

## 멀티 스레드 그룹 기반 캐스팅 서버 설계 및 구현

김연정, 조동섭  
이화여자대학교 과학기술대학원 컴퓨터학과

### Design and Implementation of the Multi-threaded Casting Server based Group

Yeun-jung Kim, Dong-Sub Cho  
Dept of Computer Science and Engineering Ewha Womans University

**Abstract** - 멀티 스레드 기법을 사용하여 사용자가 원하는 데이터를 그룹별로 전송해 주는 블로트캐스팅 서버를 구현하였다. 서버는 멀티 스레드로 사용자를 받아들이고 사용자가 만들기 원하는 그룹을 생성하며 데이터를 전송한다. 또한 각 사용자와 생성된 그룹들을 연결 리스트를 사용하여 관리하므로 서버의 효율을 높였다. 제안된 서버는 새로운 데이터를 만들어 저장하고 전송하는 부분과 그 데이터를 받아 저장하고 웹 서버에 전달하는 부분으로 나누어진다. 이 서버는 데이터를 전송하는데 메시지 전송과 파일 전송의 두 가지 방법을 사용하여 매끄럽게 블로트캐스팅하도록 하였으며 서버와 클라이언트 기반으로 서버와 클라이언트가 서로 정보를 주고받을 수 있다.

#### 1. 서 론

서버가 생긴 이후 클라이언트에게 더 나은 서비스를 제공하기 위해 서버의 성능을 향상하기 위한 많은 연구가 진행되어 왔다. 서버의 성능에 영향을 미치는 요소에는 저장장치 시스템 부분과 서버 시스템 부분, 네트워크 시스템 부분 그리고 어플리케이션 부분 등이 있다. 어플리케이션 부분에서 서로 비슷한 특성을 가지고 있는 사용자들을 묶어 그룹으로 관리하는 것이 더 효율적이다. 이 논문에서는 그룹별로 사용자가 원하는 데이터를 전송해 주는 블로트캐스팅 서버를 설계하고 그 중 메시지 전달 부분을 구현하였다. 서버의 특징으로는 어떠한 사용자가 새로운 그룹을 만들기를 원하면 멀티 스레드로 그룹을 생성하고 각 사용자와 생성된 그룹을 연결 리스트를 사용하여 관리한다는 것이다. 제안된 서버는 새로운 데이터를 만들어 전송하고 저장하는 뉴스 캐스팅 부분과 그 데이터를 전송 받아 저장하고 웹 서버에 전달하는 뉴스 캐스팅 클라이언트 부분으로 나누어진다.

이 서버는 데이터를 전송하는데 실시간으로 데이터 전송은 메시지로 전송하고 기존의 데이터는 파일로 전송하는 두 가지 방법을 사용하여 매끄럽게 블로트캐스팅하고 있다. 서버와 클라이언트 기반으로 서버와 클라이언트가 서로 정보를 주고받을 수 있다. 클라이언트는 자신이 얻고 싶은 정보를 고를 수 있다. 서버는 이것을 지원하기 위해 생성된 파일과 클라이언트의 정보(로그)와 데이터베이스에 저장하여 관리한다.

2장에서는 파일 전송과 메시지 전송을 모두 하는 그룹 기반 다중 스레드 서버에 적합한 프로토콜과 그 프로토콜을 프로그래밍할 때 필요한 소켓대해 알아보고 3장에서는 그룹 기반 멀티 스레드 서버의 전체적인 디자인과 각 부분별 기능과 필요한 점을 알아보고 구현한 방법에 대해 알아보겠다. 4장에서는 지금까지 구현한 결과를 살펴보고 5장에서는 앞으로 구현 해야할 캐스팅 서버의 부분과 매끄럽고 빠른 전송을 위해 필요한 서버의 보안 점 등의 향후 연구과제를 알아보는 것으로 이 논문을 마칠 것이다.

본 연구는 2003년도 두뇌한국 21산업에 의하여 지원되었음

#### 2. 데이터 전송 방법

##### 2.1 소켓이란

소켓은 TCP/IP 프로토콜의 프로그래머 인터페이스이다. 소켓은 동작하는 프로세스들 사이의 데이터를 주고받는 것을 가능하게 하는 프로세스간 상호 통신방식이다. 이들 프로세스는 서로가 동일한 컴퓨터 내에 있거나 다른 컴퓨터에 있어도 가능한데 이런 네트워크를 통한 통신 능력이 소켓이 대단히 유용하게 사용되는 이유이다.

소켓을 경유한 프로세스간의 통신은 클라이언트-서버 모델에 기초로 한다. 서버 프로세스는 그 컴퓨터에서 유일하게 할당된 번호의 소켓을 생성한다. 클라이언트 프로세스는 이 번호의 소켓을 통해서 서버 프로세스와 대화할 수 있다. 소켓은 양방향(bidirectional)으로 연결에 성공하면 서버와 클라이언트 모두에게 각각 하나의 소켓 기술자가 반환되는데 이것을 이용하여 읽기/쓰기를 할 수 있다.

포트는 물리적인 전송선은 하나지만 그것을 여러 개의 응용 프로그램이 같이 사용하기 위해서 만들어 졌다. 한 컴퓨터 내의 모든 서버 프로세스는 별도의 포트번호를 갖는 소켓을 갖는다. 이 포트 번호는 TCP/IP가 지원하는 상위계층의 응용 프로그램을 구분하기 위해 사용되는 번호이다. 주로 많이 사용되고 있는 TCP/IP 기반의 프로토콜에는 다른 사람의 호스트 컴퓨터를 원격지에서 액세스할 수 있도록 해주는 Telnet(23번 포트)과 사용자에 관한 정보를 제공하는 프로토콜 Finger(79번 포트) 그리고 파일 전송에 사용되는 Ftp(20/21번 포트) 등이 있다.[5]

##### 2.2 소켓 동작 모드

소켓의 동작 모드에는 두 가지가 있는데 그중 하나가 블로킹 모드(blocking mode)이다. 블로킹 모드에서는 accept, recv, send를 호출했을 때 연결하고자 하는 클라이언트가 없거나, 읽을 데이터가 없거나 혹은 보낼 데이터를 즉시 쓸 수 없을 때 그것이 처리될 때까지 대기한다. 반면 언블로킹 모드(non-blocking mode)는 언블로킹 모드에서는 accept, recv, send를 호출했을 때 함수가 호출되었을 때 함수의 원하는 동작이 완료되는 것과 무관하게 일단 함수가 즉시 리턴되는 소켓을 말한다. 즉 error 예외처리가 발생한다. 소켓이 처음 만들어진다면 blocking 모드가 되는데 프로그래머는 필요에 따라 blocking 모드의 소켓을 non-blocking 모드로 바꿀 수 있다.

그룹 기반 멀티 스레드 서버에서는 데이터를 전달할 때 실시간으로는 메시지를 전송하고 기존의 데이터는 파일 전송을 하므로 텔넷 프로토콜이 사용하는 포트는 23 포트를 이용하였다. 멀티 스레드를 구현하여 각각의 프로세스들이 독립적으로 수행되므로 한 프로세스가 블로킹되어 있어도 다른 프로세스들은 계속 수행될 수 있기 때문에 블로킹 모드로 구현하였다.

### 3. 그룹 기반 캐스팅 서버

#### 3.1 그룹 기반 캐스팅 서버의 디자인

본 논문에서 제안하는 그룹 기반 캐스팅 서버란 사용자가 자신이 원하는 주제를 다루는 그룹에 가입하여 원하는 정보를 보다 쉽고 편하게 제공받을 수 있도록 디자인한 서버이다.

그룹 기반 캐스팅 서버의 전체적인 구조는 그림1과 같은 크게 두 부분으로 나누어진다. 각 부분은 새로운 데이터를 만들어 전송하는 부분과 그 데이터를 받아 저장하고 웹 서버에 전달하는 부분이다.

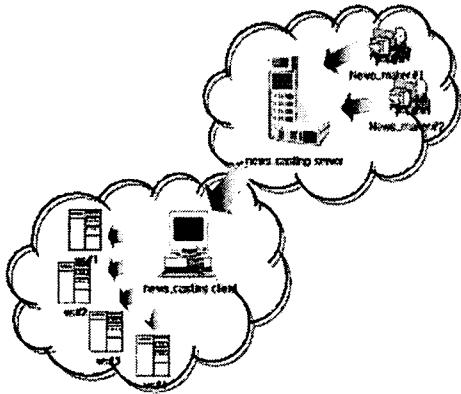


그림 1 뉴스 캐스팅 서버의 디자인

##### 3.1.1 뉴스 캐스팅 서버

뉴스 캐스팅 서버(news\_casting server)가 새로운 데이터를 만들어 전송하는 기능을 담당하고 있다. 이 부분에서는 뉴스 캐스팅 서버에 접속한 두 명의 뉴스 메이커가 자신들이 원하는 주제를 다룰 하나의 그룹을 만들고 그 주제에 맞는 새로운 데이터를 뉴스 캐스팅 서버를 통해 자신의 그룹에 속한 뉴스 캐스팅 클라이언트에게 블로드캐스팅한다.

이 기능을 원활하게 수행하기 위해서는 몇 가지 추가적인 기능을 필요로 하는데 첫째로 데이터 전달 방식이다. 클라이언트가 접속을 유지하고 있을 때에는 실시간으로 데이터를 메시지로 전송한다. 그러나 클라이언트가 그룹에 가입해 있으나 잠시 접속을 해지한 동안의 데이터는 클라이언트가 접속을 해지하더라도 뉴스 캐스팅 서버는 데이터를 유지하고 있다가 다시 접속했을 때 전달한다. 이를 위해 데이터를 파일로 만들어 보관한다.

두 번째는 이렇게 만들어진 파일을 원활하게 해당 클라이언트에게 전달하기 위해 파일과 클라이언트의 로그 관련 정보를 데이터베이스에 저장하여 관리한다. 데이터베이스에는 각 클라이언트가 속한 그룹명과 파일의 위치 접속시간 접속을 해지한 시간들을 저장한다. 데이터베이스의 이 정보들을 이용하여 해지된 동안의 데이터를 뉴스 캐스팅 클라이언트에게 원활하게 전달할 수 있다.

##### 3.1.2 뉴스 캐스팅 클라이언트

뉴스 캐스팅 클라이언트(news\_casting client)부분은 뉴스 캐스팅 서버에서 만들어진 데이터를 전송 받아 웹 서버에게 보내는 역할을 한다. 또한 자신이 접속하기 전에 블로드캐스팅된 데이터를 서버에 요청하는 역할과 캐쉬 서버로서의 역할을 동시에 수행한다. 뉴스 캐스팅 서버에서 데이터를 전송 받은 뉴스 캐스팅 클라이언트는 자신이 받은 뉴스를 자신의 그룹에 속해 있는 웹 서버에 전송하고 웹 서버는 그 뉴스를 데이터베이스에 저장한다.

웹 서버는 사용자의 요청을 듣고 데이터베이스로부터

새로운 뉴스를 사용자에게 전송한다. 사용자의 요청을 받아들이고 처리하는 것은 웹 서버가 한다.

뉴스 캐스팅 클라이언트가 접속을 해지한 후 다시 접속을 했을 때는 클라이언트의 권한으로 현재의 전송 중 데이터만을 가지고 올지 아니면 자신의 접속하지 않은 동안의 데이터를 가지고 올지를 결정한다. 그림2가 재접속 후 데이터를 전송할 때 알고리즘이다.

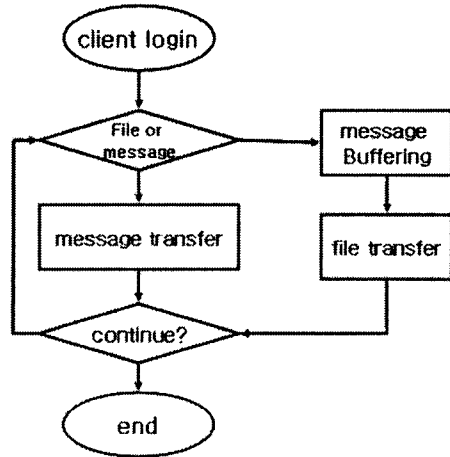


그림2 데이터 전송

클라이언트가 항상 접속을 하고 있지 않기 때문에 매끄럽게 데이터를 전하는데 생기는 여러 가지 문제들을 해결해야한다. 첫째로 클라이언트가 다시 접속했을 때 파일을 전송하는 동안 새로 만들어진 데이터 처리를 고려해야 한다. 이것은 실시간 전송을 중요시하기 때문에 그 시간 동안 새로 만들어진 데이터는 서버가 버퍼에 저장해 두었다가 전송을 해 주어야 한다.

#### 3.2 뉴스 캐스팅 서버의 구현

현재에는 news\_casting server와 news\_casting client중에서 news\_casting server부분의 그룹간의 메시지 전송부분이 구현되었다. news\_casting server는 새로운 사용자를 받아들이고 그 사용자가 그룹은 생성할 때 각각의 그룹을 멀티 스레드로 만들어지고 유지된다.

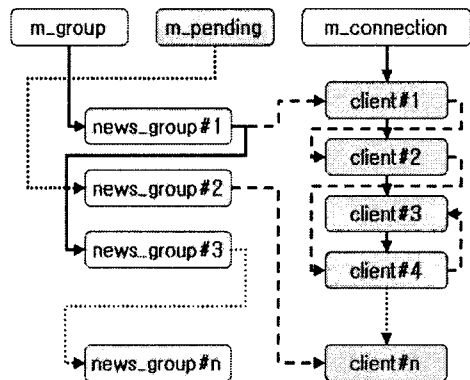


그림 3 연결 리스트를 이용한 데이터 관리

서버는 각각의 사용자가 접속하는지를 스레드를 생성하여 감시하고 계속해서 다른 사용자를 받아들인다. 접속된 사용자는 멀티 스레드를 이용하여 새로운 그룹을 만든다.

이 서버에 접속하는 각 사용자(news\_casting client, news maker)와 각 뉴스 그룹 등은 메모리를 동적으로 할당받을 수 있는 연결 리스트로 구현하여 메모리를 절약하며 통합적으로 관리하도록 하였다. 그림3에서 보여지는 것처럼 각각의 접속자를 관리하는 연결리스트와 생성되어 뉴스를 블로트캐스팅하는 그룹과 생성되었지만 클라이언트가 없어 아직 블로트캐스팅을 하고 있지 않은 그룹으로 나누어 연결리스트로 관리한다. 생성된 그룹은 자신의 사용자(new\_casting client, news maker)가 누구인지를 연결 리스트를 이용하여 구별하고 있다.

#### 4. 구현 결과

아래의 두 그림(그림4, 그림5)은 실제로 구현한 그룹 기반 캐스팅 서버의 실행 화면이다.

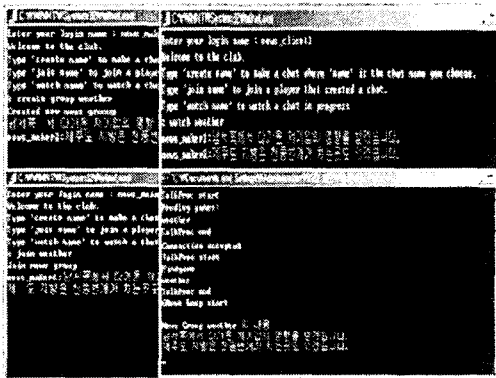


그림 4 실행 화면 I

그림4는 하나의 그룹 내에서 자신이 원하는 데이터를 전달하는 실행 화면이다. 한 명의 뉴스 메이커가 뉴스 그룹을 만들고 또 다른 뉴스 메이커와 함께 새로운 뉴스를 만들어 서버로 전송하면 서버는 그것을 받는다. 그리고 서버는 이 그룹에 속한 뉴스 캐스팅 클라이언트에게 새로운 뉴스를 블로트캐스팅하는 것을 볼 수 있다.

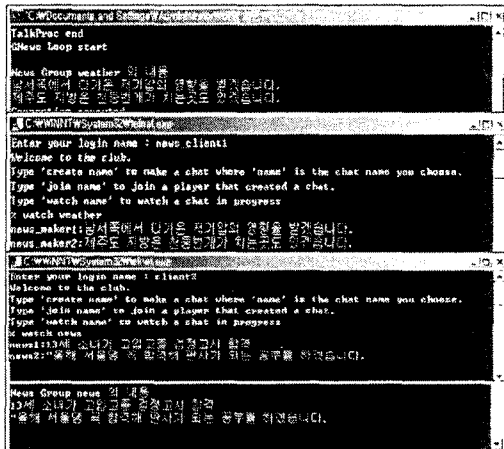


그림 5 실행 화면 II

그림5는 각각 다른 주제를 가지고 있는 두 개의 그룹이 각각 자신의 그룹을 만들어 자신의 그룹끼리 데이터를 전송하는 모습을 보여 주고 있다. 그림5의 서버 부분에서 알 수 있듯이 하나의 서버를 통해 각 뉴스 그룹

으로 다른 그룹에 영향을 주지 않고 데이터를 전송하고 있다.

#### 5. 결론

그룹 기반 캐스팅 서버란 자신이 원하는 주제를 다루는 그룹에 가입하여 쉽고 편하게 정보를 얻을 수 있도록 한 서버로 새로운 그룹을 만들어 서버를 통해 뉴스를 블로트캐스팅하는 news\_casting server 부분과 이렇게 만들어진 뉴스를 전송 받아 웹 서버에게 보내는 news\_casting client 부분이 있다. 데이터를 전달하는 방법에는 메시지 전달과 파일 전달하는 방식을 사용하였다. 실시간으로 전달할 때에는 메시지 전송을 사용하고 중간에 전달받지 못한 데이터를 전달할 경우에는 파일 전송을 사용한다. 현재에는 news\_casting server 부분의 메시지 전달 부분이 연결 리스트와 멀티 스레드로 구현하였다.

사용자는 서버와는 다르게 항상 접속하고 있는 것이 아니므로 사용자가 접속을 하지 않은 동안 블로트캐스팅 되지 못한 새로운 데이터는 서버에 요청하여 다시 받아오는데 사용하기 위한 캐쉬 기법이 필요로 한다. 또한 뉴스 캐스팅 클라이언트는 뉴스 캐스팅 서버로부터 전달 받은 데이터를 저장하고 있다가 요청자에게 전달하는 캐쉬 서버의 역할을 하기 때문에 각 그룹의 선호도에 따른 캐쉬 서버 정책이 또한 연구되어질 필요성이 있다.

#### (참 고 문 헌)

- [1] Aaron Cohen & Mike Woodring, "Win32 MultiThreaded Programming", O'reilly, 1997.9
- [2] Schmit Douglas C, Huston, Stephen D., "C++ and Network Programming", Addison-Wesley 2002.11
- [3] 김연정, "다중 스레드 서버의 설계 및 구현", 한국멀티미디어 학회 춘계학술대회 논문집, 상권, 2003.5
- [4] 김정훈, "TCP/IP 소켓 프로그래밍", 교학사, 2003.4
- [5] 이상성, "소켓 프로그래밍", 소켓이용하기, <http://www.python.or.kr:8080/python/Internet/socket/socket.html>, 2000.3.23
- [6] "인터넷 방송 기술" <http://myhome.hanafos.com/~kdyacht/ingi.htm>
- [7] 저자명, "논문제목", 논문지명, 권호, 페이지, 출판년도