네트워크 프로그래밍

Socket

[2013년 10월 14일]

숭실대학교 ICN 연구실 최종석
jschoi@ssu.ac.kr

목차

I 네트워크 개요

II 소켓 개요

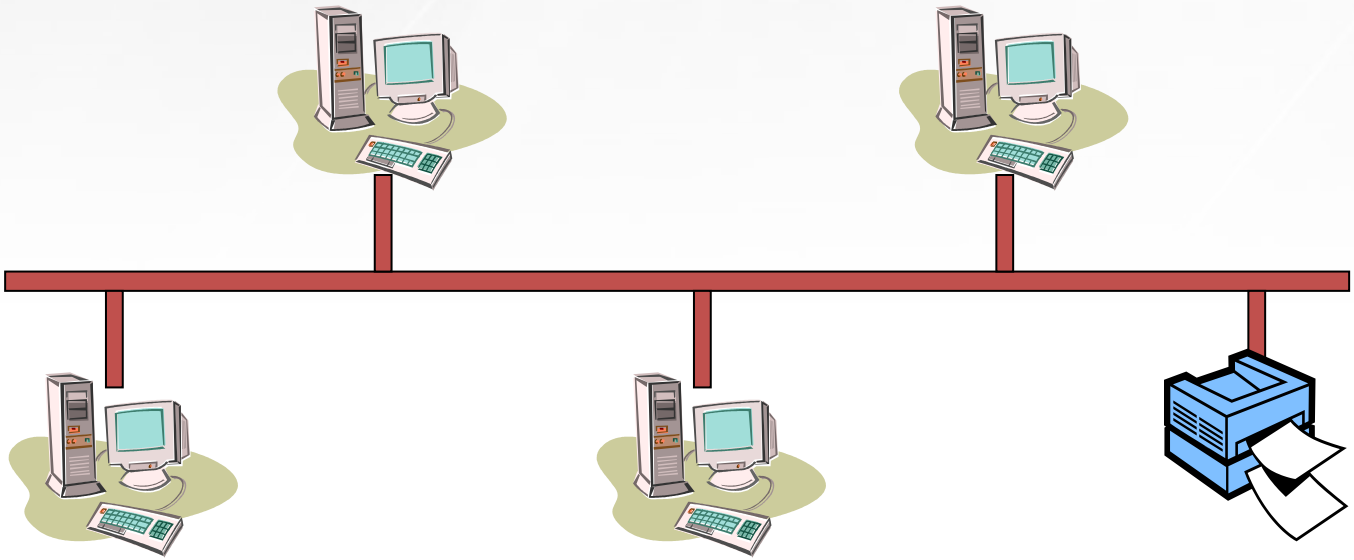
III 소켓 프로그래밍(TCP, UDP)
JAVA , C

1. 네트워크 개요

- **네트워크(Network)**

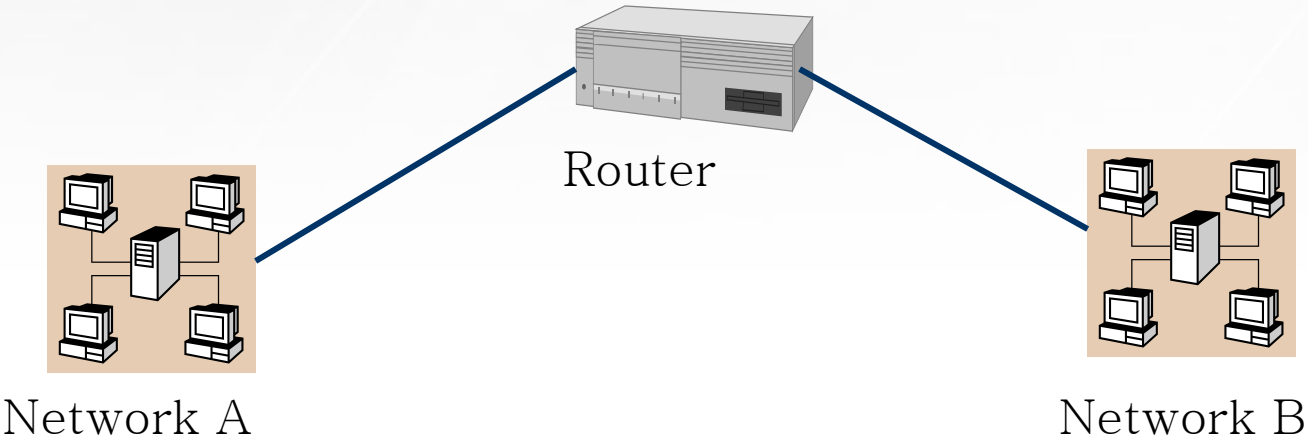
  - 네트워크란 각 호스트(End-System)들을 연결하는 하나의 단일 System을 의미

    ex) 개인용 PC, 워크스테이션, 스마트폰, 프린터 등



1. 네트워크 개요

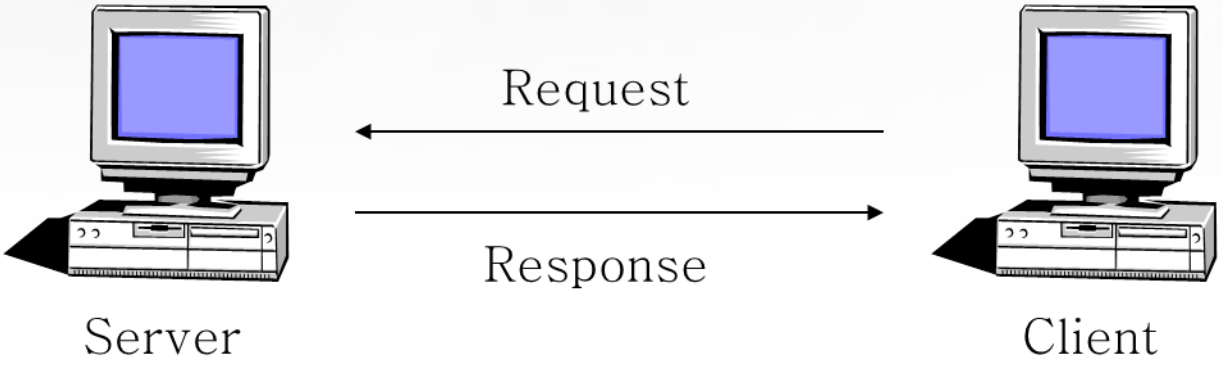
- **인터넷(Internet)**
  - 서로 멀리 떨어진 둘 이상의 네트워크가 연결된 거대한 네트워크를 의미
  - 인터넷을 구축하기 위해서는 서로 다른 네트워크를 연결하는 장비(=라우터)가 필요



1. 네트워크 개요

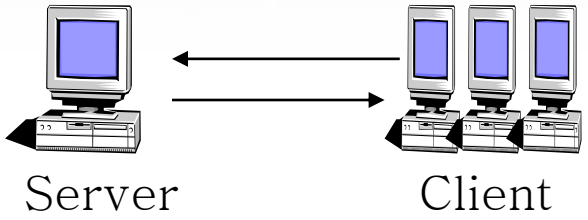
- **클라이언트/서버(Client/Server) 모델**
  - **Server/Client = 단일 프로그램**
  - **Server는 Client의 연결요청 대기 → 정보 및 서비스 제공**

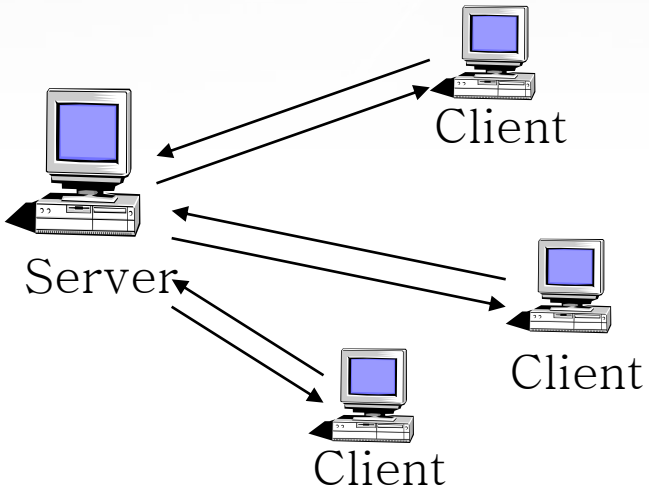
Client는 Server에 정보 및 서비스의 제공을 요청하고 응답을 기다리는 호스트를 의미



1. 네트워크 개요

- **서버 종류**
  - Server는 Client에게 서비스를 제공하기 때문에 Client에 비해 복잡한 제어와 구조를 포함
  - Server 종류
    : Iterative server : 서비스를 한 순간에 하나의 클라이언트에게 제공
    : Concurrent Server : 서비스를 동시에 여러 클라이언트에게 제공





1. 네트워크 개요

- **네트워크 프로그래밍**

  - 원거리 사용자간의 원활하고 빠른 의사 소통을 위해 활용
  - 네트워크로 연결되어 있는 두 호스트간의 데이터 송수신
  - 파일 입/출력과의 차이점은 데이터를 주고 받는 대상
  - 소켓(Socket)을 사용하여 프로그래밍

    : 원격 호스트를 연결시켜 주는 매개체 역할을 수행

    사용 예 → Messenger, On-line 게임, ftp, telnet 등 다양한 분야에 사용

2. 소켓 개요

- **What is socket?**

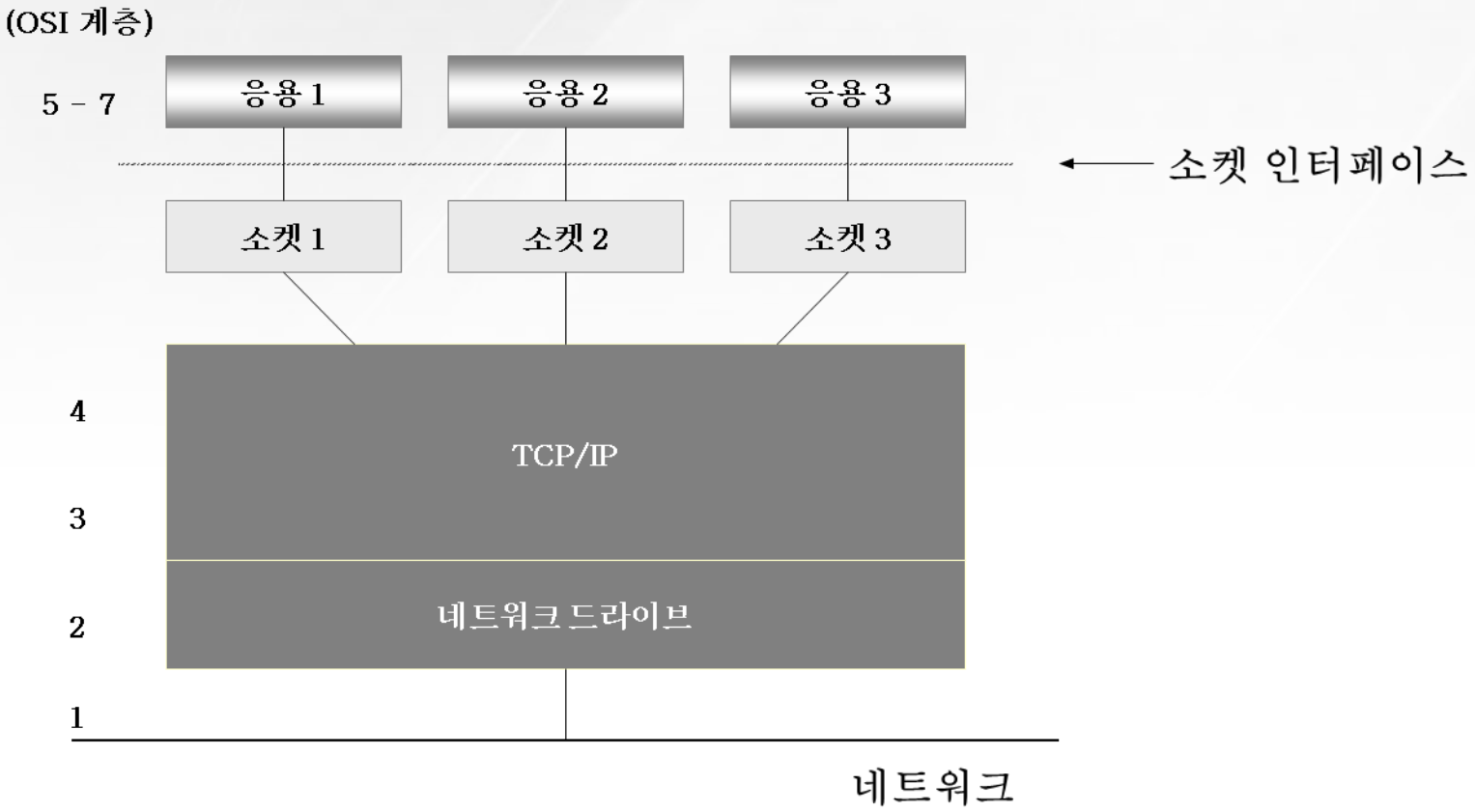
  - 소켓(socket)은 네트워크를 통한 입/출력을 하기 위해 사용자에게 필요한 수단을 제공하는 응용 프로토콜 인터페이스

  - 소켓을 활용한 네트워크 응용 프로그램을 통해 네트워크상에서 데이터를 송/수신

  - 네트워크 입/출력을 위한 요소
    - : 프로토콜(Protocol)
    - : 소스 IP 주소(Source IP Address)
    - : 소스 포트 번호(Source Port Address)
    - : 목적지 IP 주소(Target IP Address)
    - : 목적지 포트 번호(Target Port Address)

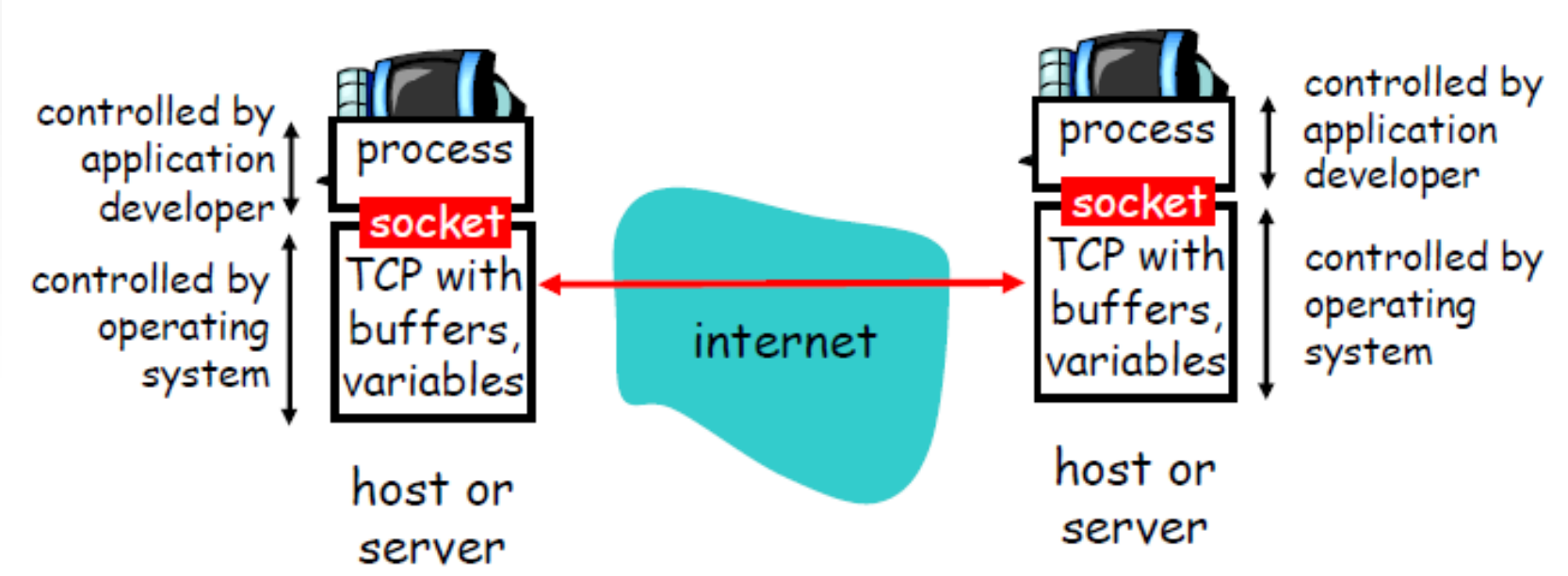
2. 소켓 개요

- **Socket 인터페이스 위치**



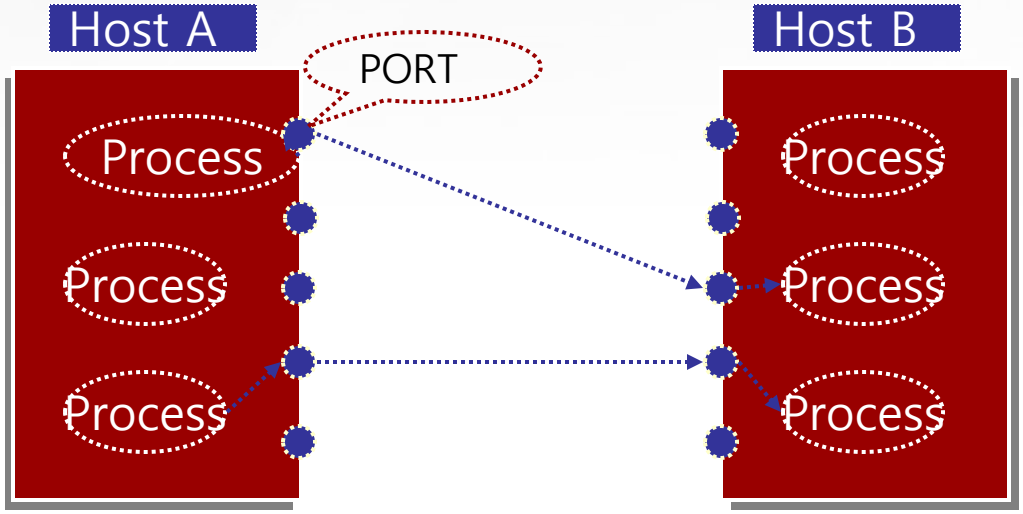
2. 소켓 개요

- 응용 프로그램과 소켓



2. 소켓 개요

- **포트**
  - 호스트 내에 실행되고 있는 프로세스(Process)를 구분 짓기 위한 16비트의 논리적 할당
  - 값의 범위 : 0 ~ 65536



2. 소켓 개요

- well known 포트

  - TCP 또는 UDP 에서 쓰이는 0 번부터 65535 번까지의 포트(port) 번호 중에서 IANA(Internet Assigned Numbers Authority)에 의해서 할당된 0 번부터 1023 번까지의 포트

  http://www.iana.org/assignments/port-numbers

Keyword	Decimal	Description
ftp-data	20/tcp	File Transfer [Default Data]
ftp	21/tcp	File Transfer [Control]
telnet	23/tcp	Telnet
smtp	25/tcp	Simple Mail Transfer
nameserver	42/tcp	Host Name Server
ni-ftp	47/tcp	NI FTP
tftp	69/tcp	Trivial File Transfer
http	80/sctp	HTTP
pop3	110/tcp	Post Office Protocol - Version 3
sftp	115/tcp	Simple File Transfer Protocol

2. 소켓 개요

- 연결 지향형 소켓(SOCK_STREAM, TCP 소켓)

- SOCK_STREAM 소켓 유형

  스트림 방식의 소켓 생성

  UNIX의 파이프 개념과 동일

  연결형(스트림) 서비스 선택 시 사용

- SOCK_STREAM 소켓의 특성

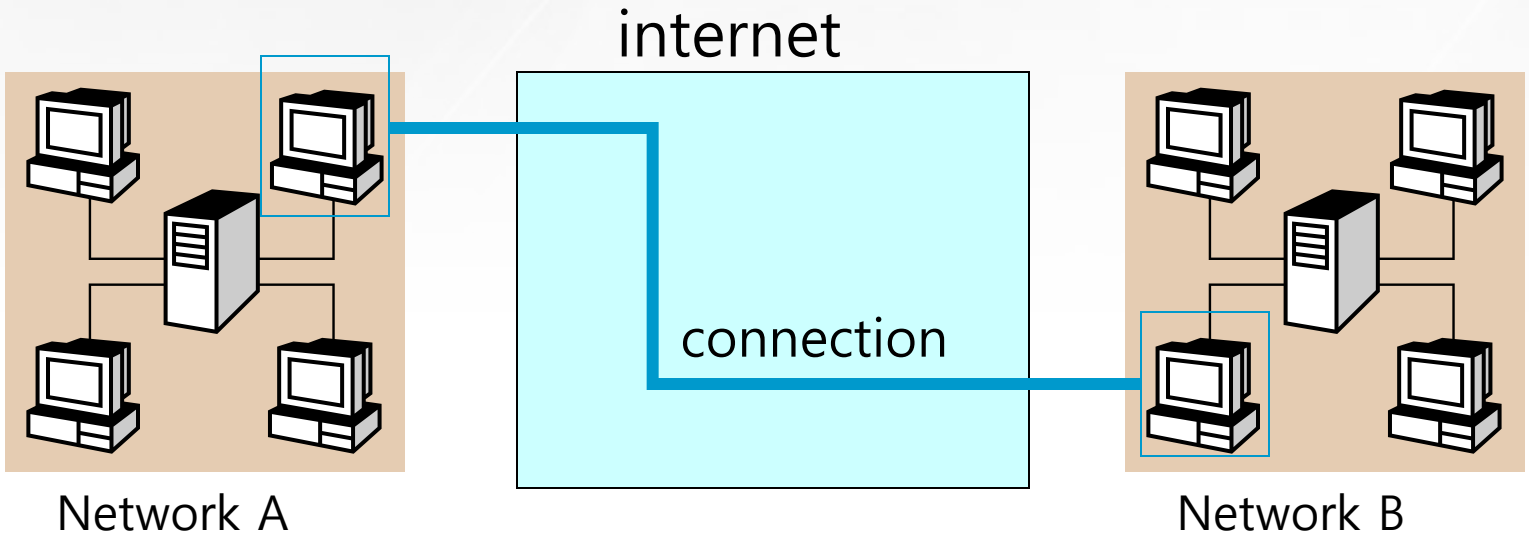
  메시지 경계가 유지되지 않음

  전달된 순서대로 수신됨

  전송된 모든 데이터는 에러없이 원격지에 도달

2. 소켓 개요

- 연결 지향형 소켓(SOCK_STREAM, TCP 소켓)



2. 소켓 개요

- 연결 지향형 소켓(SOCK_STREAM, TCP 소켓)

- SOCK_STREAM 소켓 유형

  스트림 방식의 소켓 생성

  UNIX의 파이프 개념과 동일

  연결형(스트림) 서비스 선택 시 사용

- SOCK_STREAM 소켓의 특성

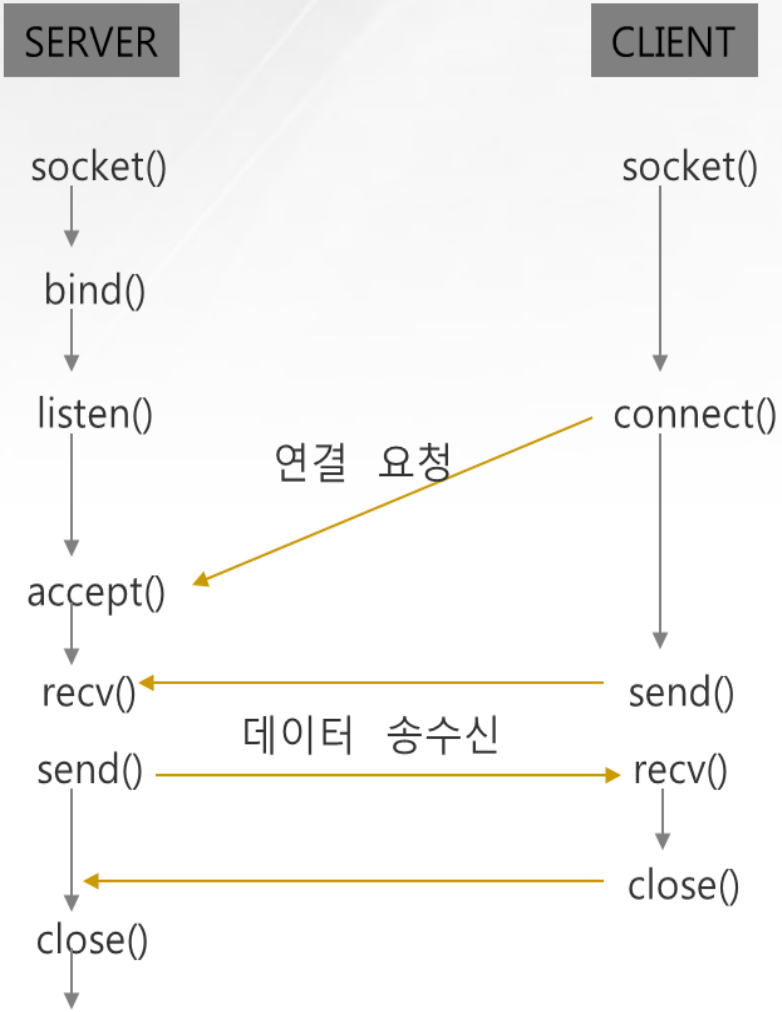
  메시지 경계가 유지되지 않음

  전달된 순서대로 수신됨

  전송된 모든 데이터는 에러없이 원격지에 도달

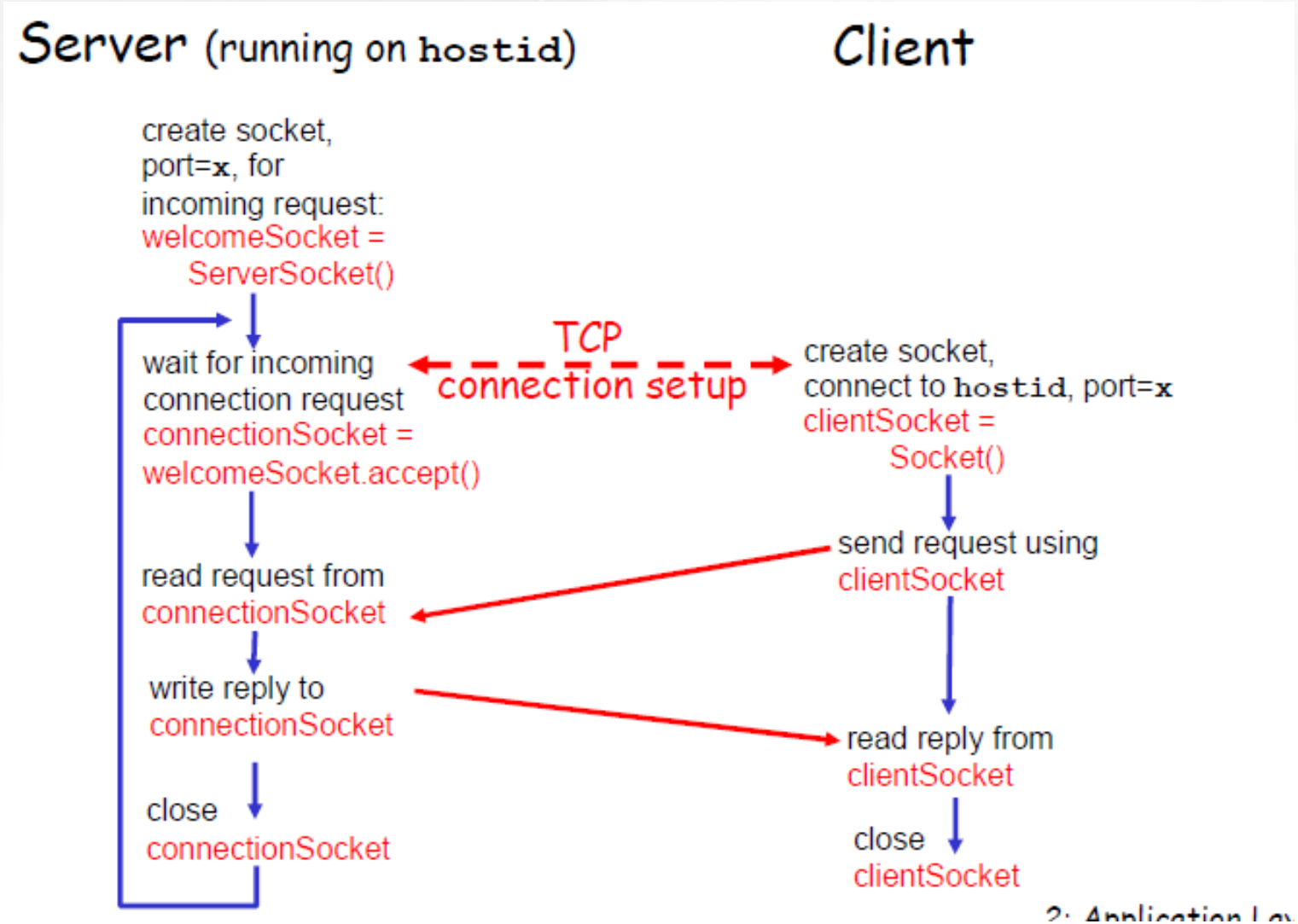
2. 소켓 개요

- **TCP 소켓 – JAVA 와 C Socket 비교(C Socket)**



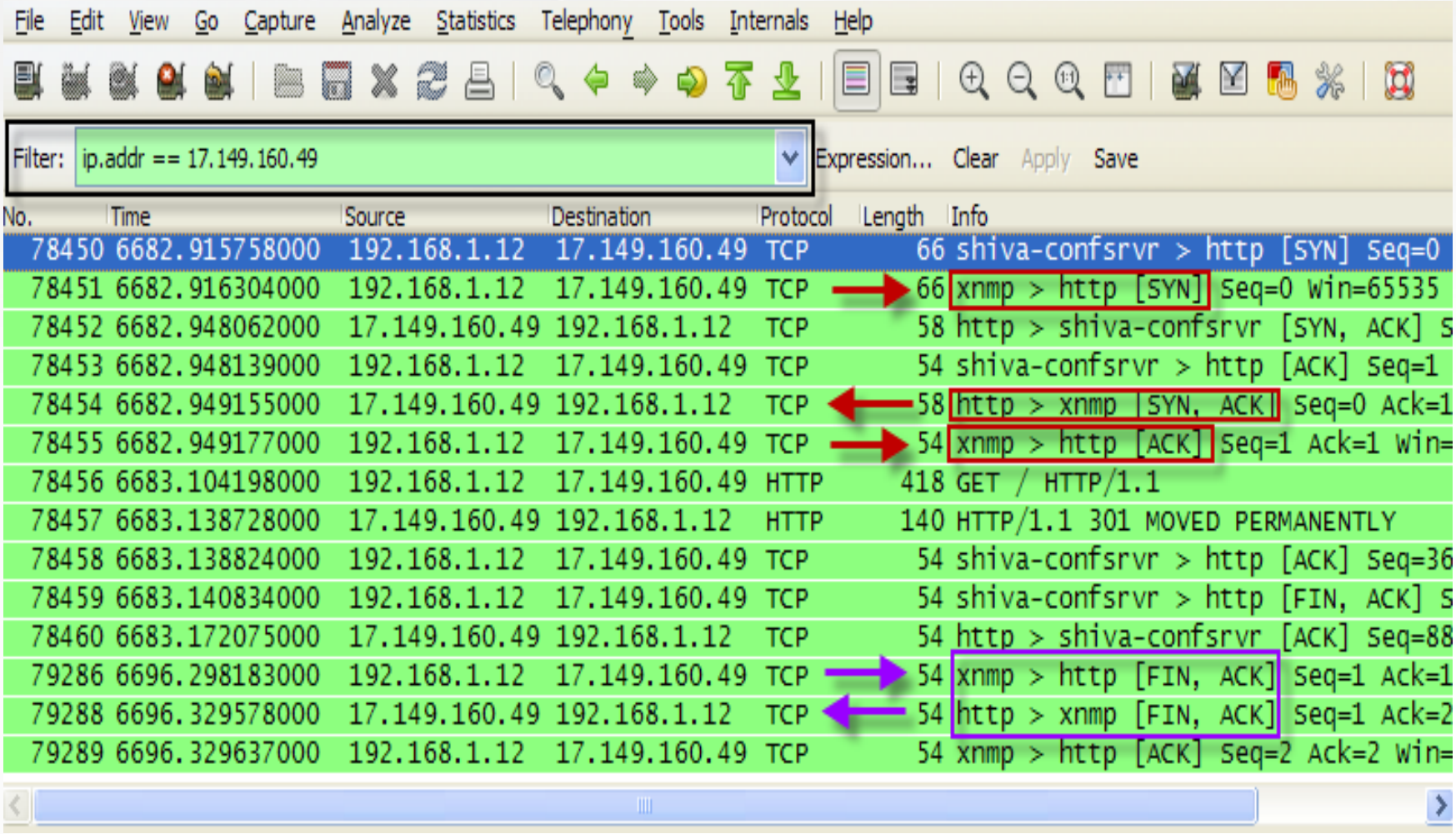
2. 소켓 개요

- **TCP 소켓 – JAVA 와 C Socket 비교(JAVA Socket)**



2. 소켓 개요

- 연결 지향형 소켓(SOCK_STREAM, TCP 소켓)
  - 3 way hand shaking



2. 소켓 개요

- 비연결 지향형 소켓(SOCK_DGRAM, UDP 소켓)

- SOCK_DGRAM 소켓 유형

  데이터그램 방식의 소켓 생성

  개별적으로 주소가 쓰여진 패킷 전송 시 사용

  비연결형(데이터그램) 서비스 선택 시 사용

- SOCK_DGRAM 소켓 유형의 특성

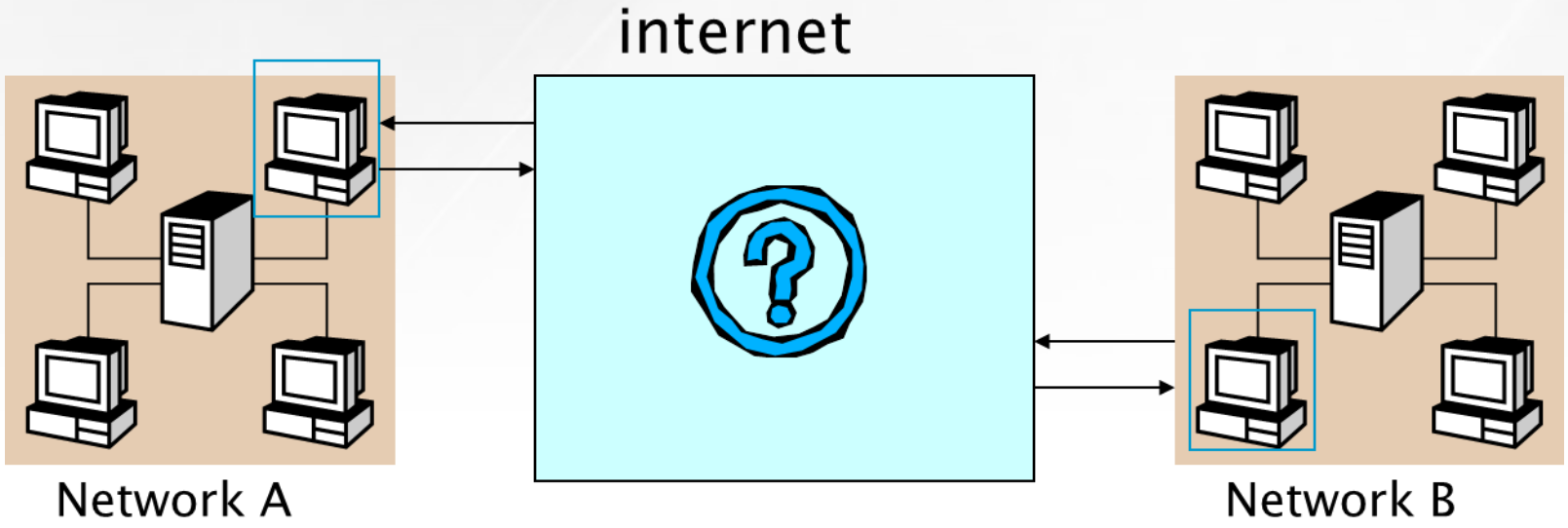
  패킷은 전달된 순서대로 수신되지 않음

  에러복구를 하지 않음 (즉, 신뢰성이 없음)

  데이터그램 패킷의 크기 제한

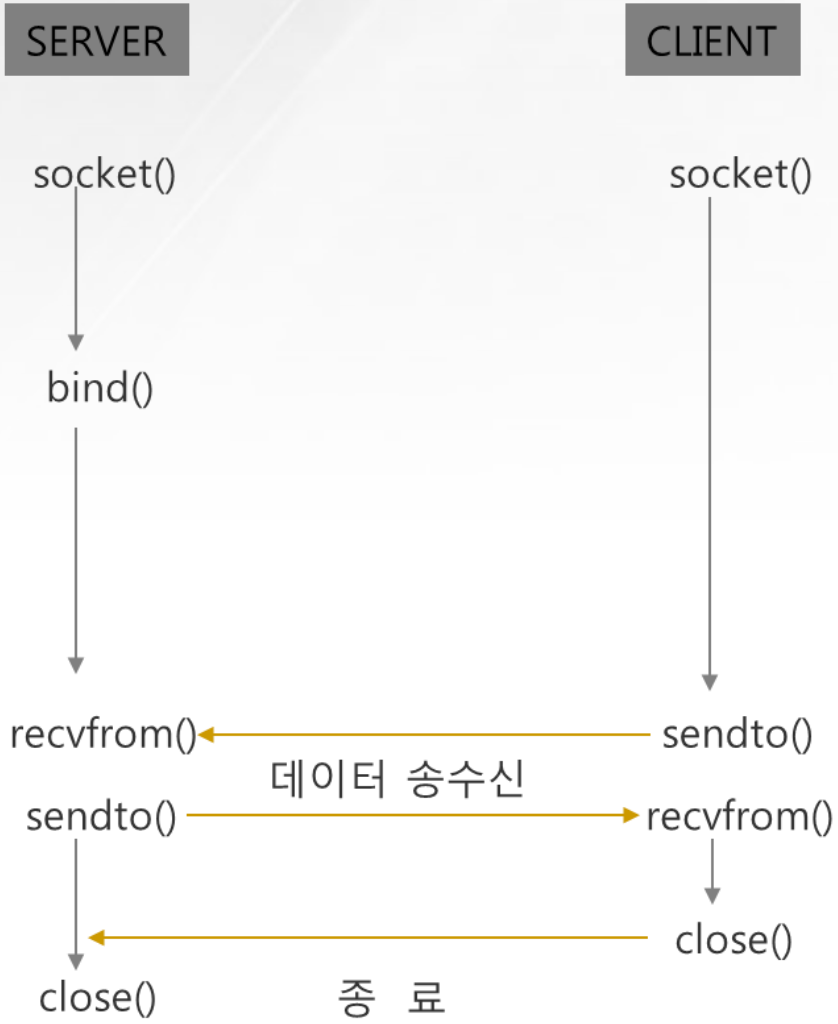
2. 소켓 개요

- 비연결 지향형 소켓(SOCK_DGRAM, UDP 소켓)



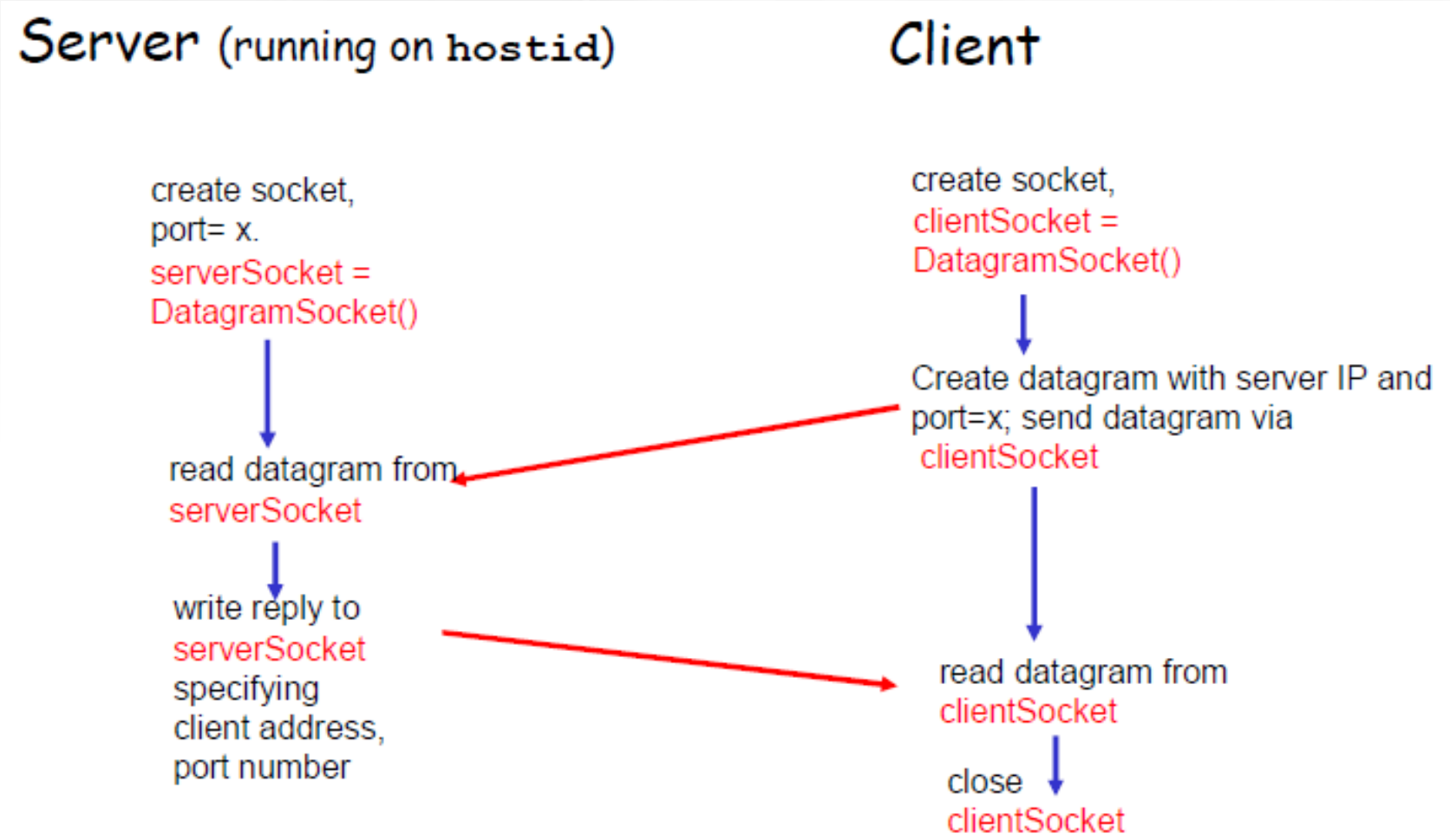
2. 소켓 개요

- **UDP 소켓 – JAVA 와 C Socket 비교**



2. 소켓 개요

- **UDP 소켓 – JAVA 와 C Socket 비교(JAVA Socket)**



3. 소켓 프로그래밍(TCP) with C

- 소켓 함수
  - 소켓 프로그래밍에서 통신 창구 역할
  - 통신을 위한 end-point 생성
  - 소켓 번호 리턴

  - 소켓 생성(socket())

```
int socket( int domain,int type, int protocol );
```

    - int domain
      - PF_INET : 인터넷 프로토콜 체계 사용
      - PF_INET6 : IPv6 프로토콜 체계 사용
      - PF_UNIX : 유닉스 방식의 프로토콜 체계 사용
    - int type
      - SOCK_STREAM : 스트림 방식의 소켓 생성 (TCP)
      - SOCK_DGRAM : 데이터그램 방식의 소켓 생성 (UDP)
      - SOCK_RAW : 사용자가 직접 정의
    - int protocol
      - 구체적인 프로토콜을 정의 할때 사용
      - 대부분의 응용 프로그램에서 '0' 사용

3. 소켓 프로그래밍(TCP) with C

- 소켓 함수
  - 소켓 연결(connect())
    - 서버에 접속 요청
    - 서버의 IP 주소와 포트 번호가 포함되어 있는 소켓 구조체

```
int connect( int sockfd,                          // 소켓 번호
             struct (sockaddr*) &serv_addr,       // 서버 소켓 구조체
             int addr_len );                      // 구조체 크기
```

sockfd : 소켓 기술자

*serv_addr : 접속하고자 하는 서버의 정보가 있는 소켓 구조체

addr_len : 구조체 sockaddr의 크기

3. 소켓 프로그래밍(TCP) with C

- 소켓 함수
  - 주소 할당(bind())
    - 소켓 번호와 소켓 주소 연결
      - 소켓 번호 : 응용 프로그램
      - 소켓 주소 : 네트워크 시스템
    - 클라이언트와 서버가 통신하기 위해

```
int bind( int sockfd,                          // 소켓 번호
        (struct sockaddr*) &sv_addr,           // 서버 소켓 구조체
        int addr_len );                        //구조체 크기
```

sockfd : socket()이 반환된 소켓 기술자
*sy_addr : 소켓 주소체로 자신의 주소와 프로토콜, 포트 번호를 지정한 후 사용
addr_len : 구조체 sockaddr의 크기

3. 소켓 프로그래밍(TCP) with C

- 소켓 함수
  - 연결요청 대기 상태(listen())
    - 클라이언트로부터 연결 요청을 기다리는 수동 대기 모드

```
int listen( int sockfd,          // 소켓 번호
        int log );               // 연결을 기다리는 최대 클라이언트 수
```

sockfd : 소켓 기술자
log : 접속 요청을 동시에 받아들일 수 있는 큐의 크기를 지정하며, n이 5일 경우 서버는 동시에 5개의 클라이언트 접속 요청을 받을 수 있다.

3. 소켓 프로그래밍(TCP) with C

- 소켓 함수
  - 연결 요청 수락(accept())
    - 소켓에 연결을 받아 들임
    - 새로운 소켓 번호 생성 >> 클라이언트와 통신

```
int accept( int sckfd,                          // 소켓 번호
            struct sockaddr   *addr,      // 클라이언트 소켓 구조체
            int *addr_len );                    // 구조체 크기
```

sockfd : 소켓 기술자
*addr : 접속을 허가해 준 클라이언트에 대한 소켓 구조체로 클라이언트의 정보를 알 수 있다.
addr_len : 구조체 sockaddr의 크기

3. 소켓 프로그래밍(TCP) with C

- 소켓 함수
  - 데이터 전송(send())
    - TCP 소켓을 통해 데이터를 송신

```
int send( int sockfd,          // 소켓 번호
        char *buf,             // 전송할 메세지가 저장된 버퍼
        int buflen,            // 버퍼 길이
        int flags );           // 송신 방법 (보통 0을 사용)
```

sockfd : 소켓 기술자
*buf : 전송할 데이터가 저장된 버퍼의 포인터
buflen : buf의 크기
flags : 데이터에 대한 처리 요구 지정(대개의 경우, 0)

3. 소켓 프로그래밍(TCP) with C

- 소켓 함수
  - 데이터 수신(recv())
    - TCP 소켓으로부터 데이터를 수신

```
int recv( int sockfd,          // 소켓 번호
        char *buf,             // 수신 메시지를 저장할 버퍼
        int buflen,            // 버퍼 길이
        int flags );           // 수신 방법 (보통 0을 사용)
```

sockfd : 소켓 기술자
*buf : 수신 데이터를 저장할 버퍼의 포인터
buflen : buf의 크기
flags : 데이터에 대한 처리 요구 지정(대개의 경우, 0)

3. 소켓 프로그래밍(TCP) with C

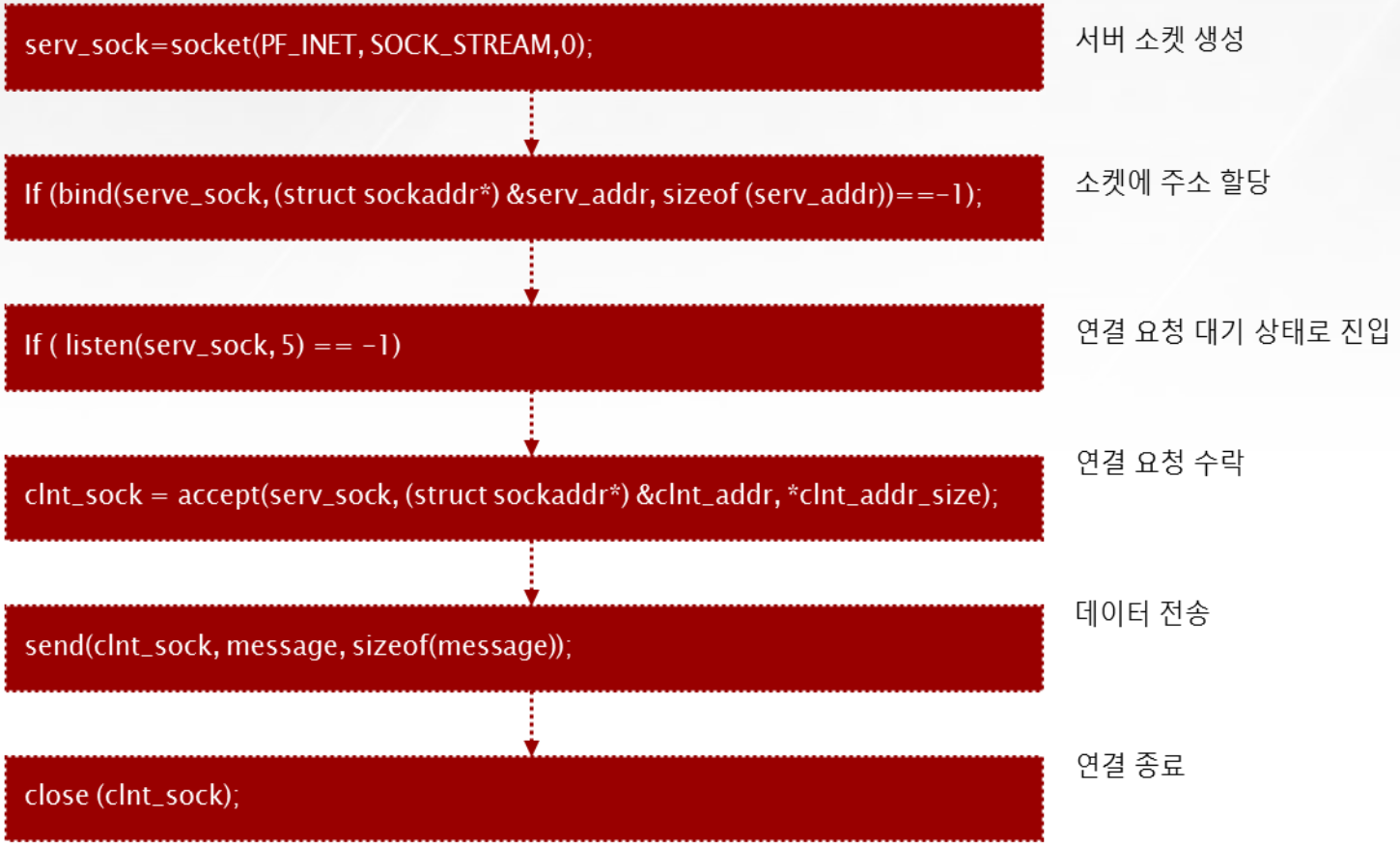
- 소켓 함수
  - 소켓 종료(closesocket())
    - 소켓을 종료
    - Socket()에 의해 생성된 sockft와 관련된 접속을 닫는다. 대부분의 경우 클라이언트가 서버와의 접속을 끊을 대 사용하며, 네트워크 입출력에서는 반드시 close()를 통해 소켓에 대한 작업을 종료한다.

```
int close( int sockfd );                    //  닫을 소켓 번호
```

sockfd : socket()이 반환된 소켓 기술자

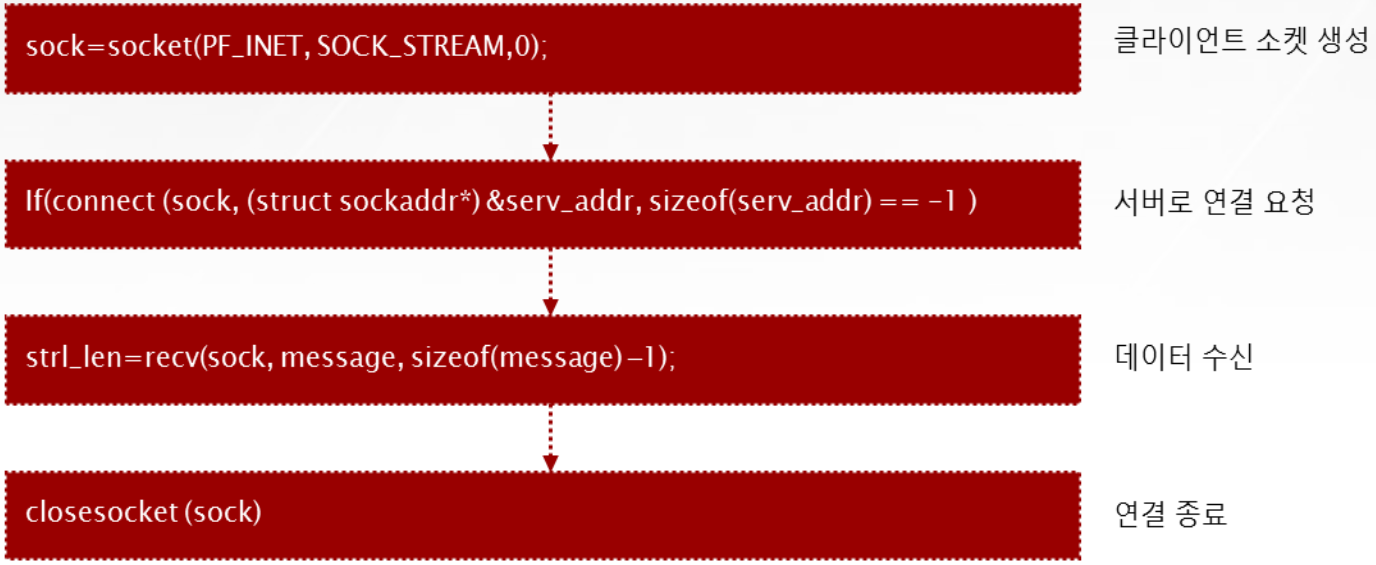
3. 소켓 프로그래밍(TCP) with C

- “Hello World!” 프로그램 – 서버



3. 소켓 프로그래밍(TCP) with C

- “Hello World!” 프로그램 – 클라이언트



3. 소켓 프로그래밍(UDP) with C

- 소켓 함수
  - 데이터 전송(sendto())

```
int sendto (int sock, const void *msg, int len, unsigned flags
                      const struct sockaddr *addr, int addrlen)
```

리턴값 : 성공 시 전송된 바이트 수, 실패 시 -1

sock: 소켓의 파일 디스크립터.
msg: 전송하고자 하는 데이터를 저장해 놓은 버퍼
len: 보낼 데이터의 크기
flags: 옵션(일반적으로 0)
addr: 전송하고자 하는 호스트의 소켓주소 구조체
addrlen: 소켓주소 구조체(addr)의 크기

3. 소켓 프로그래밍(UDP) with C

- 소켓 함수
  - 데이터 수신(recvfrom())

```
int recvfrom (int sock, const void *buf, int len, unsigned flags
                        struct sockaddr *addr, int *addrlen)
```

리턴값 : 성공 시 수신한 바이트 수, 실패 시 -1

sock: 데이터를 수신할 소켓의 파일 디스크립터
buf: 수신할 데이터를 저장할 버퍼
len: 수신 할 수 있는 최대 바이트 수
flags: 옵션
addr: 전송한 호스트의 소켓주소 구조체
addrlen: addr이 가리키는 구조체 변수의 크기

3. 소켓 프로그래밍 – JAVA 레포트

- 레포트
  - 내용 : JAVA를 사용하여 TCP HelloWorld 프로그램 구현하기
  - 내용 : JAVA를 사용하여 UDP 1: 1 채팅 프로그램 구현하기
  - 제출방법 : 10월 30일(수요일), 정보과학관 407호 앞 레포트 박스
  - 작성한 TCP, UDP프로그램 코드, 결과 스크린샷을 인쇄하여 제출
  - 단, TCP와 UDP의 특징과 코드설명을 코드 안에 주석으로 설명

- 강의자료 Chapter2 Application Layer

  : 2.7 Socket programming with TCP 참조

  : 2.8 Socket programming with UDP 참조


Q&A