

망 분리 환경에서 스토리지 기반의 데이터 공유 시스템 설계 및 구현

정희원 조인휘*, 이석순*

Design and Implementation of Storage-based Data Sharing System in the Separate Network Environment

Inwhee Joe*, Suksoon Lee* *Regular Members*

요 약

본 논문에서는 망이 분리된 환경에서 스토리지 기반의 데이터 공유 시스템을 구축하는 경우 데이터 전송의 효율을 개선하기 위하여 어떻게 시스템을 설계하고 구현해야 하는지에 대해서 제안하고자 한다. 기존 시스템은 사용자 PC에서 수신된 단위 패킷별로 파일을 생성하고 스토리지를 통하여 생성된 파일의 수만큼 상대편 망으로 전송을 한다. 스토리지를 통해서 다수의 파일을 전송하는 경우 파일에 할당된 블록 외에 추가적으로 불필요한 블록도 함께 읽어야 함으로써 전송 시 비효율이 발생했다. 제안 시스템은 이러한 비효율을 제거하기 위해서 동일한 시점에 수신되는 패킷에 대해서 스냅 샷 개념을 도입하여 하나의 파일에 저장하고 생성되는 파일의 수를 최대한 줄이고 블록 사이즈에 최적화 되도록 하여 파일을 생성했다. 이렇게 생성된 파일은 불필요한 블록을 읽어 들이는 것을 최소화하여 전송함으로써 기존의 방식보다 사용자 응답속도가 개선되었다.

Key Words : Storage, Data Sharing, Separate Network, Snapshot

ABSTRACT

In this paper, we propose the design and implementation of the storage-based data sharing system in the separate network environment to improve efficiency of data transmission. The previous system generates files per received packet and transmit files to another network through storage. This system causes inefficiency by reading unnecessary blocks, when it transmits a number of files through storage. Our proposed system deals with this inefficiency by adopting concept of snapshot. Consequently, we create one file with snapshot so that the number of files can be reduced and the file size can be optimized according to the block size. The proposed system improves the response time significantly with the minimized reading of unnecessary blocks, compared to the previous system.

I. 서 론

일반적으로 망이 완전히 분리된 환경에서 인터넷 망으로부터 획득한 자료를 내부 업무 망으로 가져오기 위해서 보안 USB를 이용하게 된다. 보안 USB를 이용하는 이유는 그 외의 다른 외부 저장매체는 원칙

적으로 사용을 할 수 없기 때문이다. 보안 USB를 이용하는 경우 매번 USB를 대상 PC에 연결해야 하는 번거로움이 있으며 USB를 분실하게 되면 이로 인한 데이터 유출이 발생할 소지가 있다. 그리고 데이터에 대한 암호·복호화 모듈이 보안 USB 자체에 탑재된 것도 아직 없는 것으로 보인다. 이러한 보안 USB의 단

* 한양대학교 컴퓨터공학부 이동네트워크 연구실 (iwjoe@hanyang.ac.kr)

논문번호 : KICS2010-08-422, 접수일자 : 2010년 8월 30일, 최종논문접수일자 : 2011년 4월 21일

점을 보완하기 위하여 내부 업무 망과 인터넷망간의 데이터를 공유할 수 있도록 하는 시스템이 구축되기 시작되었다. 이렇게 되면 보안 USB를 사용하지 않고도 인터넷 망 PC의 데이터를 내부 업무 망 PC로 공유할 수 있게 된다. 현재 출시된 데이터 공유 시스템의 아키텍처는 비슷한 형태인데 크게 분류를 하면 표 1과 같다. 표 1에서 소개된 2가지 방법 중 게이트웨이 서버에 랜 카드나 1394 포트를 이용하여 구축한 경우는 각각의 망이 서로 연결되었다고 볼 수 있으므로 인터넷망의 게이트웨이 서버에서 해킹이 발생하는 경우 내부 업무망의 게이트웨이 서버로 접근할 수 있는 경로를 확보하게 된다.

따라서 각 망에 연결이 가능해지게 되므로 망을 분리한 목적을 잃어버리게 된다. 반면 스토리지를 이용해서 데이터를 공유하는 경우는 인터넷망의 게이트웨이 서버를 해킹한다 하더라도 서버가 분리되었기 때문에 콘솔을 이용하여 내부 업무망의 서버로 접근하여 조작할 수 있는 방법이 없다. 또한 SAN 환경의 스토리지를 활용하여 전송 시 Fiber Channel을 이용하게 되므로 보안성을 확보할 수 있다. 따라서 보안적인 측면에서 보면 스토리지를 활용하여 구축하는 경우가 가장 보안성이 높다고 할 수 있다.

본 논문에서는 이와 같이 스토리지를 기반으로 하여 데이터 공유 시스템을 구축하는 경우에 기존의 구축된 시스템에 발생하는 파일 전송 시의 비효율을 살펴보고 기존의 방식보다 효율적으로 데이터를 전송할 수 있는 방안에 대해서 설계하고 구현하고자 한다.

표 1. 데이터 공유 시스템 구축 방식 분류

번호	분류
1	인터넷망의 게이트웨이 서버와 내부 업무망의 게이트웨이 서버 간 스토리지를 이용해서 연결
2	인터넷망의 게이트웨이 서버와 내부 업무망의 게이트웨이 서버 간 랜카드/1394 포트로 연결

II. 제안 시스템 설계

2.1 제안 시스템 개요

기존 시스템의 문제점을 살펴보면 사용자 PC에서 이벤트가 발생할 때마다 별도로 게이트웨이 서버와 연결을 맺어서 데이터를 전송하고 연결을 끊어 버리기 때문에 매번 연결을 맺어야 하는 단점이 있다. 또한 사용자 PC에서 수신된 패킷을 가지고 수신된 단위 패킷별로 게이트웨이 서버에 파일을 생성하게 된다.

동시 접속자 수가 많을 경우 사용자 PC에서 사용자 PC로 오고가는 패킷의 수가 증가하게 되는데 이 때문에 생성되는 파일의 수도 급격히 증가한다. 이로 인해서 전송해야 할 파일의 개수가 많아지게 되어 결과적으로 성능이 느려지게 되는 이유가 된다. 따라서 사용자 입장에서 보면 어떤 액션을 취했을 경우 응답속도가 늦어지게 되는 것이다. 원인을 살펴보면 크게 두 가지가 있는데 그 중 하나가 후지쯔에서 제공되는 XL-Datamover 솔루션의 특성 때문이며 다른 하나는 connection 관리 때문이다.

데이터 공유 시스템의 파일 전송에 사용되는 XL-Datamover 솔루션 command 중 mtfSEND는 64KB 블록 단위로 파일을 읽어서 전송하게 된다.¹⁾ 즉 파일의 크기가 64KB가 안되더라도 최소 64KB의 블록을 읽어야 한다는 의미가 된다. 따라서 파일 전송 시 읽어 들이는 블록의 비효율이 존재하지 않도록 파일의 크기를 최적화해야 할 것이다. 또한 mtfSEND는 전송할 파일을 개별로 하나씩 지정해서 상대편 스토리지의 특정 영역으로 이동시킨다. 그러므로 전송해야 할 파일이 하나만 존재한다면 문제가 될 것이 없지만 동시 사용자 수가 많아져서 전송해야 할 파일의 수가 많아진다면 매번 mtfSEND를 실행시키게 되고 결과적으로 mtfSEND 프로세스가 여러 번 생성되게 되며 system call 요청이 잦아지게 된다는 의미가 된다. 이는 시스템에 악영향을 미치게 되어 사용자 입장에서 응답속도가 늦어지게 되는 것과 직결된다.

따라서 데이터 전송 시 효율을 최대한 높으려면 표 2와 같은 방법으로 파일을 생성해야 가장 이상적인 모델이 될 것이다. 이를 위해서 본 시스템에서는 스냅샷 개념을 이용하여 일정 시간 단위로 모니터링 하면서 동 시간대에 들어온 패킷을 하나의 파일로 저장하여 생성되는 파일의 수를 최대한 줄이며 가능하면 파일을 64KB 또는 64KB 배수 단위의 크기로 생성될 수 있도록 한다. 이렇게 하면 기존보다 프로세스 생성을 최소화 할 수 있고 읽어 들이는 블록의 비효율을 최소화 할 수 있다. 또한 데이터 전송 시 raw device의 개수만큼 전송 채널이 존재하므로 각 채널을 효율

표 2. 수신된 패킷을 전송하기 위한 파일 생성 방법

번호	내용
1	파일 생성시 파일의 크기를 64KB 단위로 생성
2	파일 생성시 파일의 크기를 64KB의 배수로 생성
3	생성되는 파일의 최소화

1) Fujitsu Korea, Softek XL-Datamover User's Guide

적으로 사용하기 위해서 round-robin 방식을 이용하여 시스템을 구현한다.

2.2 제안 시스템 설계

내부 업무 망 PC와 인터넷 망 PC 사이에서 주고받는 패킷은 크게 2가지로 나눌 수 있다. 하나는 정보를 교환하기 위해서 주고받는 패킷이 있으며, 다른 하나는 파일을 송·수신하기 위해서 파일의 내용을 전송하는 패킷이 있다. 후자의 경우 패킷의 종류는 하나뿐이며 전자의 경우 패킷의 종류에는 표 3과 같은 것들이 있다.

본 시스템은 인터넷 망 PC 또는 내부 업무 망 PC로부터 수신된 패킷을 게이트웨이 서버 간 전송하기 위하여 그림 1과 같은 단위 패킷의 구조를 사용한다. 단위 패킷은 header 부분과 data 부분으로 나눌 수 있다. header 부분은 대칭키를 암호화한 부분 값을 이용해서 암호화)과 통신 헤더(대칭키로 암호화)로 구분되며 header의 크기는 고정되어 있다. data 부분은 상황에 따라서 유동적으로 변경될 수 있으며 대칭키로 암호화되어 있다. 내부 업무망의 게이트웨이 서버에서는 수신된 패킷을 하나의 파일에 append 방식으로 계속 덧붙여쓰기를 하여 인터넷망의 게이트웨이 서버로 파일을 전송한다.

인터넷망의 게이트웨이 서버에서 수신된 패킷을 각각 인터넷 망 PC로 전송하기 위해서 고정된 크기인 header 부분을 먼저 읽어서 앞으로 읽어야 할 데이터의 총 크기를 구하고 패킷을 전송할 대상 PC를 선별한다. 그리고 header 부분에서 구해진 데이터 크기만큼 계속 읽어서 선별된 인터넷 망 PC에 데이터를 전송하게 된다. 인터넷망의 게이트웨이 서버에서도 같은 방식으로 파일을 생성하며 생성된 파일을 내부 업무 망 게이트웨이 서버로 전송한다.

표 3. 패킷 종류

패킷 구분	설명
인증 요청	로그인시 사용
folder 정보 요청	내부 업무 망 PC에서 인터넷 망 PC의 공유 폴더/파일 정보를 요청하는데 사용
download 정보 요청	내부 업무 망 PC에서 인터넷 망 PC의 파일을 다운받기 위해서 사용

Header		대칭키로 암호화된 Data
비대칭키로 암호화된 대칭키	기타 통신 헤더	

그림 1. 파일 저장 구조

그림 2는 내부 업무 망과 인터넷망의 게이트웨이 서버가 동작하는 전체 개요를 나타내며 내부 업무 망 PC에서 인터넷망의 PC로 패킷을 전송하는 과정을 보여준다. 인터넷망의 PC는 listen 상태로 대기하며 인터넷망의 게이트웨이 서버에서 connect 요청을 하게 된다. 반면 내부 업무망의 게이트웨이 서버는 listen 상태로 대기하며 업무 망 PC에서 connect 요청을 하게 된다. 그림 2에서 업무 망 게이트웨이 서버의 데몬 프로세스는 3개의 스레드(thread)를 생성하며 각각의 스레드에서 하는 역할은 다음과 같다.

- 1) acceptThread에서는 내부 업무 망 PC에서 connect 요청이 들어오면 연결을 수락하고 Socket Pool 해당 오브젝트(구조체)를 포함하고 연결 상태를 관리하게 된다.
- 2) socketRecvThread에서는 Socket Pool의 오브젝트로부터 내부 업무 망 PC에서 수신된 패킷이 있는지 검사하여 인터넷 망으로 전송해야 하는 패킷을 파일에 저장하며, 전송 buffer에 내부 업무 망 PC로 전송해야 할 데이터가 있으면 패킷을 전송한다.
- 3) storageReadThread에서는 생성되는 파일을 관리하며, 인터넷 망으로부터 수신된 데이터가 있는지 확인하여 전송할 데이터가 있으면 Socket Pool에서 해당 오브젝트의 전송 buffer에 전송 데이터를 저장한다.
- 4) main loop에서는 인터넷 망으로 전송해야 하는 데이터(파일)가 있는지 주기적으로 확인하여 데이터가 존재하면 인터넷 망으로 전송한다.

그리고 인터넷망의 게이트웨이 서버에서 실행되는 데몬 프로세스는 2개의 스레드를 생성하고 1개는 필요시마다 스레드를 생성하며 각각의 스레드에서 하는 역할은 다음과 같다.

- 1) socketRecvThread에서는 Socket Pool을 검사하여 인터넷 망 PC로부터 수신된 패킷이 있으면

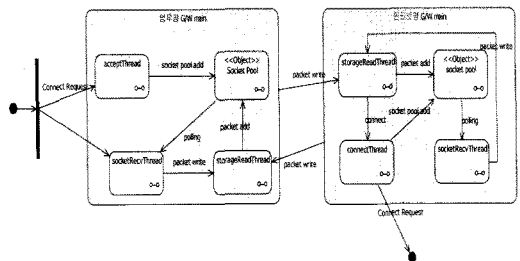


그림 2. 데몬 프로세스의 프로세스 구성

- 파일에 저장하고 전송 buffer에 전송할 데이터가 있으면 인터넷 망 PC로 데이터를 송신한다.
- 2) storageReadThread에서는 생성되는 파일을 관리 하며, 내부 업무 망으로부터 수신된 데이터가 있는지 확인하여 전송할 데이터가 있으면 해당 인터넷 망 PC와 연결을 맺고 데이터를 송신한다.
- 3) connectThread에서는 각각의 인터넷 망 PC와 최초 연결 시에 생성되며 연결을 시도하여 성공 하면 Socket Pool에 오브젝트를 포함하고 스레드가 종료된다.
- 4) main loop에서는 업무 망으로 전송해야 하는 데이터(파일)가 있는지 주기적으로 확인하여 데이터가 존재하면 업무 망으로 전송한다.

그림 3은 내부 업무 망 PC에서 사용자가 공유하고자 하는 파일의 목록을 확인한 뒤 인터넷 망 PC로부터 특정 파일을 요청하여 다운로드하는 흐름을 나타낸 것이다. 업무 망 게이트웨이 서버와 인터넷 망 게이트웨이 서버의 “수신 패킷 저장” 부분에서 동일한 시점에 수신된 패킷을 하나의 파일에 저장하게 된다. 전체적인 흐름은 다음과 같다.

- 1) 업무 망 PC에서 인터넷 망으로부터 다운로드하고자 하는 파일을 선택하여 다운로드 요청을 한다.
- 2) 업무 망 게이트웨이 서버에서 현재 연결이 유효한지 확인하여 유효하다면 업무 망 PC로부터 수신된 패킷을 파일로 저장한다.(동일한 시점에 수신된 패킷은 하나의 파일에 저장됨)
- 3) 업무 망 게이트웨이 서버에서 패킷이 저장된 파일을 인터넷망의 게이트웨이 서버로 전송한다.
- 4) 인터넷 망 게이트웨이 서버는 수신된 파일의 패킷을 분석하여 인터넷 망 PC로 전송한다.

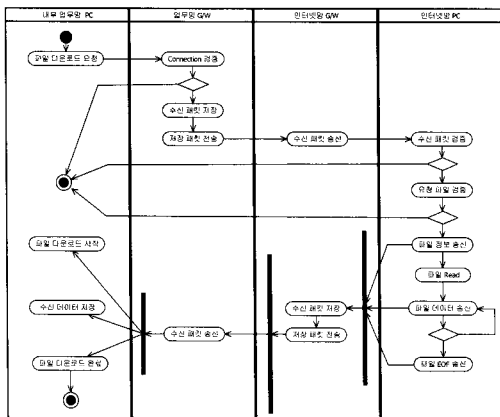


그림 3. 파일 전송 흐름

- 5) 인터넷 망 PC는 업무 망 PC부터 요청된 파일에 대해서 인터넷 망 게이트웨이 서버로 전송을 시작한다.
- 6) 인터넷 망 PC에서 수신된 패킷을 인터넷 망 게이트웨이 서버에 파일로 저장하고 해당 파일을 업무 망 게이트웨이 서버로 전송한다.(동일한 시점에 수신된 패킷은 하나의 파일에 저장됨)
- 7) 수신된 파일을 업무 망 게이트웨이 서버에서 분석하여 해당 업무 망 PC로 패킷을 전송한다.
- 8) 파일 전송을 마칠 때까지 5-7과정을 반복한다.

III. 구현 및 실험

3.1 데이터 공유 시스템 구현

개발 환경은 표 4와 같고 개발 언어는 서버 프로그램은 C/C++를 사용 하고, 클라이언트 프로그램은 VC++를 사용했다.

게이트웨이 서버는 Linux, Solaris등의 Unix machine 환경에서 동작하게 된다. 내부 업무망의 게이트웨이 서버에서 동작하는 서버 프로그램의 main 코드 중 스냅샷 부분은 표 5와 같다.

표 5의 usleep에 인수로 제공된 숫자만큼의 시점까지 수신된 모든 패킷을 한번에 전송하는 것이다. 인터넷망의 게이트웨이 서버에서 동작하는 서버 프로그램

표 4. 개발 환경

분류	환경
업무망 게이트웨이 서버 / 인터넷망 게이트웨이 서버	Desktop PC 운영체제: CentOS 5.1 개발언어: C/C++
업무망 클라이언트 / 인터넷망 클라이언트	Desktop PC 운영체제: Windows XP sp3 개발언어: Visual C++

표 5. main loop 코드

```

for(;;)
{
    // 0.5초 단위로 스냅샷 생성
    usleep( 500 * 1000 ) ;

    // 수신된 패킷이 있는지 확인
    if( g_pStorageManager->getPacket
        Count() == 0 )
        continue ;

    // 수신된 패킷 전송
    g_pStorageManager->sendPacket() ;
}
    
```

의 코드도 표 5와 유사한 구조를 가지고 있다. 데이터 공유 클라이언트는 윈도우 계열의 환경에서 동작을 하게 되며 사용자 UI를 가지게 된다. 그림 4는 내부 업무 망 PC에서 실행되는 데이터 공유 클라이언트 프로그램의 화면이다. 화면은 수평으로 분할이 되어 있는데 일반적으로 사용되는 FTP 프로그램들과 유사한 화면 구조를 가지고 있다.

상단 부분은 인터넷 망 PC에서 공유된 폴더와 파일의 목록을 보여주며 하단 부분은 내부 업무 망 PC의 폴더를 보여준다. 상단에 공유된 파일 중 필요한 파일 또는 폴더를 하단으로 드래그 & 드롭을 하면 내부 업무망 PC로 다운로드할 수 있다.

그림 5는 인터넷 망 PC에서 실행되는 데이터 공유 클라이언트 프로그램의 화면이다. 내부 업무 망 PC의 클라이언트 화면과 동일하게 분할되어 있다. 하단 부분은 인터넷 망 PC의 폴더와 파일을 나타내며 상단 부분은 내부 업무 망으로 공유된 폴더와 파일의 목록을 보여준다. 하단의 파일 중 내부 업무 망 PC로 공유하고자 하는 폴더 또는 파일을 상단으로 드래그 & 드롭을 하면 원하는 파일을 공유할 수 있다.

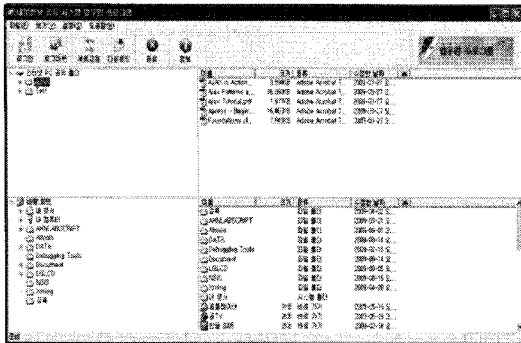


그림 4. 내부 업무망 PC 클라이언트

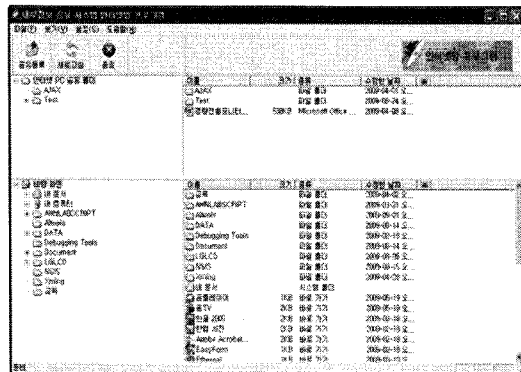


그림 5. 인터넷망 PC 클라이언트

3.2 실험 및 결과

본 논문에서 제안한 설계 방법을 이용하여 구현한 데이터 공유 시스템을 가지고 성능평가를 진행하였다. 스토리지에서 파일 전송을 하기 위해서 XL-Datamover를 이용하였으며 구현 후 실험 환경은 표 6에서 제시된 환경에서 진행하였다.

상기 표의 실험 환경에서 진행된 실험은 스토리지 내에서 파일을 전송하는데 걸리는 시간을 기준으로 측정해 보았고 걸린 시간은 소수점 셋째 자리에서 반올림하였다. 그림 6은 수신된 데이터의 총 크기가 120KB인 파일을 스토리지 내에서 전송하는 경우 걸리는 시간을 산출하기 위하여 수백 회 반복 후 평균을 구한 결과이다.

상기 그림과 같이 120KB를 한 번에 모아서 전송하는 경우 총 64KB블록을 두 번 읽어서 128KB 블록만큼 읽은 것이며 두 번에 나눠서 전송하는 경우 동일하게 128KB를 읽었지만 수행속도에 차이를 보이는 것은 mftsend 프로그램을 실행시키면서 초기화하는데 약간의 시간이 소요된 것으로 보인다. 반면 기존의 방식처럼 10KB씩 총 12회에 걸쳐서 전송을 하는 경우 읽어 들인 블록은 총 768KB이며 걸린 시간도 약 4배

표 6. 실험 환경

장비명칭	사양	비고
PRIMERGY RX300 S5	CPU: xeon quad core 2.26 * 2 cpu RAM: 8GB OS: Redhat Linux Enterprise Server	업무망 게이트웨이 서버
PRIMERGY RX300 S5	CPU: xeon quad core 2.26 * 2 cpu RAM: 8GB OS: Redhat Linux Enterprise Server	인터넷망 게이트웨이 서버
ETERNUS DX60	N/A	스토리지 장비

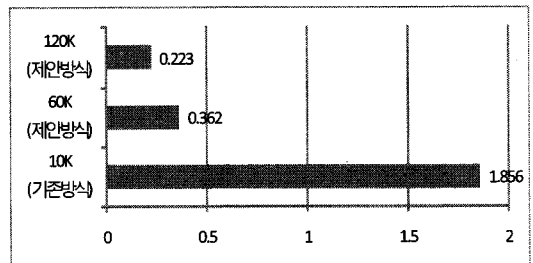


그림 6. 원본 파일의 크기가 120KB인 경우 전송하는데 걸린 시간 측정

이상 차이가 발생했다. 즉 동시 사용자 수가 100명이라 가정할 경우 동일한 시점에 수신된 패킷을 각각의 파일로 저장하여 전송하는 것보다 하나의 파일로 모아서 전송하는 것이 성능 상 더 좋다고 할 수 있다.

생성되는 파일의 크기에 따라서 전송시간이 어떻게 달라지는지 측정하기 위해서 수십 회 반복 후 평균을 구한 결과는 그림 7과 같다. 여기에서 생성되는 파일의 크기가 비교적 작은 경우에는 대부분 0.2초 내외로 소요되었으나 8MB가 되는 순간 약 50%정도 시간이 더 소요되었다.

1MB 크기의 파일 8개를 전송하기 위해서 동시에 채널을 1~8개를 이용하는 경우에 대한 시간을 측정해 보았다. 이 경우 파일을 전송하는 thread를 8개를 띄운 뒤 1개의 채널에 동시에 데이터를 보내는 것과 8개의 채널에 각각 1개씩 데이터를 보내는 것과 차이를 알아보기 위해서 테스트를 한 결과가 그림 8과 같다. 즉 하나의 채널에 동시에 8개의 데이터를 전송한 경우와 다수의 채널을 사용하여 동시에 8개의 데이터를 전송한 경우의 시간 차이가 크지 않았다.

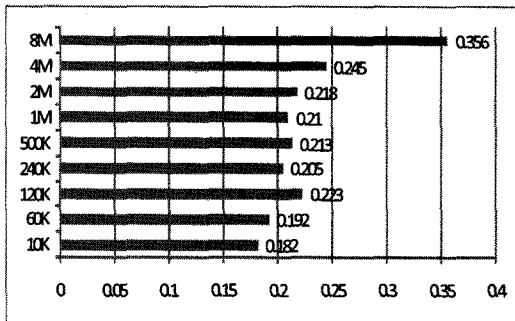


그림 7. 제안방식: 파일 사이즈에 따른 파일 전송시간 측정

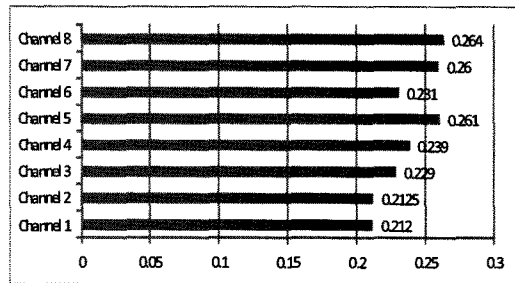


그림 8. 제안방식: 채널수에 따른 파일 전송 시간 측정

IV. 결 론

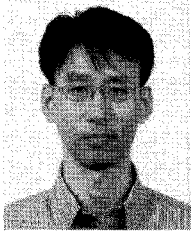
여러 명의 사용자가 동시에 접속하는 것을 가정하고 실험을 한 결과는 본 제안 시스템이 기존 시스템보다 파일 전송에 소요되는 시간이 줄어들었다. 그러나 한 명의 사용자가 접속하는 경우를 보면 기존의 시스템과 차이가 나지 않았다. 그 이유는 동일한 시점에 수신된 패킷이 하나 밖에 없고 수신된 단위 패킷으로 저장되어 기존 시스템과 차이가 없기 때문이다. 또한 데이터 공유 시스템은 실시간 형태로 서비스가 되어야 하기 때문에 한 명의 사용자가 접속하는 경우 파일을 일정 크기로 만들기 위해서는 계속 패킷을 수신 받아야 하는데 그렇게 된다면 사용자 입장에서는 대기 상태가 생겨나게 된다.

참 고 문 헌

- [1] 국가정보원, “국가정보보호백서”, 2010.
- [2] 신흥순, “공공기관의 정보유출 방지를 위한 정보 보호 방안에 관한 연구: 효과적인 망분리를 중심으로”, 성균관대학교 정보통신대학원 석사학위논문, 2009. 08.
- [3] 이은배 외, “망 분리기반의 정보보호에 대한 고찰”, 정보보호학회지, 제20권 제1호, pp.39-46, 2010. 02.
- [4] 이기혁 외, “내부정보 유출 징후 분석을 통한 유출방지체계 구축에 관한 연구”, 정보보호학회지, 제19권 제3호, pp.70-79, 2009. 06.
- [5] 최주호, “정보유출의 심각성과 이에 대한 보안방법”, 국회도서관보, 제44권 제6호 통권 제337호, pp.53-64, 2007. 06.
- [6] 최창학, “공공기관의 효율적인 정보보호 방안에 관한 연구: 서울시 내부문서 자료유출 방지 분야를 중심으로”, 서울터전: 시정연구논총, 제16권 통권 제43호, pp.729-752, 2009.
- [7] 정명수, “SAN 환경에서 선택적 자료 Backup 시스템 설계 및 구현”, 2008. 02.
- [8] 김석경, “SAN(Storage Area Network) 구축 방안에 관한 연구”, 2002. 12.
- [9] 인터넷침해대응 센터, “인터넷 침해사고 동향 및 분석 월보”, 2010. 10.

조 인 휘 (Inwhee Joe)

정회원



1983년 2월 한양대학교 전자
공학과

1994년 12월 미국 University of
Arizona, Electrical and Com-
puter Engineering, M.S.

1998년 9월 미국 Georgia Tech,
Electrical and Computer

Engineering, Ph.D.

1992년 12월 (주) 테이콤 종합연구소 선임연구원

2000년 6월 미국 Oak Ridge 국립연구소 연구원

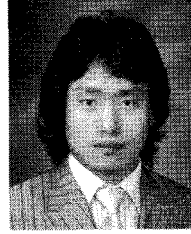
2002년 8월 미국 Bellcore Lab (Telcordia) 연구원

2002년 9월 ~현재 한양대학교 컴퓨터공학부 부교수

<관심분야> Sensor Networks, Cellular System and
PCS, Network Security, Mobility Management

이 석 순 (Suksoon Lee)

정회원



2001년 2월 충청대학 멀티미디
어과 졸업

2010년 8월 한양대학교 공학대
학원 컴퓨터공학과 졸업

2005년 11월 (주) 시큐랩 개발
팀 팀장

2010년 04월~현재 (주) 엔코아
통합건설팅사업본부

<관심분야> Data Communication, Network Security,
Data Modeling, Data Architecture