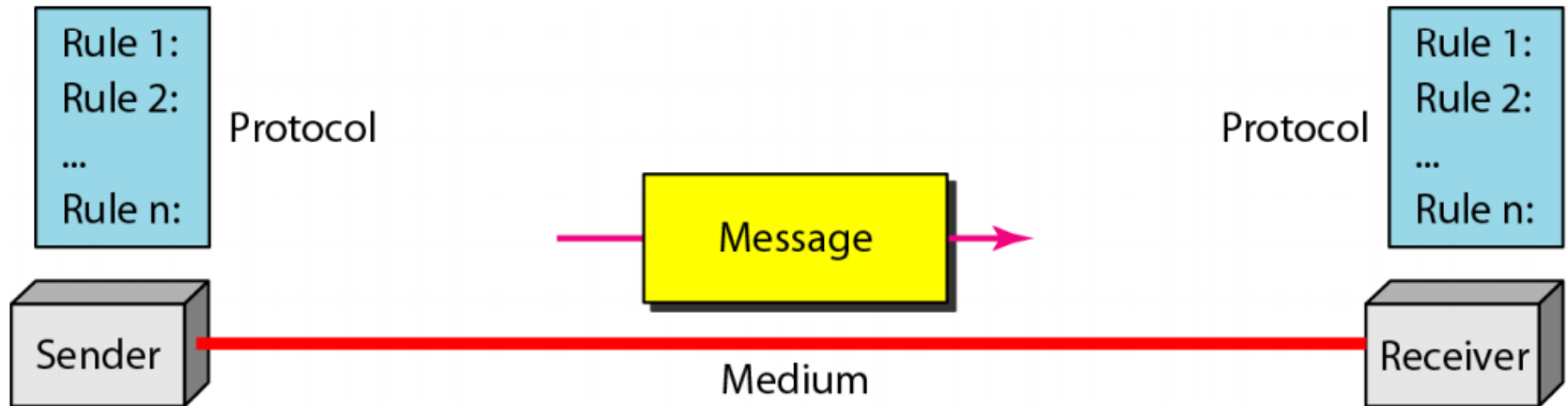


C언어로 구현하는 TCP/IP 소켓 프로그래밍

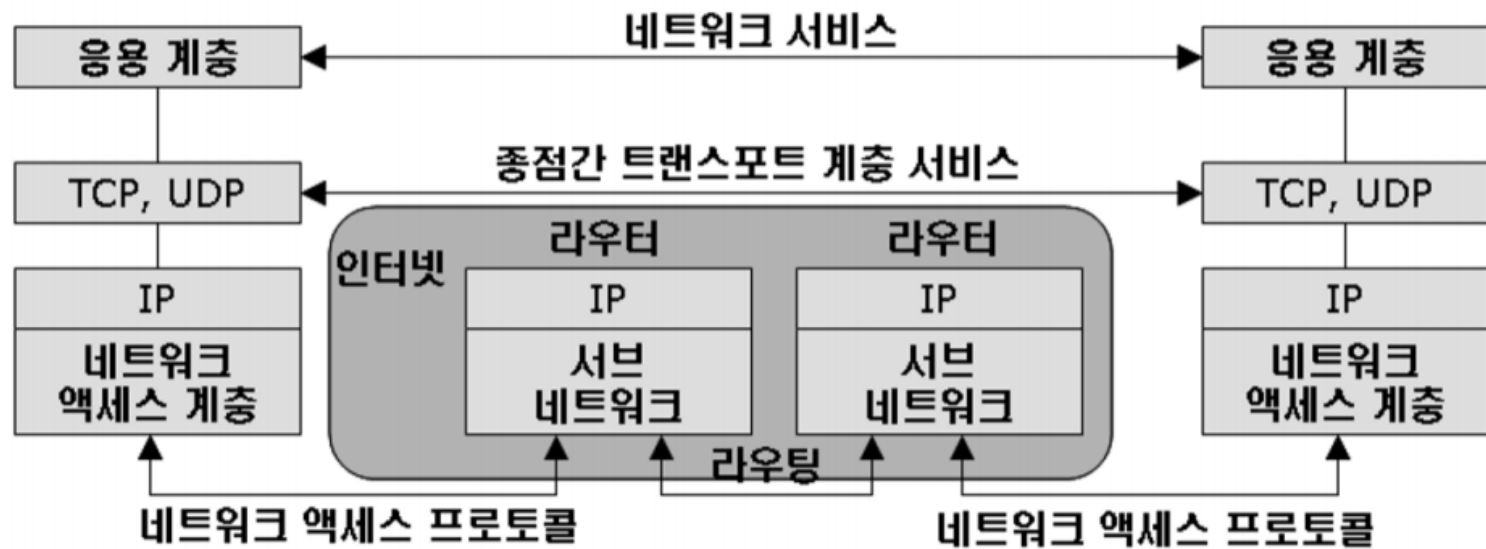


통신의 구성요소

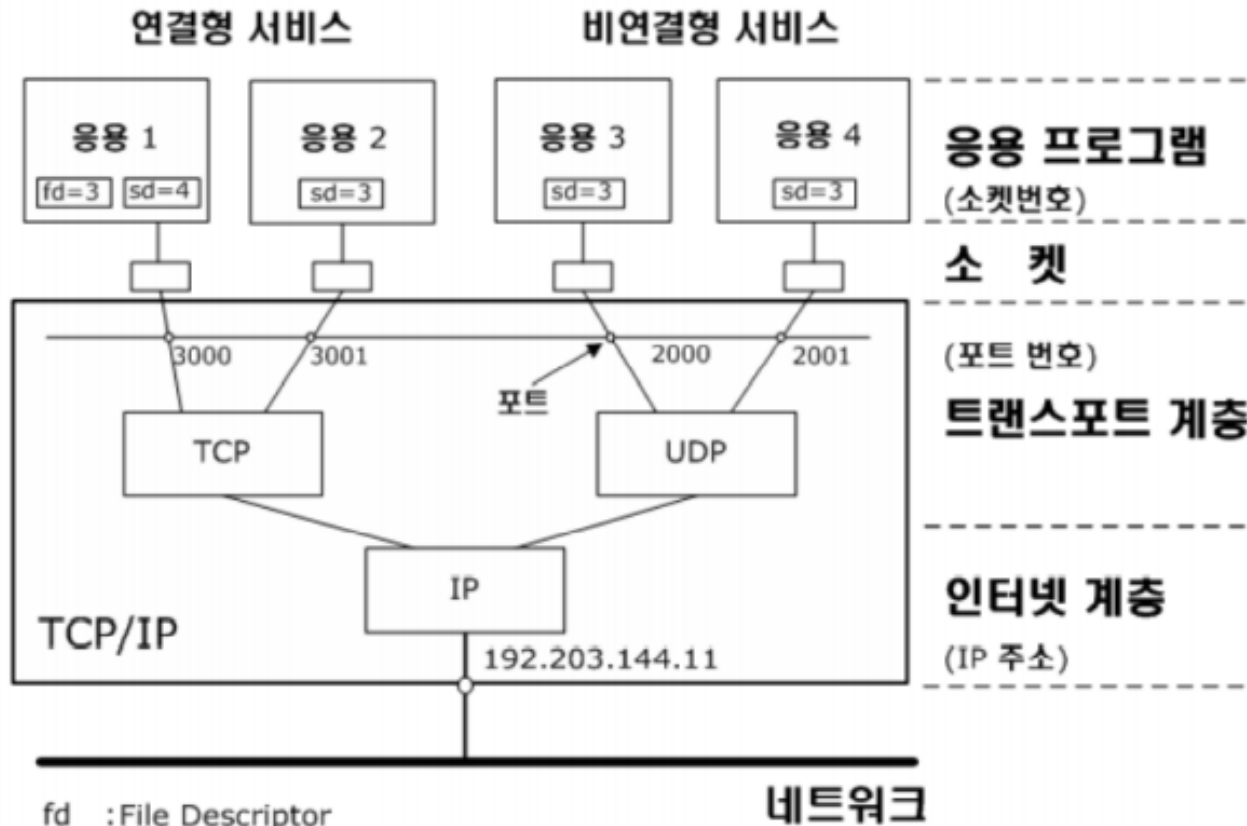


- 송신자(Sender): 메시지를 보내는 장치 ex. 컴퓨터, 전화기, 비디오 카메라 등
- 수신자(Receiver): 메시지를 받는 장치 ex. 컴퓨터, 전화기, TV
- 전송매체(Medium): 송신자에서 수신자까지 메시지를 전달하는 물리적인 경로
ex) 꼬임선(twisted pair wire), 동축선(coaxial cable), 광케이블(fiber-optic cable), 마이크로파 등
- 프로토콜(Protocol): 데이터 통신 수행 규칙들의 집합(상호 합의)
- 메시지(Message): 전송되는 정보(데이터) ex. 문자, 숫자, 소리, 영상, 그림, 이들의 조합

통신의 구성요소



통신의 구성요소

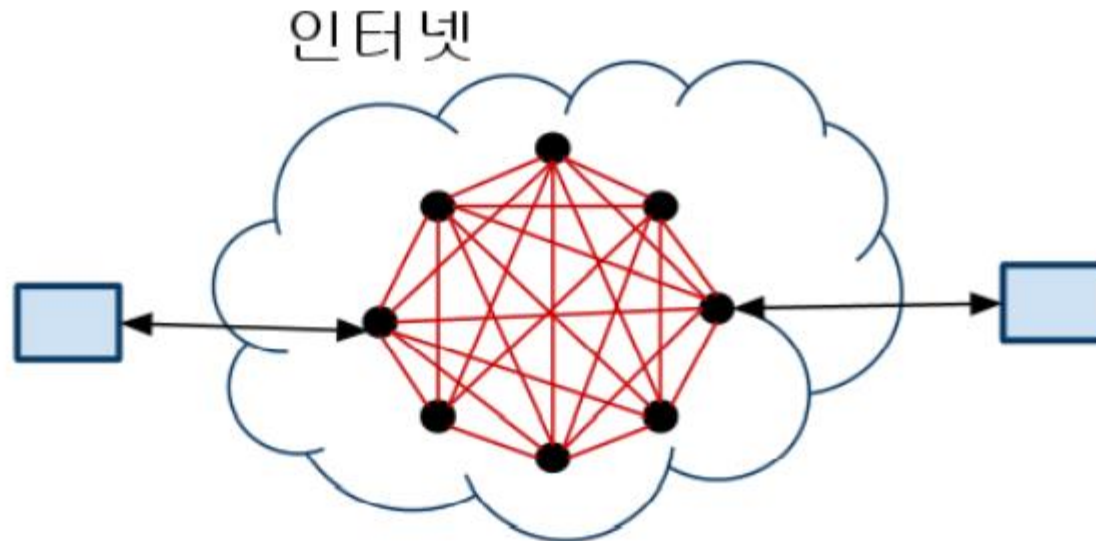


fd : File Descriptor
sd : Socket Descriptor
IP : Internet Protocol
TCP : Transmission Control Protocol
UDP : User Datagram Protocol

네트워크 개념

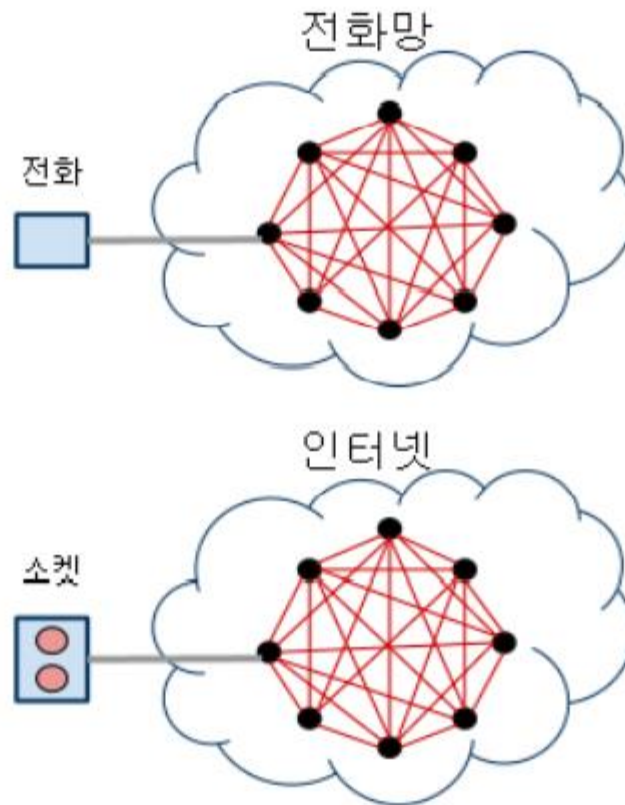
- 인터넷 구성요소

- 컴퓨터 노드 : 송신자 혹은 수신자의 역할
- 인터넷 : 전령 혹은 매체의 역할



네트워크 개념

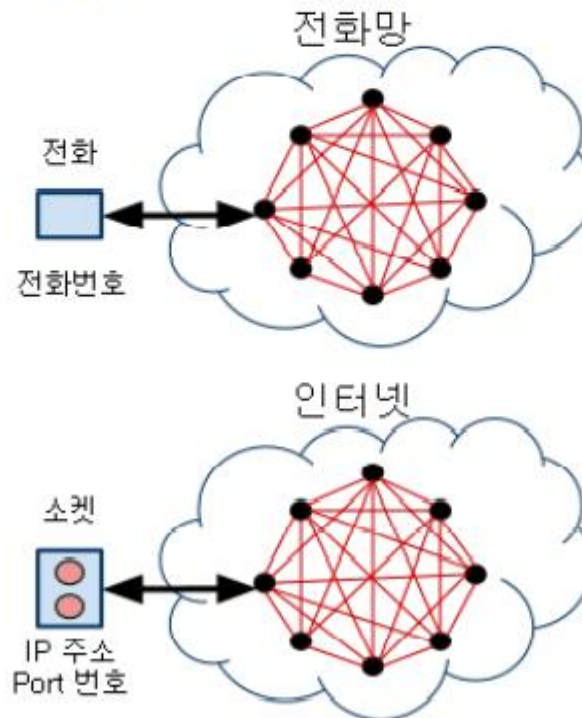
- 전화망과 동일한 흐름
- **소켓** : 전화기의 역할을 함. 인터넷과의 접점 혹은 관문



네트워크 개념

- Node를 인터넷에 연결하기

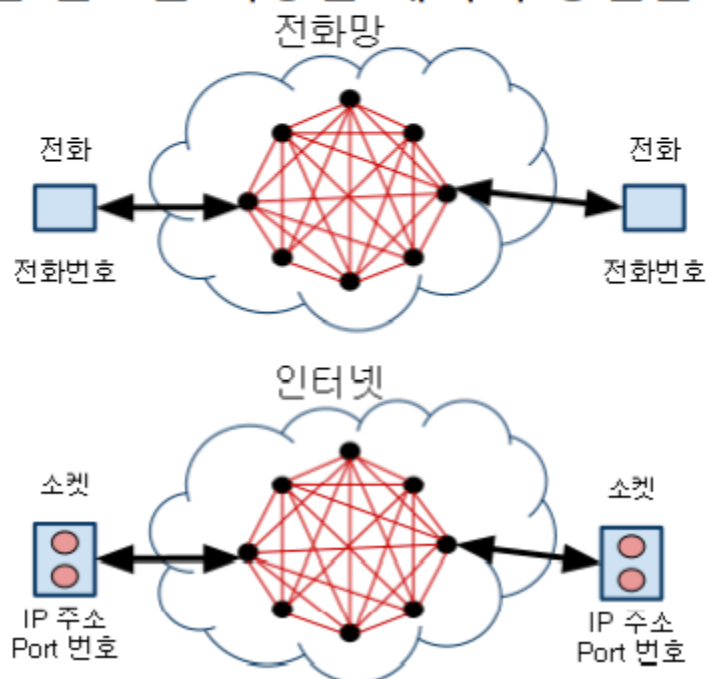
- 소켓에 전체 인터넷 노드에서 유일하게 식별가능한 주소(IP)를 부여한다.
- 프로그램을 찾을 수 있도록 포트 번호를 bind 한다.



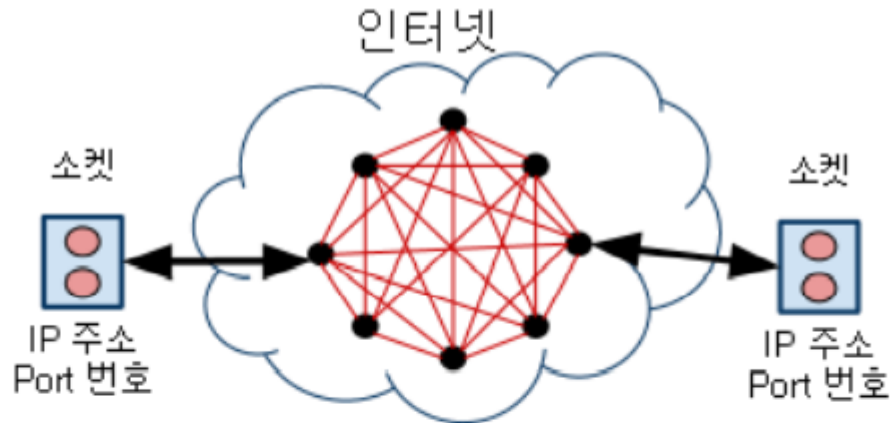
네트워크 개념

- 원격지의 Node 찾기

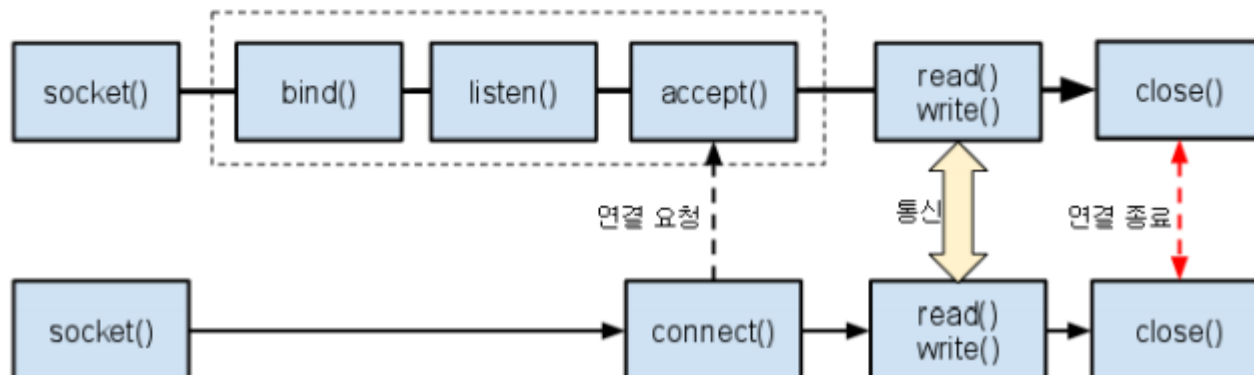
- 원격지의 Node도 IP 주소와 Port 번호를 가지고 있다.
 - IP를 이용해서 컴퓨터의 위치를 찾고
 - Port 번호를 이용해서 프로그램을 찾는다.
- 만들어진 통신 선로를 이용한 데이터 통신을 한다.



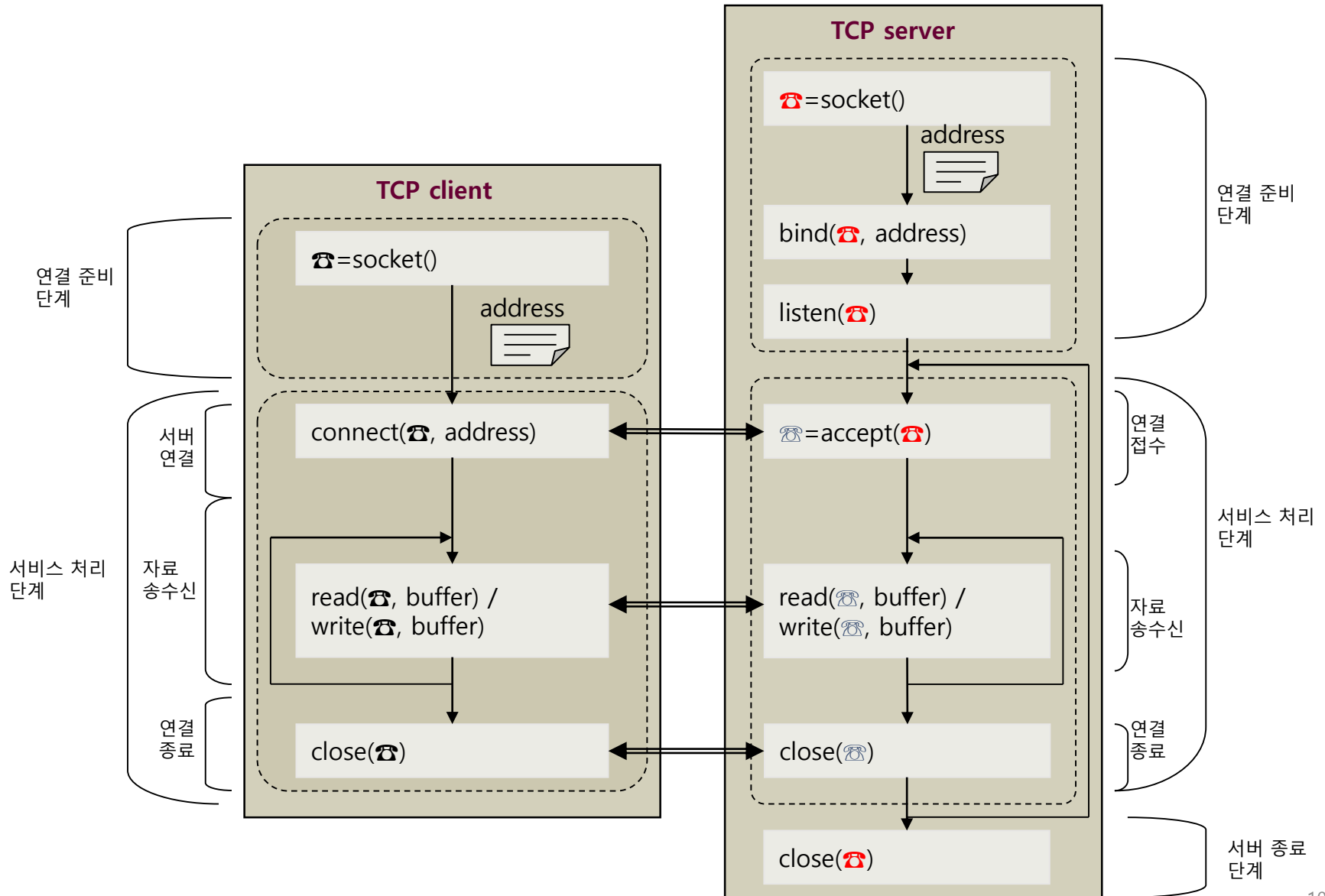
네트워크 개념



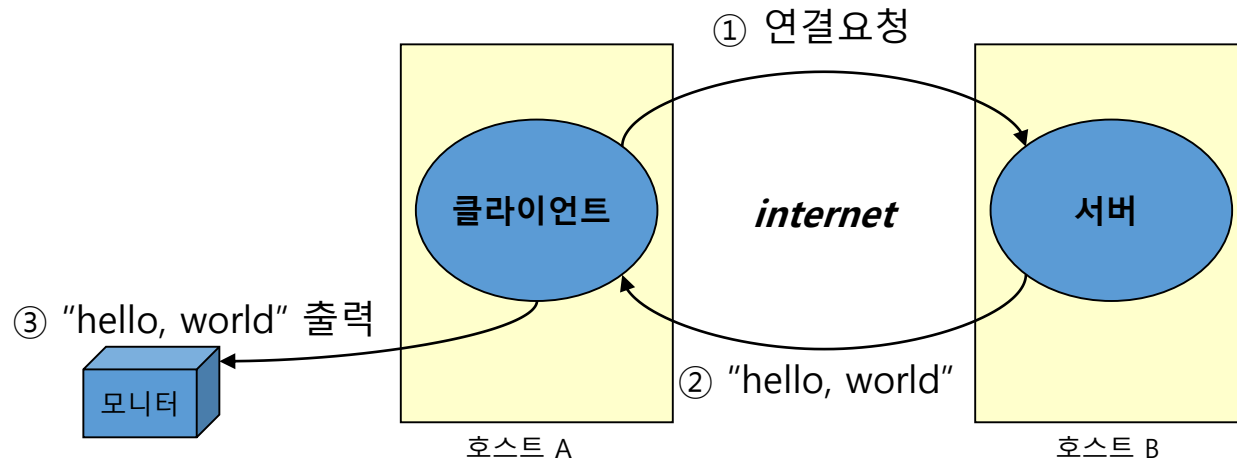
• 서버와 클라이언트 네트워크 프로그램 흐름



TCP 소켓 프로그래밍



hello, world 출력 소켓 프로그램



- | | | |
|----------|----------------------------|----------------|
| ① 연결요청 | : 서버 프로그램에 연결 요청 | (클라이언트 -> 서버) |
| ② 문자열 전송 | : 문자열 'hello, world' 전송 | (서버 -> 클라이언트) |
| ③ 화면 출력 | : 문자열 'hello, world' 화면 출력 | (클라이언트 -> 모니터) |

hello, world 출력 소켓 프로그램 / 서버

```
1. #include <stdio.h>
2. #include <string.h>
3. #include <arpa/inet.h>
4. #include <sys/socket.h>

5. #define PORT 9001

6. char buffer[BUFSIZ] = "Hello World\n";

7. int main(void){

8.     int s_socket, c_socket;
9.     struct sockaddr_in s_addr, c_addr;
10.    int len;
11.    int n;

12.    s_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP); ①
13.
14.    memset(&s_addr, 0, sizeof(s_addr)); ②
15.    s_addr.sin_addr.s_addr = htonl(INADDR_ANY);
16.    s_addr.sin_family = AF_INET;
17.    s_addr.sin_port = htons(PORT);
```

```
20. if(bind(s_socket, (struct sockaddr *)&s_addr, sizeof(s_addr)) == -1){ ③
21.     printf("Can not Bind\n");
22.     return -1;
23. }

24. if(listen(s_socket, 5) == -1){ ④
25.     printf("listen Fail\n");
26.     return -1;
27. }

28. while(1){
29.     printf("Server waiting...\n");
30.     len = sizeof(c_addr);
31.     c_socket = accept(s_socket, (struct sockaddr*)&c_addr, &len); ⑤
32.     printf("Connected IP : %s\n", inet_ntoa(c_addr.sin_addr));
33.
34.     n = strlen(buffer); ⑥
35.     write(c_socket, buffer, n);
36.
37.     close(c_socket); ⑦
38. }

39.
40. close(s_socket); ⑧
41. return 0;
42. }
```

① 소켓을 생성

② 연결 요청을 수신할 주소 설정

③ 소켓을 주소와 포트에 연결

④ 수신대기열 생성

⑤ 클라이언트 연결 요청 수신

⑥ 클라이언트 요청 서비스 제공

⑦ 클라이언트와 연결 종료

⑧ 서버 종료

- s_socket : 클라이언트의 연결 요청을 처리하는 듣기소켓
- c_socket : 연결된 클라이언트의 소켓과 직접 통신하는 연결소켓

hello, world 출력 소켓 프로그램 / 클라이언트

```
1. #include <stdio.h>
2. #include <string.h>
3. #include <arpa/inet.h>
4. #include <sys/socket.h>

5. #define PORT 9001
6. #define IPADDR "127.0.0.1"

7. int main(void){

8.     int c_socket;
9.     struct sockaddr_in c_addr;
10.    int len;
11.    int n;

12.    char rcvBuffer[BUFSIZ];

13.    c_socket = socket(AF_INET, SOCK_STREAM, IPPROTO_TCP); ①
14.
15.    memset(&c_addr, 0, sizeof(c_addr)); ②
16.    c_addr.sin_addr.s_addr = inet_addr(IPADDR);
17.    c_addr.sin_family = AF_INET;
18.    c_addr.sin_port = htons(PORT);
```

```
19.    if(connect(c_socket, (struct sockaddr*)&c_addr, sizeof(c_addr)) == -1){ ③
20.        printf("Can not connect %Wn");
21.        perror("Error:");
22.        close(c_socket);
23.        return -1;
24.    }

25.    if((n = read(c_socket, rcvBuffer, sizeof(rcvBuffer))) < 0){ ④
26.        return -1;
27.    }

28.    rcvBuffer[n] = '\0'; ⑤
29.    printf("received Data : %sWn", rcvBuffer);

30.    close(c_socket); ⑥

31.    return 0;
32. }
```

- ① 소켓을 생성
- ② 연결할 서버의 주소 설정
- ③ 소켓을 서버에 연결
- ④, ⑤ 서비스 요청과 처리
- ⑥ 소켓 연결을 종료

참고문헌

- 정석용의 TCP/IP 소켓 프로그래밍, 정석용 저, 프리렉
- 뇌를 자극하는 TCP/IP 소켓 프로그래밍, 윤상배 저, 한빛미디어
- 열혈강의 TCP/IP 소켓 프로그래밍, 윤성우 저, 오렌지미디어
- 리눅스 네트워크 & 시스템 프로그래밍, 한국소프트웨어진흥원