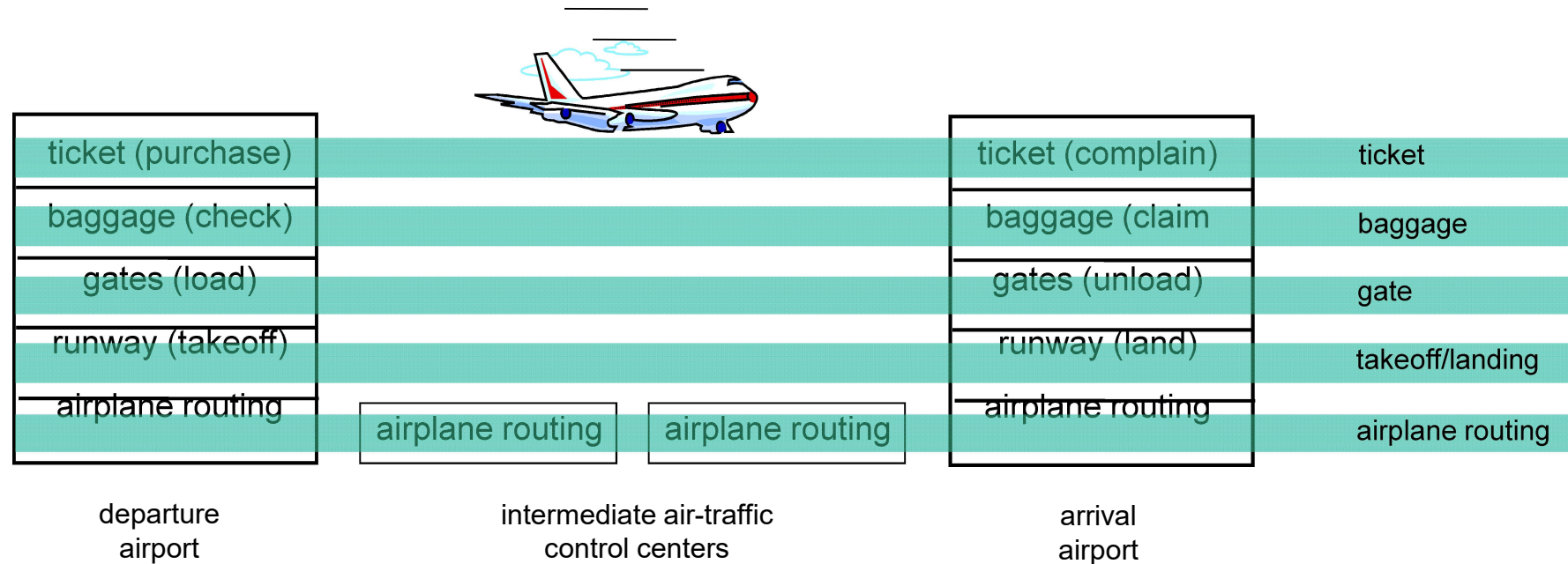


## 7. 네트워크 모델: 계층 구조의 개념 -1

### 1. Layering of airline functionality



**Layers:** each layer implements a service

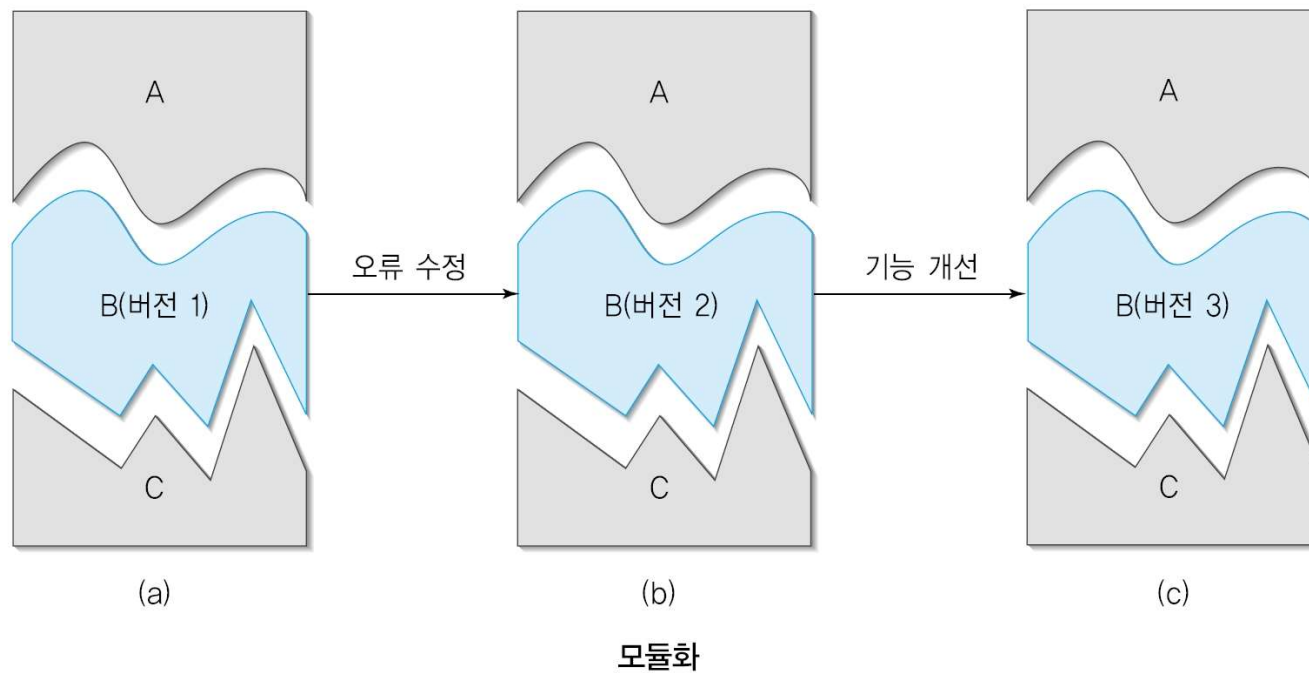
- A. via its own internal-layer actions
- B. relying on services provided by layer below

# 7. 네트워크 모델: 계층 구조의 개념 -2

## 2. 계층적 모듈 구조

### A. 모듈화

- 크고 복잡한 시스템을 기능별로 여러 개 작고 단순한 **모듈로 독립화**
- 모듈 사이의 적절한 **인터페이스**가 필요
- 예: C 프로그래밍 언어의 함수 개념

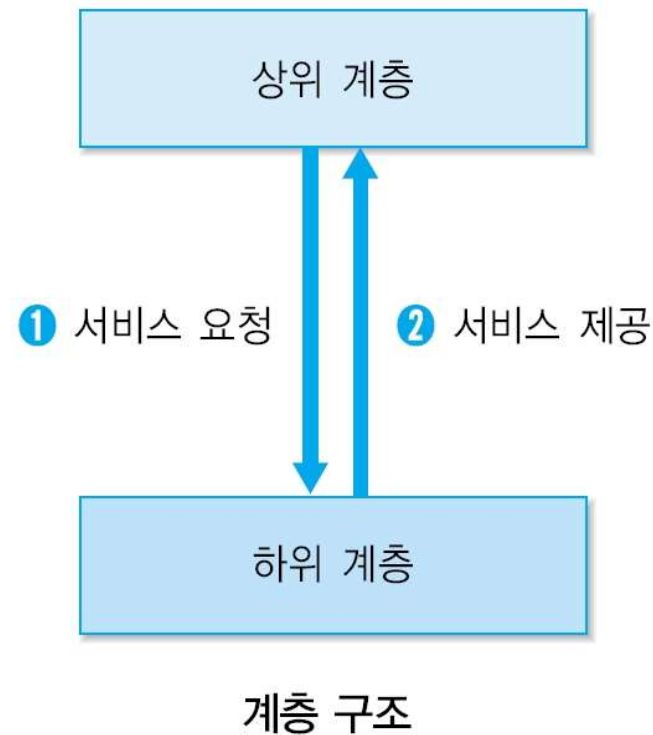


## 7. 네트워크 모델: 계층 구조의 개념 -3

### 3. 계층적 모듈 구조

#### A. 계층 구조

- 상위 모듈이 하위 모듈에게 **서비스를 요청**
- 하위 모듈은 **서비스를 실행**하고 그 결과를 상위 모듈에 **통보**



## 7. 네트워크 모델: 계층 구조의 개념 -4

### 4. 계층적 모듈 구조: 장점

- A. 전체 시스템을 이해하기 쉽고, 설계 및 구현이 용이
- B. 모듈간 표준 인터페이스가 단순하면 모듈 독립성을 향상  
시스템 구조를 단순화시키는 장점이 됨
- C. 대칭 구조에서는 동일 계층 사이의 인터페이스인  
프로토콜을 단순화시킬 수 있음
- D. 특정 모듈의 외부 인터페이스가 변하지 않으면  
내부 기능 변화가 전체 시스템 동작에 영향을 미치지 않음

## 7. 네트워크 모델: 계층 구조의 개념 -5

### 5. 프로토콜 설계시 고려사항

#### A. 주소 표현, 오류 제어, 흐름 제어, 데이터 전달 방식

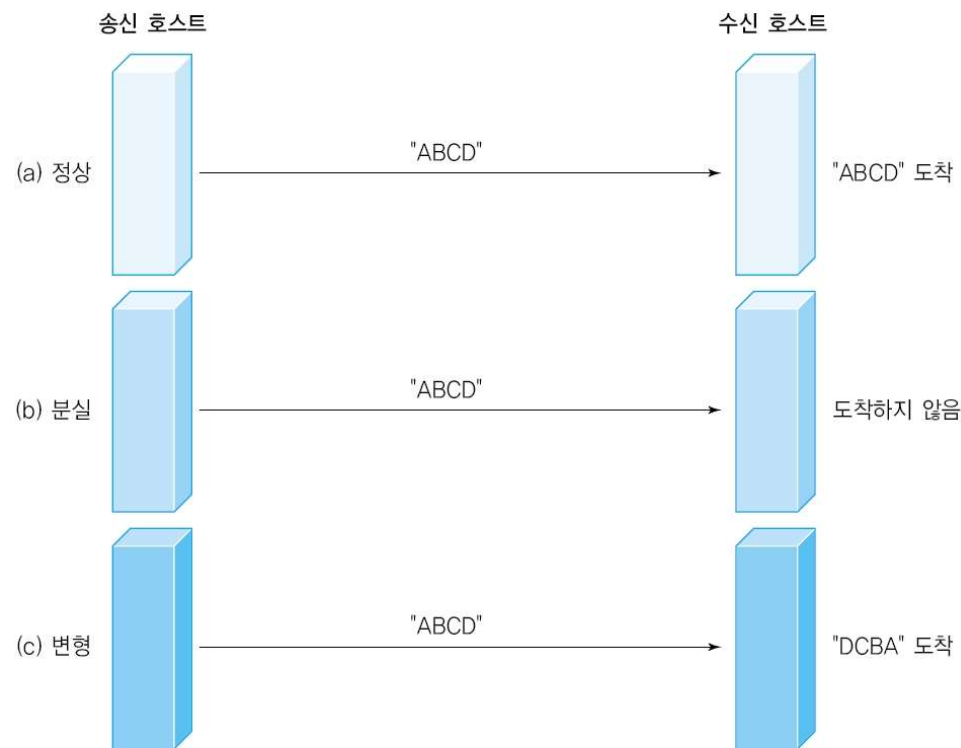
#### \* 고려사항1: 주소 표현

- A. 주소의 역할: 서로를 구분
- B. 주소의 활용도를 높이기 위하여 **구조적 정보**를 포함
  - 전화번호: 국가 코드 - 지역 코드 - 번호
  - 주민번호: yymmdd-abcdefgh
- C. 1:many 통신을 지원
  - **브로드캐스팅(Broadcasting)**: 모든 호스트에 데이터 전달
  - **멀티캐스팅(Multicasting)**: 특정 호스트에게 데이터 전달

## 7. 네트워크 모델: 계층 구조의 개념 -6

### \* 고려사항2: 오류 제어(Error control)

- A. 데이터 **변형** 오류: 데이터가 깨져서 수신자에게 도착
- B. 데이터 **분실** 오류: 데이터가 수신자에게 도착하지 못함
- C. 오류제어는 **데이터링크 계층의 주요기능**으로 재전송 방식 사용함

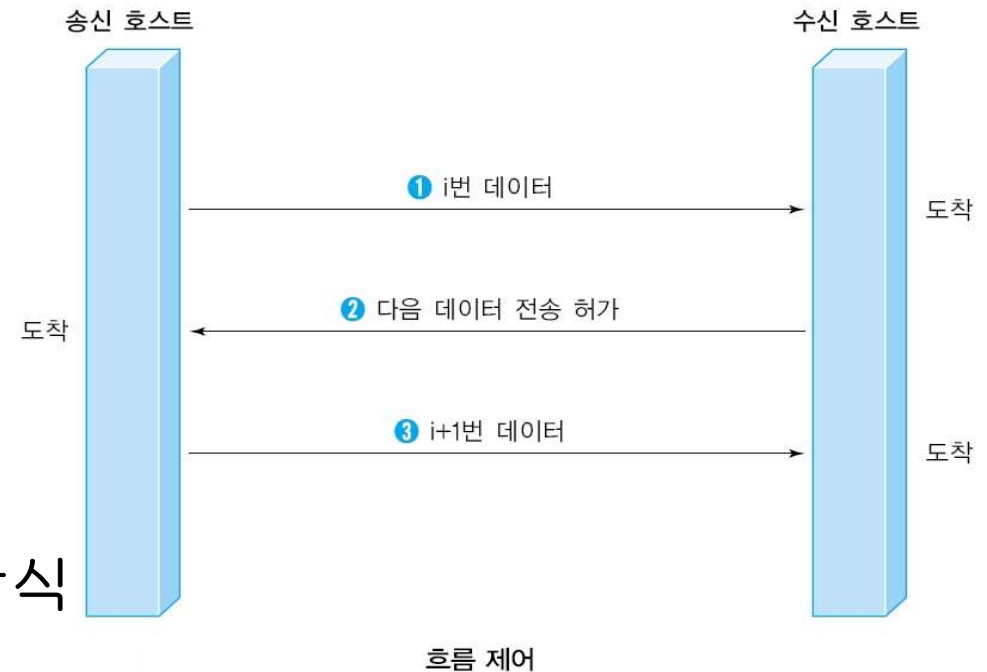


가능한 전송 결과 유형

## 7. 네트워크 모델: 계층 구조의 개념 -7

### \* 고려사항3: 흐름 제어(Flow control)

- A. 수신자 처리능력에 비해 너무 빨리 데이터를 전송하지 않도록 제어
- B. 수신 버퍼가 부족하면 수신자는 데이터를 수신 처리할 수 없다



### \* 고려사항4: 데이터 전달 방식 (Data transfer)

단방향(Simplex): 데이터를 한쪽 방향으로만 전송

전이중(Full Duplex): 데이터를 양쪽에서 동시에 전송

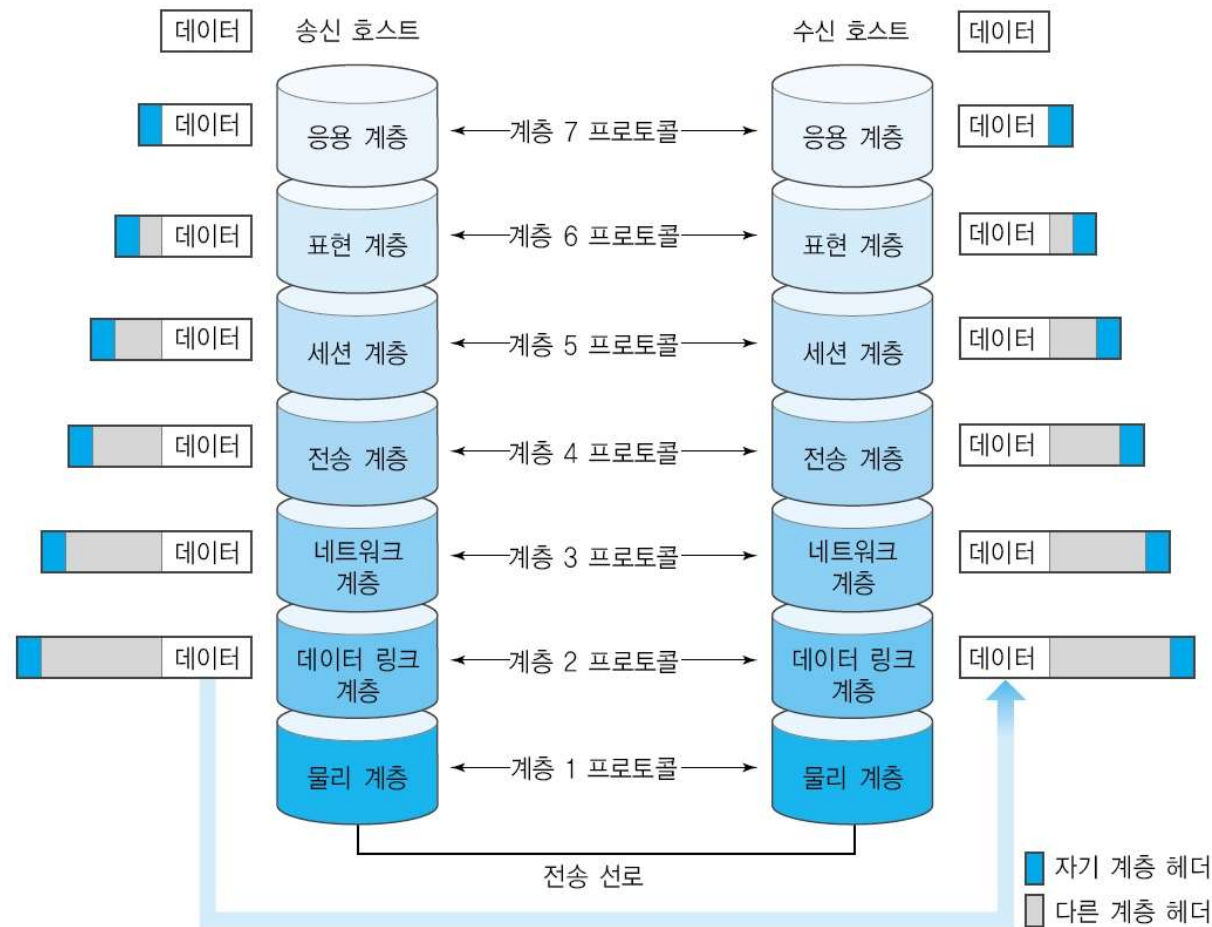
반이중(Half Duplex): 양방향으로 전송할 수 있지만,

특정 시점에서는 한쪽 방향으로만 전송

## 8. 네트워크 모델: OSI 참조 모델 -1

### 1. OSI 7 계층모델

(Open Systems Interconnection 7 Layer-model)



OSI 7계층 모델의 동작



## 8. 네트워크 모델: OSI 참조 모델 -2

### 1. 용어정의

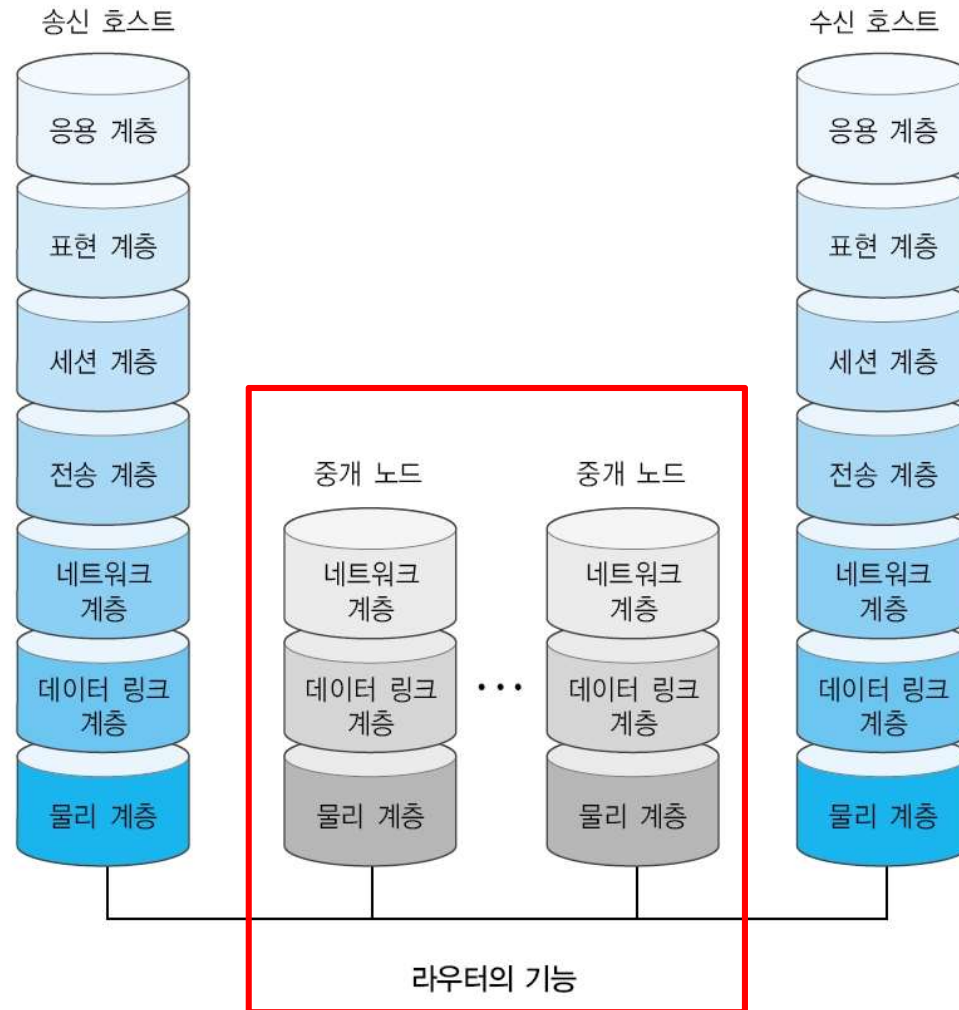
- A. 계층n 프로토콜: 계층n 모듈끼리 사용하는 통신 규칙
- B. 동료 프로세스: 동일 계층에 위치한 통신 양단 프로세스
- C. 인터페이스: 상하위 계층 사이의 접속 방법
- D. 서비스: 상위 계층이 하위 계층을 사용하는 방법

### 2. 헤더 정보

- A. 송신호스트(Sender):  
데이터가 상위계층에서 하위계층으로 갈수록 **헤더 추가**
- B. 수신호스트(Receiver):  
데이터가 하위계층에서 상위계층으로 갈수록 **헤더 제거**

## 8. 네트워크 모델: OSI 참조 모델 -3

### 1. 중계 기능



## 8. 네트워크 모델: OSI 참조 모델 -4

### 1. 계층별 기능

#### A. 물리 계층(Physical Layer)

- 데이터 전송 속도, 클럭 동기화 방법, 물리적 연결 형태 등

#### B. 데이터링크 계층(Datalink Layer)

- 물리 계층의 **물리적 전송 오류** 문제를 해결
- 프레임(Frame): 전송 데이터의 명칭

#### C. 네트워크 계층(Network Layer)

- 데이터의 **전송 경로**를 결정
- **호스트 구분을 위한 주소 개념 필요** (예: IP 주소)
- 패킷(Packet): 전송 데이터의 명칭

## 8. 네트워크 모델: OSI 참조 모델 -5

### 1. 계층별 기능 (계속)

#### A. 전송 계층(Transfer Layer)

- 송수신 프로세스 사이의 단대단 통신 기능을 지원
- 프로세스 구분을 위한 주소 개념 필요 (예: 포트 번호)

#### B. 세션 계층(Session Layer)

- 송수신자 사이에 상위적 연결 개념인 세션을 지원

#### C. 표현 계층(Presentation Layer)

- 데이터의 의미와 표현 방법을 처리
- 암호화/압축 기능도 처리

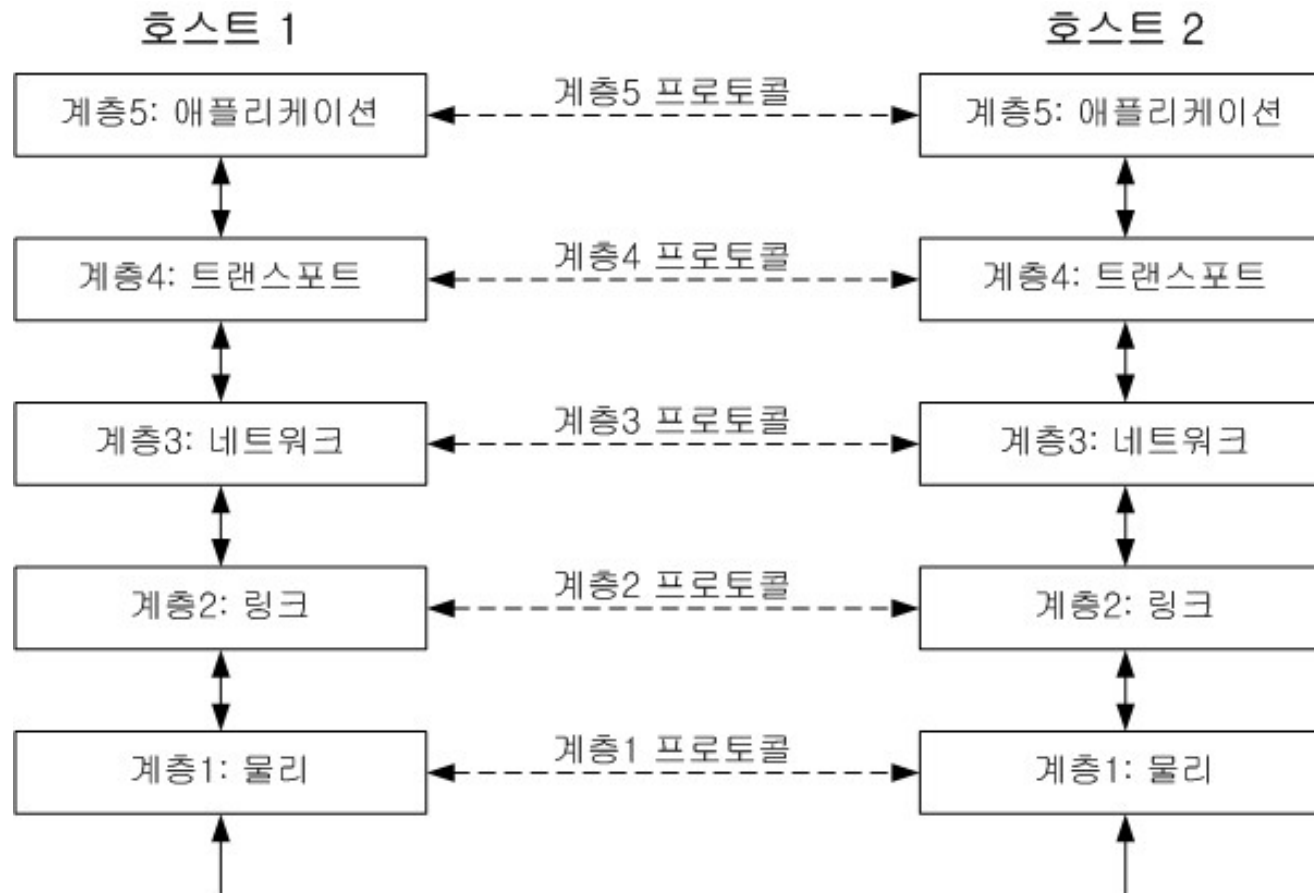
#### D. 응용 계층(Application Layer)

- 대표적인 인터넷 서비스: FTP, Telnet, 전자 메일

## 8. 네트워크 모델: TCP/IP 모델 (인터넷) -6

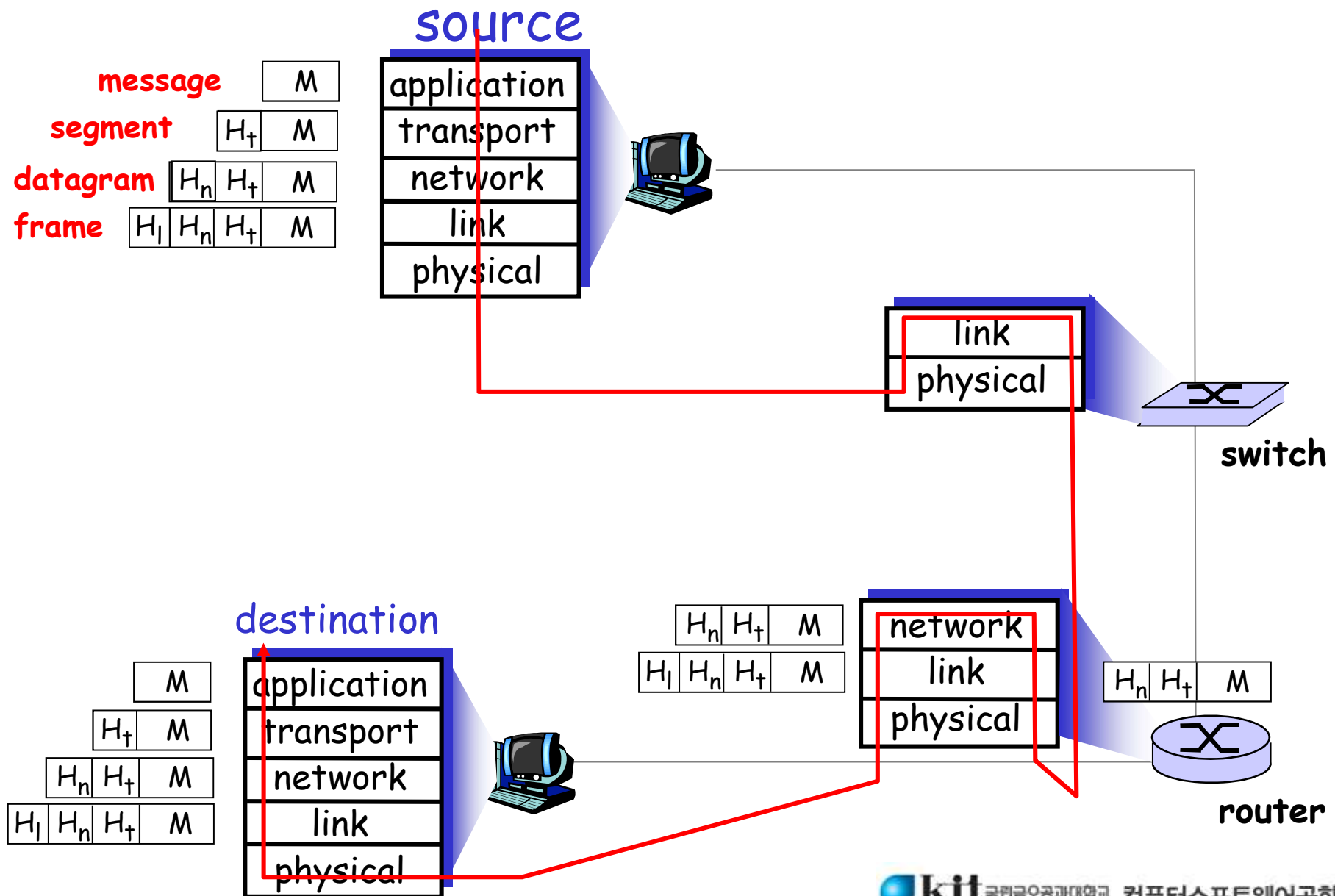
### 1. 5 계층으로 구성

ARPANET  
미국, 국방부



**TCP** – Transmission Control Protocol  
**IP** – Internet Protocol

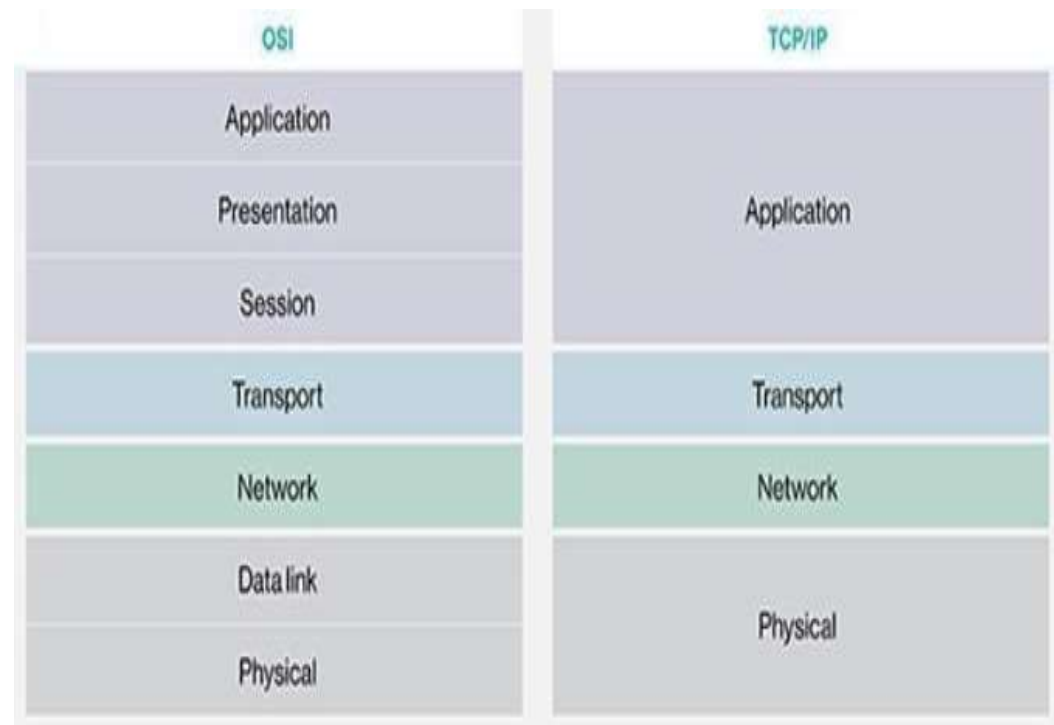
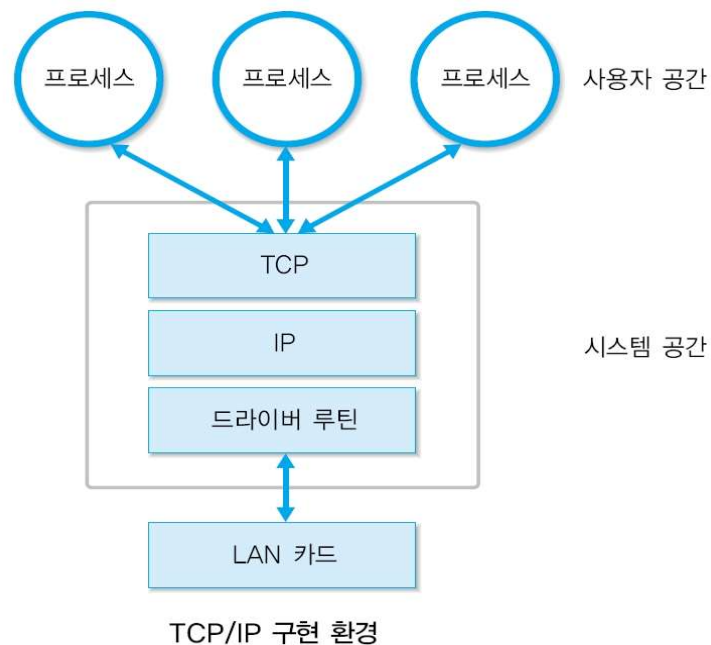
## 8. 네트워크 모델: TCP/IP 모델 (인터넷) -7



## 8. 네트워크 모델: TCP/IP 모델 (인터넷) -8

### 1. 구현 환경 (OSI 7 Layer 모델과 관계)

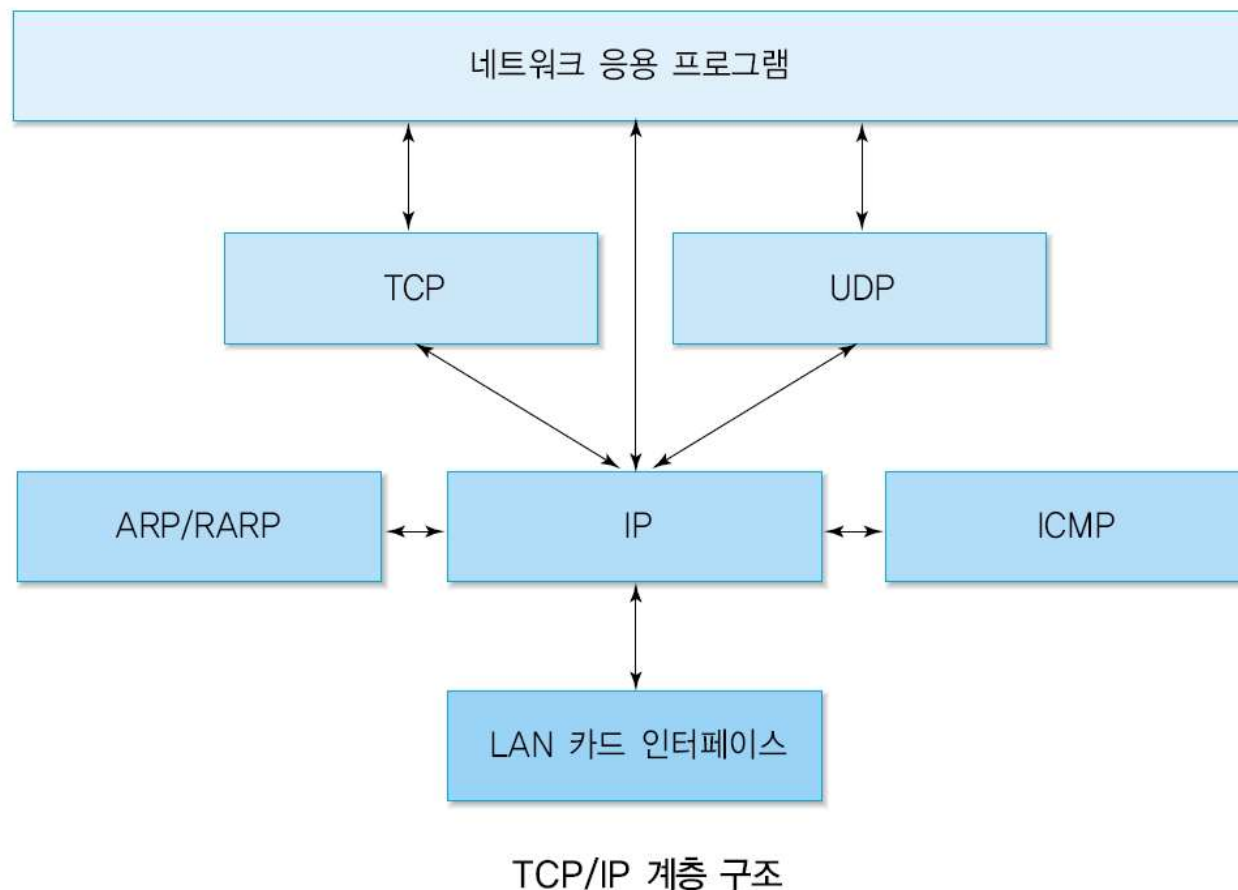
- A. 시스템 공간(계층 1 ~ 4): 운영체제에서 동작
- B. 사용자 공간(계층 5 ~ 7): 사용자 프로그램으로 동작



## 8. 네트워크 모델: TCP/IP 모델 (인터넷) -9

### 1. 프로토콜

#### A. TCP/IP 계층 구조





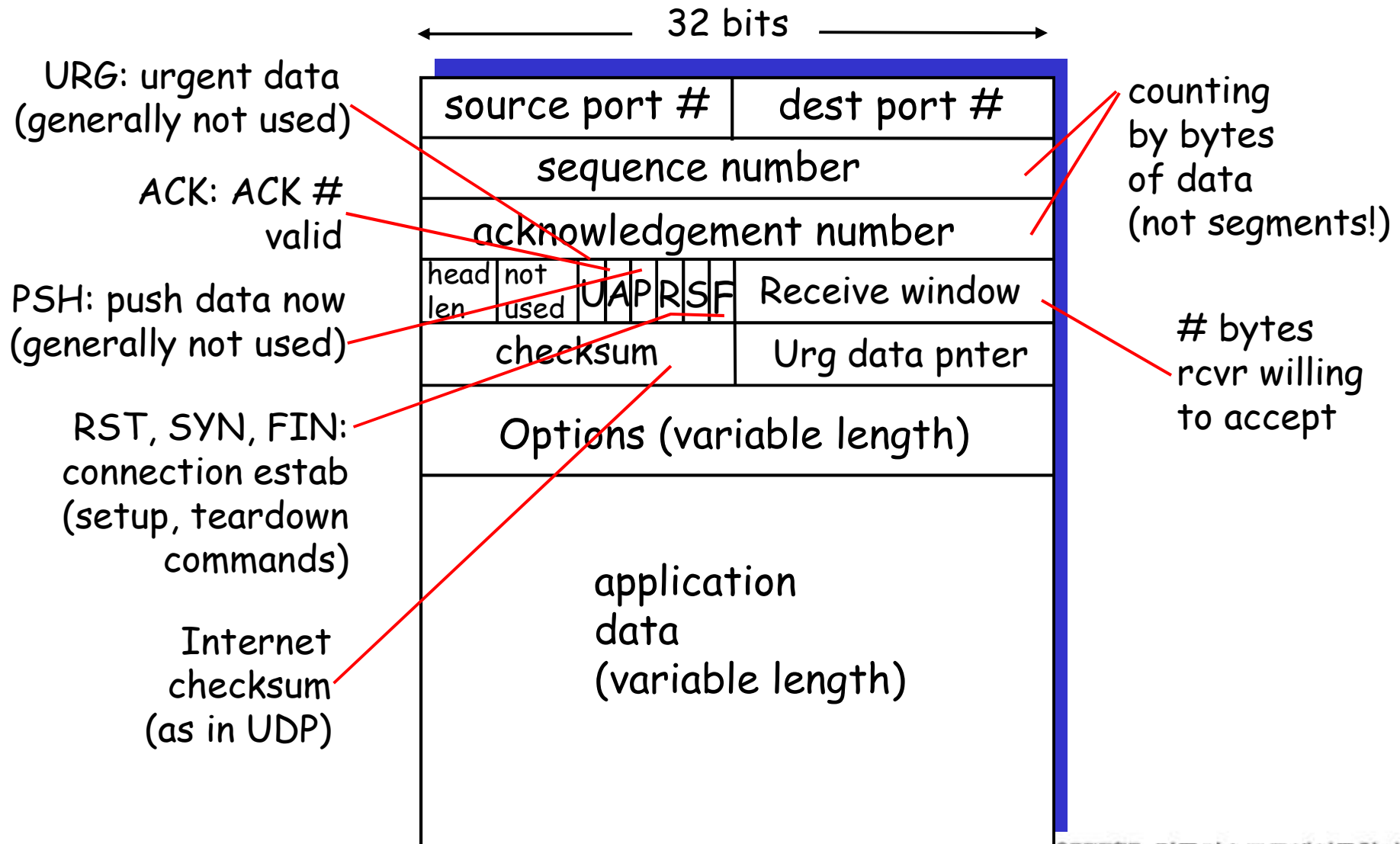
## 8. 네트워크 모델: TCP/IP 모델 (인터넷) -10

### 1. TCP (Transmission control protocol)

- A. 연결형 서비스를 지원
- B. 전이중 방식의 양방향 가상 회선을 제공
- C. 신뢰성 있는 데이터 전송을 보장
- D. 흐름 제어
  - 수신자의 처리량을 초과하여 전송하지 않음
- E. 혼잡 제어
  - 네트워크 라우터 처리량을 초과하여 전송하지 않음
- F. 오류 제어
  - 데이터 변형, 데이터 분실 오류
  - 재전송 기능으로 복구

## 8. 네트워크 모델: TCP/IP 모델 (인터넷) -11

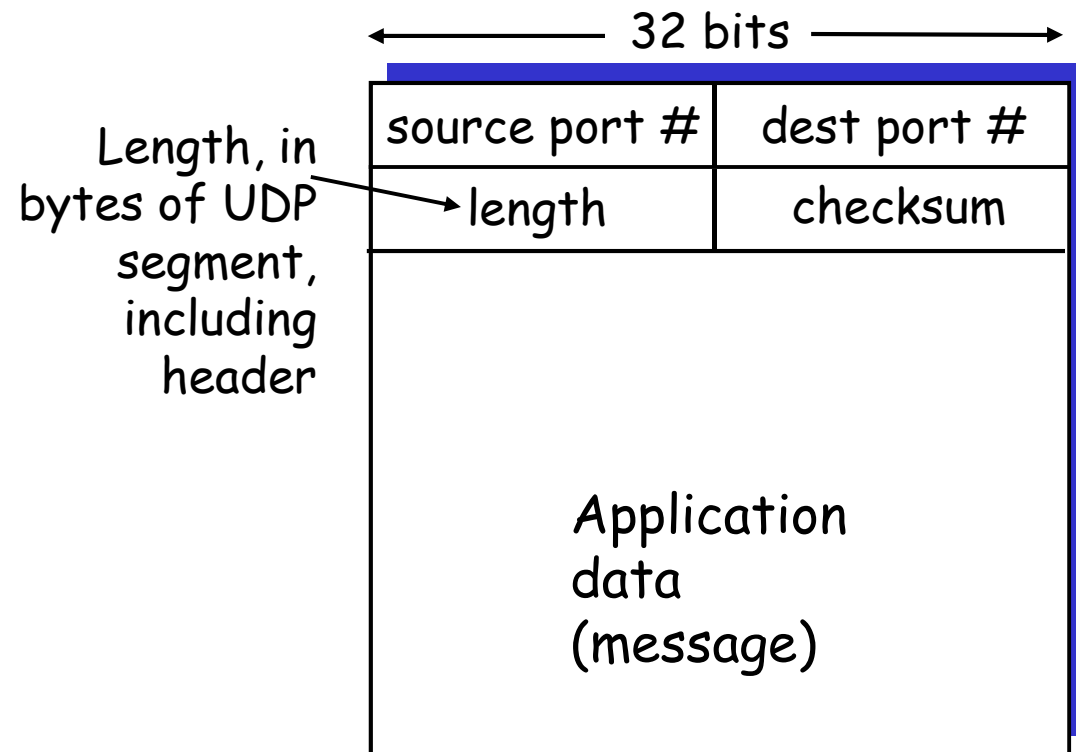
### 1. TCP (Transmission control protocol)



## 8. 네트워크 모델: TCP/IP 모델 (인터넷) -12

### 2. UDP (User datagram protocol)

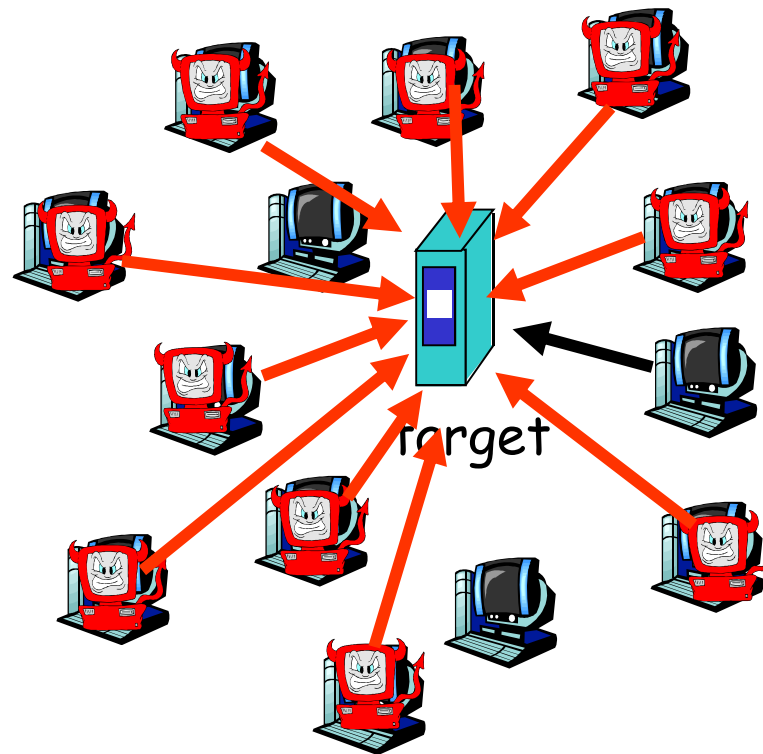
- A. 비연결형 서비스를 제공
- B. 헤더와 전송 데이터에 대한 체크섬 기능을 제공
- C. Best Effort 전달 방식을 지원



## 9. 네트워크 보안 -1

### 1. Denial of service (DOS) attacks

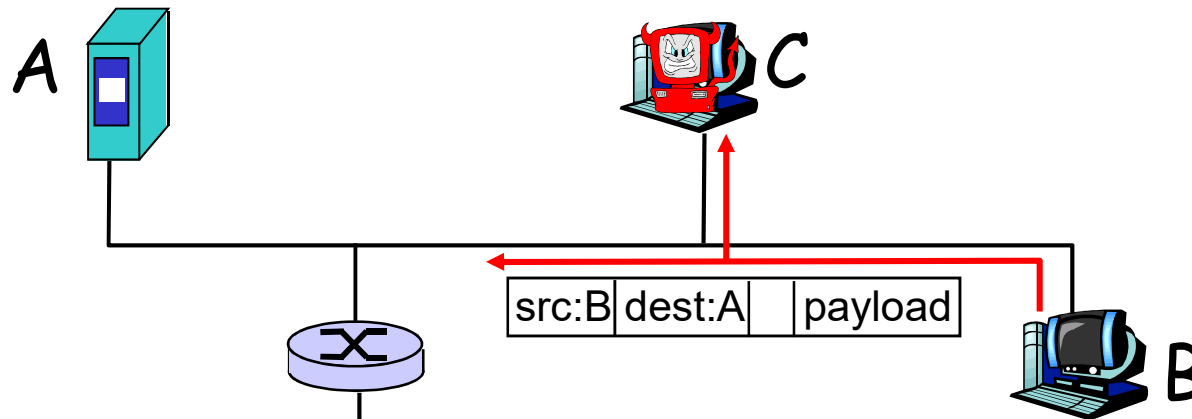
- A. attackers make resources (server, bandwidth) **unavailable** to legitimate traffic by overwhelming resource with bogus traffic
- B. DDOS (Distributed Denial of Service)



## 9. 네트워크 보안 -2

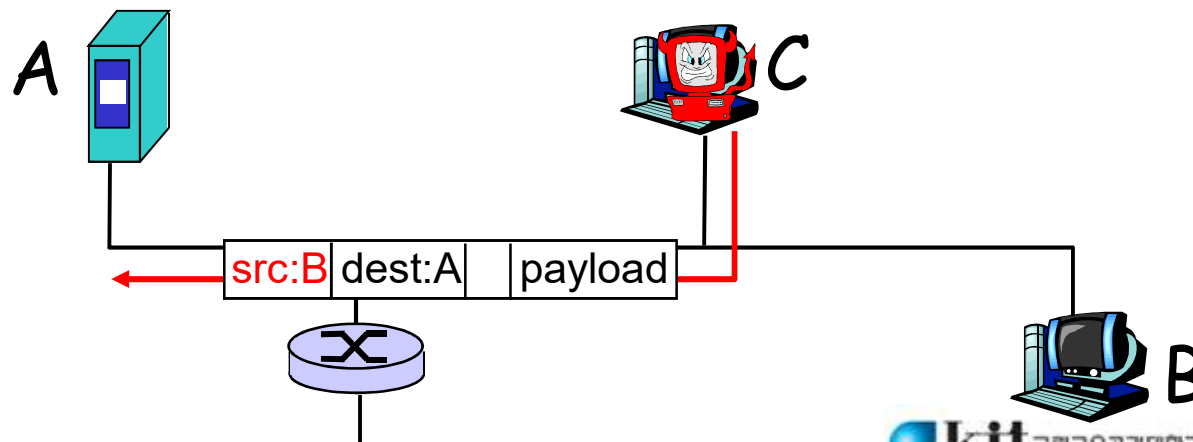
### 2. 패킷 스니핑 (Packet sniffing)

A. 수동적으로 메시지를 기록한 후에 분석



### 3. IP 스푸핑 (IP Spoofing)

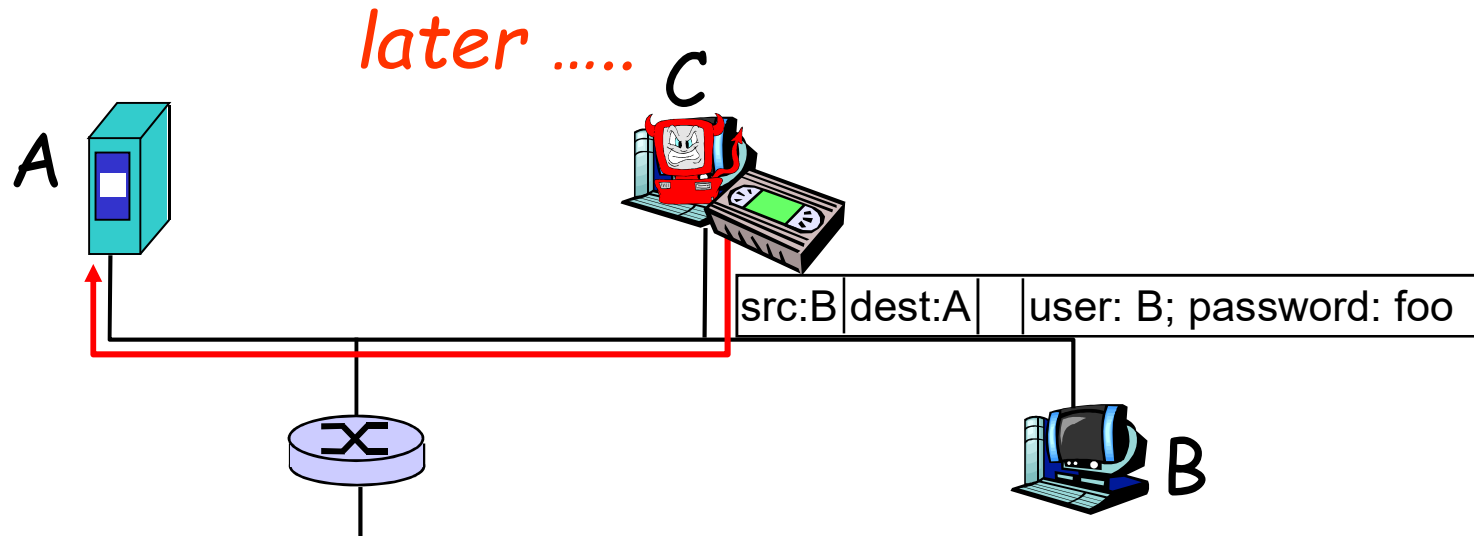
A. 거짓 소스 주소를 갖는 패킷을 전송



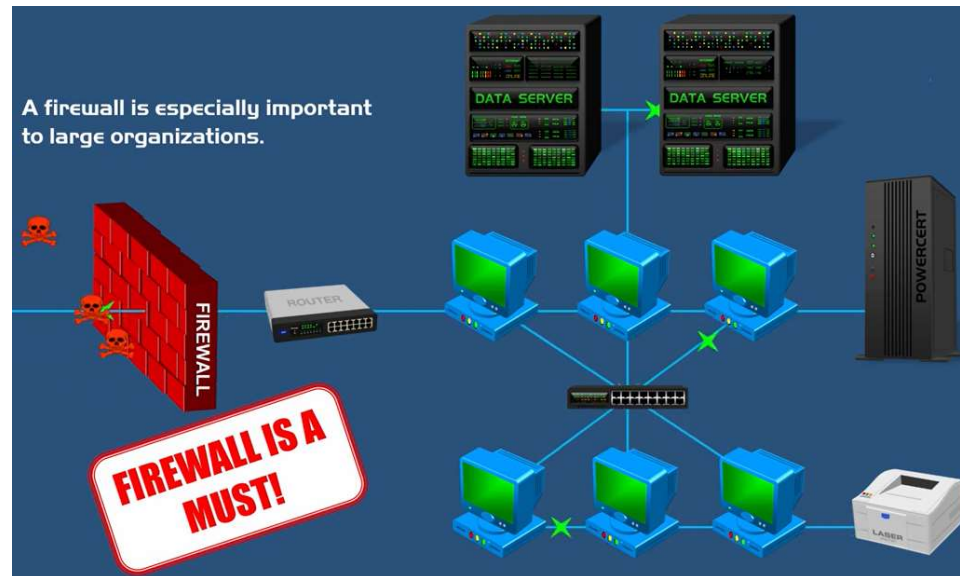
## 9. 네트워크 보안 -3

### 4. record-and-playback

A. 민감한 정보를 저장해두었다가 차후에 사용하는 방법



## 9. 네트워크 보안: Firewall -4



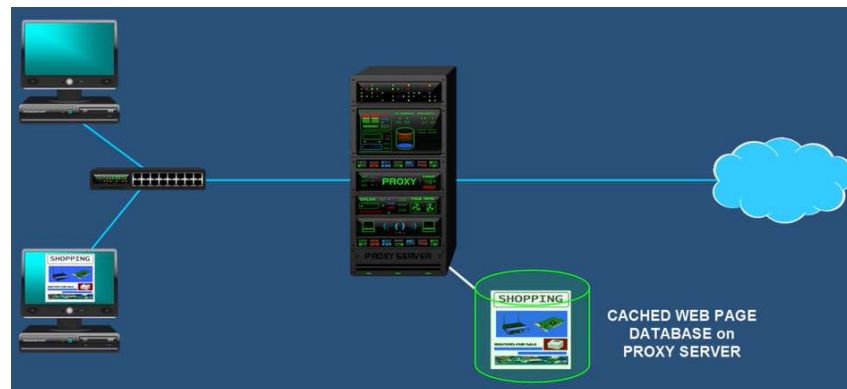
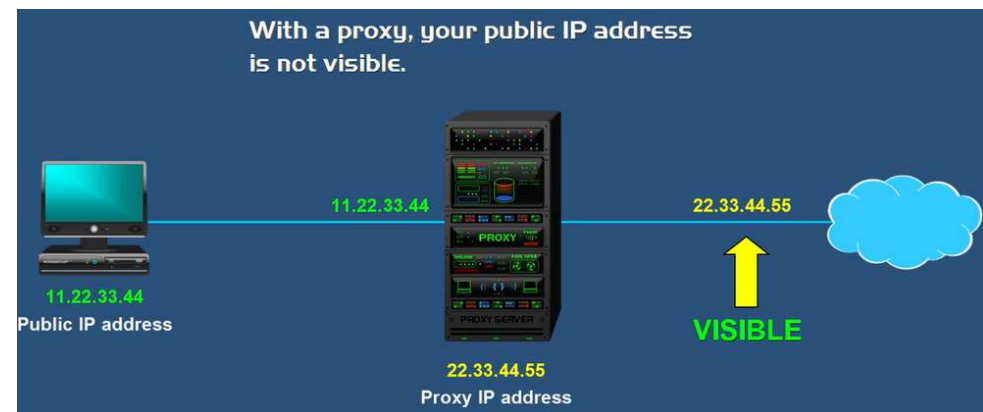
Firewall rules can be based on:

- IP addresses
- Domain names
- Protocols
- Programs
- Ports
- Key words

**ACCESS CONTROL LIST**

Permission	IP Address	Protocol	Destination	Port
ALLOW	162.213.214.14	TCP	10.10.10.2	80
ALLOW	54.21.66.112	TCP	ANY	80
DENY	192.168.1.1	TCP	ANY	80
ALLOW	65.252.1.2	TCP	ANY	80
DENY	ANY	TCP	ANY	80
ALLOW	ANY	TCP	ANY	80
DENY	ANY	UDP	10.10.10.1	23
DENY	255.255.255.0	TCP	ANY	25
ALLOW	10.10.0.1	TCP	ANY	110

## 9. 네트워크 보안 Proxy Server -5



### Proxy benefits:

- Privacy – allows you to surf the internet anonymously.
- Speed
- Saves bandwidth
- Activity logging



# 10. WWW(World Wide Web): 웹 시스템 구조 -1

## 1. 웹 서비스: TCP 포트 80번

참고명령

\$ ps -A

다양한 daemon

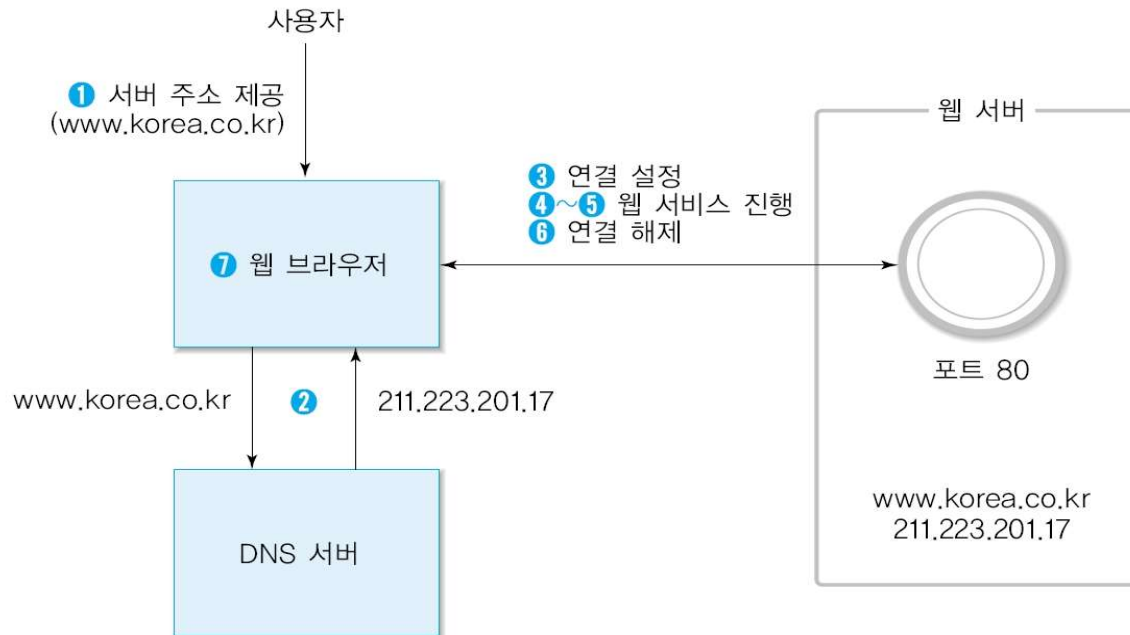
## 2. 클라이언트-서버 모델

### A. URL (Uniform Resource Locator)

- 웹 서버의 자원 명칭
- 사용하는 프로토콜, 서버의 호스트 이름, 서버 내부의 파일 경로명으로 구성
- 예: <http://www.korea.co.kr/welcom.html>
- UNIX/Linux 시스템
  - ❖ 로그인 이름 hong
  - ❖ 홈 디렉토리/public\_html/index.html
  - ❖ <http://www.korea.co.kr/~hong>

# 10. WWW: 웹 시스템 구조 -2

## 3. 연결 설정과 해제

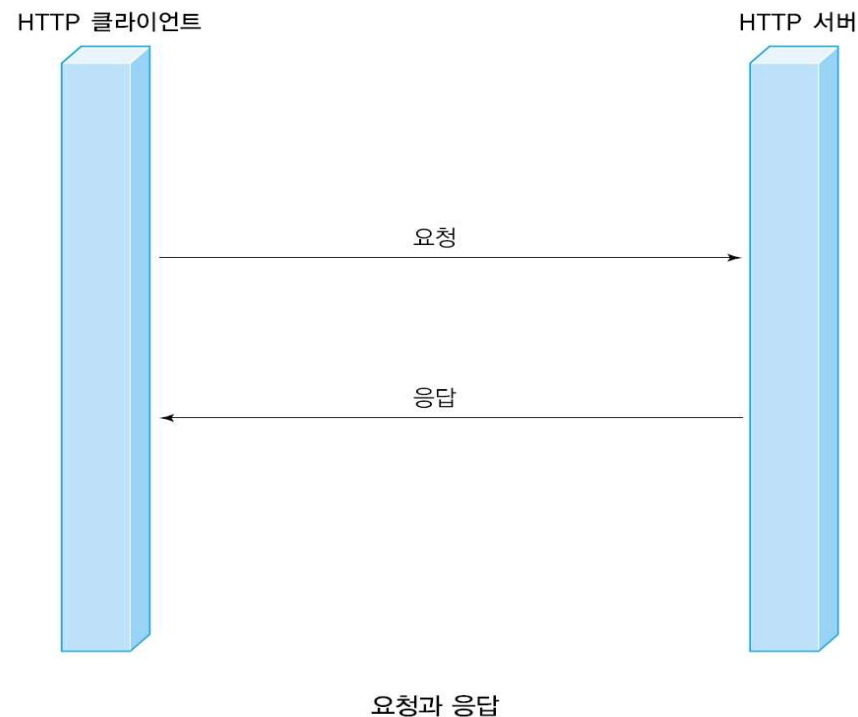


- A. 사용자가 웹 브라우저에게 웹 서버의 URL 주소 입력
- B. 웹 브라우저가 DNS 서버에게
- C. 웹 서버의 호스트 이름을 IP 주소로 변경 요청
- 웹 브라우저가 <IP 주소 + 포트 80번>의 웹 서버와 TCP 접속 시도
- D. 웹 브라우저가 웹 서버에게 최초 화면을 위한 GET 명령 전송
- E. 웹 서버가 웹 브라우저에게 요청한 웹 문서를 회신
- F. 웹 브라우저와 웹 서버 사이의 연결 해제
- G. 웹 브라우저가 사용자 화면에 웹 문서를 출력

# 10. WWW: HTTP (Hypertext transfer protocol) -3

## 1. 요청과 응답: 비연결(Connectionless)/비상태(Stateless) 특징

- A. 비연결: 클라이언트 요청에 응답한 후에 바로 연결을 끊음
- B. 비상태: 서버의 상태에 상관없이 요청 수행



# 10. WWW: HTTP (Hypertext transfer protocol) -4

## 2. 요청과 응답: 요청 메시지

### A. 요청문

- 요청 메소드

- ❖ 요청 메소드의 종류

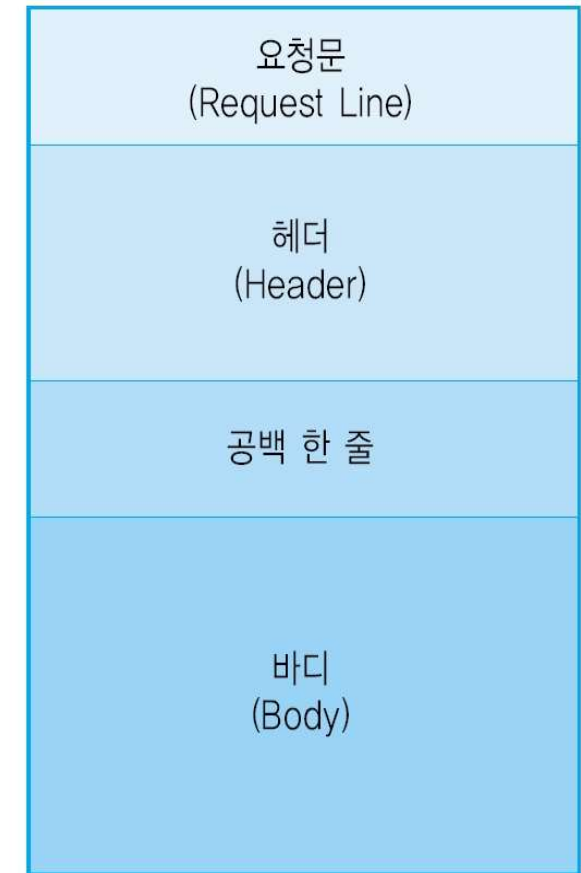
- » GET: 클라이언트가 서버에 웹 문서를 요청
    - » HEAD: 문서 내용보다 문서 정보를 요청
    - » POST: 클라이언트가 서버에 정보 전송
    - » PUT: 클라이언트가 서버에 문서 전송

- URL

- HTTP 버전

### B. 헤더 (Header)

### C. 바디 (Body)



요청 메시지

# 10. WWW: HTTP (Hypertext transfer protocol) -5

## 3. 요청과 응답: 응답 메시지

### A. 상태문

- HTTP 버전
- 상태 코드

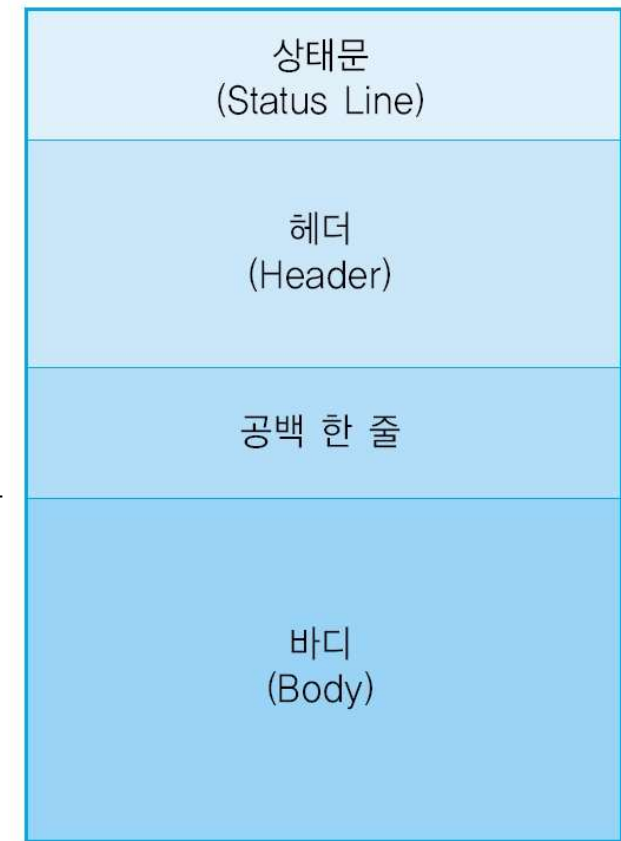
#### ❖ 상태 코드의 종류

- » 200 OK: 요청이 성공적으로 수행
- » 202 Accepted: 요청이 수신되었으나, 즉각 실행되지 않고 있음
- » 400 Bad Request: 요청 메시지의 문법 오류
- » 401 Unauthorized: 요청의 실행에 필요한 권한이 없음
- » 403 Forbidden: 요청이 거부됨
- » 404 Not Found: 원하는 문서를 찾을 수 없음
- » 500 Internal Server Error: 서버에 오류 발생
- » 501 Not Implemented: 요청을 수행할 수 없음

- 상태 이름

### B. 헤더 (Header)

### C. 바디 (Body)

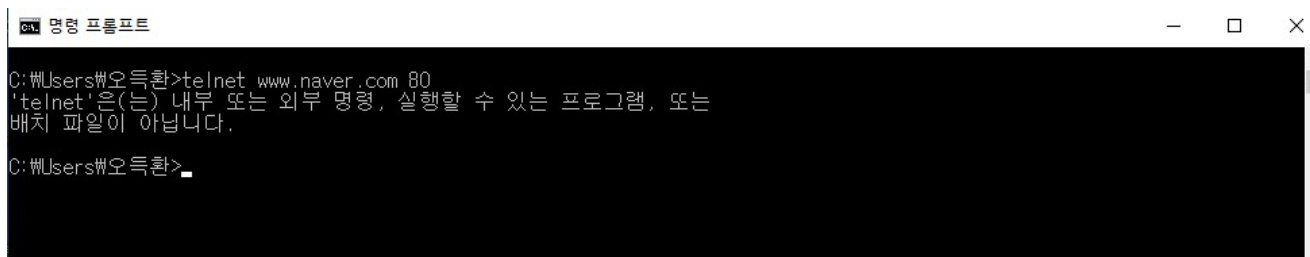


응답 메시지

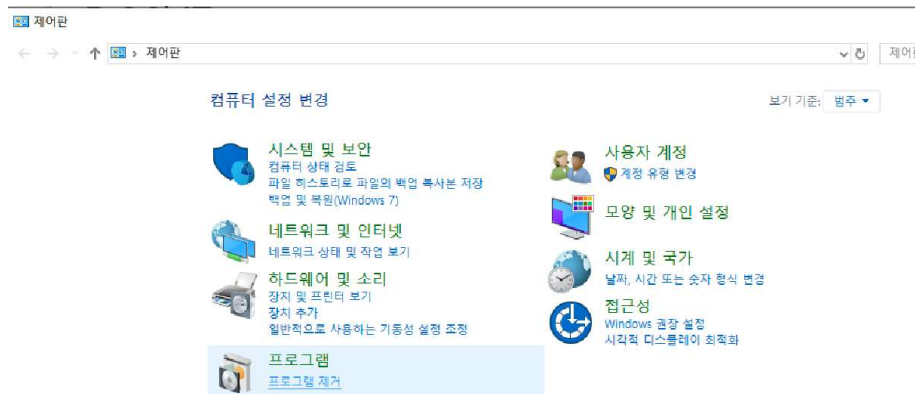
# 10. WWW: HTTP (Hypertext transfer protocol) -6

## 4. Telnet을 사용하여 HTTP 접속 테스트를 위한 설정

- A. 이번 실습은 HTTP의 동작을 telnet이라는 원격 연결(터미널)프로그램을 이용하여 살펴본다.
- B. 먼저 Windows 10에서 telnet을 사용할 수 있도록 설정해줘야 함.



### telnet Client 사용 설정이 안된 경우

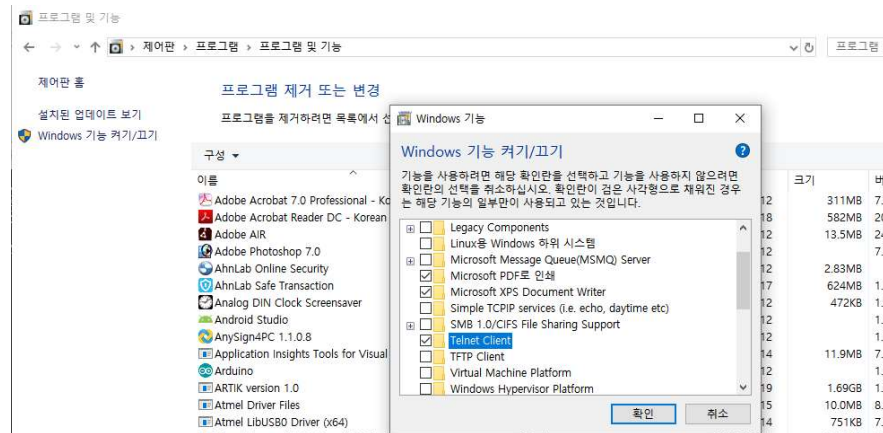


1. 제어판에서 프로그램 클릭

# 10. WWW: HTTP (Hypertext transfer protocol) -6

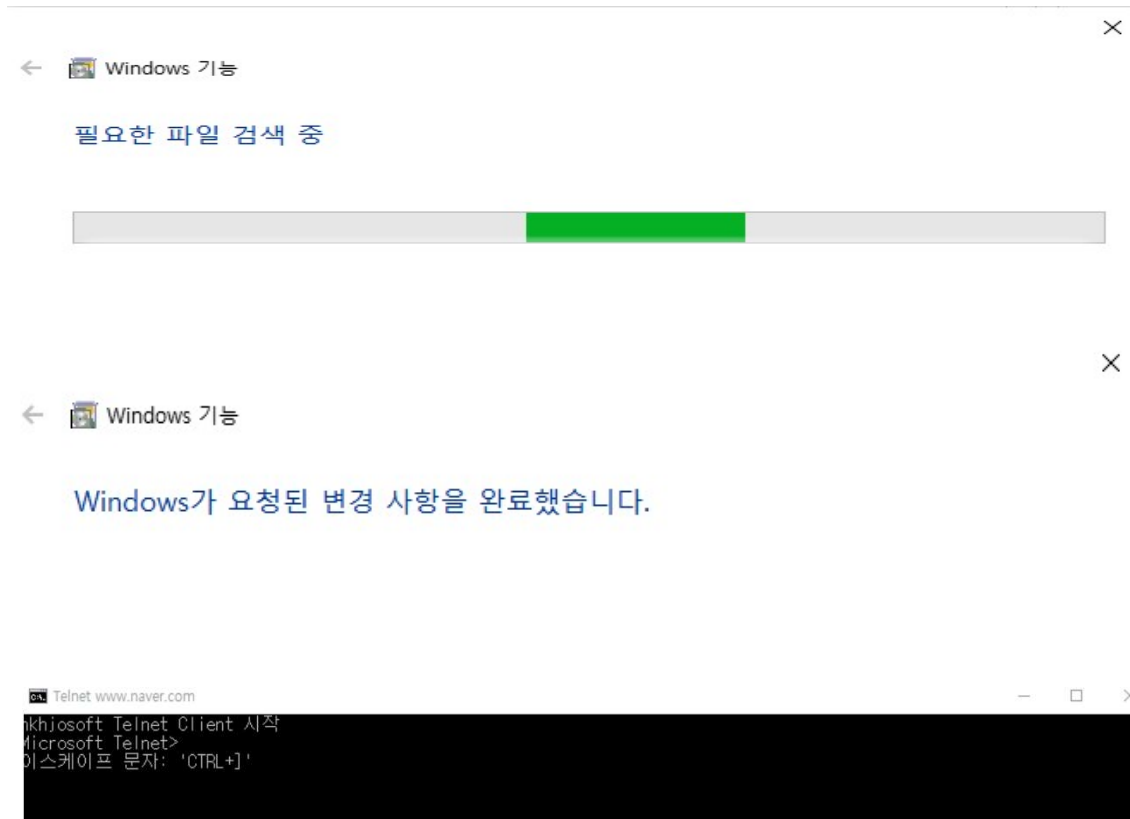


## 2. Windows 기능 켜기/끄기



## 3. telnet Client – 클릭 - 확인

## 10. WWW: HTTP (Hypertext transfer protocol) -6



telnet Client 사용 설정된 경우



# 10. WWW: HTTP (Hypertext transfer protocol) -6

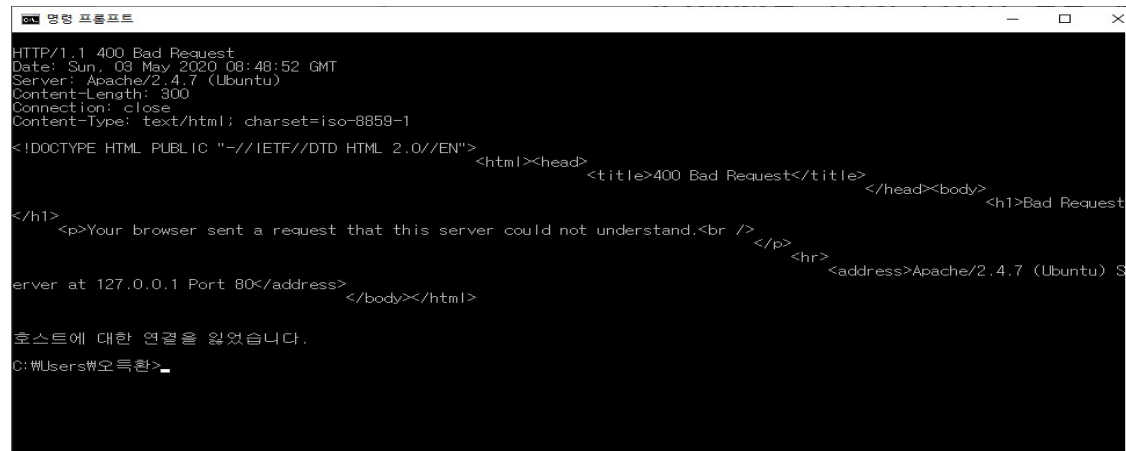
## 5. Telnet을 사용하여 HTTP 접속 테스트

A. telnet 사용을 설정 하였기에

B. [윈도우 cmd 창에서] **telnet se.kumoh.ac.kr 80** [ENTER]

이 명령은 se.kumoh.ac.kr host에 80번 port로 telnet을 연결하는 명령

C. 검게 보이는 [윈도우 cmd 창에서] **GET/HTTP/1.1** [ENTER]



```
명령 프롬프트
HTTP/1.1 400 Bad Request
Date: Sun, 03 May 2020 08:48:52 GMT
Server: Apache/2.4.7 (Ubuntu)
Content-Length: 300
Connection: close
Content-Type: text/html; charset=iso-8859-1

<!DOCTYPE HTML PUBLIC "-//IETF//DTD HTML 2.0//EN">
<html><head>
  <title>400 Bad Request</title>
</head><body>
  <h1>Bad Request</h1>
  <p>Your browser sent a request that this server could not understand.<br />
  </p>
  <hr>
  <address>Apache/2.4.7 (Ubuntu) S
</body></html>

호스트에 대한 연결을 잃었습니다.
C:\Users\W오득환>
```

요청에 대한 파일을  
제공한 후에  
연결이 종료됨.

D. 추가 [윈도우 창 창에서]

1. telnet www.naver.com 80 [ENTER]

GET

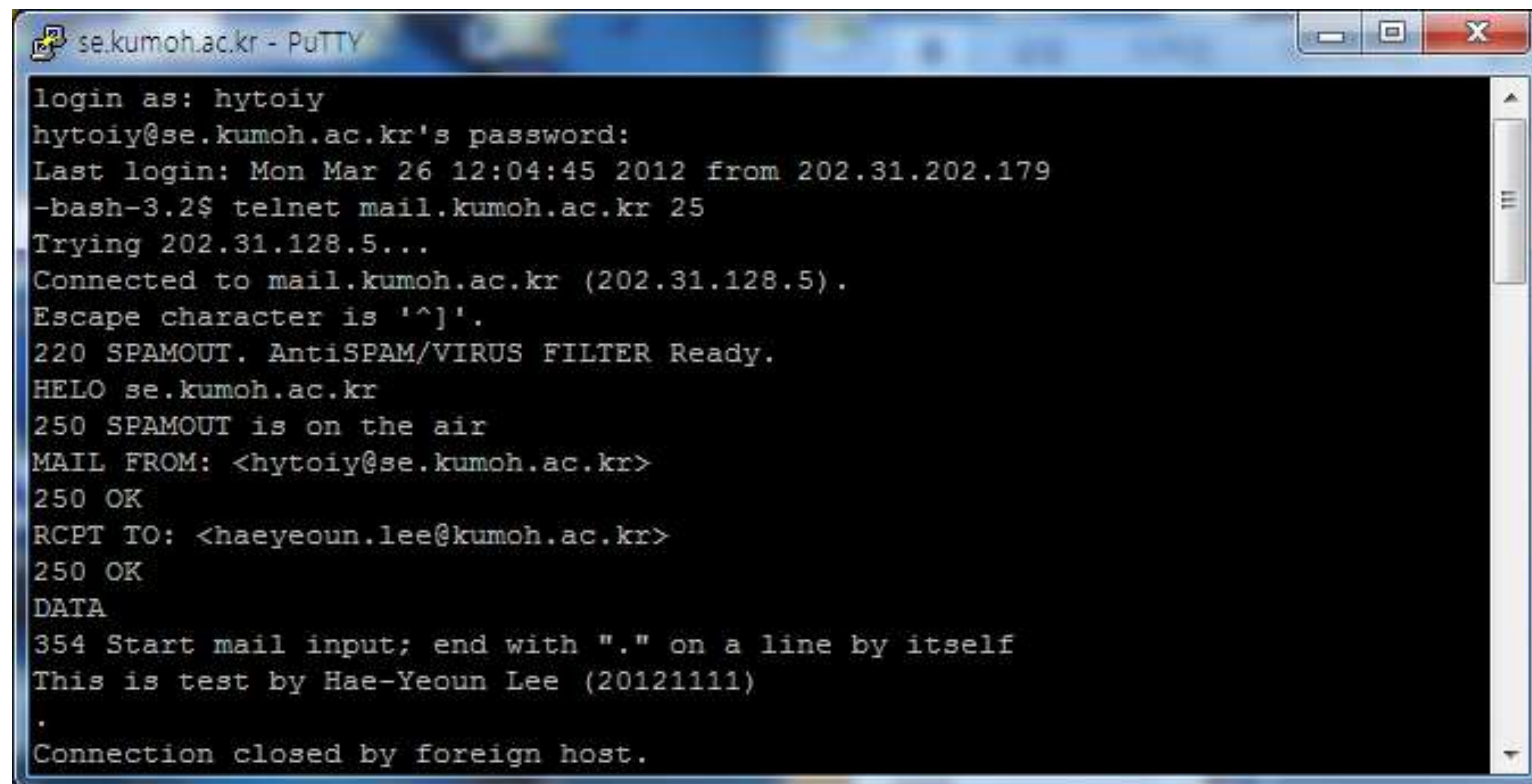
2. telnet kumoh.ac.kr 80

GET /index.html HTTP/1.1

# 11. Try SMTP interaction for yourself: -1

1. telnet **servername** 25
2. see 220 reply from server
3. enter HELO, MAIL FROM, RCPT TO, DATA, QUIT commands

**SMTP – Simple Mail  
Transfer Protocol**



```
se.kumoh.ac.kr - PuTTY
login as: hytoiy
hytoiy@se.kumoh.ac.kr's password:
Last login: Mon Mar 26 12:04:45 2012 from 202.31.202.179
-bash-3.2$ telnet mail.kumoh.ac.kr 25
Trying 202.31.128.5...
Connected to mail.kumoh.ac.kr (202.31.128.5).
Escape character is '^]'.
220 SPAMOUT. AntiSPAM/VIRUS FILTER Ready.
HELO se.kumoh.ac.kr
250 SPAMOUT is on the air
MAIL FROM: <hytoiy@se.kumoh.ac.kr>
250 OK
RCPT TO: <haeyeoun.lee@kumoh.ac.kr>
250 OK
DATA
354 Start mail input; end with "." on a line by itself
This is test by Hae-Yeoun Lee (20121111)
.
Connection closed by foreign host.
```

## 11. Try SMTP interaction for yourself: -2

HELO se.kumoh.ac.kr

MAIL FROM: 이름-학번@se.kumoh.ac.kr

RCPT TO: dhoh@kumoh.ac.kr

DATA

To: dhoh@kumoh.ac.kr

From: 이름-학번@se.kumoh.ac.kr

Subject: Computer Basics Class

Type 내 이름은 누구누구 입니다.

Type 학번-2020xxxxxx.

<enter>

<enter>

.

매일 전송/수신 확인