

Chapter 11. 멀티캐스트

Mingyu Lim

**Collaborative Computing Systems Lab,
School of Internet & Multimedia Engineering
Konkuk University, Seoul, Korea**

■ 학습 목표

- 멀티캐스트 방식이란
- 멀티캐스트 백본: Mbone
- 간단한 인터넷 방송 클라이언트와 서버 구현
- 채팅 프로그램 작성

멀티캐스트 방식이란

- 한번의 패킷 전송으로 라우터에 의해 네트워크 내의 여러 호스트들이 수신하도록 하는 전송방식
 - 라우터와 호스트는 IGMP (Internet Group Management Protocol)을 지원해야 함
- 패킷의 목적지는 하나의 호스트가 아닌 멀티캐스트 그룹으로 지정
 - 클래스 D IP 주소 (244.0.0.0 ~ 239.255.255.255)
- UDP 기반
 - 전송 패킷이 유실되거나 순서 바뀔 수 있음

멀티캐스트 백본: Mbone

□ Mbone

- 멀티캐스트를 지원하지 않는 라우터들 사이에 멀티캐스트가 가능하도록 해주는 가상 멀티캐스트 프레임워크
- 주로 실시간 오디오 및 동영상 전송에 이용
- 패킷은 멀티캐스트 지원이 안되는 인터넷에서는 유니캐스트로 전송 (터널링)

□ 멀티캐스트 vs 유니캐스트

- 한번의 전송으로 여러 호스트에 수신 가능하므로 전송시 필요한 대역폭 감소

간단한 인터넷 방송 클라이언트와 서버 구현

□ MulticastServer (예제 11-1)

- UDP 서버 프로그래밍과 거의 유사
- 클라이언트 IP를 멀티캐스트 주소로 지정하는 부분이 차이

```
String address = "239.0.0.1";  
...  
channel = InetAddress.getByName(address);  
packet = new DatagramPacket(b, b.length, channel, port);  
socket.send(packet);
```

간단한 인터넷 방송 클라이언트와 서버 구현

□ MulticastClient (예제 11-2)

- UDP 클라이언트 프로그래밍과 유사
- DatagramSocket 대신 MulticastSocket 사용, 멀티캐스트 그룹에 참여 루틴 필요

```
String address = "239.0.0.1";  
...  
receiver = new MulticastSocket(port);  
channel = InetAddress.getByName(address);  
packet = new DatagramPacket(b, b.length);  
receiver.joinGroup(channel);
```

- 멀티캐스트 패킷 수신: receive()

```
for (int i=0; i<3; i++) {  
    receiver.receive(packet);  
    String notice = new String(packet.getData());  
    System.out.println(notice);  
}  
receiver.leaveGroup(channel);  
receiver.close();
```

멀티캐스트 채팅

□ MulticastChatClient (예제 11-3)

– 사용자 입력 문자열 전송

◆ 로그인후 메인 창에서 사용자 입력 대기: actionPerformed 이용

```
public void actionPerformed(ActionEvent e) {  
    if(e.getSource() == idTF){  
        ...  
    }else if(e.getSource() == input){  
        String msg = input.getText();  
        sendMsg(id + ":" + msg + "\n");  
        if(msg.equals("/quit")){  
            try{  
                socket.close();  
            }catch(Exception ex){}  
            sendMsg(id + " 님이 종료합니다.");  
            System.out.println("종료합니다.");  
            System.exit(1);  
        }  
        ...  
    }  
}
```

멀티캐스트 채팅

– 멀티캐스트 수신 메시지 출력

◆ 로그인후 별도의 스레드로 메시지 수신 대기

```
public void actionPerformed(ActionEvent e) {
    if(e.getSource() == idTF){
        id = idTF.getText();
        if(id == null || id.trim().equals("")){
            System.out.println("아이디를 다시 입력하여 주세요.");
            return;
        }
        sendMsg(id + " 님이 입장하였습니다.\n");
        WinInputMulticastThread wit = new WinInputMulticastThread();
        wit.start();
        cardLayout.show(this, "main");
        input.requestFocus();
    }
}
```


멀티캐스트 채팅

– 멀티캐스트 전송 (sendMsg())

```
public void sendMsg(String msg) {  
    byte[] b = new byte[2000];  
    try {  
        b = msg.getBytes(); // 바이트 배열로 만듦  
        schannel = InetAddress.getByName(saddress);  
        spacket = new DatagramPacket(b, b.length, schannel, sport);  
        socket.send(spacket);  
    } catch (IOException e) {  
        e.printStackTrace();  
    }  
}
```

– 현재 코드의 문제점?

- ◆ 사용자의 입장/종료 메시지 전송 시점

– 현재 코드를 같은 머신에서 여러 클라이언트를 실행시킬 수 있도록 변경?