNP에 필요한 소프트웨어의 종류

NNP의 소프트웨어 기능은 크게 고유기능, 가입자 서어비스, 네트워크 운영기능의 세 가지로 나뉜다. 여기서 NNP의 고유기능이라 함은 NNP의 bootstrapping, 소프트웨어의 초기 loading, 주입부의 scheduling 등의 OS와 관련되는 기능을 말한다. 또 가입자 서어비스라 함은 가입자 링크 제어(PAD, X.25 프레임 레벨 프로토콜), 가입자 통화로(로지컬 채널)의 제어(X.25 패킷 레벨, 인터럽트 서어비스등), 여러 가지 에뮬레이션 기능의 운용 등 가입자의 통화와 직접 관련되는 기능을 뜻한다. 끝으로 네트워크 운영기능은 NCC의 한 부분으로 동작하는 NNP의 기능이라 말할 수 있으며, 루팅 테이블의 운용, 고장에 대한 자동복구기능, NNP 주변기기의 제어 등의 기능이 포함된다. 이러한 기능들을 보다 세부적으로 나열하면 다음과 같다.

i) NNP의 고유기능

ii) 커뮤니케이션 리소스의 경영

- 링크의 매개변수 제어
- 루프시럽(모뎀, 레이어별 루프시럽)
- 가입자 선로의 조정, 단절 및 수정

iii) 링크의 형성에 관련되는 소프트웨어

- X.25 프레임 레벨의 규정
- X.21 또는 RS-232-C의 규정
- iv) Virtual circuit(VC)경영
 - Call의 형성, 단절, 재구성
 - 패킷 스위칭 및 플로우 콘트롤
- v) 루팅 경영
 - 루팅 테이블의 계산
 - 루팅 테이블에 따른 버퍼링 프로그램
- vi) 데이터 포맷의 변환
 - PAD 기능 및 프로토콜 변환
- vii) 가입자 경영
 - 가입자 변수의 등록, 삭제 및 수정
 - 패스워드를 통한 가입자 인지
- viii) 내부 프로토콜의 마련
 - NCC로부터 오는 명령의 수행 및 결과 보고
 - 루팅에 관한 정보의 교환
- ix) 소프트웨어 경영
 - 각종 자료의 수정 및 첨가
- x) 측정 경영
 - 통화로 사용도에 관한 측정
 - 에러에 관한 측정
- xi) 과금자료의 수집

소프트웨어의 개발이란 결국 이상의 제반 기능들을 각각 구현하고 전체를 모아 일관성 있게 운용되게 하는 일이 된다. 이중, 초기 개발단계에서 가장 중점을 둔 부분은 가입자 서어비스 부분이다. 앞에서도 언급하였듯이 이 부분은 주로 CCITT X series 규정에 의거하여 개발함이 바람직하며 따라서 이 규정들의 기본 구조를 파악하는 일은 매우 중요하다.

3. X.25 프레임 레벨 및 패킷 레벨 프로토콜의 구조

그림 6은 패킷 통신망의 구조를 X.25 프로토콜을 기준으로 도해한 그림이다. 가입자가 전송한 산발적인 가입자가 전송한 산발적인 데이터는 PAD에서 패킷으로 포장되고 그것이 패킷 레벨, 프레임 레벨을 거쳐 전송선로에 실린다. 이 데이터가 몇 개의 중간 NNP들을 거쳐 마침내 수신 NNP에 도착하면 프레임 레벨, 패킷 레벨, PAD의 역과정에 의해 처음 전송한 형태의 데이터로 재생되어 수신가입자에게 전달되는 것이다.

본 란에서는 네트워크 설계에 근간이 되는 프레임 레벨과 패킷 레벨의 구조를 살펴보기로 한다.

먼저, 링크 어세스 과정을 규정한 프레임 레벨 프로토콜은 asynchronous response mode의 규정(LAP 프로토콜)과 asynchronous balanced mode의 규정

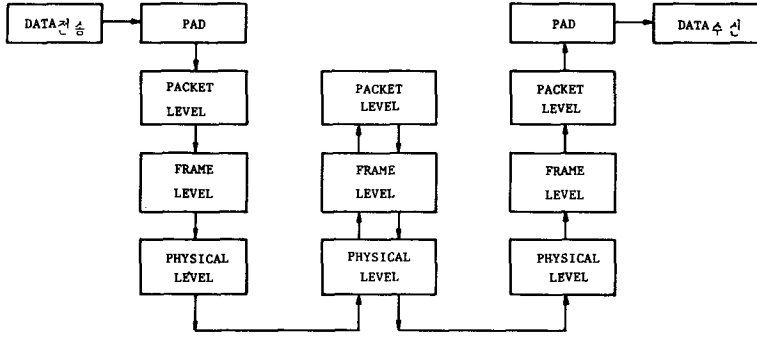
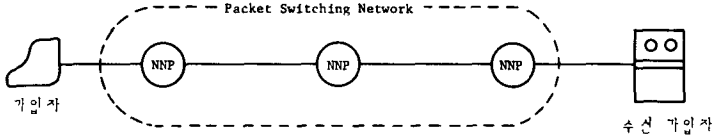


그림 6. X.25 프로토콜을 중심으로 본 네트워크의 구조

(LAPB 프로토콜)의 두 가지로 구성된다. 전자는 마스터와 슬레이브가 미리 규정되어 있는 시스템에 쓰이는데, 이 경우 슬레이브는 마스터측의 어떤 명령이 있을 때만 응답과 아울러 정보 전송을 할 수 있다는 제약이 따른다. 반면 후자의 경우에는 양편이 모두 고유한 전송권을 가지고 있어서 full-duplex(FDX) 선로를 통하여 언제나 자유롭게 데이터의 송·수신을 할 수 있다. 프레임 레벨의 요체는 에러 콘트롤과 플로우 콘트롤 양식에 있다. 기본적인 에러 콘트롤 양식은 수신한 프레임에 대하여 잘 받았으면 acknowledgement (ACK), 에러가 발생했으면 negative acknowledgement (NACK)를 보내는 방식을 채택한다. 여기에 윈도우 콘트롤이라 불리는 대표적인 플로우 콘트롤 양식을 도입하여 프레임 하나 하나마다 ACK 응답을 해야하는 비효율성을 막아주고 있다. 즉, V(S), N(S), V(R), N(R)의 네 가지 변수를 정의하여 전송되는 각 프레임에 모듈 8 또는 모듈 128로 순열수를 매김으로써 하나의 ACK로 몇 개의 프레임에 대한 응답을, 동시에 하도록 마련한 것이다. 그림 7은 패킷이 담긴 인퍼메이션 프레임의 포맷을 도해한 것이다.

한편 패킷 레벨 프로토콜은 송신 NNP와 수신 NNP 사이에 일정한 로지컬 채널을 형성하여 가입자가 만드는 데이터 패킷 또는 제어 패킷의 전송을 경영하는 기능을 정의한다. 정의되는 로지컬 채널에는 전송 형태에 따라 datagram(DG), virtual circuit(VC), permanent virtual circuit(PVC)의 세 가지가 있다. DG

방식은 송신측과 수신측 사이에 실제적인 회로가 구성되지 않고 하나 하나의 패킷에 송·수신측 주소, 이용할 설비, 우선권 등에 관한 모든 정보를 실어서 전송하고, 각 노드가 이것을 해독하여 수신단까지 전달하도록 설계하는 방식이다. 반면 VC 방식은 송·수신단 사이에 논리적인 전송로를 정의하여 양자간의 데이터 교환은 그 전송로를 통하여 이루어지도록 하는 방식이다. VC는 call이 끝나면 지워진다. 이에 대하여 PVC는 call이 끝나도 논리적인 전송로는 지워지지 않

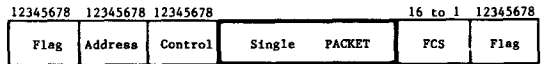


그림 7. 인퍼메이션 프레임의 포맷

(주 : FCS; frame check sequence)

는다는 점이 특징이다. 패킷 레벨 프로토콜에서는 세부적인 에러 콘트롤은 마련되지 않는다. 플로우 콘트롤은 프레임 레벨에서와 같이 윈도우 콘트롤 방식을 운용하며, 추가적으로 D bit, M bit, Q bit 등을 마련하여 통화로 이용의 범주를 넓혀주고 있다.^[10]

종합하여 이상의 프레임 레벨 및 패킷 레벨의 프로토콜은 스테이트 머신 구조로 프로그래밍함으로써 구현할 수 있어서 소프트웨어 개발의 첫단계는 각 레벨의 스테이트들은 규명하고 각 스테이트 사이의 데이터 흐름을 정의하는 일이 된다.^[11] 한 예로 패킷 레벨의 제반 스테이트를 그림으로 표현하며 그림 8과 같다.

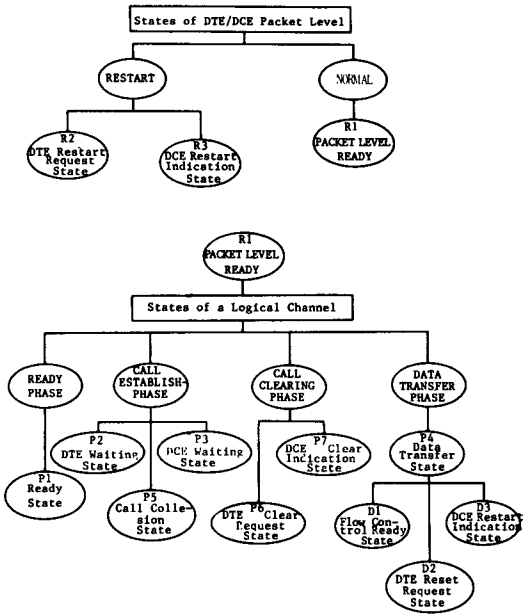


그림 8. 패킷 레벨의 제반 스테이트

IV. NNP 하드웨어의 구조

제 III장에서 규정한 소프트웨어 기능이 NNP의 업무를 규정짓는 것이라면 하드웨어란 그 업무를 수행할 수 있는 주변환경이 된다. 아울러 데이터의 입력, 처리, 출력 등의 기능이 기본되는 하드웨어 기능위에 운용될 것이므로 하드웨어 설계는 NNP의 성능에 일차적인 연관을 가진다. NNP가 갖추어야 할 I/O 용량을 KORNET NNP의 SPEC을 기준으로 살펴보면 아래와 같다.

- i) 다른 NNP와 연결되는 트렁크 용량
56kbps × 4trunks (최대) = 224kbps
 - ii) NC를 연결하는 링크의 용량
20kbps × 4 links (최대) = 80kbps
 - iii) 호스트를 연결하는 라인의 용량
9600bps × 4 lines (최대) = 40kbps
 - iv) 터미날을 연결하는 라인의 용량
2400bps × 64terminals (최대) = 155kbps
- 총 499kbps

설계에 앞서 선행되어야 할 일은 NNP 하드웨어에 필요한 전제 조건들의 검토이다. 첫째, 신호처리 속도가 고려되어야 한다. 즉, 데이터를 받아서 다시 내보내는 데까지 걸리는 시간에 제한이 있으므로 하드웨어는 신호처리 속도가 비교적 빠른 마이크로컴퓨터나 최신의 16비트 마이크로프로세서등을 도입함이 바람직하다.

둘째, 하드웨어 기능을 모듈별로 분리함으로써 장래의 NNP 확장에 적응성이 크도록 하여야 한다. 셋째, 소프트웨어 구조가 상하체계를 가진 층상구조이므로 이 구조에 맞추어 하드웨어의 모듈라이제이션을 다시 고려하여야 한다. 넷째, 하드웨어 일부 또는 전부의 고장 발생시의 혼란을 예상하여 하드웨어 모듈의 중복 등의 방식으로 신뢰도를 높여야 한다.

이러한 전제조건들을 종합해 보면 결국 16비트 마이크로프로세서를 이용한 멀티-프로세서 구조가 NNP 하드웨어에 가장 바람직하다는 결론을 얻는다. 이 구조는 개발중이거나 최근에 개발된 선진국들의 패킷 교환망에서 많이 쓰이고 있는 구조이다. 참고로 KOR-NET NNP의 하드웨어 구조를 도시하면 그림9와 같다.

V. 네트워크 시뮬레이션

일반적으로 패킷 교환망의 개발에 임하여 고려하여야 할 문제는 상당히 많고 또 복잡하다. 이러한 복잡한 문제들의 분석과, 개발될 통신망의 성능예측은 시뮬레이션을 통하여 어느 정도 가능하다. 일반적으로 시뮬레이션은 몇 개의 과정으로 구분된다. 즉,

- i) 문제의 정의 및 시뮬레이션 대상 선정
 - ii) 수학적 모듈의 구성
 - iii) 프로그램 작성 및 결과 검토
- 등의 단계를 거치게 되는데, 우리가 네트워크 시뮬레이션으로 얻고자하는 정보는 다음과 같이 요약될 수 있다.
- i) 각 선로 및 CPU, 버퍼 등의 이용도 계산
 - ii) 네트워크의 신뢰도 측정
 - iii) 구현된 프로토콜의 성능 평가
 - iv) 네트워크에서 발생하는 시간 지연의 계산 등

시뮬레이션 프로그램을 짜는 일은 FORTRAN, PL/1, BASIC, ALGOL, COBOL 등 일반 프로그래밍 언어의 사용으로도 가능하나, 프로그램의 확장에 대한 적응성, 프로그래밍의 난이도, 실제 시스템과의 연관성 등을 고려하여 GPSS나 SIMSCRIPT 등과 같은 특별한 시뮬레이션용 언어를 사용함이 바람직하다. 네트워크 시뮬레이션은 NNP에 대한 펀크션 모델을 기본요소로 하여 수행된다. 한편 NNP의 펀크션 모델은 다음의 다섯가지 블록으로 분류된다.

- i) Polling
 - ii) NNP 입력과정
 - iii) 스위치 블록
 - iv) NNP 출력 과정
 - v) 서브루틴
- 우선 폴링 블록에서는 각 입력 라인을 주어진 시간

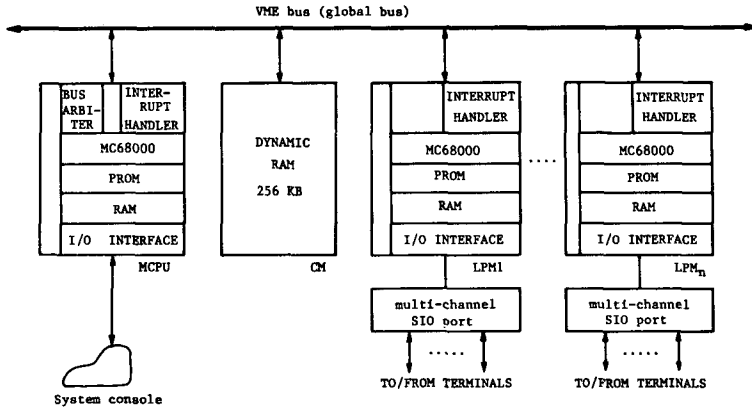


그림 9. NNP 하드웨어의 구조

(주: MCPU - master CPU, CM - common memory, LPM - local processing module)

동안 차례로 체크하여 각 라인의 이용도에 관한 통계를 구한다. 동시에 NNP의 입력 블럭에서는 주어진 분포함수에 따라 가상적인 메시지를 발생시키며 이 메시지는 각 라인의 입력으로 할당된다. 라인의 종류에 따라 메시지는 서로 다를 프로토콜의 지배를 받도록 설정된다. 입력 버퍼에 들어온 메시지는 스위칭 되어 NNP 출력 블럭으로 전달되는데 이를 위해 스위칭 블럭에서는 각 프로토콜에 맞는 루팅 기능을 갖추고 있다. NNP 출력 블럭에서는 각각의 메시지가 전송되어야 할 라인으로 메시지를 내보낸다. 끝으로 서브루틴 블럭은 주로 버퍼의 상태점검 및 ACK/Response 양식의 메시지의 처리를 담당하는 프로그램들이 마련된다. 이러한 다섯 블럭은 하나의 모니터 프로그램으로 통제되며 결국 concurrent programming 기법으로 NNP의 펀크션 모델을 프로그래밍 할 수 있다. 이것을 기본으로 네트워크 전체의 흐름을 표현하는 큰 프로그램을 짜게 되는데 기본적인 접근 방식은 이것과 동일하다.

이러한 시뮬레이션 과정은 패킷 통신망의 개발업무와 병행하여 진행되어야 하며 양자간에 긴밀한 협조가 이루어질 때 비로소 성공적인 개발이 보장되는 것이다.

VI. 결론

지금까지 KORNET 개발 현황을 중심으로 패킷 스위칭 네트워크에 관련된 이론적, 실제적인 제반사항들을 살펴보았다. 패킷 스위칭 네트워크에 관한 연구는 70년대 초반부터 점진적으로 진행되어 왔으며, 데이터 통신, 정보사회의 구현등에 필수적인 요건이 됨에 따라 최근에는 전 세계적으로 하나의 붐을 형성하고 있

다. 여기에 발맞추어 국책사업의 하나로 추진되고 있는 KORNET의 개발은, 전세계 표준 네트워크와의 호환성을 유지한 한국형 네트워크의 개발이라는 점에서 의미가 깊다. 특히, CCITT의 최신 규정을 구현하고, 소프트웨어의 개발에 SDL, CHILL, MML 등^[4] 세계적으로 공인된 소프트웨어 개발 방법론을 도입하고 있어서 개발된 KORNET은 명실공히 up-to-date한 시스템이 될 것이다. 아울러 이 프로젝트를 통한 기술의 축적으로 장래 수요 증가에 의한 네트워크 확장시에도 우리힘으로 문제 해결에 임할 수 있음을 큰 자량이 된다.

결론으로, 마이크로프로세서 응용, 소프트웨어 프로그래밍 첨단 분야의 전자공학등이 집적화된 패킷 통신망의 개발은 곧, 새로운 형태의 산업구조 형성을 예시하고 있으므로 본 논고에서 소개한 제반 기술에 대하여 보다 깊이 검토하고 여러 가지 응용분야를 창안해 냄으로써 장래의 국내 산업분야를 선도해 갈 수 있기를 믿어마지 않는다.

참고 문헌

- [1] P.E. Jackson and C.D. Stubbs, "A study of multiaccess computer communications," *AFIPS Conf. Proc.*, 34, pp. 491-504, 1969
- [2] A.S. Tanenbaum, *Computer Networks*: Prentice-Hall, 1981.
- [3] H. Frank and W. Chou, "Topological optimization of computer networks," *Proc. IEEE*, vol. 60, pp. 1385-1397, Nov. 1972.

- [4] R.R. Boostyn & Frank, "Large scale network topological optimization," *IEEE Trans. Commun.*, vol. COM-25, pp. 29-47, Jan. 1977.
- [5] M. Schwartz and J.E. Stern, "Routing techniques used in computer communication network," *IEEE Trans. Commun.*, vol. COM-28, pp. 539-552, Apr. 1980.
- [6] R. Gallager, "An optimal routing algorithm using distributed computation," *IEEE Trans. Commun.*, vol. COM-25, pp. 73-85, Jan. 1977.
- [7] L. Pouzin, "Methods, tools and observations on flow control in packet-switched data networks," *IEEE Trans. Commun.*, vol. COM-29, no. 4, pp. 413-426, 1981.
- [8] M. Gerlá and L. Kleinlock, "Flow control: A comparative survey," *IEEE Trans. Commun.*, vol. COM-28, no. 4, pp. 553-574, 1980.
- [9] V. Ahuja, "Routing and flows control in SNA," *IBM Syst. J.*, vol. 18, no. 2, pp. 298-314, 1979.
- [10] CCITT Recommendations X.21, X.21 bis, X.3, X.28, X.29, X.25, ammended at Geneva, 1980.
- [11] Telenet X.25 Documentation Service, GTE Telenet Co.
- [12] Schriber, T.J., *Simulation Using GPSS*. John Wiley & Sons, 1974.
- [13] G. Gordon, *System Simulation*. Prentice-Hall, 1978.
- [14] Claudio Carreli and D.J. Roche, "CCITT languages for SPC switching systems," *IEEE Trans. Commun.*, vol. COM-30, no. 6, pp. 1304-1309, June 1982. ***

略語解説

TIC REP (toll incoming repeater 市外入리피터 (交換))

A型 端局에 설치된 市外台로 부터의 出中繼 接續에 사용하는 리피터 (REP)이다.

TIDF (toll intermediate distributing frame 市外用 中繼配線盤 (交換))

市外局에 있어서 各種 中繼線裝置를 市外台 등에 수용하는 경우의 接續, 負荷配分 등을 위하여 裝置와 台間에 설치되는 配線盤이다.

TILS (toll incoming and local switch 市外着信 市内併合交換機 (通信網))

市外着信交換機 (TIS)와 市内交換機 (LS)의 기능을 병합한 交換機이다.

TIM (timing and dating circuit 日時作成回路 (電信))

加入電信의 自動交換記錄裝置 (ASR)에 있어서 通話終了의 日時分을 레코오더 (REC)에 傳送하는 裝置이다.

TIMS (toll incoming and tandem switch 市外着信 中繼併合交換機 (通信網))

市外着信交換機 (TIS)와 탄덤交換機 (MS)의 기능을 병합한 交換機이다.

TIS (toll incoming switch 市外着信交換機 (通信網))

集中局階位에 설치하고 그 구역에 着信하는 市外呼를 端局에 배분하여 接속하는 交換機이다.

TKAMP (talkie amplifier 토오키 增幅裝置 (交換))
各種 토오키 서어비스에 사용하는 增幅裝置로, 토오키 音源을 他局에서 받는 경우 등에 쓰인다.

TKE (talkie equipment 토오키裝置 (交換))
各種 토오키 서어비스에 사용하는 錄音再生裝置로, 有料서어비스, 通知서어비스, 發呼局識別用 등에 대해서 각종이 있다.

TKT (talkie trunk 토오키 트렁크 (交換))
토오키 서어비스를 하는 트렁크로, 氣象, 時報서어비스 등 有料토오키 서어비스 回線用과 空番號通知 등 無料토오키 서어비스 回線用的 트렁크가 있다.