論 文

공중망을 이용한 화일전송 프로토콜의 설계 및 구현에 관한 연구

正會員 李 仁 行* 正會員 梁 海 權** 正會員 金 東 龍***

Design and Implementation of File Transfer Protocol Through Public Network

In Haeng LEE,* Hae Kwon YANG**, Dong Yong KIM*** *Regular Members*

要 約 이 연구는 공중망에 연결된 PC들 사이의 파일전송프로토콜의 설계및 구현에 관한 것이다.
3B20S/UNIX을 호스트컴퓨터로 하여 공중망에 접속된 MS-DOS하의 PC들 사이에 파일처리시스템을 구현한 것으로 이때의 프로토콜로는 파일전송을 지원하는 Kermit프로토콜에 윈도우기능및 파일관련 기능을 갖도록 확장한 것이다. 사용자들은 PC로부터 호스트컴퓨터로 데이터를 송신하고 호스트컴퓨터의 파일을 수신, 조회, 삭제할 수 있게 된다.

ABSTRACT This paper presents the design and implementation of a File Transfer Protocol between PC's which are connected through public network. The file handling system was implemented under 3B20S/UNIX which has connections to IBM-PC under MS-DOS. In designing the Protocol we use Kermit Protocol which supports a file transfer between several kinds of PC's, and we extended the Kermit Protocol to have window function and file managing system.

Users prepare data at PC and are able to send it to host computer and can receive, inquire, and delete file in host computer.

I. 서 론

컴퓨터 분야의 발전으로 자원을 공유하거나 상호 정보 교환을 위해 많은 방법들이 시도되고 있으며, 여러 형태의 네트워크가 구성되어 있다. 특히 호스트와 연결하여 사용하는 PC의 이용은 자원의 이용율을 향상시켜 준다. 이에 본 논문에서는 IBM-PC 호환기종/MS-DOS를 사용하는 PC들이 공중 패켓망과 호스트 시스템의 사용자가 되고, 각 PC들 간에 전달해야 할 화일들을 보관하는 중간 매체로 호스트 시스템을 사용하였다. 사용자들은 필요한 화일을 PC에서 작성하여 공중 패켓망을 통하여 호스트 시스템으

* (株)韓國데이타통신
DACOM Engineer Division
** 群山大學 情報通信工學科
Kunsan National Univ.
*** 全北大學校 工科大學 電氣工學科
Dept. of Electrical Engineering, Chonbuk National Univ.
論文番號 : 90-20(接受1989. 11. 7)

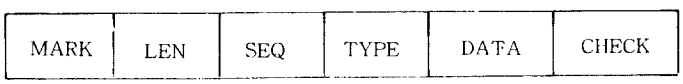
로 송신하고, 호스트 시스템에 저장되어 있는 자신의 화일들을 PC로 수신해 올 수도 있고, 삭제, 조회할 수 있도록 구현하였다.

또한 실제 운영에서 발생할 수 있는 여러장애 요인에 대한 처리 기능이 추가되었으며 화일전송 프로토콜은 전송 효율을 고려하여 기존의 PC용 통신 패키지에서 일반적으로 사용하는 KERMIT 프로토콜을 수정하여 사용하였다.

Ⅱ. Kermit Protocol 분석 및 수정

1) Kermit 프로토콜의 특징

화일 전송을 위한 데이타는 패켓을 단위로 하여 주고 받으며(그림 1) 패켓의 최대 길이는 96 바이트이다.



MARK : 패켓의 시작 바이트, SOH(Start-of-Header)
LEN : 이 영역뒤의 패켓의 바이트 수(최대 94)
SEQ : 패켓의 sequence 숫자 (modulo-64)
TYPE : 패켓의 종류
'D' Data
'Y' Acknowledge (ACK)
'N' Negative Acknowledge (NAK)
'S' Send initiate (exchange parameters)
'B' Break Transmission (EOT)
'F' File Header
'Z' End of file (EOF)
'E' Error
DATA : 패켓의 실제 내용
CHECK : 에러 검출에 사용하는 block check 값

Fig. 1 패켓의 형태

기본적인 Kermit 프로토콜은 화일의 송신단에서 여러가지 패켓을 보내며, 수신단은 각각의 패켓에 대한 ACK나 NAK 패켓을 보냄으로써 응답해야 한다. (Stop and Wait ARQ 방식) 즉 송신단은 하나의 패켓을 전송한 뒤 ACK를 받을 때까지 기다려야 한다. 수신단의 응답이 송신단에 도착할 때까지 다른 데이타 패켓을 보낼 수 없다.

2) Kermit 프로토콜의 수정

Stop-and-Wait ARQ 방식을 사용하는 KERMIT 프로토콜은 공중패켓망을 통하여 화일 전송을 하는 경우, 실제 화일전송 속도는 선로속도의 약 19% 밖에는 되지 않는다. 이러한 비효율적인 면을 개선하기 위해 기본적인 Kermit 프로토콜에 두가지 확장이 가능하다. 첫번째 방법은 최대 패켓의 길이를 증가시키는 것이다. 이러한 확장은 half duplex 라인이나 잡음이 없고 wait가 없는 (wait-less) 케이블을 통해 송신측과 수신측이 직접적으로 연결되어 있을 때 가장 잘 동작할 수 있다. 그러나 이 방법은 패켓을 재전송할 경우 시간이 걸리기 때문에 에러에 의해 크게 영향을 받는다. 두번째 방법은 sliding window 기능을 갖도록 확장하는 것이다. 이 방법은 패켓을 연속적으로 전송하여 송신측이 window 크기라는 어떤 interval내에서 응답하도록 하는 것이다. 이러한 방법은 많은 에러가 있거나 network delay가 있는 full-duplex 시스템

에서 Kermit의 효율을 크게 향상시킨다.

본 논문에서는 비효율적인 면을 개선하기 위하여 silding window 확장을 도입하여 프로토콜을 수정하였다. 즉 window 크기가 1로 정해져 있는 stop-and wait ARQ 방식을 continuous ARQ 방식을 사용하도록 프로토콜을 수정하여 효율을 증대시켰다. continuous ARQ 중에 일반적인 것에는 go-back-N ARQ와 selective-repeat ARQ 가 있다. 두 ARQ 방식중에 selective repeat ARQ 방식은 go-back-N ARQ 방식을 사용할 경우보다 수신단에는 각 패켓을 저장해 놓을 영역이 많이 필요하고 적절한 순서로 패켓을 정렬하는 논리가 있어야 하며, 송신단에는 수순에 관계없이 자유자재로 패켓을 재전송할 수 있는 복잡한 논리가 필요하다. 이러한 복잡성 때문에 go-back-N 알고리즘을 일반적으로 많이 사용하므로 본 논문에서도 Kermit의 프로토콜을 수정하는데 있어서 go-back-N ARQ를 채택하였다. 사용자는 네트워크와 전송 선로의 상태에 따라 window 크기를 1~8로 지정하여 사용할 수 있다.

Send-Init(S) 패켓의 데이타 영역에는 configuration 정보에 관한 문자열이 포함된다. Send-Init(S) 패켓을 위한 ACK 패켓의 데이타 영역에는 수신단의 configuration 파라미터가 들어가야 한다.

MAXL	TIME	NPAD	PADC	EOL	QCTL	QBIN	CHKT	REPT	WIND	reserved

Fig. 2 The data field of a 'S'(Send-Init) packet

그림 2의 데이타 영역의 configuration 파라미터의 의미는 다음과 같다(A가 송신단이고, B는 수신단이다).

- MAXL : A가 받기를 원하는 최대 패켓 길이이다. 94까지 가능하다. B는 A에게 보내고자 하는 최대 패켓 길이를 A에게 보낸다. 이것은 버퍼 크기, 전송매체의 조건 등에 의해 조정될 수 있다.
- TIME : B가 A로부터 패켓을 기다리는 동안의 time-out 되어야 하는 시간을 말하는 것으로서, A가 B에게 먼저 보낸다. B는 A가 time-out 시켜야 하는 시간을 A에게 보내준다. 이것은 양측으로 하여금 서로 다른 선로 속도 또는 timing 문제를 발생시키는 제반 요소들을 조화있게 조절할 수 있도록 하는 데 도움이 된다.
- NPAD : A가 원하는, 각각의 들어오는 패켓에 선행되어 붙을 수 있는 padding 문자의 갯수이다. B도 마찬가지이다. 이는 반이중 시스템에서 전송방향의 변경을 위한 시간만큼의 전송지연을 위한 것이다.
- PADC : padding을 시키고자 하는 제어 문자이다. ctl( )에 의해 변형되어 프린트 가능하게 바뀐다. 주로 null(ASCII 0)을 사용하지만 시스템에 따라 DEL (ASCII 127)을 사용하기도 한다. npad 를 0로 한다면 이 영역은 무시된다.
- EOL : 들어오는 패켓을 종결지우는 문자이다. 터미날 입력을 위한 라인 종결문자를 요구하는 대부분의 시스템이 CR을 허용한다. ctl( )을 사용하여 프린트 가능하게 변형시킨다.
- QCTL : 제어 문자임을 말해주기 위한 prefix 문자이다.
  default로 '#'이다.
- QBIN : 8번째 비트가 세트되어 있는 문자임을 알려주는 prefix 문자이다. Parity 비트가 데이타로 사용될 수 없는 2진 화일의전송시에 사용된다.

• CHKT : 에러 검사 종류이다.
  1) 1 : 1-character-checksum
  2) 2 : 2-character-checksum
  3) 3 : 3-character-checksum(CRC--CCITT)
  default는 "1"이다.

• REPT : 반복되는 문자임을 말해주는 prefix 문자이다. 그 범위는 ASCII 33-62, 또는 96-126이다.

• WIND : 화일의 데이타를 전송하는 중에 window 크기이다. 0부터 8까지 사용자가 크기를 정할 수 있으며, 정한 window 크기만큼은 ACK를 받지 않고 계속하여 패켓을 전송할 수 있다.
  default는 "1"이다.

Send-Init 영역은 선택적이므로, 그 데이타 영역이 모두 비어있을 수도 있으며 Default Send-Init 영역은 다음과 같다.

maxl : 80
npad : 0, no padding
padc : 0 (null)
eol : CR (carriage return)
qctl: "#"
qbin : none, 8비트 quoting을 하지 않는다.
chkt : "1", single-character-checksum
rept : repeat count processing을 하지 않는다.
wind : "1"

Ⅲ. 구 현

1) 네트워크 구성

여러 곳에 산재해 있는 140여 개의 PC들이 공중전화망과 공중패켓망 (DACOMNET)의 5개 노드와 15개의 access point를 통하여 연결된다.

호스트(3B20S) 시스템은 2개의 R-PAD를 사용하여 공중 패켓망에 접속되어 있다(Fig. 3).

네트워크의 S/W는 호스트(2B20S) 시스템과 PC에 각각 구현되어 있다.

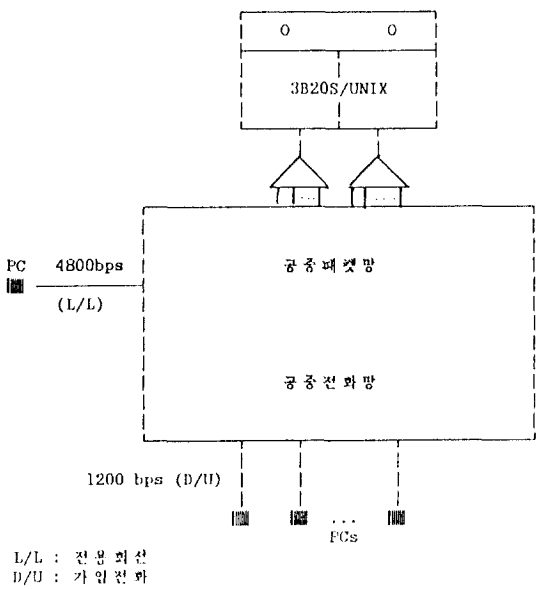


Fig. 3 네트워크 구성

2) PC에서의 시스템 구성

사용자들이 사용하는 PC는 IBM-PC 호환기종이며, 화일전송 기능과 사용자 interface 기능이 제공된다.

A. 화일전송 기능

네트워크를 통하여 연결되어 있는 시스템들 간에 화일을 주고 받기 위해 같은 프로토콜 (Kermit 프로토콜)을 갖는 화일 전송 프로그램이 동작한다.

호스트와 PC를 연결시킨 경우에는 호스트 시스템의 부하를 고려하여 호스트에서 timeout 값을 지정하고, timeout 검사도 호스트에서 하는 것이 일반적이다. 그러나 위와 같이 timeout을 하면 본 네트워크에서는 가입전화를 통하여 공중 패켓망과 연결하여 호스트에 들어가게 되므로 전송 중간에 hang-up이 걸리거나 또는 시스템의

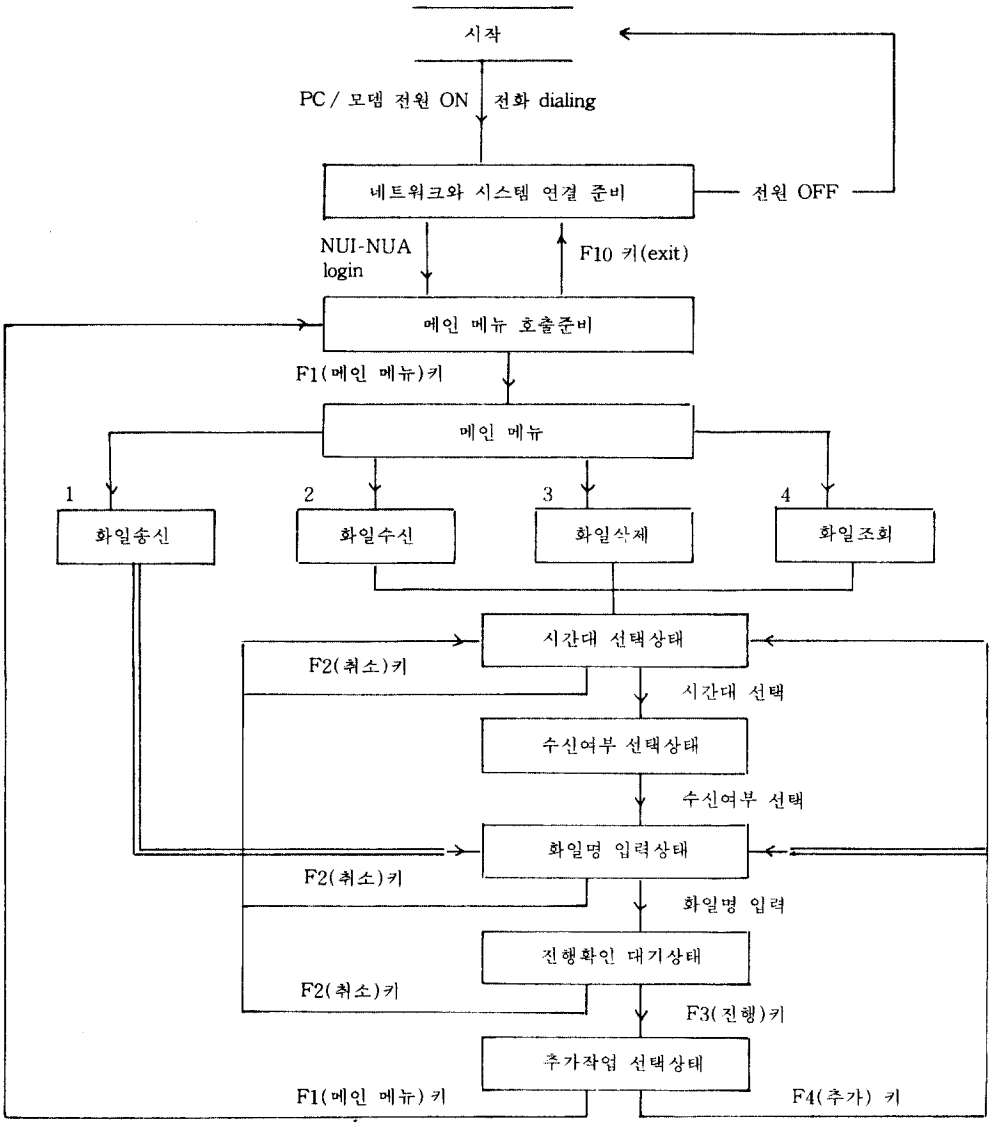


Fig. 4 사용자 interface 기능 흐름도

프로세스에 이상이 생기든지 하는 여러 경우에 대해 PC는 아무런 행동을 취하지 못하게 된다. 그래서 PC에서 timeout을 검사하여 처리하도록 수정되었다.

B 사용자 interface 기능

사용자 interface는 PC에서 모두 제공하고, 호스트 시스템은 login을 한 이후에는 tty 포트를 통하여 들어오는 패켓 내용에 따라서 모든 작업을 수행하고, 사용자의 명령을 직접 받아들이지 않는 모드로 동작하게 된다. 사용자 interface 기능은 메뉴 처리 방식으로 되어 있고, 네트워크 및 시스템과 연결을 위한 화면과 화일처리를 위한 메뉴 화면을 제공한다.

화일처리를 위한 메뉴 화면에서는 화일송신, 화일수신, 화일삭제, 화일조회를 선택할 수 있

다. 화일수신, 화일삭제, 화일조회는 시간대, 화일 수신 여부, 화일명으로 사용자가 처리하고자 하는 화일들을 지정할 수 있다. 사용자 interface 기능의 흐름은 그림 4.와 같다.

3) 호스트 시스템에서의 시스템 구성

호스트 시스템은 3B20S 시스템이며, 화일전송 기능과 화일처리 기능, 화일 정보기록 기능, back-up 기능을 갖는다.

A. 3B20S 시스템 안에서의 화일 구성

호스트 시스템내의 화일 구성은 크게 세가지로 구분할 수 있다(그림 5).

(a) DIR1 : 실제 PC들이 사용하는 디렉토리

(b) DIR2 : DIR1 디렉토리 밑의 화일에 손실이 생긴 경우 복구하는 데에 사용.

(c) DIR3 : PC들이 사용하는 실행 가능한 화일들과 시스템 관리를 하는데 사용하는 화일

모든 PC들의 login 명은 "pcs"+"수신처 코드"로 이루어지며, 각각의 HOME 디렉토리 밑에는 각 PC에게 발송된 화일을 관리하는 화일인 "fillst" 화일이 있다.

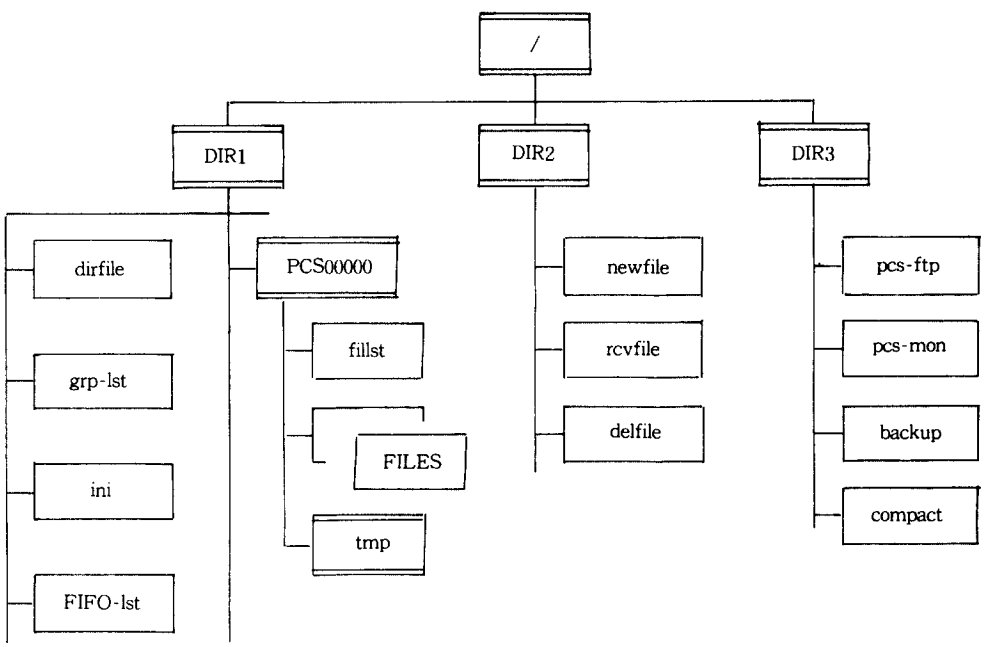


Fig 5 3B20S 시스템 안에서의 화일 구성도

B. 전송 화일명의 구성

PC에서 전송하는 화일명은 그 화일의 종류와 수신처로 구성되며 화일종류는 화일명에 따라 세가지로 구분된다(그림 6).

(a) 개별적인 수신처 코드가 들어 있는 경우 : 해당 수신처 송신

(b) 화일명에 group명을 지정하는 경우 : "grp_lst" 화일에 의해 송신.

(c) 수신처 코드부분이 dar99로 지정된 경우 : 데이타 내의 수신처로 송신

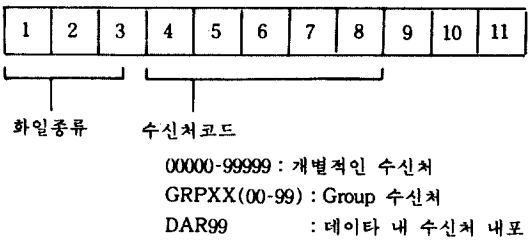


Fig. 6 시스템에서 사용하는 화일명의 구성

C. 화일전송 기능

화일전송 기능은 DIR3하에 구현되어 있으며, 그 포로토콜은 PC에서 동작하는 화일전송 기능의 프로토콜과 같고, 항상 server로 동작하여 PC에서 보내는 패켓의 내용에 따라 동작하게 된다.

D. 화일처리 기능

화일처리 기능은 DIR3하에 구현되어 있다.

화일처리 기능은 화일전송 기능에서 처리한 패켓 내용 (그림 7)을 분석하여 PC에서 선택한 다음의 기능들을 처리한다 ; 화일송신, 화일수신, 화일삭제, 화일조회

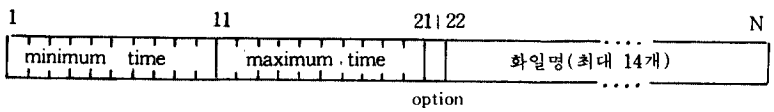


Fig. 7 화일관리를 위한 패켓의 데이터 형식

E. 화일정보 기록기능

화일정보 기록기능은 DIR3 하에 구현되어 있으며 daemon 프로세스이다. 이 기능은 nemed_pipe인 DIR1하의 "FIFO_list"를 읽어 그 pipe에 내용이 있으면 처리를 하여 해당 수신처의 "fillst" 화일에 내용을 덧붙인다.

FIFO_list의 구조는 다음과 같다.

```
struct sendinfo{
  char destination[CODLEN+20] ;
  struct filelist flst ;
  struct stat statbuf ;
  }
```

F. Back-up 기능

Back-up 기능은 DIR1의 디렉토리에 손상을 입은 경우에 복구를 하기 위하여 구현되었다.

G. "fillst" 화일의 처리

fillst는 각각의 login 디렉토리에 존재하는 화일에 대한 리스트이며, 그 구조는 다음과 같다.

```
struct file list{
  unsigned short exist ;
  char           f_name[DIRSIZ];
  Char           f_from[9];
  long           f_time;
  unsigned short f_access;
  int            f_index;
};
```

새로운 화일을 수신하여 fillst 화일의 내용을 갱신하는 경우, fillst의 맨 끝에 새로운 화일의 내용을 첨가 시킨다.

Ⅳ. 실험

1) 화일 전송 시험

프로토콜에 대한 sliding window 확장은 그들 자체의 프로토콜을 갖는 네트워크 상에서 통신할 때 매우 효과적이며, 그러한 네트워크에서 송수신 할 때 얼마동안의 지연(delay)이 생긴다. 이러한 지연동안 Kermit의 Stop-and Wait ARQ

프로토콜은 각 패켓에 대한 ACK를 기다린다. 그러나 Kermit의 continuous ARQ 하에서 송신하는 ACK를 받기 전에 지연시간 동안 8개 까지의 패켓을 보낼 수 있다.

공중 패켓망을 통하여 두 시스템간 화일전송을 시험한 결과는 표 1과 같다.

표 1 Go-back-N ARQ 방식을 사용하여 공중패켓망을 통한 화일전송 시험 결과

(전송선로의 속도는 1200 bps)

화일종류	시험조건	화일전송속도	전송효율	개선율
text	Window 크기1	230 bps	0.19	1.0
	Window 크기2	330 bps	0.28	1.4
	Window 크기4	460 bps	0.38	2.0
	Window 크기6	530 bps	0.44	2.3
	Window 크기8	580 bps	0.48	2.5
binary	Window 크기1	150 bps	0.13	1.0
	Window 크기2	240 bps	0.20	1.6
	Window 크기4	310 bps	0.26	2.1
	Window 크기6	360 bps	0.30	2.4
	Window 크기8	390 bps	0.33	2.6

2) 3B2OS 시스템 성능 문제

PC간의 화일 교환시 user CPU 점유가 크기 때문에 다수의 사용자가 동시에 사용할 경우 시스템 병목 현상이 나타났다(표 2).

서비스의 최대 정점 동시 PC수는 15대이며, 만일 3B2OS 시스템을 전용으로 사용할 경우라도 20대의 PC가 동시 사용시 정상 작업이 어렵다. 30대 이상의 PC를 동시 유치할 경우 서비스 시스템을 Super-mini에서 대형 시스템으로 변경이 불가피하다.

V. 결 론

본 논문의 PC 네트워크는 동일기종의 PC를 사용하여 사용자와의 interface를 3B2OS 시스템과 무관하게 PC에서 제공하였다. 사용자는 PC하에서 자료를 준비하여 메뉴 화면에서 제공하는 간단한 입력만을 하면 화일전송 기능과 화일처리 기능이 필요한 작업을 모두 처리하도록 시스템이 구성되어 있어서, 사용자가 시스템에 관한 지식

표 2 가입자 증가에 따른 CPU 점유 현황

CPU 점유현황 / 총사용 PC수	CUP 점유 내역				비 고
	% user	% sys	% wio	% icle	
6	22	10	0	67	* user : 사용자 CPU 점유량 * sys : 시스템 CPU 점유량 * wio : Disk I/O 발생시 CPU idle * Idle : CPU Idle ** CPU Idle이 10% 이상일 경우에 정상 작업 가능
10	37	17	1	45	
15	60	23	0	16	
17	66	23	0	12	
19	70	27	0	3	
20이상	72	27	1	0	

측정시 일반 사용자수 : 5명

이 없어도 쉽게 원하는 일을 수행할 수 있도록 설계되어 있다. 이렇게 함으로써 필요한 데이타만을 전송하여 기존의 on-line 작업보다 전체적인 통신망을 감소시킬 수 있었다.

한편 Stop-and-Wait ARQ 방식으로 동작하는 기존의 화일전송 프로토콜에 Go-back-N ARQ 방식을 채택하여 전송효율을 증대시켰다. 전송효율을 증가시키는 방안으로써 최대 패켓의 길이를 증가시키거나 continuous ARQ 방식중에서 selectiverepeat ARQ 방식을 도입하여 수정하는 방법을 차후에 고려하여야 할 것이다.

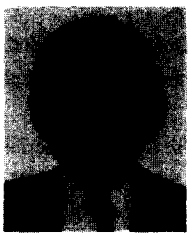
3B2OS 시스템에서는 도착하는 화일들의 내용

을 관리하기 위한 프로세스를 따로 둠으로써 여러개의 프로세스가 하나의 화일을 access하는 문제를 해결하였다. 또한 시스템에서 발생할 수 있는 데이타 손실을 대비하여 back-up 시스템을 구축하였다.

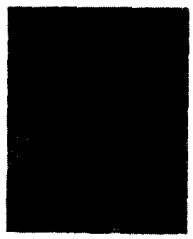
그러나 시스템의 성능(performance) 때문에 동시에 작업할 수 있는 사용자 수가 제약되었다. 따라서 이를 위해 port 수를 제한하거나 사용자의 시간대 조정, 대형시스템으로의 변경등의 방안이 고려되어야 한다.

參 考 文 獻

1. William Stallings, Data and Computer Communication, Macmillan Publishing Company, 1985.
2. Andrew S. Tanenbum, Computer Networks, Prentice-Hall, Inc., 1981.
3. IBM Technical Reference (Personal Computer XT Hardware Reference Library), 1983.
4. Microsoft, MS-DOS Programmer's Reference Manual.
5. AT & T, UNIX System V Programmer's Reference Manual 3B 20 Computers
6. AT & T, UNIX System V User Reference Manual 3B 20 Computers.
7. Robert Lafore, Assembly Language Primer for the IBM PC & XT, The Wait Group, 1984.
8. Leo J. Scanlon, Assembly Language Programming with IBM PC AT, Brady Communications Company, Inc., 1986.
9. Kernighan Ritchie, The C Programming Language, Prentice-hall software Series, 1978.
10. Frank da Cruz, KERMIT Protocol Manual Fifth Edition, Colunbia University Center for Computing Activities, 1984.
11. Frank da Cruz, KERMIT User's Guide Fifth Edition, Colunbia University Center for Computing Activities, 1984.
12. Joe Campbell, C Programmer's Guide to Serial Communications, HOWARD W. SAMS & COMPANY, 1987.
13. William Stallings, "Here is one way to get a close estimate of a data link's efficiency", Data Communication, October
14. Malcolm G. Lane, Data Communications Software Design, Boyd & Fraser Publishing Company, 1985.
15. Keith Hariland & Ben Salama, Unix System Programming, Addison-Wesley publishing company, 1987.
16. DACOM_NET 접속 기준(Ⅰ) (문자형 단말기), 한국데이타통신(주), 1988.
17. DACOM_NET 접속 기준(Ⅱ) (패켓형 단말기), 한국데이타통신(주), 1988.
18. Brian W. Kernighan / Rob Pike, The UNIX Progrramming Environment, Prentice-Hall, 1984.
19. AT & T, Internal UNIX System Calls and Libraries Using C Language-101 Student Guide, 1986.
20. Stephen Pruta, Advanced UNIX-A Programmer's Guide, Howard W. Sams & company, 1985.
21. Frank da Cruz and Bill Catchings, A File Transfer Protocol for Universities-Partl. Design considerations and specifications, Byte, June 1984.
22. Frank da Cruz and Bill Catchings, A File Transfer Protocol for Universities-Part2. States and Transitions, heuristic rules, and exampls, Byte, July 1984
23. Joe Campbell, The RS-232 Solution, SYBEX Inc, 1984.



李 仁 行(In Haeng LEE) 正會員
1954年 1月 1日生
1978年：서울大學校 電子工學科 卒業 (工學士)
1989年 2月：漢陽大學校 大學院 卒業 (工學碩士)
1978年 1月～1982年12月(株)豊山 勤務
1983年 1月～現在：(株)韓國데이타통신 技術本部 技術기준部長



梁 海 權(Hae Kwon YANG) 正會員
1953年 7月17日生
1976年 2月：서울大學校 電氣工學科 卒業(工學士)
1983年 8月：蔚山大學校 大學院 卒業 (工學碩士)
1986年 3月～現在：全北大學校 電氣工學科 博士課程 在學
1985年 9月～1987年 2月：全北産業大學 電子計算學科 專任講師
1987年 3月～現在：群山大學 情報通信工學科 助教授

金 東 龍(Dong Yong KIM) 正會員
1945年 7月31日生
1963年 3月～1967年 2月：全北大學校工科大學 電氣工學科 電氣工學 學士
1971年 3月～1973年 8月：全北大學校大學院 電子工學 博士 碩
1979年 9月～1984年：카나다 마니토바 大學校 電氣工學科 電子工學 博士
1979年～1985年：카나다 마니토바大學校 電氣工學科 研究院
1985年～現在：全北大學校 工科大學 電氣工學科 副教授
總務處 考試課 回路理論 考試委員
大韓電氣協會 教育委員, 韓國通信學會 編輯委員
1987年～現在：全北大學校 電子計算所 所長
科學技術處 教育電算網 委員