

Chapter 01 네트워크와 소켓 프로그래밍

학습목표

- TCP/IP 프로토콜의 동작 원리를 살펴본다.
- 소켓의 기본 개념을 이해한다.
- 소켓의 역사와 특징을 살펴본다.
- 소켓 프로그램을 작성하고 실행하는 과정을 이해한다.

목차

01 TCP/IP 프로토콜 개요

02 소켓의 개념

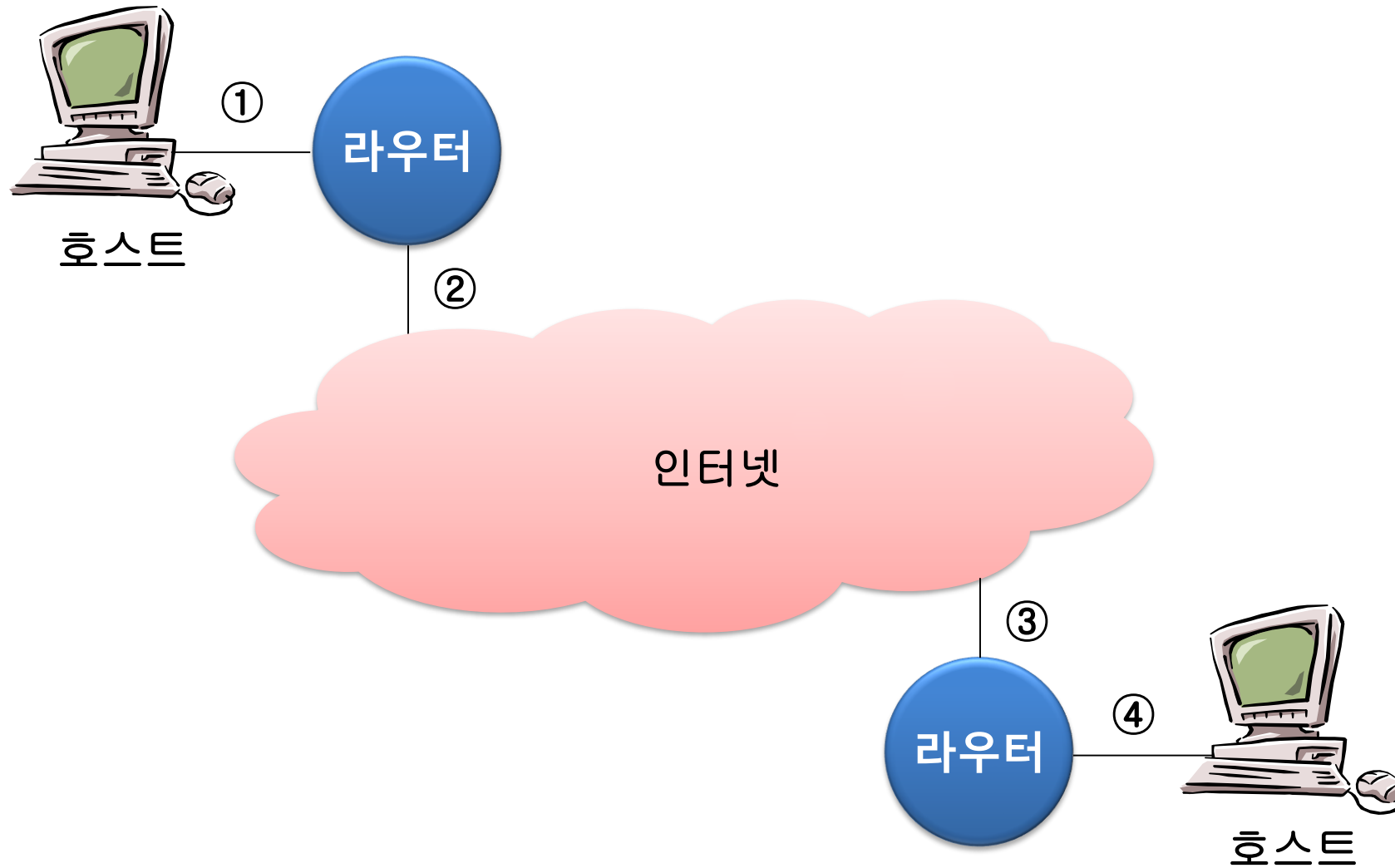
03 소켓의 특징과 구조

04 소켓 프로그램 맛보기

01 TCP/IP 프로토콜 개요



인터넷 구성 요소 (1)



인터넷 구성 요소 (2)

■ 호스트

- 최종 사용자의 응용 프로그램을 수행하는 주체

■ 라우터

- 호스트에서 생성된 데이터를 여러 네트워크를 거쳐 전송함으로써 서로 다른 네트워크에 속한 호스트 간에 데이터를 교환할 수 있게 하는 장비

■ 통신 프로토콜

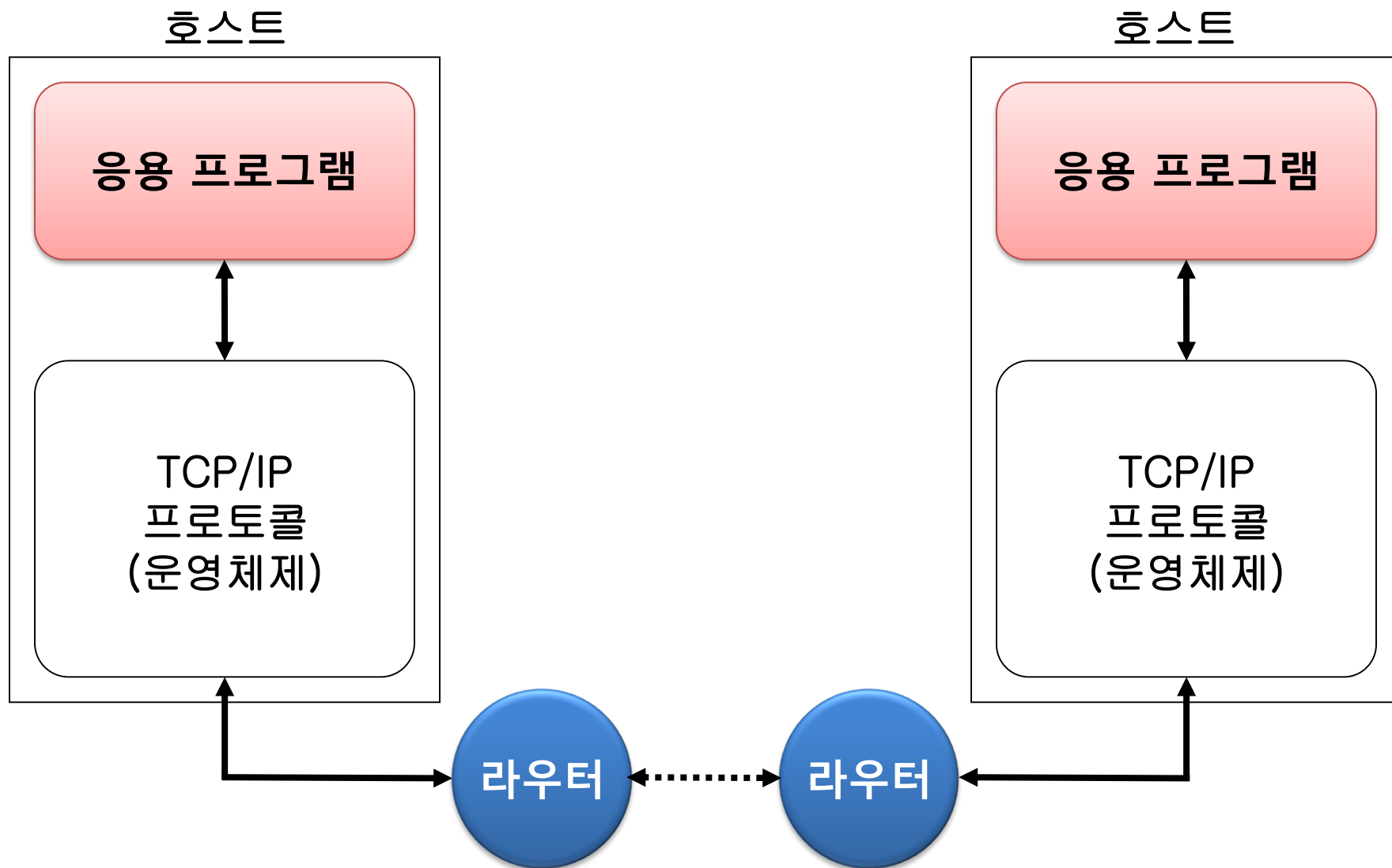
- 호스트와 라우터, 라우터와 라우터, 호스트와 호스트가 통신하기 위한 정해진 절차와 방법

TCP/IP 프로토콜 (1)

■ TCP/IP 프로토콜

- 인터넷의 핵심 프로토콜인 TCP와 IP를 비롯한 각종 프로토콜
- 운영체제의 일부로 구현되며, 응용 프로그램은 운영체제가 제공하는 TCP/IP 프로토콜의 서비스를 사용해 통신

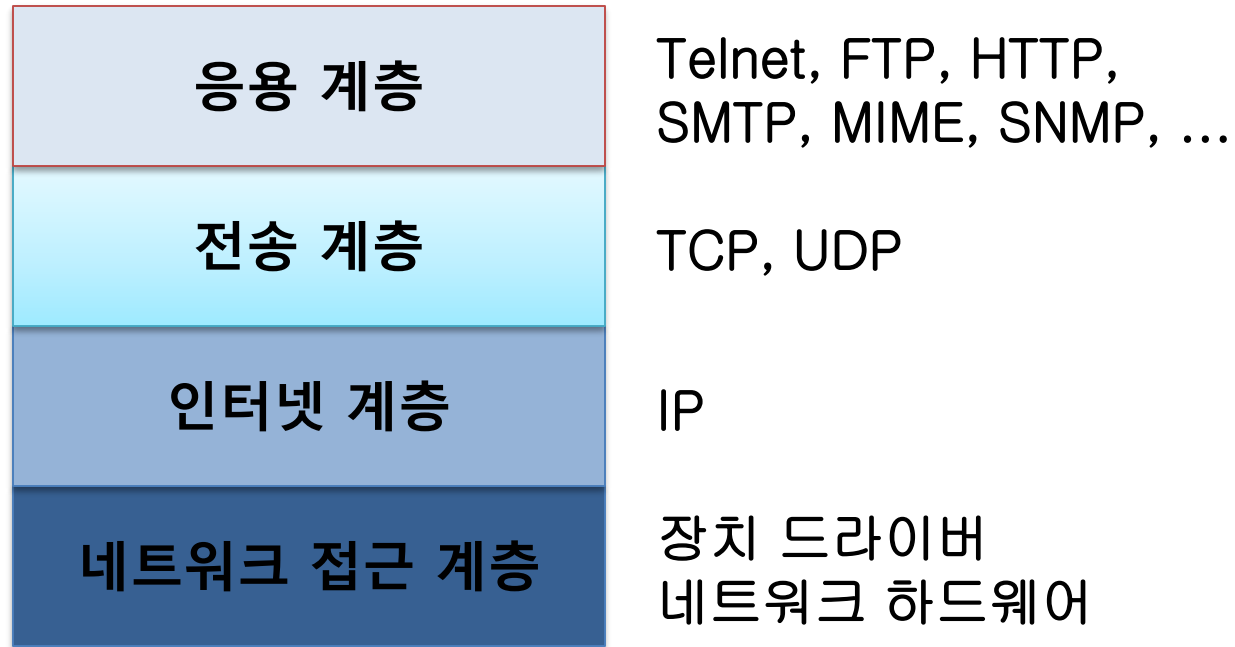
TCP/IP 프로토콜 (2)



TCP/IP 프로토콜 구조 (1)

■ TCP/IP 프로토콜 구조

■ 계층적 구조



TCP/IP 프로토콜 구조 (2)

■ 네트워크 접근 계층

■ 역할

- 물리적 네트워크를 통한 데이터 송수신

■ 구성 요소

- 네트워크 하드웨어 + 장치 드라이버

■ 주소 지정 방식

- 물리 주소
 - 예) 이더넷: 48비트 물리 주소

(a) 윈도우

```
C:\Windows\system32\cmd.exe
C:\Users\swkim>ipconfig /all

Windows IP 구성

호스트 이름 . . . . . : 2760p-en
주 DNS 접미사 . . . . . :
노드 유형 . . . . . : 혼성
IP 라우팅 사용 . . . . . : 아니요
WINS 프록시 사용 . . . . . : 아니요

이더넷 어댑터 VirtualBox Host-Only Network:

연결별 DNS 접미사 . . . . :
설명 . . . . . : VirtualBox Host-Only Ethernet Adapter
물리적 주소 . . . . . : 08-00-27-00-00-0E
```

(b) 리눅스

```
student@vm: ~
student@vm:~$ ifconfig
enp0s3: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 10.0.2.15 netmask 255.255.255.0 broadcast 10.0.2.255
    inet6 fe80::6a65:2379:ecbf:cfa7 prefixlen 64 scopeid 0x20<link>
    ether 08:00:27:26:7e:ac txqueuelen 1000 (Ethernet)
    RX packets 2616 bytes 3574416 (3.5 MB)
    RX errors 0 dropped 0 overruns 0 frame 0
    TX packets 992 bytes 104733 (104.7 KB)
    TX errors 0 dropped 0 overruns 0 carrier 0 collisions 0

enp0s8: flags=4163<UP,BROADCAST,RUNNING,MULTICAST> mtu 1500
    inet 192.168.56.101 netmask 255.255.255.0 broadcast 192.168.56.255
    inet6 fe80::f29f:f01a:276:72d2 prefixlen 64 scopeid 0x20<link>
```

TCP/IP 프로토콜 구조 (3)

■ 인터넷 계층

■ 역할

- 네트워크 접근 계층의 도움을 받아 데이터를 목적지 호스트까지 전달

■ 구성 요소

- IP 주소 + 라우팅(*or* 라우터)

■ 주소 지정 방식

- IP 주소
 - 소프트웨어적으로 정의된 논리 주소
 - 전 세계적인 유일성과 하드웨어 독립성을 가짐

■ 라우팅

- 데이터를 목적지까지 전달하는 일련의 작업
 - 라우팅에 필요한 정보 수집
 - 라우팅 정보를 기초로 데이터 전달

TCP/IP 프로토콜 구조 (4)

■ 전송 계층

■ 역할

- 최종 통신 목적지(응용 프로그램)를 지정하고, 오류 없이 데이터를 전송
 - 데이터 손실 또는 손상을 검출해 잘못된 데이터가 목적지에 전달되는 일을 방지

■ 주소 지정 방식

- 포트 번호

■ 대표 프로토콜

- TCP
- UDP

TCP/IP 프로토콜 구조 (5)

■ TCP와 UDP

TCP	UDP
연결형(connection-oriented) 프로토콜 – 연결 설정 후 통신 가능	비연결형(connectionless) 프로토콜 – 연결 설정 없이 통신 가능
신뢰성 있는 데이터 전송 – 데이터를 재전송함	신뢰성 없는 데이터 전송 – 데이터를 재전송하지 않음
일대일 통신(unicast)	일대일 통신(unicast), 일대다 통신(broadcast, multicast)
데이터 경계 구분 안 함 – 바이트 스트림(byte-stream) 서비스	데이터 경계 구분함 – 데이터그램(datagram) 서비스

TCP/IP 프로토콜 구조 (6)

■ 응용 계층

■ 역할

- 전송 계층을 기반으로 한 다수의 프로토콜과 이 프로토콜을 사용하는 응용 프로그램을 포괄

■ 대표 프로토콜

- Telnet, FTP, HTTP, SMTP, ...

데이터 전송 원리 (1)

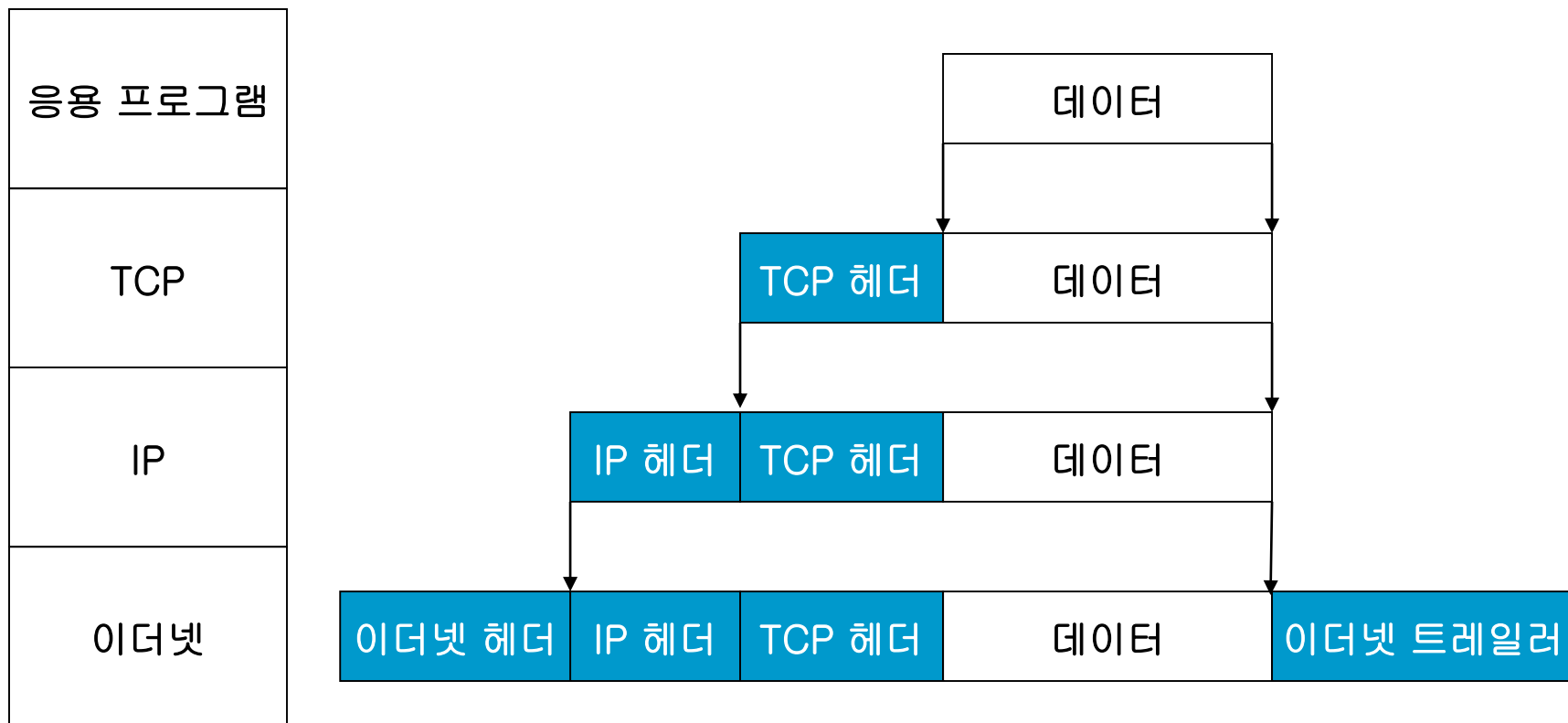
■ 패킷이란?

- 각 프로토콜에서 정의한 제어 정보(IP 주소, 포트 번호, 오류 체크 코드 등) + 데이터
- 제어 정보의 위치에 따라 앞쪽에 붙는 헤더(*header*)와 뒤쪽에 붙는 트레일러(*trailer*)로 구분

데이터 전송 원리 (2)

■ 패킷 전송 형태

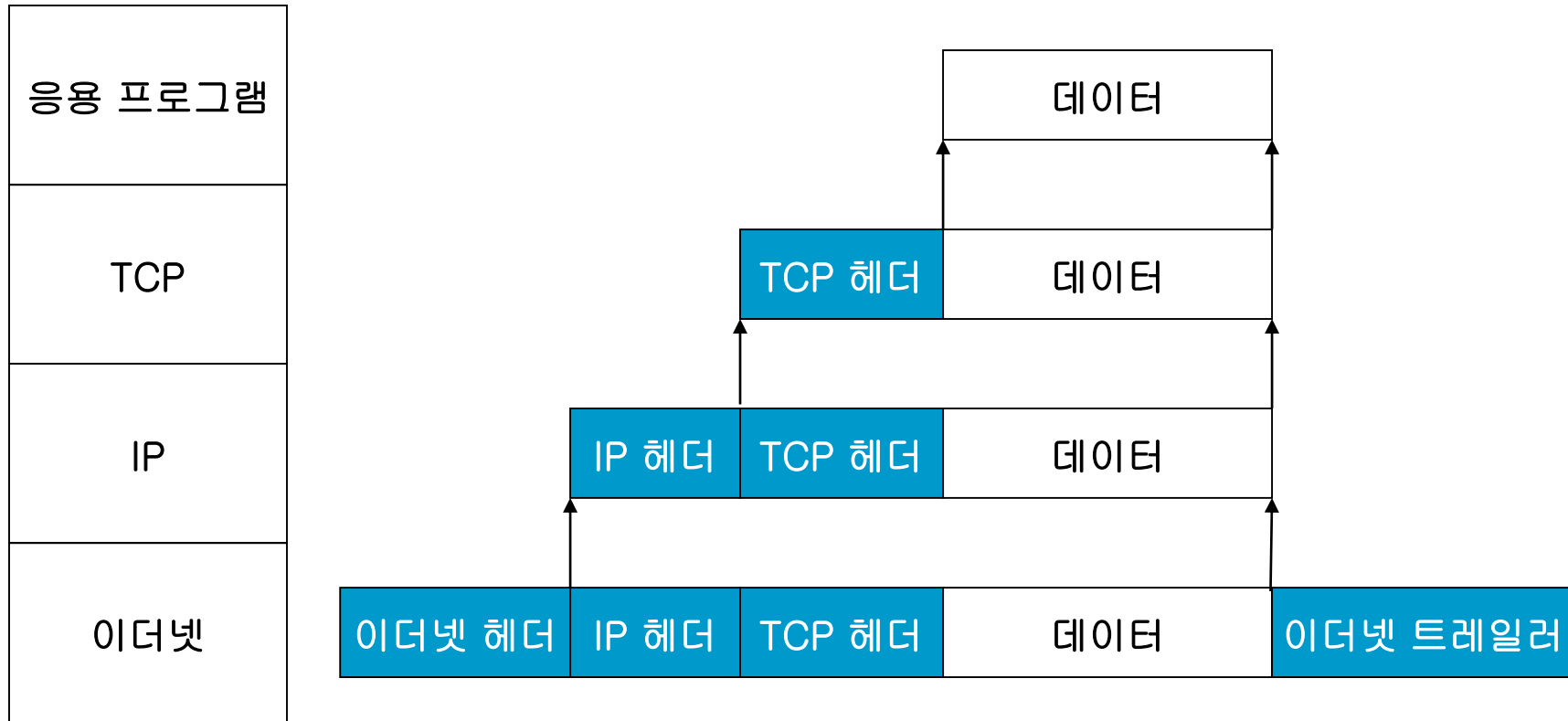
■ 송신측



데이터 전송 원리 (3)

■ 패킷 전송 형태

■ 수신측

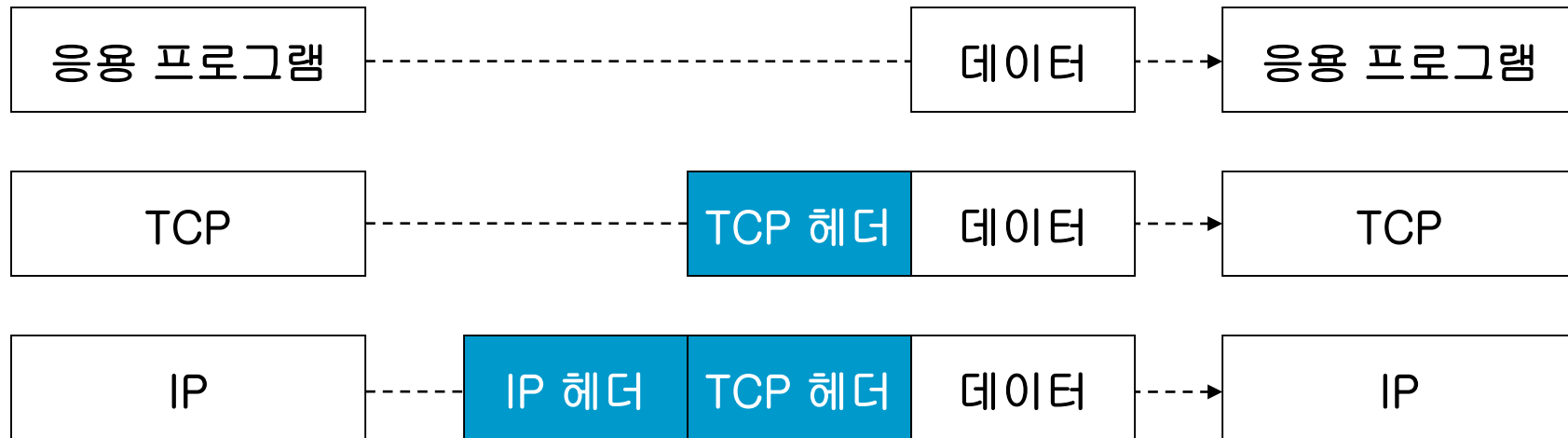


데이터 전송 원리 (4)

■ 패킷 전송 형태

■ 계층별

- 각 계층은 동일 위치의 상대 계층과 통신하는 것으로 간주

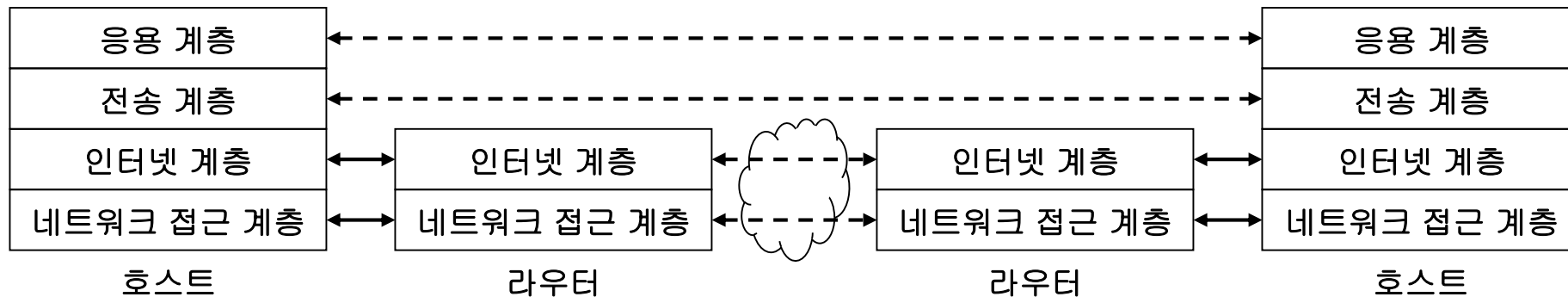


데이터 전송 원리 (5)

■ 패킷 전송 형태



■ 인터넷

- 응용 계층, 전송 계층
 - 하부 계층이 제공하는 가상적인 연결을 사용해 동작
- 인터넷 계층
 - IP 주소와 라우팅 기능을 이용해 패킷 전송 경로 결정
- 네트워크 접근 계층
 - 물리 주소를 사용해 실제 패킷 전송



IP 주소, 포트 번호 (1)

■ IP 주소

- 인터넷에 있는 호스트와 라우터의 식별자
 - 폐쇄된 네트워크이거나 IP를 공유하는 경우가 아니면 전 세계적으로 값이 유일
- IPv4는 32비트, IPv6는 128비트 사용
- IPv4는 8비트 단위로 .(dot)로 구분하여 10진수 4개로 표기  *dotted-decimal notation*
 - 예) 147.46.114.70
- IPv6는 16비트 단위로 :(colon)으로 구분하여 16진수 8개로 표기  *colon-hexadecimal notation*
 - 예) 2001:0230:abcd:ffab:0023:eb00:ffff:1111

IP 주소, 포트 번호 (2)

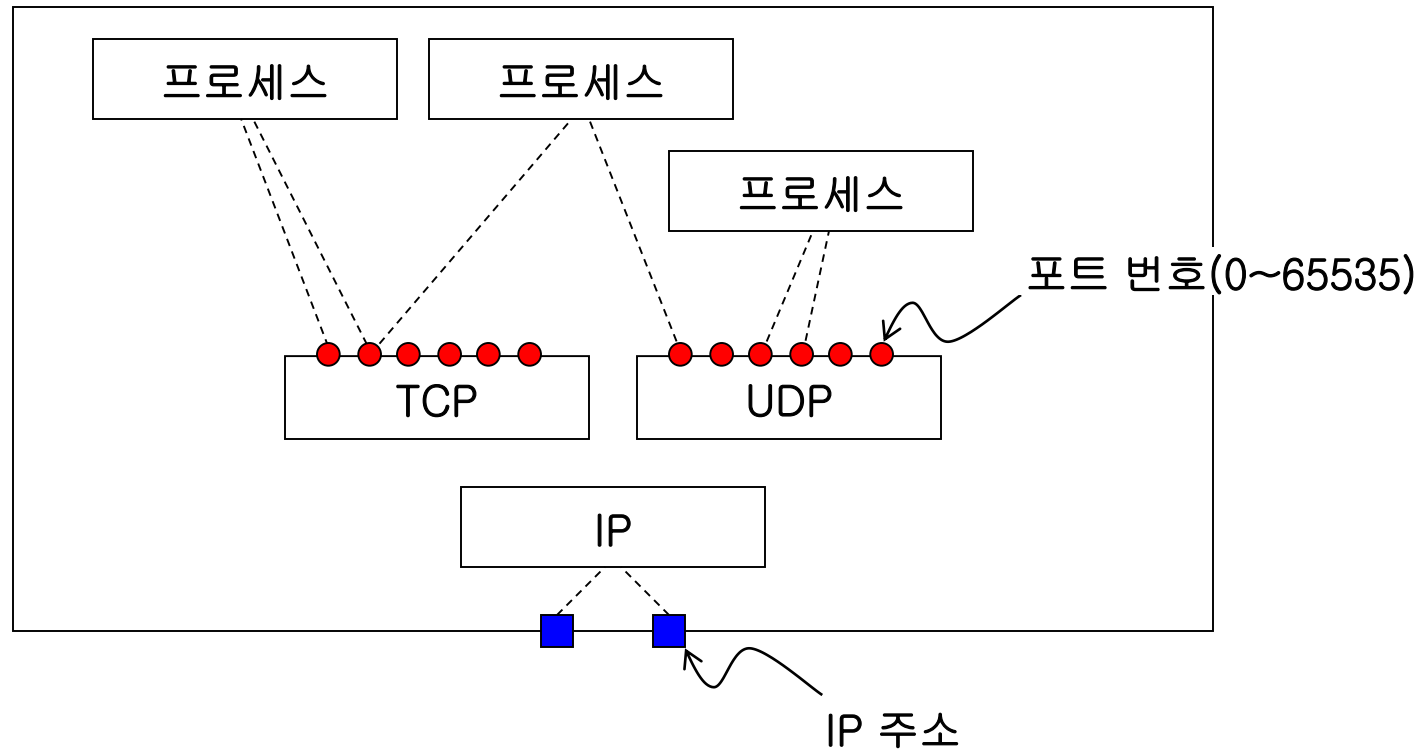
■ 포트 번호

- 인터넷 통신의 종착점(하나 혹은 여러 프로세스)을 나타내는 식별자
- TCP와 UDP는 포트 번호로 부호 없는 16비트 정수를 사용하므로 0~65535 범위가 가능
- 영역별 포트 번호

포트 번호	분류
0 ~ 1023	알려진 포트(well-known ports)
1024 ~ 49151	등록된 포트(registered ports)
49152 ~ 65535	동적/사설 포트(dynamic and/or private ports)

IP 주소, 포트 번호 (3)

■ IP 주소와 포트 번호



IP 주소, 포트 번호 (4)

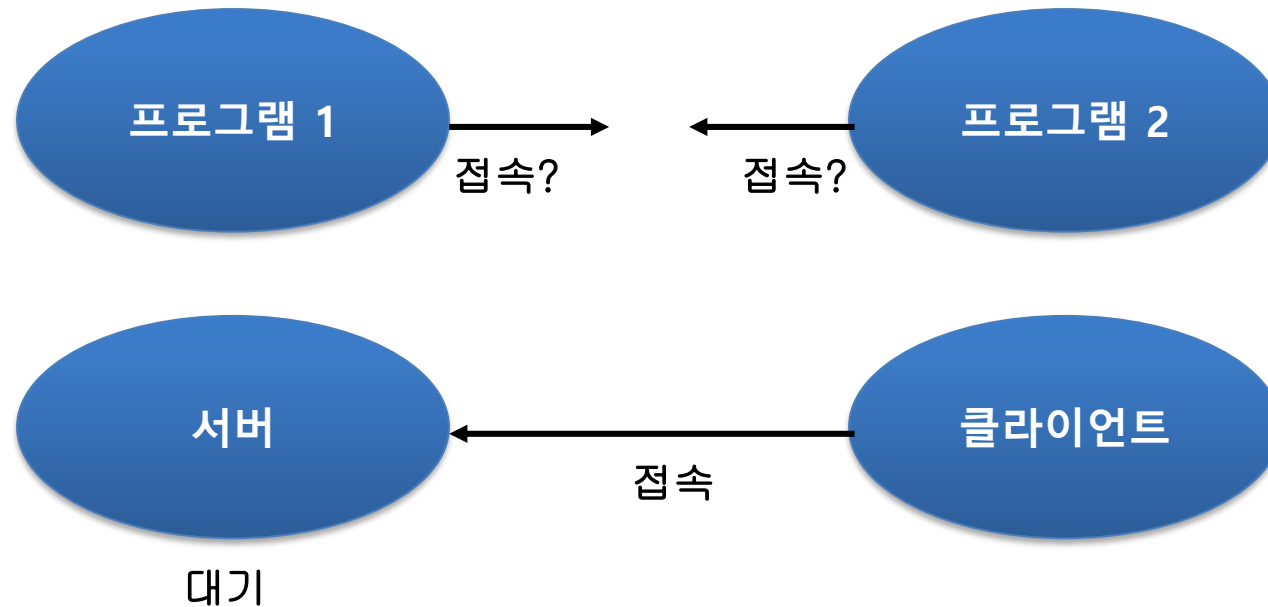
■ 도메인 이름

- IP 주소에 대한 (기억하기 쉬운) 별명
- 실제 통신할 때는 IP 주소로 변환해야 함

클라이언트-서버 모델

■ 클라이언트-서버(Client/Server) 모델

- 두 프로그램이 상호 작용하는 방식을 나타내는 용어
- 서비스를 요청하는 쪽은 클라이언트(Client), 클라이언트가 요청하는 서비스를 처리하는 쪽은 서버(Server)

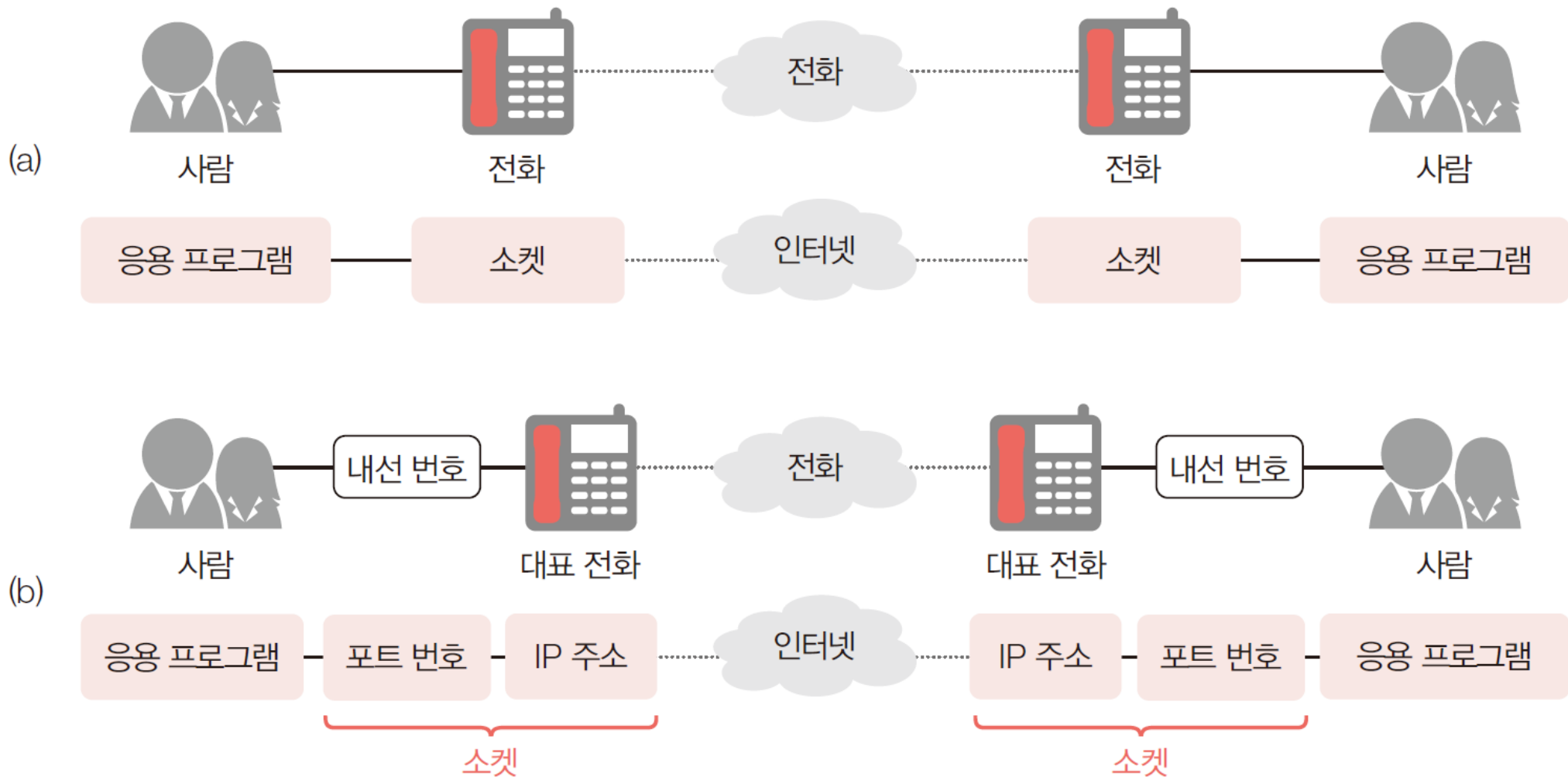


02 소켓의 개념



소켓의 개념 (1)

■ 전화 통신과 소켓 통신 비교



소켓의 개념 (2)

■ 세 가지 관점

- ① 데이터 타입
- ② 통신 종단점
- ③ 네트워크 프로그래밍 인터페이스

소켓의 개념 (3)

■ 데이터 타입

- 파일 디스크립터 혹은 핸들과 유사한 개념
- 생성과 설정 과정이 끝나면 운영체제의 통신 관련 정보를 참조해 다양한 작업을 편리하게 할 수 있는 데이터 타입

```
int fd = open("myfile", ...); // 파일 생성
...
read(fd, ...); // 데이터 읽기
write(fd, ...); // 데이터 쓰기
```

(a) 리눅스 파일 입출력

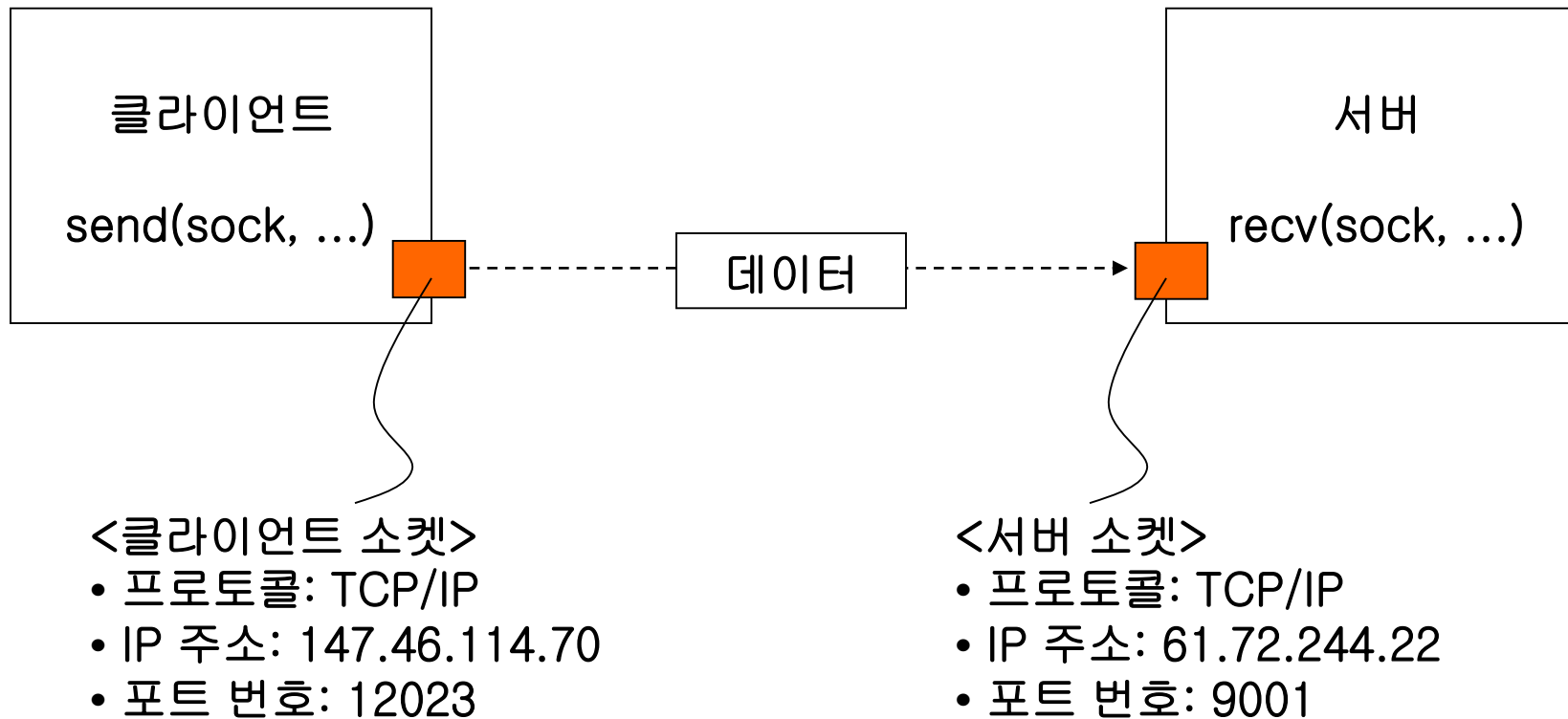
```
SOCKET sock = socket(...); // 소켓 생성
...
recv(sock, ...); // 데이터 받기
send(sock, ...); // 데이터 보내기
```

(b) 윈도우 소켓 통신

소켓의 개념 (4)

■ 통신 종단점

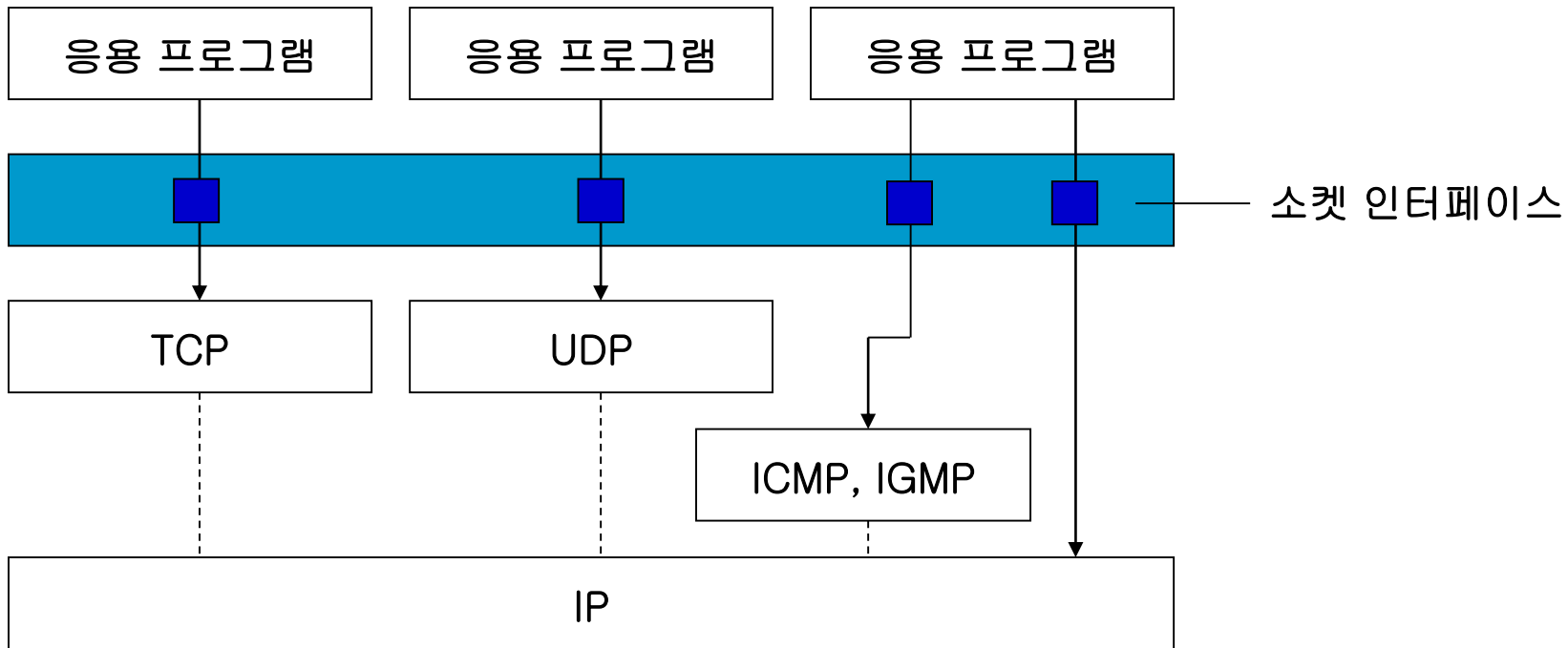
- 응용 프로그램은 자신의 소켓이 상대방의 소켓과 연결된 것으로 생각하고 데이터를 주고받음



소켓의 개념 (5)

■ 네트워크 프로그래밍 인터페이스

- 통신 양단이 모두 소켓을 사용할 필요는 없음
- TCP/IP 프로토콜에서 (일반적으로) 응용 계층과 전송 계층 사이에 위치하는 것으로 간주



03 소켓의 특징과 구조



윈도우 소켓 (1)

■ 윈도우 소켓(원속)

- 버클리 유닉스에서 개발한 네트워크 프로그래밍 인터페이스를 윈도우 환경에서 사용할 수 있게 만든 것
- 윈도우 95 버전부터 API에 정식으로 포함하여 제공

윈도우 소켓 (2)

■ 윈도우 소켓과 유닉스 소켓의 차이점

- 윈도우 소켓은 DLL을 통해 대부분의 기능이 제공되므로 DLL 초기화와 종료 작업을 위한 함수가 필요
- 윈도우 프로그램은 대개 GUI를 갖추고 메시지 구동 방식으로 동작하므로 이를 위한 확장 함수가 존재
- 윈도우는 운영체제 차원에서 멀티스레드를 지원하므로 멀티스레드 환경에서 안정적으로 동작하는 구조와 이를 위한 함수가 필요

윈도우 소켓 (3)

■ 윈도우 운영체제의 원속 지원

운영체제	원속 버전
윈도우 95	1.1 (업그레이드를 통해 2.2 버전을 지원)
윈도우 98 이후 모든 버전	2.2

■ 원속에서 지원하는 통신 프로토콜

- IPv4 - 윈도우 95 이상
- IPv6 - 윈도우 XP SP1 이상
- Bluetooth - 윈도우 XP SP2 이상
- IrDA(적외선 통신) - 윈도우 98 이상

윈도우 소켓 (4)

■ 윈속의 장점

- 유닉스 소켓과 소스 코드 수준에서 호환성이 높으므로 기존 코드를 이식하여 활용하기 쉬움
- 가장 널리 사용되는 네트워크 프로그래밍 인터페이스이므로 한번 배우면 여러 운영체제(윈도우, 리눅스 등)에서 사용 가능
- TCP/IP 외의 프로토콜도 지원하므로 최소 코드 수정으로 응용 프로그램이 사용할 프로토콜 변경 가능
- 비교적 저수준 프로그래밍 인터페이스이므로 세부 제어가 가능하며 고성능 네트워크 프로그램 개발 가능

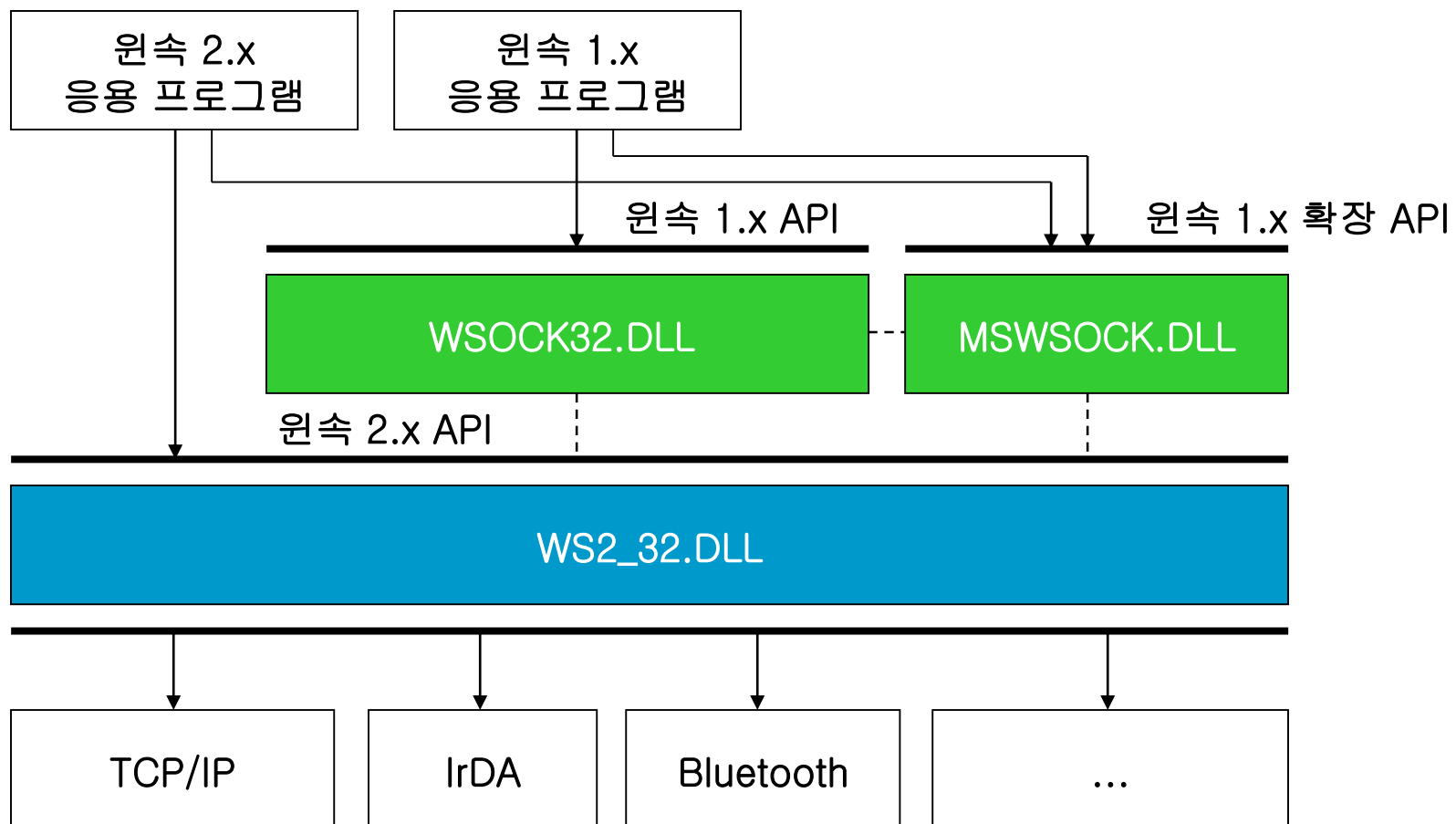
윈도우 소켓 (5)

■ 윈속의 단점

- 응용 프로그램 수준의 프로토콜을 프로그래머가 직접 설계해야 함
 - 주고받는 데이터 형식이나 전송 절차 등을 고려해 프로그래밍해야 하며, 설계 변경 시에는 코드 수정이 불가피함
- 서로 다른 바이트 정렬 방식을 사용하거나 데이터 처리 단위가 서로 다른 호스트끼리 통신할 경우, 응용 프로그램 수준에서 데이터 변환을 처리해야 함

윈도우 소켓 (6)

■ 윈속 구조



04 소켓 프로그램 맛보기



윈도우 소켓 프로그램 맛보기 (1)

■ 프로젝트 생성

새 프로젝트 구성

Windows 데스크톱 마법사 C++ Windows 데스크톱 콘솔 라이브러리

프로젝트 이름(J)

Server

위치(L)

C:\TestW

솔루션 이름(M) ⓘ

Server

☒ 솔루션 및 프로젝트를 같은 디렉터리에 배치(D)

뒤로(B) 만들기(C)

윈도우 소켓 프로그램 맛보기 (2)

■ 설정 변경

새 프로젝트 구성

Windows 데스크톱

프로젝트 이름(I)
Server

위치(L)
C:\Test\

솔루션 이름(M) ⓘ
Server

☒ 솔루션 및 프로젝트를 같...

Windows 데스크톱 프로젝트

애플리케이션 종류(T)
콘솔 애플리케이션(.exe)

추가 옵션:
☒ 빈 프로젝트(E)
☐ 미리 컴파일된 헤더(P)
☐ 기호 내보내기(X)
☐ MFC 헤더(M)

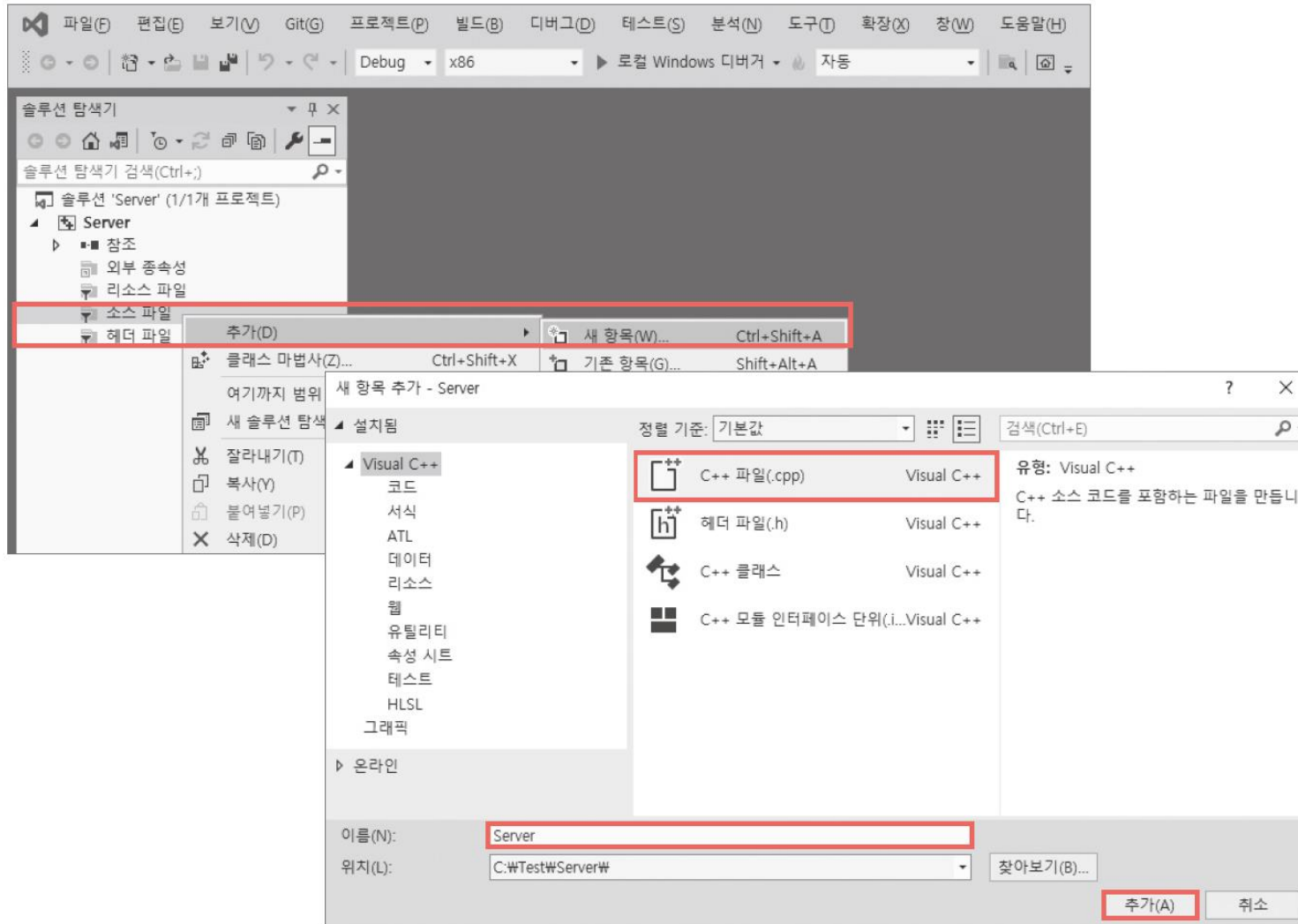
ⓘ 팁: 빈 프로젝트 템플릿을 사용하여 이 종류의 프로젝트를 만들 수도 있습니다.

확인 취소

뒤로(B) 만들기(C)

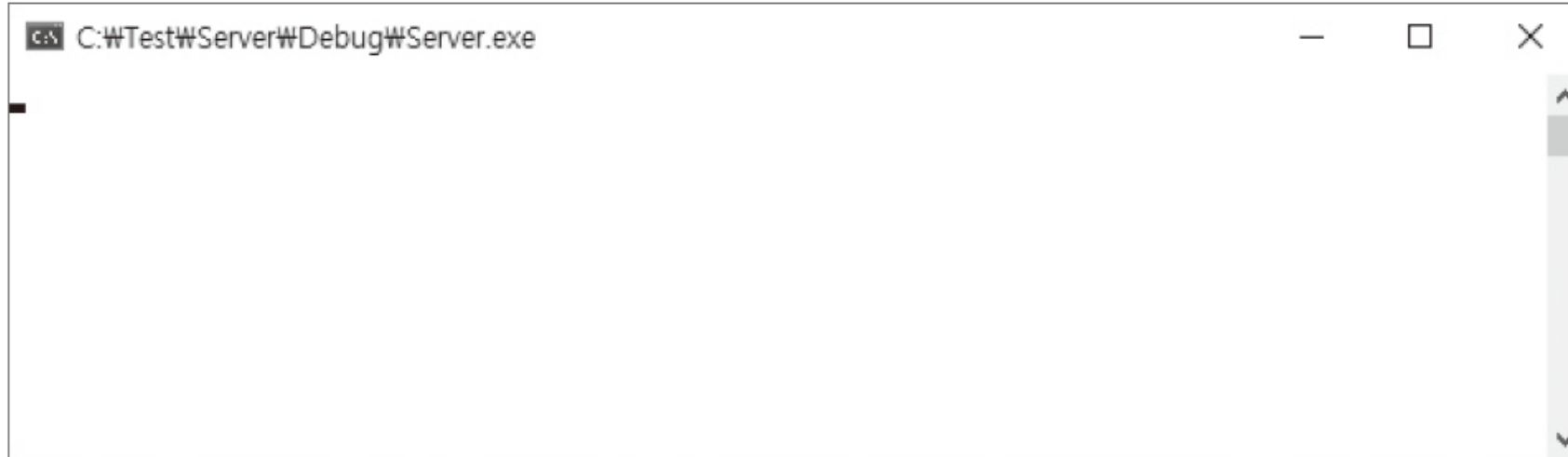
윈도우 소켓 프로그램 맛보기 (3)

■ 소스 파일 추가



윈도우 소켓 프로그램 맛보기 (4)

■ 서버 실행 화면



윈도우 소켓 프로그램 맛보기 (5)

■ 통신 테스트 화면

