

Client/Server 모델을 이용한 E-Learning 시스템의 네트워크 엔진 구현

임재우, 이진영, 최종근, 박성순
안양대학교 컴퓨터학과

Implementing a Network Engine of E-Learning System Using Client/Server Model

Jae-Woo Lim, Jin-Yeong Lee, Jong-Guen Choi, and Sung-Soon Park
Dept. of Computer Science, Anyang University

요약

컴퓨터 보급이 크게 증가하면서 모든 분야에 있어 그 활용성을 더해 가고 있으며 교육 분야에도 많은 영향을 주었다. 그로 인해 CAI(Computer Assisted Instruction)분야에 커다란 진전이 있었다. 최근 들어 멀티미디어 기술 및 컴퓨터 네트워크 기술이 획기적으로 발전되고 확산됨에 따라 같은 시간, 다른 장소에서의 E-Learning이 활발해지고 있다. 이것이 바로 Live E-Learning System이다. 이렇게 실시간으로 화상회의나, 강의가 이루어지는 E-Learning System에서 과연 어떻게 E-Learning System을 구축하고, 동작하는지에 대한 중요성이 강조 되고 있다. 본 논문에서는 Live E-Learning 시스템에서의 효과적인 네트워크 엔진 구현에 대한 방안을 제시하고자 한다.

1. 서론

많은 정보가 초단위로 증가하고 있는 현 시대 속에서 technology는 적시에 필요한 정보를 찾아 주고 학습도 가능하게 해주고 있으며 특히 인터넷의 발달은 E-Learning의 급진적 발전을 가져왔다.

E-Learning은 technology를 활용한 교육을 의미한다. 즉 컴퓨터를 기반으로 한 교육이라든지 웹을 기반으로 하는 교육 등이 E-Learning에 포함될 수 있다. 실제로는 E-Learning이 원격교육, 온라인 교육 등과 뚜렷이 구분되지 않은 채 사용되고 있지만, 엄

격히 구분하면 원격교육이 E-Learning을 포함하며, E-Learning 안에는 온라인 교육이 포함된다고 할 수 있다.

E-Learning의 큰 장점은 동일한 콘텐츠를 학습자에게 맞춤형으로 제공, 언제 어디서나 학습이 가능하다는 점이다. 또한 E-Learning은 정보를 저렴하게 전달하고 인프라의 수요를 제거하여 교육비용을 대폭 줄이는 효과를 가져 오기도 한다. 이러한 장점을 기반으로 E-Learning의 수요는 세계적으로 급증하고 있다. E-Learning의 적용은 온라인 실시간 강의, 사내교육, 사업 설명회 등 여러 분야에 활용 가능 하

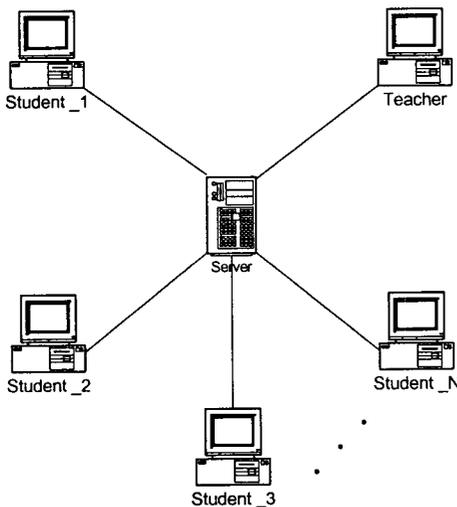
다. 정적인 온라인 교육 이외에 실시간 원격 교육시 가장 중요한 사항은 클라이언트/서버간의 모델링 및 양자간의 통신 방법이다. 이에 대한 선택에 따라 성능에 많은 차이가 일어날 수 있으므로, 상황에 맞는 선택이 요구된다. 뿐만 아니라, 서버부분에서 공유될 수 있는 자원에 대한 동기화(Synchronization), 스레드의 원활한 사용 등이 중요한 사항이 될 것이다. 클라이언트 부분에서도 멀티미디어 데이터 전송을 위한 버퍼링, 음성, 화상 및 이벤트 데이터들의 동기화 또한 중요 이슈가 될 것이다.

그 이외에도 많은 고려사항들이 있지만, 본 논문에서는 클라이언트/서버의 모델링, 그에 대한 장·단점을 분석, 기술하며, 클라이언트/서버간의 통신방법, 서버의 스레드 동기화, 자원의 효율적인 사용 및 클라이언트에서의 멀티미디어 데이터, 즉 음성, 화상 데이터에 따른 버퍼링과 TCP, UDP Protocol을 이용한 Data의 전송 및 수신 기능의 네트워크 엔진 구현 방안을 기술한다.

본 논문의 구성은 2장에서는 클라이언트/서버의 모델링 및 설계, 3장에서는 구현에 대한 설명을, 마지막으로 4장에서는 결론 및 향후 발전 가능성을 제시한다.

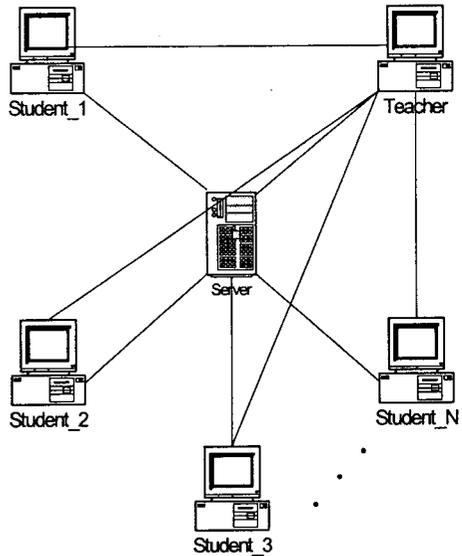
2. 설계

2.1 클라이언트/서버의 모델링



< 그림1 스타형의 구조(유형 1) >

<그림 2>의 모델은 <그림 1>보다 서버의 데이터 전송량을 줄임으로써 네트워크의 부하를 줄일 수 있다. 그러나, 원격 교육의 특성상 멀티미디어 데이터를 다루기 때문에 <그림 2>의 모델을 사용 시 Teacher에게 Student들 모두에게 데이터를 전송해야 하는 부담을 줄 수 있다. 이는 Teacher에게 과중한 네트워크 부하가 생기는 원인이 되어 통신에 큰 차질을 줄 수 있기 때문에 <그림 1>의 모델을 지향한다.



< 그림2 스타형의 구조(유형 2) >

2.2 서버의 구조 및 설계

서버의 구조는 크게 채팅 서버, 파일 서버, 멀티캐스트 서버 및 얼라이브 서버의 기능으로 구분된다. 파일서버는 강의록을 개설하는 강사가 강의에 필요한 파일들을 서버에 업로드 하여, 강의를 수강하고자 하는 학생들이 서버에 접속하여 강의 파일을 다운로드 한다. 채팅서버는 실시간 강의시 학생들과 강사의 질의 응답시 사용되며, 멀티캐스트 서버는 각 강의별로 강사로부터 발생하는 모든 이벤트, 음성, 화상들의 데이터들을 강의 강의실에 접속한 학생들에게 멀티캐스트한다. 얼라이브 서버는 서버에 접속한 클라이언트가 비정상적인 종료로 하거나, 클라이언트의 네트워크 끊김 현상 등 클라이언트들의 서버연결 유무를 체크한다.

2.3 클라이언트의 구조 및 설계

E-Learning 시스템에서 네트워크 엔진의 Client의 기능은 크게 데이터의 전송과 수신으로 나눌 수 있다. 실시간 강의시 발생하는 데이터를 서버로 전송하며, 서버로부터 수신된 데이터를 종류별로 저장하여 내부에서 가공할 수 있도록 한다.

2.3.1 데이터 전송

실시간 강의시 강의를 효과적으로 Student에게 전달하기 위하여, 제공되는 강의 자료 외에 판서도구, 채팅, Scroll, Tab 변경 등의 이벤트와 화상 및 음성이 이용된다. 이렇게 다양하게 발생하는 데이터는 다른 정보들과 함께 Packet으로 구성되어 Socket을 이용하여 전송한다.

1) Packet의 구성

실시간 강의시에 발생하는 다양한 종류의 데이터를 구분하기 위한 방법으로 Packet type과 Header를 이용하였다. Packet은 Packet type, Header, 발생한 데이터로 구성된다. Packet의 크기는 한번에 전송할 수 있는 데이터의 최대크기(1Kb)와 Packet typer, Header를 포함하기 위해 1040byte로 선언하여 사용한다.

Packet type (4 byte)	Header (12 byte)	데이터 (1024 byte)
---------------------------	-----------------------	---------------------

< 그림 3 Packet의 형식 >

◆ Packet type

발생하는 데이터의 다양성으로 인해 필요한 Packet의 종류도 다양하다. 따라서 현재 Packet이 어떠한 종류의 데이터를 저장하고 있는지를 나타내는 구분자가 필요하다. 본 시스템에서는 Packet type은 정수형으로 선언하여 사용한다.

◆ Header

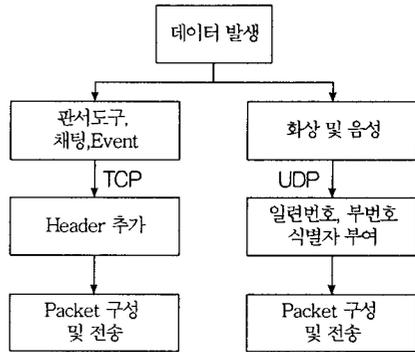
Header는 현재 데이터가 어떠한 의미를 지니고 있는지를 나타내기 위하여 이용된다. 같은 형태의 데이터를 전송하는 경우 각 데이터가 의미하는 내용이 다른 경우를 고려하여 Header는 세 개의 정수형을 구조체로 선언하여 이용한다.

```
typedef struct tagHeader
{
    int iFirst;
    int iSecond;
    int iThird;
}HEADER;
```

< Header를 나타내는 구조체 >

2) 데이터의 전송

실시간 강의시 발생하는 데이터는 서버에게 전송되어 해당 강의에 연결되어 있는 다른 Client에게 전달된다. 데이터를 전송 할 때는 데이터의 형태에 따라 TCP와 UDP Protocol 방식중 하나를 이용한다.



< 그림 4 데이터 전송 >

◆ TCP Protocol 방식을 이용

TCP Protocol 방식을 이용하여 전송하는 방법은 신뢰성이 보장되어야 하는 경우에 주로 사용한다. 본 시스템에서는 판서도구, 채팅, Event 데이터를 보낼 때 TCP Protocol 방식을 이용하여 신뢰성을 높여 데이터의 손실을 최소화 하였다.

◆ UDP Protocol 방식을 이용

UDP Protocol 방식을 이용하여 전송하는 방법은 전송하고자 하는 데이터의 크기가 크고 신뢰성이 비교적 낮아도 되는 경우에 주로 사용한다. 따라서 화상 및 음성 데이터는 UDP 방식을 이용한다.

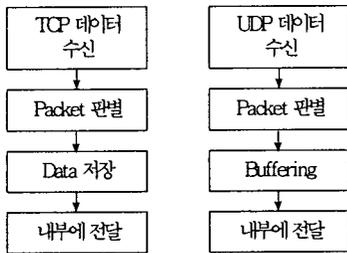
화상 및 음성의 크기는 대부분 1Kb를 초과하므로 일련번호와 부 번호를 붙여서 나누어 전송한다. 일련번호는 강의의 시작부터 종료 전까지 발생하는 모든 화상 및 음성 데이터에 순차적으로 부여된다. 부 번

호는 각 데이터를 전송할 때 부여되는데, 해당 데이터의 크기를 1Kb로 나누어 해당 개수만큼 부 번호가 부여된다.

화상 및 음성 데이터는 나누어 전송하므로 해당 데이터별로 마지막에 전송되는 Packet에는 마지막 데이터임을 나타내는 식별자를 첨부한다.

2.3.2 데이터 수신

데이터를 전송할 때 TCP Protocol과 UDP Protocol 방식을 이용하여 전송하므로, Client에 수신될 때도 해당 방식에 따라 각각 다른 함수를 호출한다.



<그림 5 데이터 수신>

1) TCP Protocol 방식을 이용한 데이터의 분류

수신된 Packet은 Packet type을 이용하여 데이터가 판서도구인지 채팅인지 또는 다른 Event 인지 판별한다. 또한 Header를 이용하여 수신된 데이터의 의미를 파악하고, 해당하는 함수를 호출하여 저장한다. 그 이후 데이터를 가공해도 된다는 Message를 내부에 전송한다.

2) UDP Protocol 방식을 이용한 데이터의 분류

수신된 Packet은 Packet type을 이용하여 데이터가 화상인지 음성인지 판별할 수 있다. 화상 및 음성 데이터는 TCP Protocol 방식을 이용한 데이터와는 달리 Header의 정보 외에 일련번호와 부 번호의 정보를 이용한다. 이 일련번호와 부 번호는 Buffering 기법에 이용된다.

◆ Buffering 기법

UDP Protocol 방식을 이용하면 Packet이 전송된 순서대로 수신이 되지 않을 수 있으며 모든 Packet

이 도착한다는 보장도 없다. 또한 데이터가 나누어 전송되므로 하나의 데이터가 모두 수신되는 동안 수신된 다른 데이터를 저장하는 공간이 필요하다.

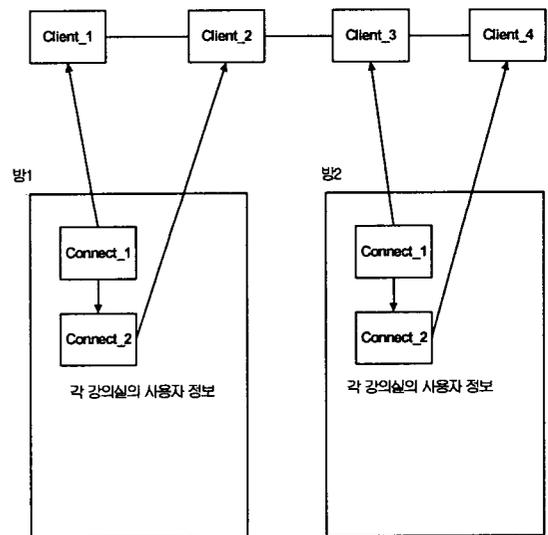
Buffer에서 동시에 다뤄야 되는 데이터의 개수를 Test를 통하여 구한 후 Buffer를 선언한다. 각 Buffer의 사용여부에 대한 Flag를 유지하여 수신된 데이터의 일련번호가 바뀔 때 마다 비어있는 Buffer를 검색 및 할당한다.

3. 구현

3.1 서버의 구현

서버구조는 각 클라이언트에 대한 정보는 하나의 리스트로 관리한다. 연결된 클라이언트가 강의에 입장 시 <그림 3>처럼 각 강의별로 하나의 리스트를 만들어 강의별 클라이언트를 관리한다. 이때, 새롭게

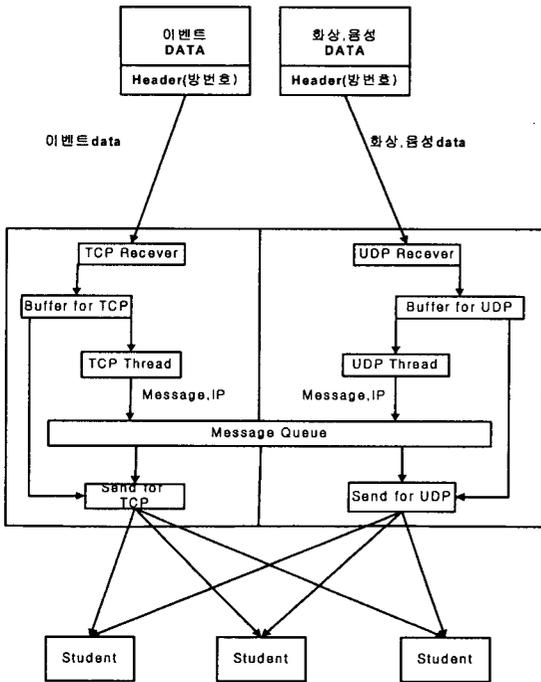
접속시 Client의 정보 구조체



<그림 6 클라이언트의 연결구조>

만드는 리스트는 처음 생성된 클라이언트의 리스트를 참조한다. 강의 종료 시 이 리스트는 소멸되며, 처음 생성된 클라이언트 리스트는 그대로 남아 있다. 처음 생성된 이 리스트는 클라이언트가 서버와의 연결을 종료 시 소멸된다.

다음으로 강의를 시작할시 데이터는 음성, 화상은 UDP 로 이벤트 및 채팅 데이터는 TCP 데이터로 분리해 <그림 4>에서와 같이 각각 방별 스레드를 하나씩



< 그림 7 강의 객체의 정보전달 구조 >

구성해 데이터의 전송을 담당한다. 각각의 스레드에서 사용하는 버퍼는 클라이언트로부터 온 데이터들을 리스트로 연결하며 데이터의 전송 후 데이터를 삭제한다.

각 강의별로 TCP,UDP 스레드 사용 시 각각 하나의 버퍼를 여러 클라이언트가 사용한다. 이때, 같은 프로세스의 스레드들의 상호 배제를 구현하기 위해 임계영역(Critical Section)을 사용하여 공유된 자원 사용 시 동기화를 구현한다..

3.2 클라이언트의 구현

3.2.1 데이터 전송

발생한 데이터는 종류별로 다른 Packet을 구성한다. Packet은 문자형 배열 1040크기로 선언하며 Packet type, Header, 발생한 데이터는 memcpy를 이용하여 배열에 저장한다.

구성된 Packet은 Socket을 이용하여 전송한다. 본 시스템은 MFC의 CAsyncSocket Class를 이용하여 전송하였다. Socket은 Protocol 방식에 따라 각각 전송된다.

3.2.2 데이터 수신

데이터의 수신은 Protocol 방식에 따라 각각 호출

된 함수에서 이루어진다.

TCP 데이터는 데이터 종류별로 분류하여 내부에서 데이터를 가공할 수 있도록 저장한 후에 Message를 발생시켜 가공할 수 있도록 데이터가 준비되었음을 알린다.

UDP 데이터는 화상인지 음성인지 구분하여 해당 Buffer에 저장한다. 일련번호와 이미 저장되어 있는 Buffer의 일련번호를 비교하여 해당번호가 있을 경우에는 저장되어 있는 데이터의 뒤에 연속하여 수신된 데이터를 저장한다. 해당번호가 없는 경우는 새로운 데이터가 수신된 것이므로 비어있는 Buffer를 할당하여 데이터를 저장한다.

만약 마지막 데이터를 가리키는 식별자가 도착한 경우에는 해당 Buffer의 내용을 Sorting하며 이때 데이터의 소실유무를 동시에 검사한다. 모든 데이터가 소실되지 않고 수신된 경우에는 Message를 발생시켜 내부에서 데이터를 가공할 수 있도록 한다.

4장. 결론

이제까지 E-Learning 시스템에 대한 구현 및 설계에 대하여 기술하였다. 위의 논문을 기반으로 하여 여러 대의 서버를 확장하여 사용자의 수를 늘리거나, 고가용성을 지닌 서버를 구축하기 위하여 클러스터링의 도입 등 상황에 맞게 여러 시스템을 구축할 수 있을 것이다. 무엇보다 가장 중요한 것은 이제 E-Learning 시스템은 통신망의 급속한 발전의 영향으로 학원의 원격강의 및 온라인 교육, 회사의 사내 교육 이외에도 많은 부분으로 그 범위를 확장하고 있다.

현재의 E-Learning 시스템은 통신망의 더 많은 보급과 속도 및 안정성의 향상, 멀티미디어의 전송 및 수신에 대한 기술 향상 등으로 인해 더욱 많은 곳에서 활용될 수 있을 것이며, 특히 E-Learning 시스템을 이용한 원격강의는 새로운 교육의 한 형태로 자리 잡아 사이버대학의 좋은 발판이 될 것으로 예상된다.

[참고문헌]

- [1] Win32 Multithreaded Programming, O'REILLY, Aaron Michael Cohen, Mike Woodring
- [2] VISUAL C++ 6 완벽 가이드, 영진출판사, 김용성 저

- [3] Visual C++ 정복 6.0 가남사 김상형 저
- [4] Win32 네트워크 프로그래밍, 대림, Ralph Davis
- [5] Unix Network Programming. Vol 1,
PRENTICE HALL, W. Richard Stevens
- [6] Inside Secret Visual C++ 6, 삼각프레스,
곽준기, 김소영 저
- [7] 컴퓨터 네트워크 프로그래밍 김화종 교수 저
- [8] TCP/IP Illustrated Volume 1 (the Protocols)
Addison Wesley, W.Richard Stevens