



윤성우의 열혈 TCP/IP 소켓 프로그래밍

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판

Chapter 14. 멀티캐스트 & 브로드캐스트

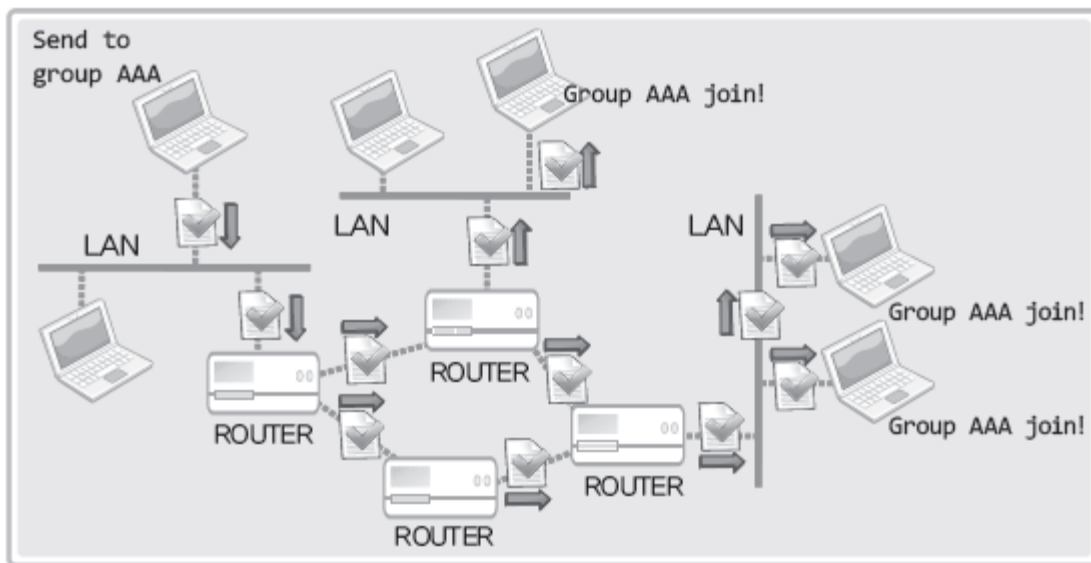


Chapter 14-1. 멀티캐스트(Multicast)

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판

멀티캐스트의 데이터 전송방식과 트래픽 이점

- 멀티캐스트 서버는 특정 멀티캐스트 그룹을 대상으로 데이터를 딱 한번 전송한다.
- 딱! 한번 전송하더라도 그룹에 속하는 클라이언트는 모두 데이터를 수신한다.
- 멀티캐스트 그룹의 수는 IP주소 범위 내에서 얼마든지 추가가 가능하다.
- 특정 멀티캐스트 그룹으로 전송되는 데이터를 수신하려면 해당 그룹에 가입하면 된다.



멀티캐스트 그룹을 대상으로 하는 데이터의 수신을 위해서는 가입의 절차를 거쳐야 한다.

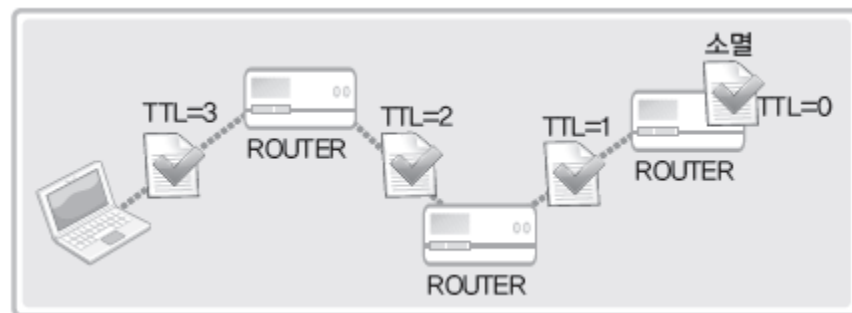
모든 호스트에 데이터 전송을 해야 할 때, TCP 또는 UDP보다 발생하는 트래픽의 양이 적다

멀티캐스트는 연결의 개념이 존재하지 않는다. 따라서 UDP 소켓을 기반으로 전송된다.

라우팅과 TTL

▶ TTL(Time to Live)

- ▶ 패킷을 얼마나 멀리 보낼 것인가를 결정하는 요소
- ▶ TTL은 정수로 표현되며, 라우터를 하나 거칠 때마다 1씩 감소.
- ▶ TTL이 0이 되면, 해당 패킷은 소멸된다.



```

int send_sock;
int time_live=64;
. . . .
send_sock=socket(PF_INET, SOCK_DGRAM, 0);
setsockopt(send_sock, IPPROTO_IP, IP_MULTICAST_TTL, (void*)&time_live, sizeof(time_live));
. . . .
  
```

**TTL을
64로 설정**

TTL의 설정과 관련된 프로토콜의 레벨은 **IPPROTO_IP**이고, 옵션의 이름은 **IP_MULTICAST_TTL**이다.

그룹으로의 가입방법

```
int recv_sock;
struct ip_mreq join_adr;
. . . .
recv_sock=socket(PF_INET, SOCK_DGRAM, 0);
. . . .
join_adr.imr_multiaddr.s_addr="멀티캐스트 그룹의 주소정보";
join_adr.imr_interface.s_addr="그룹에 가입할 호스트의 주소정보";

setsockopt(recv_sock, IPPROTO_IP, IP_ADD_MEMBERSHIP, (void*)&join_adr, sizeof(join_adr));
```

멀티캐스트 그룹의 가입은 소켓의 옵션정보 변경을 통해서 이뤄진다. 그룹의 가입에 사용되는 프로토콜 레벨은 **IPPROTO_IP**이고, 프로토콜의 이름은 **IP_ADD_MEMBERSHIP**이다. 그리고 그룹의 정보는 **ip_mreq** 구조체 변수를 이용해서 표현한다.

```
struct ip_mreq
{
    struct in_addr imr_multiaddr;
    struct in_addr imr_interface;
}
```

멀티캐스트 Sender

news_sender.c의 일부

```
send_sock=socket(PF_INET, SOCK_DGRAM, 0);
memset(&mul_addr, 0, sizeof(mul_addr));
mul_addr.sin_family=AF_INET;
mul_addr.sin_addr.s_addr=inet_addr(argv[1]); // Multicast IP
mul_addr.sin_port=htons(atoi(argv[2])); // Multicast Port
setsockopt(send_sock, IPPROTO_IP,
    IP_MULTICAST_TTL, (void*)&time_live, sizeof(time_live));
if((fp=fopen("news.txt", "r"))==NULL)
    error_handling("fopen() error");
while(!feof(fp)) /* Broadcasting */
{
    fgets(buf, BUF_SIZE, fp);
    sendto(send_sock, buf, strlen(buf),
        0, (struct sockaddr*)&mul_addr, sizeof(mul_addr));
    sleep(2);
}
```

sender에서는 특정 멀티캐스트 그룹을 향해 파일에 저장된 데이터를 전송하고 있다.

이 데이터는 라디오 방송과 마찬가지로 receiver가 그룹에 가입을 하는순간부터 수신을 시작한다.

```
root@my_linux:/tcpip# gcc news_sender.c -o sender
root@my_linux:/tcpip# ./sender
Usage : ./sender <GroupIP> <PORT>
root@my_linux:/tcpip# ./sender 224.1.1.2 9190
```

실행 결과

멀티캐스트 Receiver

news_receiver.c의 일부

```
recv_sock=socket(PF_INET, SOCK_DGRAM, 0);
memset(&adr, 0, sizeof(adr));
adr.sin_family=AF_INET;
adr.sin_addr.s_addr=htonl(INADDR_ANY);
adr.sin_port=htons(atoi(argv[2]));
if(bind(recv_sock, (struct sockaddr*)&adr, sizeof(adr))==-1)
    error_handling("bind() error");

join_adr.imr_multiaddr.s_addr=inet_addr(argv[1]);
join_adr.imr_interface.s_addr=htonl(INADDR_ANY);
setsockopt(recv_sock, IPPROTO_IP,
    IP_ADD_MEMBERSHIP, (void*)&join_adr, sizeof(join_adr));

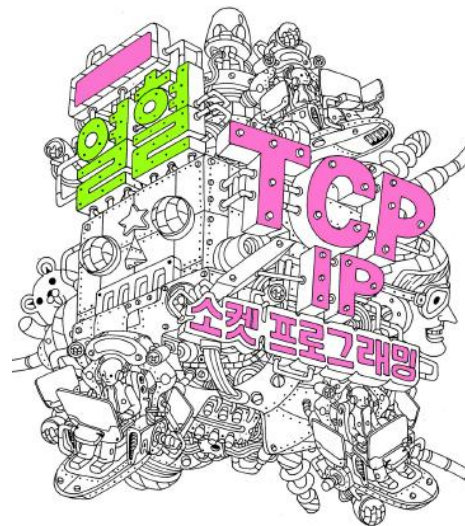
while(1)
{
    str_len=recvfrom(recv_sock, buf, BUF_SIZE-1, 0, NULL, 0);
    if(str_len<0)
        break;
    buf[str_len]=0;
    fputs(buf, stdout);
}
```

receiver가 멀티캐스트 그룹에 가입을하는 순간부터 데이터를 수신하게 된다. 따라서 가입 이전에 sender가 전송한 데이터는 수신할 수 없다.

실행 결과

```
root@my_linux:/tcpip# gcc news_receiver.c -o receiver
root@my_linux:/tcpip# ./receiver
Usage : ./receiver <GroupIP> <PORT>
root@my_linux:/tcpip# ./receiver 224.1.1.2 9190
```

The government, however, apparently overlooked a law that requires a CPA to receive at least two years of practical training at a public accounting firm.



Chapter 14-2. 브로드캐스트

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판

브로드캐스트의 이해와 구현의 방법

▶ 브로드캐스트

- ▶ 동일한 네트워크 내에 존재하는 호스트에게 데이터를 전송하는 방법
- ▶ 데이터 전송의 대상이 호스트가 아닌 네트워크이다.
- ▶ 멀티캐스트와 마찬가지로 **UDP** 소켓을 기반으로 한다.
- ▶ IP에서 네트워크 주소를 제외한 호스트 주소를 모두 1로 해서 전송하면, 해당 네트워크로 데이터가 전송된다 – **Directed 브로드캐스트**
- ▶ 255.255.255.255로 데이터를 전송하면, 전송한 호스트가 속한 네트워크로 데이터가 전송된다 – **Local 브로드캐스트**

브로드캐스트가 가능하려면,
소켓의 **SO_BROADCAST** 옵션정보를 1로 변경해야한다.

```
int send_sock;
int bcast=1;    // SO_BROADCAST의 옵션정보를 1로 변경하기 위한 변수 초기화
. . . .

send_sock=socket(PF_INET, SOCK_DGRAM, 0);
. . . .
setsockopt(send_sock, SOL_SOCKET, SO_BROADCAST, (void*)&bcast, sizeof(bcast));
. . . .
```

브로드캐스트 Sender

```
send_sock=socket(PF_INET, SOCK_DGRAM, 0);
memset(&broad_adr, 0, sizeof(broad_adr));
broad_adr.sin_family=AF_INET;
broad_adr.sin_addr.s_addr=inet_addr(argv[1]);
broad_adr.sin_port=htons(atoi(argv[2]));

setsockopt(send_sock, SOL_SOCKET,
    SO_BROADCAST, (void*)&so_brd, sizeof(so_brd));
if((fp=fopen("news.txt", "r"))==NULL)
    error_handling("fopen() error");

while(!feof(fp))
{
    fgets(buf, BUF_SIZE, fp);
    sendto(send_sock, buf, strlen(buf),
        0, (struct sockaddr*)&broad_adr, sizeof(broad_adr));
    sleep(2);
}
```

옵션의 설정, 그리고 전송에 사용되는 IP의 주소 이외에는 일반적인 UDP 프로그램과 차이가 없다.

```
root@my_linux:/tcpip# gcc news_sender_brd.c -o sender
root@my_linux:/tcpip# ./sender 255.255.255.255 9190
```

실행 결과



브로드캐스트 Receiver

```
recv_sock=socket(PF_INET, SOCK_DGRAM, 0);
memset(&adr, 0, sizeof(adr));
adr.sin_family=AF_INET;
adr.sin_addr.s_addr=htonl(INADDR_ANY);
adr.sin_port=htons(atoi(argv[1]));

if(bind(recv_sock, (struct sockaddr*)&adr, sizeof(adr))==-1)
    error_handling("bind() error");
while(1)
{
    str_len=recvfrom(recv_sock, buf, BUF_SIZE-1, 0, NULL, 0);
    if(str_len<0)
        break;
    buf[str_len]=0;
    fputs(buf, stdout);
}
```

브로드캐스트 Receiver의 경
우는 일반적인 UDP
Receiver와 동일하다.

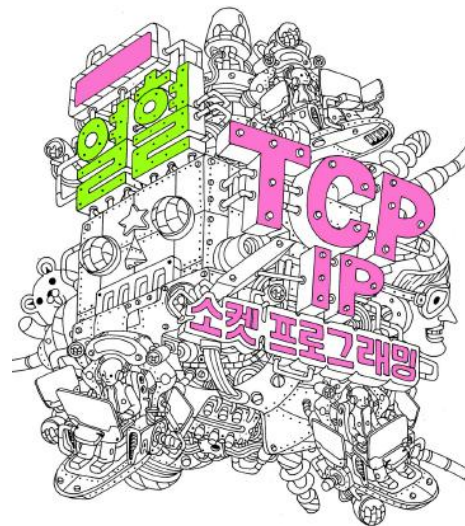
```
root@my_linux:/tcpip# gcc news_receiver_brd.c -o receiver
root@my_linux:/tcpip# ./receiver 9190
accountants say that the committee will in all likelihood tackle
issues that have been previously raised by them with the ministry.
```

실행 결과



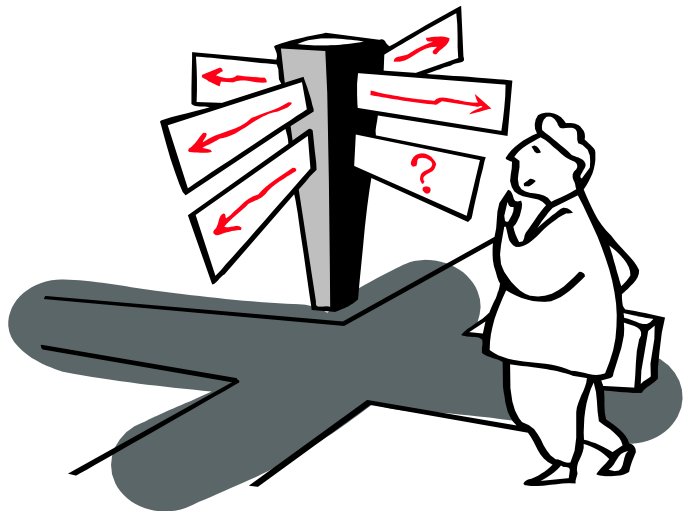
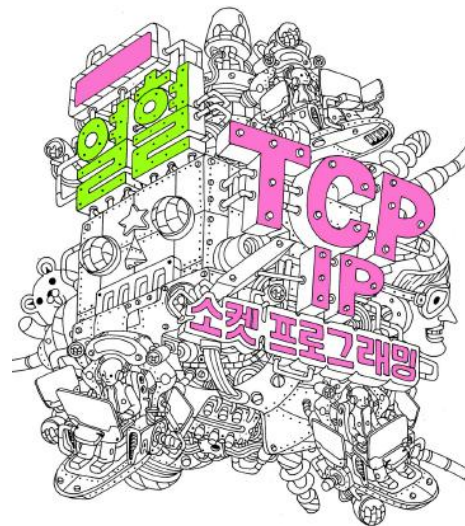


멀티캐스트에 사용되는 `IP_MULTICAST_TTL`과 `ip_mreq`의 사용을 위해서 헤더파일 `<ws2tcpip.h>` 추가하는 것 이외에 차이가 없다. 따라서 추가적인 언급을 생략하겠다.



Chapter 14-3. 윈도우 기반으로 구현하기

윤성우 저 열혈강의 TCP/IP 소켓 프로그래밍 개정판



Chapter 14가 끝났습니다. 질문 있으신지요?