

# 네트워크 프로그래밍

숭실대학교 ICN 연구실 최종석  
jschoi@ssu.ac.kr

# 목차



**네트워크 개요**



**소켓 개요**



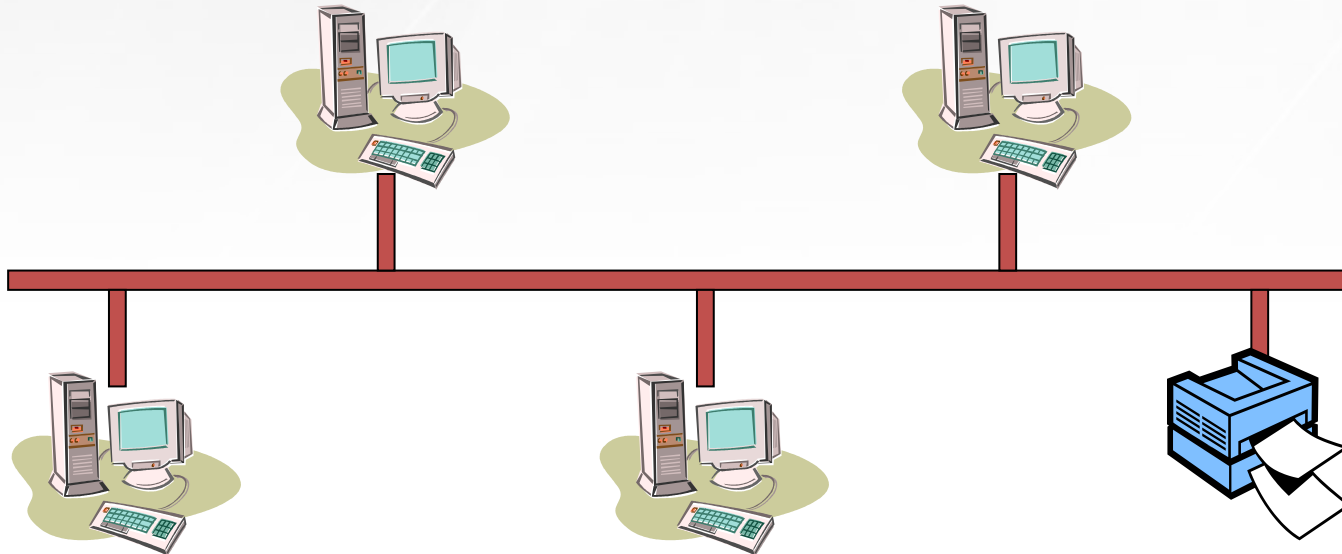
**소켓 프로그래밍(TCP, UDP)  
JAVA , C**



# 1. 네트워크 개요

## 네트워크(Network)

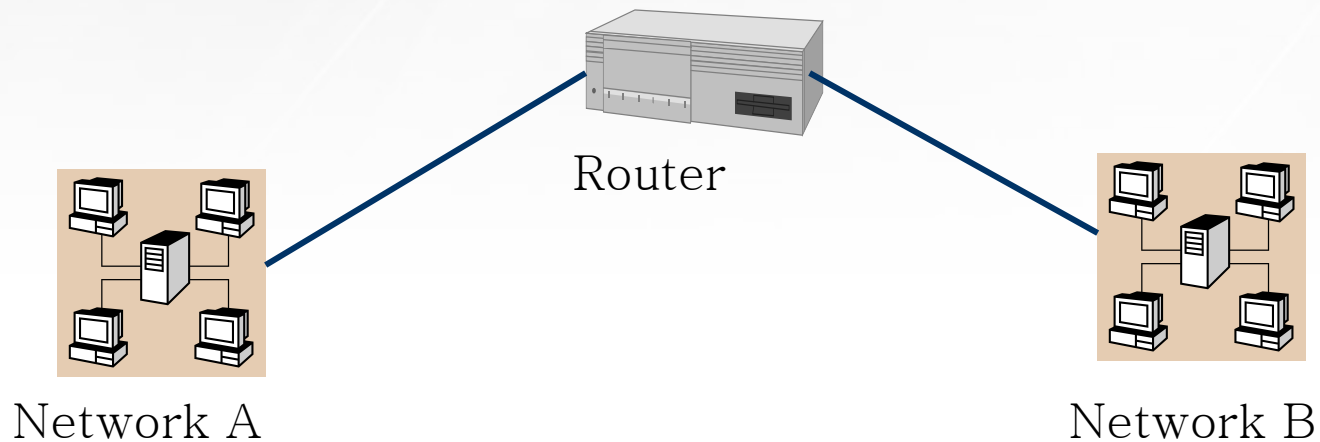
- 네트워크란 각 호스트(End-System)들을 연결하는 하나의 단일 System을 의미  
ex) 개인용 PC, 워크스테이션, 스마트폰, 프린터 등



# 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

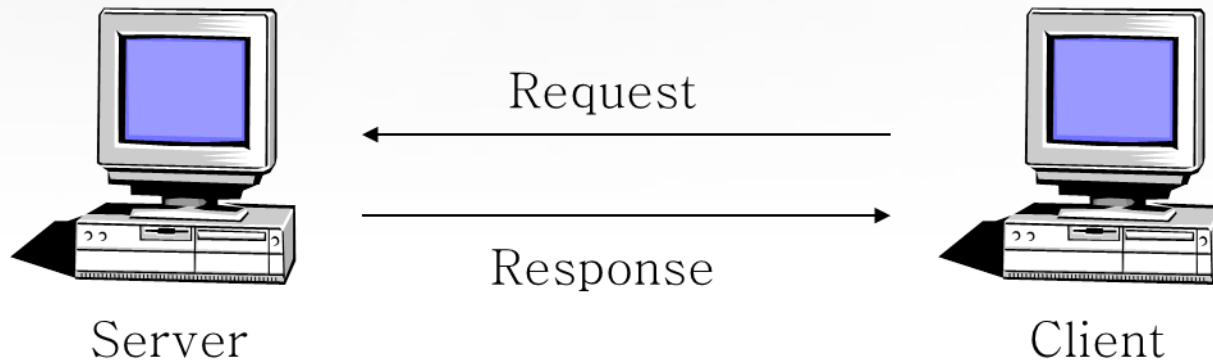


# 1. 네트워크 개요

- 클라이언트/서버(Client/Server) 모델

- Server/Client = 단일 프로그램
- Server는 Client의 연결요청 대기 → 정보 및 서비스 제공

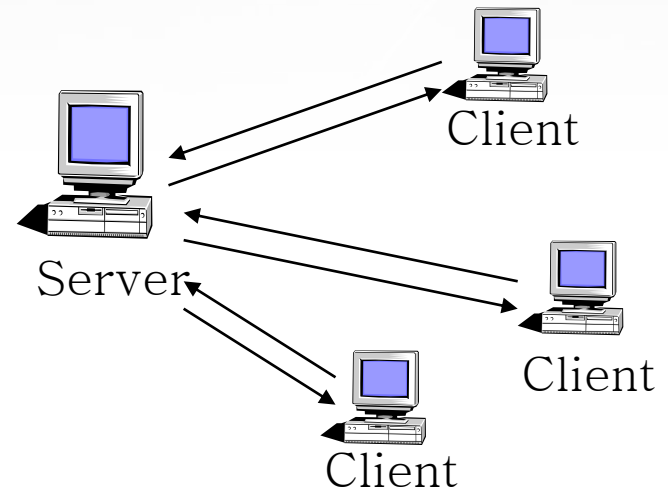
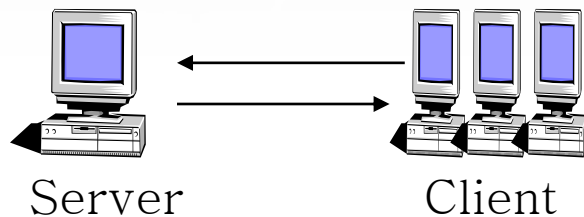
Client는 Server에 정보 및 서비스의 제공을 요청하고 응답을 기다리는 호스트를 의미



# 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/출력과 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?

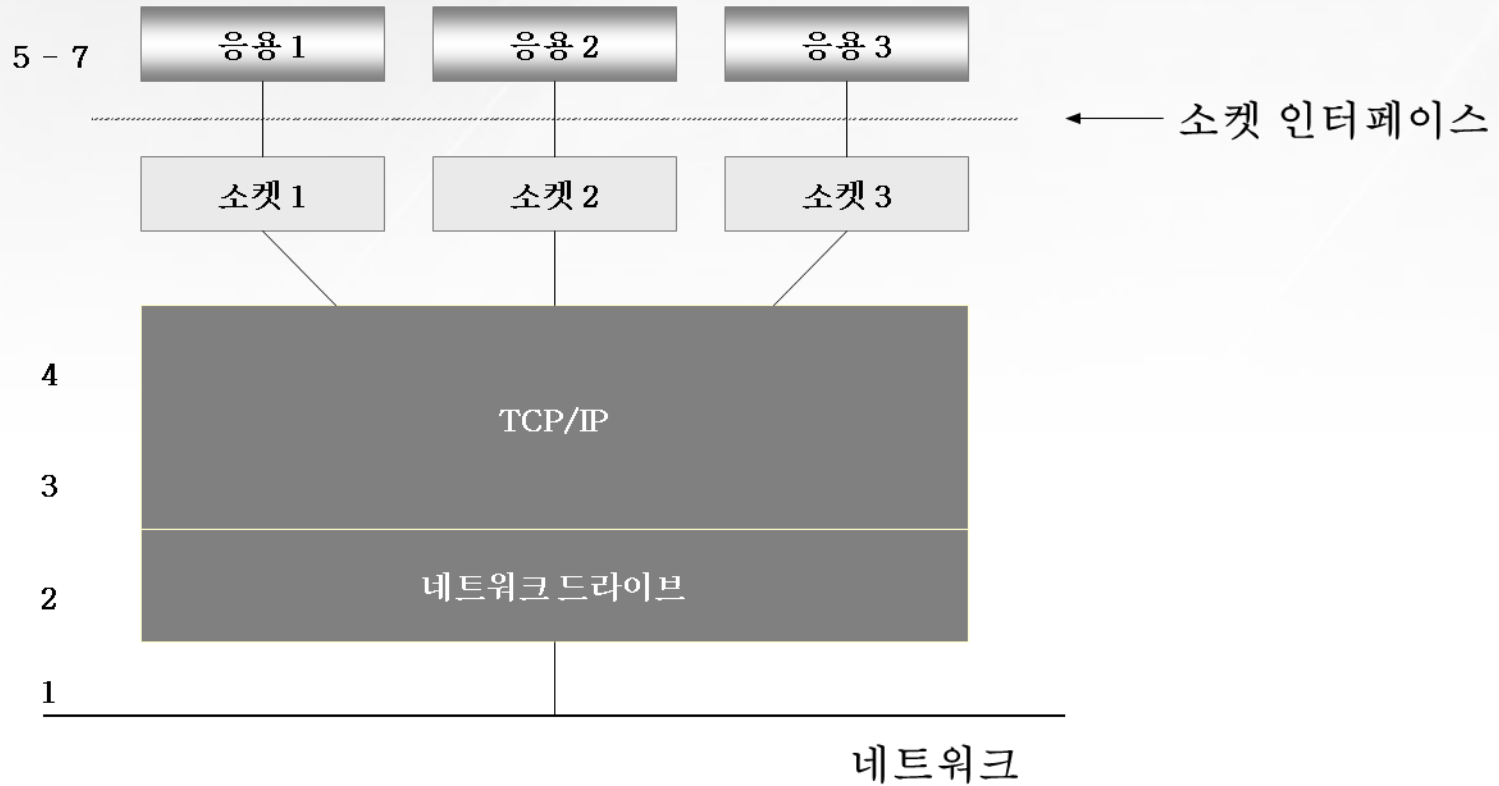
- 소켓(socket)은 네트워크를 통한 입/출력을 하기 위해 사용자에게 필요한 수단을 제공하는 응용 프로토콜 인터페이스
- 소켓을 활용한 네트워크 응용 프로그램을 통해 네트워크상에서 데이터를 송/수신
- 네트워크 입/출력을 위한 요소
  - : 프로토콜(Protocol)
  - : 소스 IP 주소(Source IP Address)
  - : 소스 포트 번호(Source Port Address)
  - : 목적지 IP 주소(Target IP Address)
  - : 목적지 포트 번호(Target Port Address)



## 2. 소켓 개요

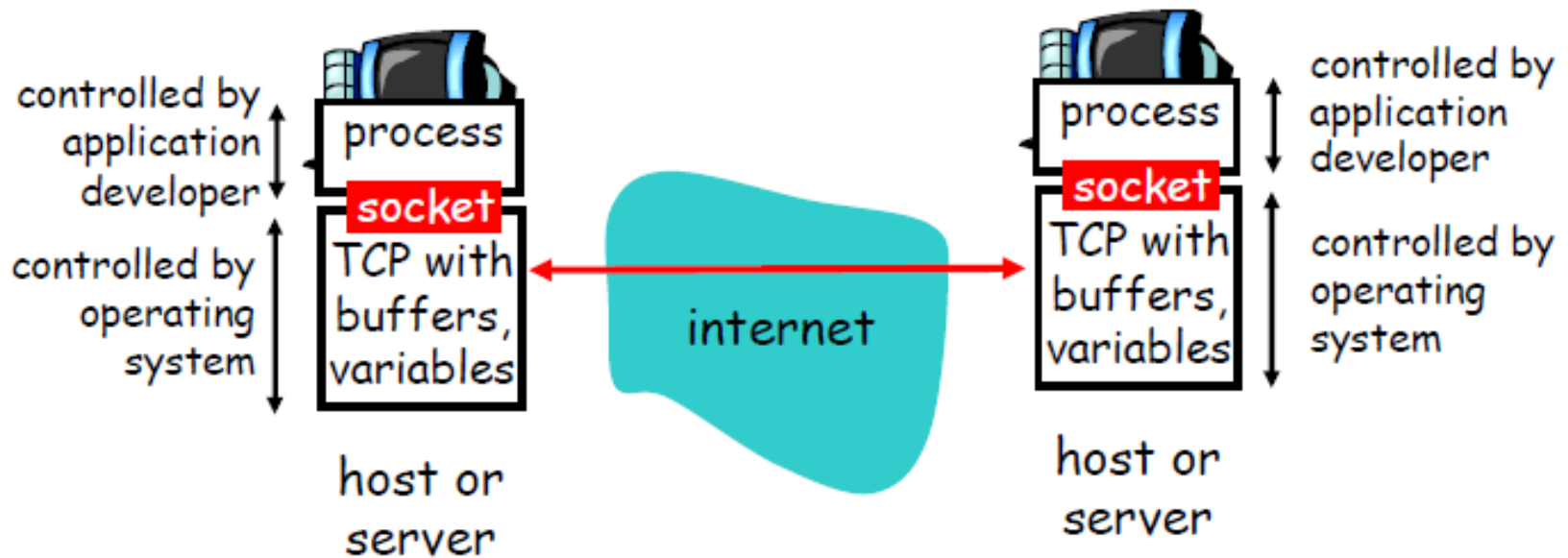
### Socket 인터페이스 위치

(OSI 계층)



## 2. 소켓 개요

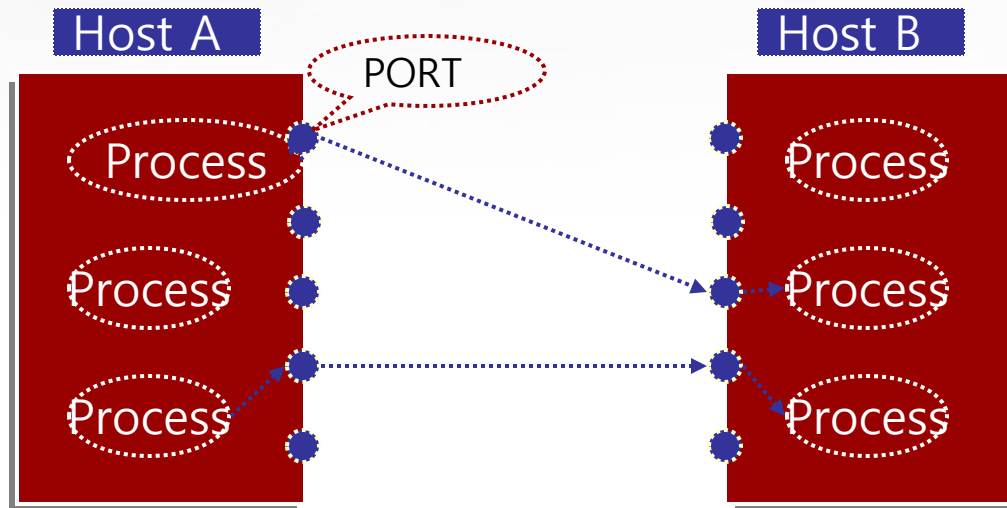
- 응용 프로그램과 소켓



## 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

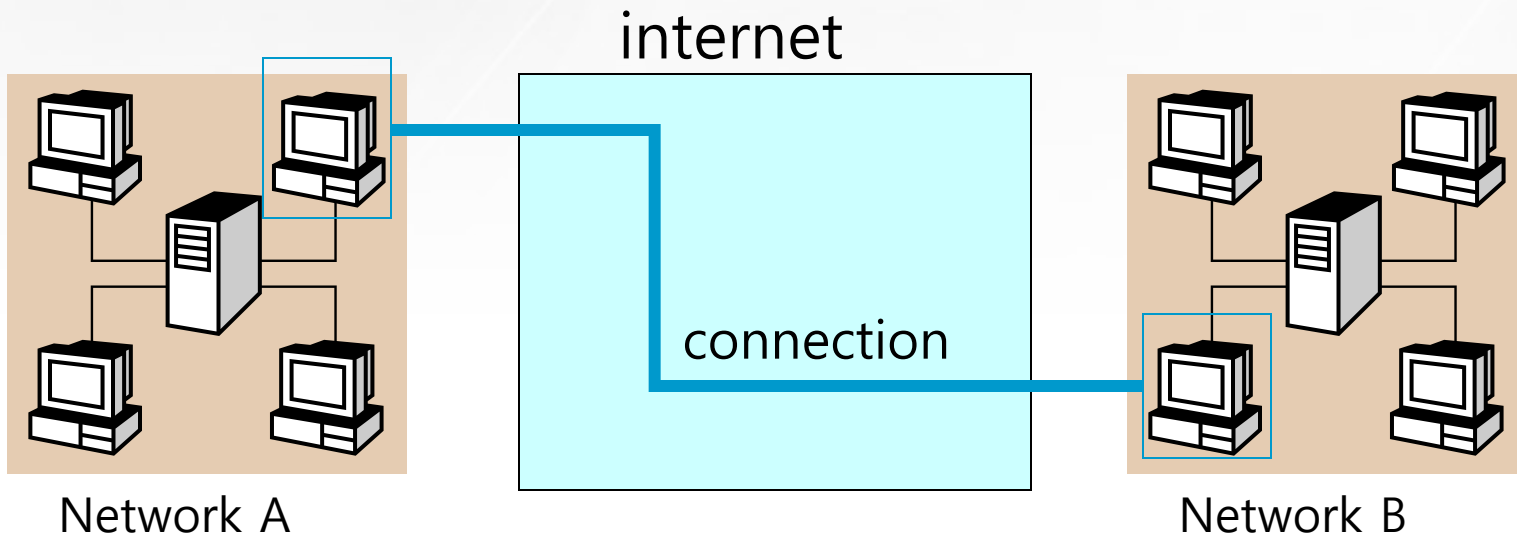
- 메시지 경계가 유지되지 않음

- 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

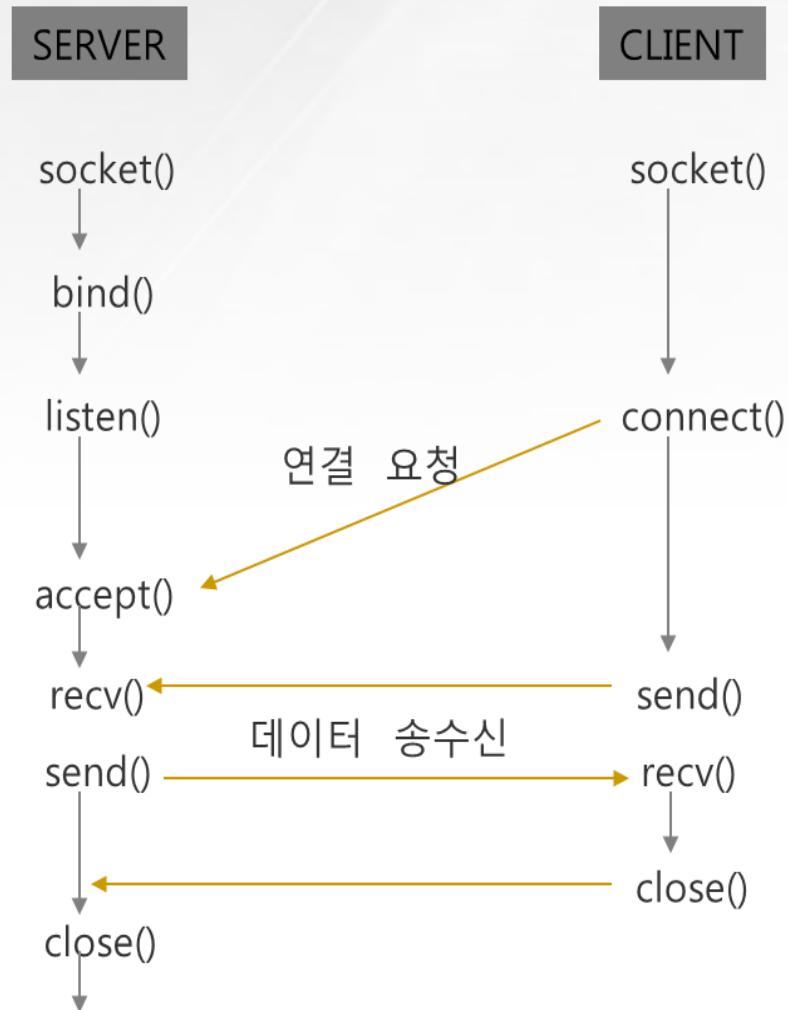
- 메시지 경계가 유지되지 않음

- 전달된 순서대로 수신됨

- 전송된 모든 데이터는 에러없이 원격지에 도달

## 2. 소켓 개요

### • TCP 소켓 – JAVA 와 C Socket 비교(C Socket)



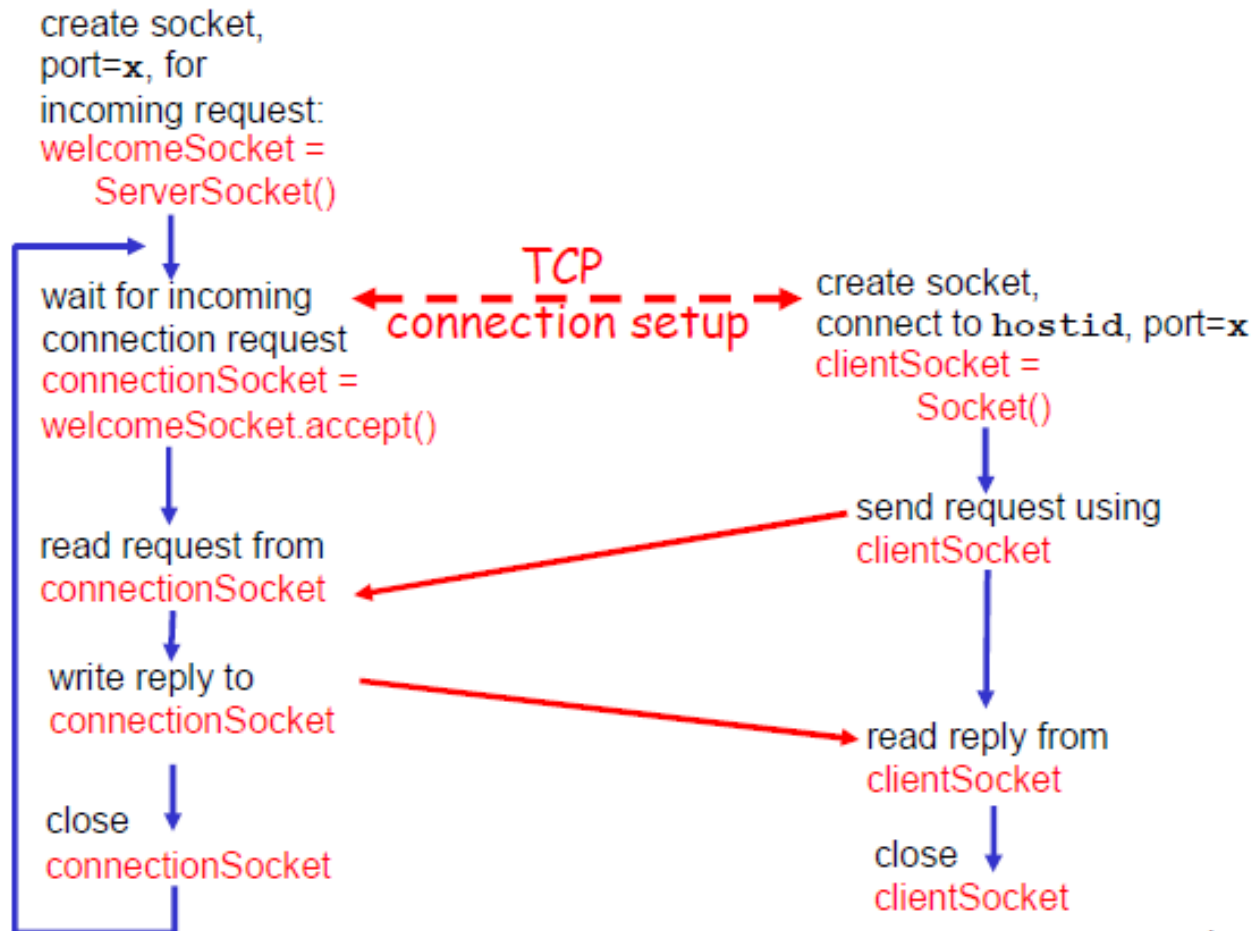


## 2. 소켓 개요

### • TCP 소켓 – JAVA 와 C Socket 비교(JAVA Socket)

Server (running on `hostid`)

Client



## 2. 소켓 개요

### 연결 지향형 소켓(SOCK\_STREAM, TCP 소켓)

#### - 3 way hand shaking

File Edit View Go Capture Analyze Statistics Telephony Tools Internals Help										
Filter: ip.addr == 17.149.160.49 Expression... Clear Apply Save										
No.	Time	Source	Destination	Protocol	Length	Info				
78450	6682.915758000	192.168.1.12	17.149.160.49	TCP	66	shiva-confsrvr > http [SYN] Seq=0				
78451	6682.916304000	192.168.1.12	17.149.160.49	TCP	66	xnmp > http [SYN] Seq=0 win=65535				
78452	6682.948062000	17.149.160.49	192.168.1.12	TCP	58	http > shiva-confsrvr [SYN, ACK] Seq=0 Ack=1				
78453	6682.948139000	192.168.1.12	17.149.160.49	TCP	54	shiva-confsrvr > http [ACK] Seq=1				
78454	6682.949155000	17.149.160.49	192.168.1.12	TCP	58	http > xnmp [SYN, ACK] Seq=0 Ack=1				
78455	6682.949177000	192.168.1.12	17.149.160.49	TCP	54	xnmp > http [ACK] Seq=1 Ack=1 win=				
78456	6683.104198000	192.168.1.12	17.149.160.49	HTTP	418	GET / HTTP/1.1				
78457	6683.138728000	17.149.160.49	192.168.1.12	HTTP	140	HTTP/1.1 301 MOVED PERMANENTLY				
78458	6683.138824000	192.168.1.12	17.149.160.49	TCP	54	shiva-confsrvr > http [ACK] Seq=36				
78459	6683.140834000	192.168.1.12	17.149.160.49	TCP	54	shiva-confsrvr > http [FIN, ACK] Seq=88				
78460	6683.172075000	17.149.160.49	192.168.1.12	TCP	54	http > shiva-confsrvr [ACK] Seq=88				
79286	6696.298183000	192.168.1.12	17.149.160.49	TCP	54	xnmp > http [FIN, ACK] Seq=1 Ack=1				
79288	6696.329578000	17.149.160.49	192.168.1.12	TCP	54	http > xnmp [FIN, ACK] Seq=1 Ack=2				
79289	6696.329637000	192.168.1.12	17.149.160.49	TCP	54	xnmp > http [ACK] Seq=2 Ack=2 win=				

## 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

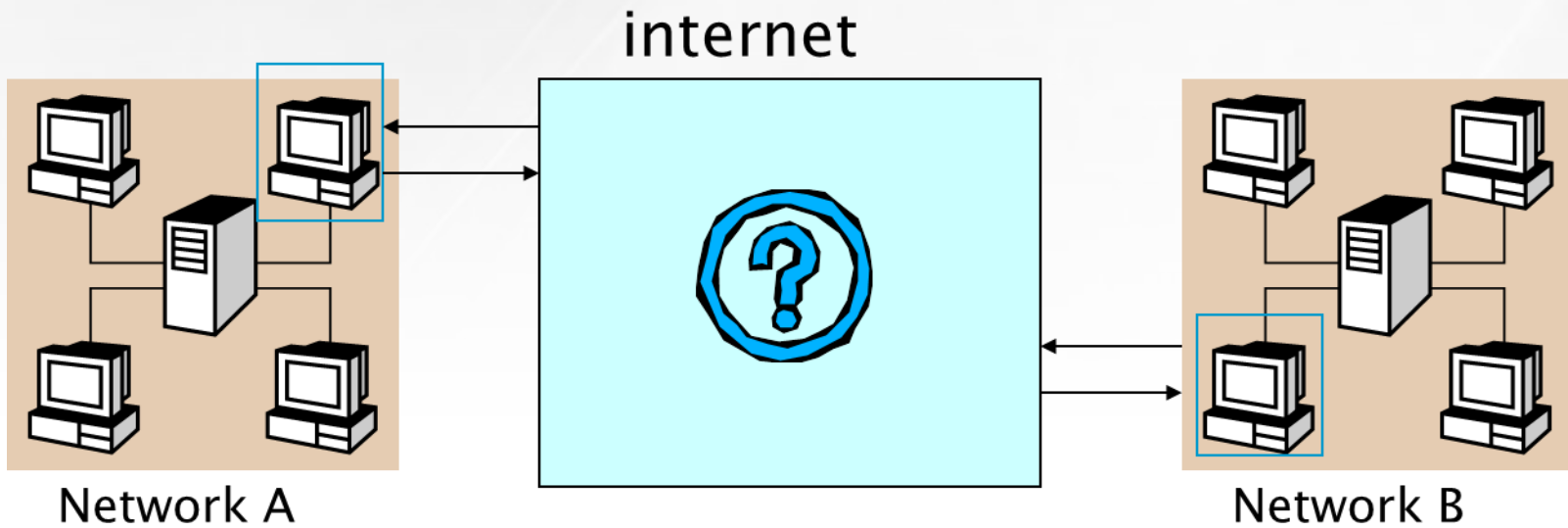
- 패킷은 전달된 순서대로 수신되지 않음

- 에러복구를 하지 않음 (즉, 신뢰성이 없음)

- 데이터그램 패킷의 크기 제한

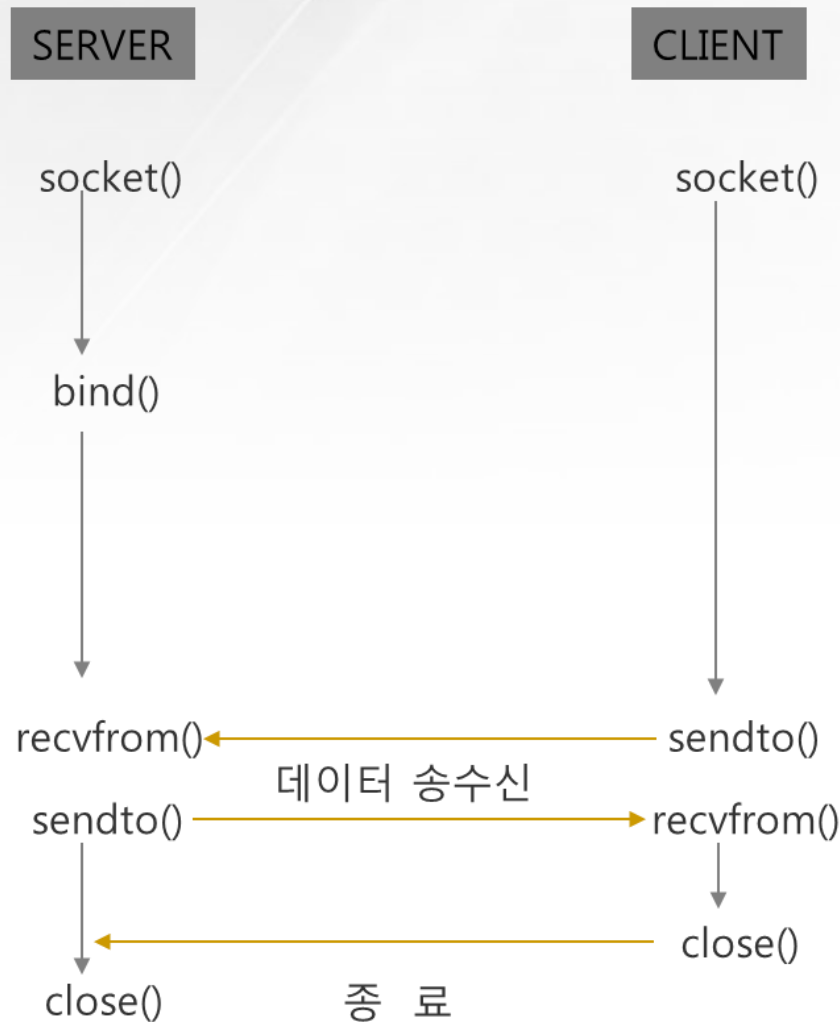
## 2. 소켓 개요

- 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)

Server (running on `hostid`)

Client

create socket,  
port= x.  
`serverSocket =`  
`DatagramSocket()`

read datagram from  
`serverSocket`

write reply to  
`serverSocket`  
specifying  
client address,  
port number

create socket,  
`clientSocket =`  
`DatagramSocket()`

Create datagram with server IP and  
port=x; send datagram via  
`clientSocket`

read datagram from  
`clientSocket`

close  
`clientSocket`

