



12 소켓을 이용한 네트워크 프로그래밍

쉽게 배우는 데이터 통신과 컴퓨터 네트워크

3절. 서버-클라이언트 프로그래밍

□ 연결형 서비스

- 소켓: 네트워크 통신을 위한 소프트웨어 교신점
- 서버의 동작 : **Well-known 포트에서 대기**
 - 동작 과정
 1. 서비스 교신점 (호스트의 IP 주소, 포트 번호) 공개
 2. 클라이언트로부터 발생하는 서비스 요구 대기
 3. 클라이언트에 서비스 제공
 4. 해당 클라이언트에 서비스 제공 완료
 5. 단계 2로 이동
- 클라이언트의 동작 : **서버의 Well-known 포트에 접속을 시도**
 - 동작 과정
 1. 원하는 서비스를 제공하는 서버 확인
 2. 해당 서버와 연결 시도
 3. 서버에 서비스 요청
 4. 서버에 서비스 요구 완료



3절. 서버-클라이언트 프로그래밍

□ 연결형 서비스

■ TCP를 이용한 통신 절차 [그림 12-1]

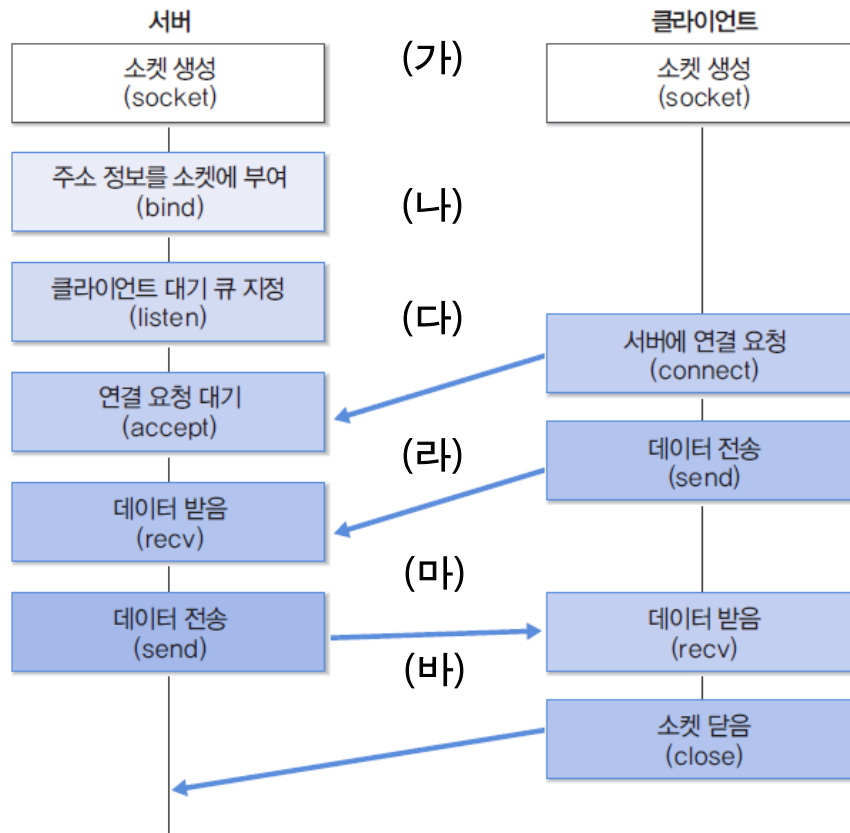


그림 12-3 TCP를 이용한 통신 절차



3절. 서버-클라이언트 프로그래밍

□ 연결형 서비스

- 소켓 함수의 컴파일
- 예) 현재의 시간 정보를 제공하는 서버 프로그램과 서버로부터 시간 정보 서비스를 받아 화면에 출력하는 클라이언트 프로그램
 - 컴파일 명령의 -l 옵션은 소켓 함수를 지원하는 라이브러리를 지칭하는 것임
 - 컴파일이 성공하면 time_client와 time_server라는 실행 파일이 현재 디렉토리에 생성됨
 - 클라이언트와 서버 프로그램을 따로 컴파일 한 후 서버, 클라이언트 순으로 실행

```
% cc -o time_client time_client.c -lsocket -lnsl  
% cc -o time_server timer_server.c -lsocket -lnsl
```

삭제할것

■ 클라이언트 프로그램

- 호스트 서버의 IP 주소 **211.223.201.30**과 포트 번호 5010은 실행 환경에 따라
서 적당한 값으로 변경
- 서버 프로그램
 - 자신의 호스트 주소를 설정하는 부분에 INADDR_ANY를 사용하기 때문에 IP 주
소는 변경할 필요가 없음



3절. 서버-클라이언트 프로그래밍

□ 연결형 서비스

■ 클라이언트 예제

```
# define TIME_SERVER    "211.223.201.30"
# define TIME_PORT      5010

main ()
{
    int sock;
    struct sockaddr_in server;
    char buf [256];

    PF_INET
    sock = socket (AF_INET, SOCK_STREAM, 0);

    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl (inet_addr (TIME_SERVER));
    server.sin_port = htons (TIME_PORT);

    connect (sock, (struct sockaddr *)&server, sizeof(server));

    if (recv (sock, buf, sizeof (buf), 0) == -1)
        exit (1);
    printf ("Time information from server is %s", buf);
    close (sock);
}
```



3절. 서버-클라이언트 프로그래밍

□ 연결형 서비스

■ 서버 예제

```
# define TIME_PORT    5010
main ()
{
    int sock, sock2;
    struct sockaddr_in server, client;
    int len=sizeof(client); /* 수정 */
    char buf [256];
    time_t today;
    PF_INET
    sock = socket (AF_INET, SOCK_STREAM, 0);
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl (INADDR_ANY);
    server.sin_port = htons (TIME_PORT);
    bind (sock, (struct sockaddr *)&server, sizeof (server));
    listen (sock, 5);
    while (1) {
        sock2 = accept (sock, (struct sockaddr *)&client, &len);
        time (&today);
        strcpy (buf, ctime (&today));
        send (sock2, buf, strlen (buf) + 1, 0);
        close (sock2);
    }
}
```



3절. 서버-클라이언트 프로그래밍

□ 비연결형 서비스

- UDP를 이용한 통신 절차 [그림 12-4]

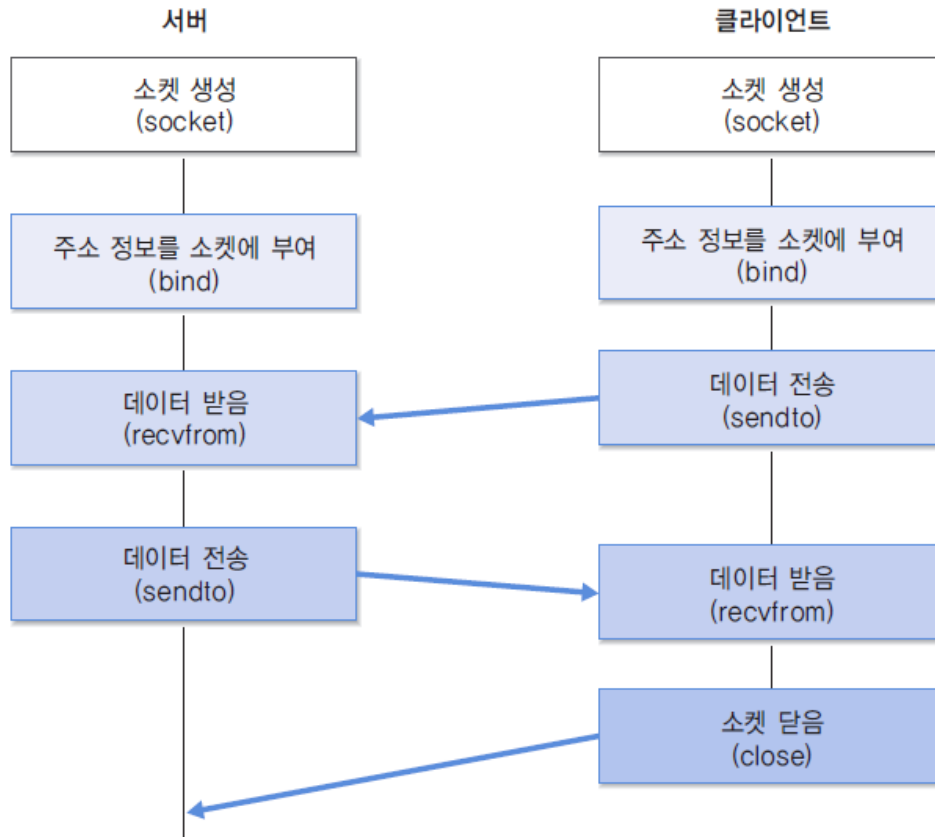


그림 12-4 UDP를 이용한 통신 절차



3절. 서버-클라이언트 프로그래밍

□ 비연결형 서비스

■ 클라이언트 예제

```
# define TIME_SERVER    "211.223.201.30"
# define TIME_PORT      5010
```

```
main ()
```

```
{
```

```
server.sin_family = AF_INET;
```

```
server.sin_addr.s_addr = htonl (net_addr (TIME_SERVER));
```

```
server.sin_port = htons (TIME_PORT);
```

PF_INET

```
sock = socket (AF_INET, SOCK_DGRAM, 0);
```

```
client.sin_family = AF_INET;
```

```
client.sin_addr.s_addr = htonl (INADDR_ANY);
```

```
client.sin_port = htons (0);
```

```
bind (sock, (struct sockaddr *)&client, sizeof (client));
```

```
buf[0] = '?'; buf[1] = '\0';
```

```
buf_len = sendto (sock, buf, strlen(buf) + 1, 0, (struct sockaddr *)&server, server_len);
```

```
if (buf_len < 0) exit (1);
```

```
buf_len = recvfrom (sock, buf, 256, 0, (struct sockaddr *)&server, server_len);
```

```
if (buf_len < 0) exit (1);
```

```
printf ("Time information from server is %s", buf);
```

```
}
```

삭제할것

클라이언트 자신의 주소를 바인딩

서버 주소로 송신하고 수신함

교재 수정

p. 407 프로그램 34줄

(struct sockaddr *)&server, server_len);

3절. 서버-클라이언트 프로그래밍

□ 비연결형 서비스

■ 서버 예제

```
main ()
{
    time_t today;
    PF_INET
    sock = socket (AF_INET, SOCK_DGRAM, 0);

    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl (INADDR_ANY);
    server.sin_port = htons (TIME_PORT);

    bind (sock, (struct sockaddr *)&server, sizeof (server));

    while (1) {
        buf_len = recvfrom (sock, buf, 256, 0, (struct sockaddr *)&client, &client_len);
        if (buf_len < 0)
            exit (1);
        printf ("Server: Got %s\n", buf);

        time (&today);
        strcpy (buf, ctime (&today));
        sendto (sock, buf, strlen (buf) + 1, 0, (struct sockaddr *)&client, client_len);
    }
}
```

교재 수정
p. 409 프로그램 13줄
int client_len=sizeof(client)

서버 자신의 주소를 바인딩

클라이언트 주소로 수신하고 송신함



3절. 서버-클라이언트 프로그래밍

□ 교재에서 오류난 부분

1. 모든 프로그램에서 헤더 파일 추가

- `#include <stdio.h> #include <stdlib.h> #include <string.h>`

2. `htonl(inet_addr(TIME_SERVER))` 수정

p. 402-19, p. 407-16 → `inet_addr(TIME_SERVER)`

3. 오타 : p. 404-13 → `int len=sizeof(client);`

p. 407-30 → `(struct sockaddr*)&server, (int *)server_len);`

4. `sock=socket(AF_INET, SOCK_STREAM, 0)` 수정

p. 402-15, p. 404-17, p. 409-18 → `AF_INET` 를 `PF_INET`

5. 비교 구문의 삽입 : p. 404-24, p. 404-27 → `if` 문내에 `== -1`을 삽입

6. 컴파일시 `-lsocket` 옵션 제거(p. 402)



3절. 서버-클라이언트 프로그래밍

□ 실행시 주의 사항

- Root 권한으로 방화벽 제거 후 실행

□ TCP 프로토콜(서버) : 대기후 시간 정보를 보냄

```
[root@infosec-server server_source]# vi time_server.c
[root@infosec-server server_source]# gcc -o time_server time_server.c -lnsl
[root@infosec-server server_source]# ./time_server
^C
[root@infosec-server server_source]#
```

□ TCP 프로토콜(클라이언트) : 연결 후 시간 정보를 받음

```
[root@infosec-server client_source]# ping 192.168.182.102
PING 192.168.182.102 (192.168.182.102) 56(84) bytes of data.
64 bytes from 192.168.182.102: icmp_seq=1 ttl=64 time=0.264 ms
64 bytes from 192.168.182.102: icmp_seq=2 ttl=64 time=0.201 ms
^C
--- 192.168.182.102 ping statistics ---
2 packets transmitted, 2 received, 0% packet loss, time 1517ms
rtt min/avg/max/mdev = 0.201/0.232/0.264/0.035 ms
[root@infosec-server client_source]# vi time_client.c
[root@infosec-server client_source]# ./time_client
Time information from server is Sat Apr 28 05:38:56 2012
```



```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

```
#include <time.h>
```

```
#define TIME_PORT 5010
```

```
main()
{
```

```
    int sock, sock2;
```

```
    struct sockaddr_in server, client;
```

```
    int len;
    char buf[256];
    time_t today;
```

```
    if((sock = socket(PF_INET, SOCK_STREAM, 0)) == -1)
        exit(1);
```

```
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl(INADDR_ANY);
    server.sin_port = htons(TIME_PORT);
```

```
    if(bind(sock, (struct sockaddr *)&server, sizeof(server)) == -1)
        exit(1);
```

```
    if(listen(sock, 5) == -1)
        exit(1);
```

```
    while(1)
    {
```

```
        if((sock2 = accept(sock, (struct sockaddr *)&client, &len)) == -1)
            exit(1);
```

```
        time(&today);
        strcpy(buf, ctime(&today));
        send(sock2, buf, strlen(buf) + 1, 0);
        close(sock2);
```

```
    }
```

```
}
```

```
#include <stdio.h>
#include <stdlib.h>
#include <string.h>
```

```
#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>
```

```
#define TIME_SERVER "192.168.182.102"
```

```
#define TIME_PORT 5010
```

```
main()
{
```

```
    int sock;
    struct sockaddr_in server;
```

```
    char buf[256];
```

```
    if((sock = socket(PF_INET, SOCK_STREAM, 0)) == -1)
        exit(1);
```

```
    server.sin_family = AF_INET;
    server.sin_addr.s_addr = inet_addr(TIME_SERVER);
    server.sin_port = htons(TIME_PORT);
```

```
    if(connect(sock, (struct sockaddr *)&server, sizeof(server)))
        exit(1);
```

```
    if(recv(sock, buf, sizeof(buf), 0) == -1)
        exit(1);
```

```
    printf("Time information from server is %s\n", buf);
```

```
    close(sock);
```

```
}
```



3절. 서버-클라이언트 프로그래밍

□ UDP 프로토콜(클라이언트)

□ ? 를 보내고 시간 정보를 받음

```
[root@infosec-server client_source]# vi udp_time_client.c
[root@infosec-server client_source]# gcc -o udp_time_client udp_time_client.c -l
nsl
[root@infosec-server client_source]# ./udp_time_client
Time information from server is Sat Apr 28 09:20:58 2012
[root@infosec-server client_source]#
```

□ UDP 프로토콜(서버)

□ ? 를 받고 시간 정보를 보냄

```
[root@infosec-server server_source]# vi udp_time_server.c
[root@infosec-server server_source]# gcc -o udp_time_server udp_time_server.c -l
nsl
[root@infosec-server server_source]# ./udp_time_server
Server : Got ?
```



```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#include <time.h>

#define TIME_PORT 5010

main()
{
    int sock;
    struct sockaddr_in server, client;

    int server_len;
    int client_len = sizeof(client);

    char buf[256];

    int buf_len;

    time_t today;

    if((sock = socket(PF_INET, SOCK_DGRAM, 0)) == -1)
        exit(1);

    server.sin_family = AF_INET;
    server.sin_addr.s_addr = htonl(INADDR_ANY);
    server.sin_port = htons(TIME_PORT);

    if(bind(sock, (struct sockaddr *)&server, sizeof(server)))
        exit(1);

    while(1)
    {
        buf_len = recvfrom(sock, buf, 256, 0, (struct sockaddr *)&client,
        &client_len);

        if(buf_len < 0)
            exit(1);

        printf("Server : Got %s\n", buf);

        time(&today);
        strcpy(buf, ctime(&today));

        sendto(sock, buf, strlen(buf) + 1, 0, (struct sockaddr *)&client,
        client_len);
    }
    close(sock);
}

```

```

#include <stdio.h>
#include <stdlib.h>
#include <string.h>

#include <sys/types.h>
#include <sys/socket.h>
#include <netinet/in.h>

#define TIME_SERVER "192.168.182.102"
#define TIME_PORT 5010

main()
{
    int sock;

    struct sockaddr_in server, client;

    int server_len = sizeof(server);
    char buf[256];
    int buf_len;

    server.sin_family = AF_INET;
    server.sin_addr.s_addr = inet_addr(TIME_SERVER);
    server.sin_port = htons(TIME_PORT);

    if((sock = socket(PF_INET, SOCK_DGRAM, 0)) == -1)
        exit(1);

    client.sin_family = AF_INET;
    client.sin_addr.s_addr = htonl(INADDR_ANY);
    client.sin_port = htons(0);

    if(bind(sock, (struct sockaddr *)&client, sizeof(client)) < 0)
        exit(1);

    buf[0] = '?';
    buf[1] = '\0';

    buf_len = sendto(sock, buf, sizeof(buf) + 1, 0, (struct sockaddr *)&server,
    server_len);

    if(buf_len < 0)
        exit(1);

    buf_len = recvfrom(sock, buf, 256, 0, (struct sockaddr *)&server, &server_len);

    if(buf_len < 0)
        exit(1);

    printf("Time information from server is %s", buf);

    close(sock);
}

```

