TCP/IP 교과서

# 6장 전송 계층

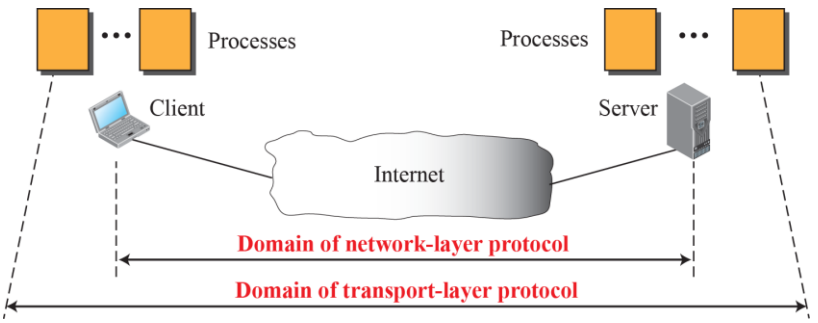


## 6.1 전송 계층 소개

TCP/IP 교과서

» 전송 계층 프로토콜의 역할

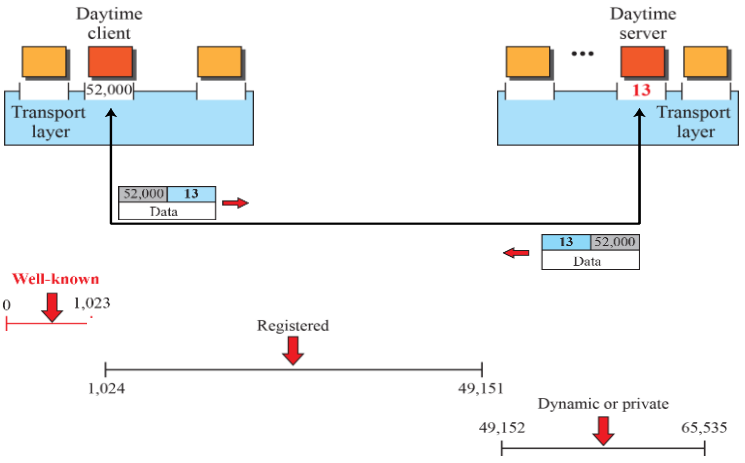
- **네트워크 애플리케이션 인터페이스:** 프로세스-프로세스간 통신 제공



# 6.1 전송 계층 소개

» 전송 계층 프로토콜의 역할

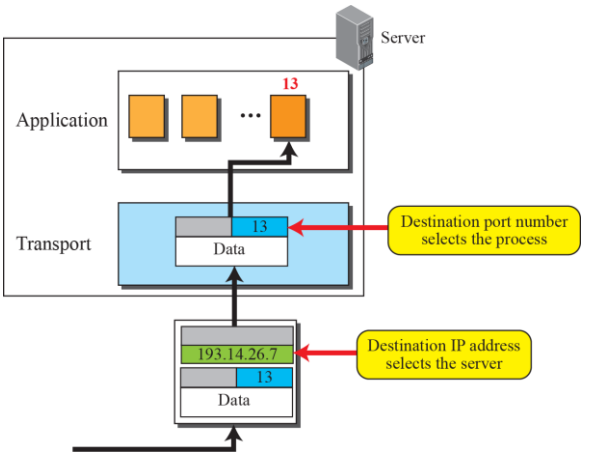
**포트번호**




# 6.1 전송 계층 소개

» 전송 계층 프로토콜의 역할

**IP주소와 포트번호**

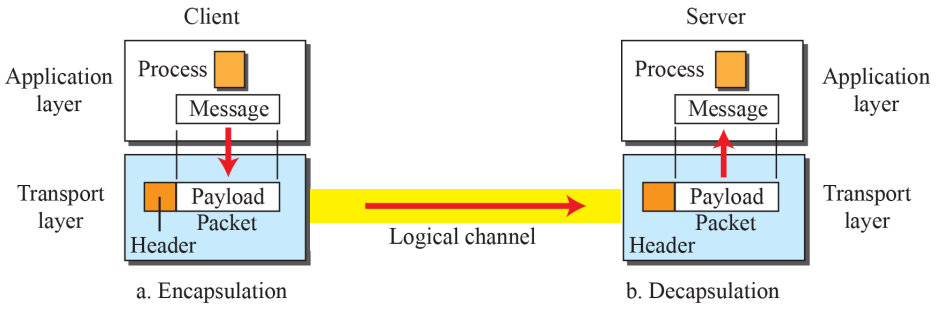


# 6.1 전송 계층 소개

TCP/IP 교과서

» 전송 계층 프로토콜의 역할

**전송계층 패킷의 캡슐화/역캡슐화**




# 6.1 전송 계층 소개

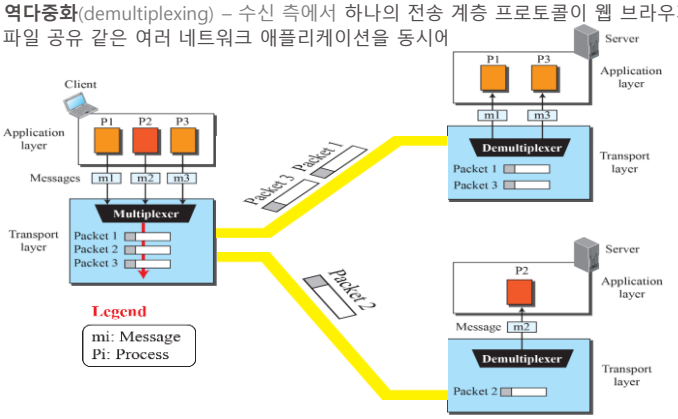
TCP/IP 교과서

» 전송 계층 프로토콜의 역할

- **다중화/역다중화 – 소켓 주소 이용**

**다중화**(multiplexing) – 송신 측에서 하나의 전송 계층 프로토콜이 서로 다른 애플리케이션들에서 데이터를 받아서 처리하는 기능. 즉, 여러 소스의 입력을 하나의 출력으로 묶는 작업

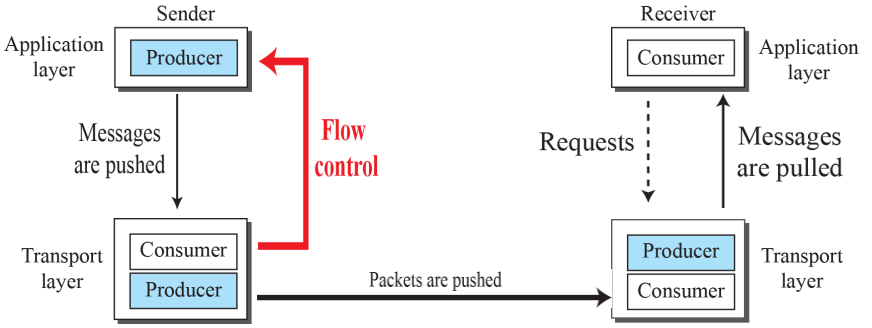
**역다중화**(demultiplexing) – 수신 측에서 하나의 전송 계층 프로토콜이 웹 브라우저, 이메일, 파일 공유 같은 여러 네트워크 애플리케이션을 동시에



# 6.1 전송 계층 소개

» 전송 계층 프로토콜의 역할

- **흐름 제어(flow control)** – 생산율과 소비율 사이의 균형 유지를 위한 제어
  - 버퍼 이용




# 6.1 전송 계층 소개

» 전송 계층 프로토콜의 역할

- **오류 제어(error control)**
  - 훼손된 패킷의 감지 및 폐기
  - 손실되거나 제거된 패킷을 추적하고 재전송
  - 중복 수신 패킷을 확인하고 폐기
  - 순서에 어긋나게 들어온 패킷을 버퍼에 저장

# 6.1 전송 계층 소개

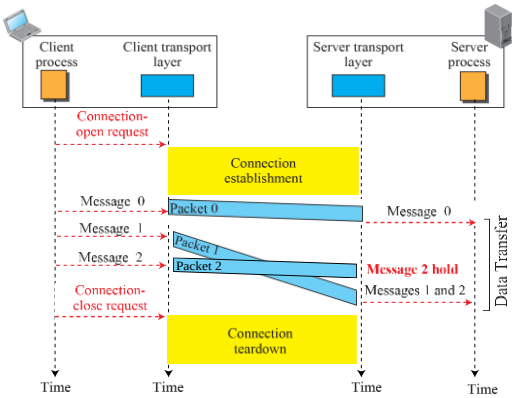
## 전송 계층 프로토콜의 역할

- **오류 제어(error control)**
  - **순서번호(sequence number)**
    - ✓ 오류제어를 수행하기 위해서 송신측 전송계층은 어떤 패킷이 재전송되어야 하는지 알아야 함
    - ✓ 또한 수신측 전송계층은 어떤 패킷이 중복 수신되었는지 또는 어떤 패킷이 순서에 어긋나게 도착되었는지 알아야 함
    - ✓ 패킷의 헤더에는 순서번호를 위한 필드가 설정되어야 함
      - ✓ 순서번호 필드의 크기가 m 비트라면, 그 범위는 $0 \sim 2^m-1$ (모듈로 $2^m$)
  - **확인응답(acknowledgement)**
    - ✓ 오류제어를 위해서 긍정과 부정신호 모두 사용될 수 있으나 전송계층에서는 일반적으로 긍정신호를 사용함
    - ✓ 수신측에서 오류 없이 잘 수신한 패킷에 대해서만 확인응답 (ACK)을 전송
    - ✓ 송신측은 타이머를 사용하여 패킷의 손실을 감지할 수 있음

# 6.2 전송 계층의 개념

## 연결 지향 및 비연결 프로토콜

- **연결 지향 프로토콜:** 전송 과정에서 통신하는 컴퓨터 간의 연결을 설정 및 유지하고 해당 연결 상태를 모니터링. 송신 측은 각 패킷이 오류 없이 전달되었는지 확인하며 필요하다면 패킷을 재전송. 전송이 완료되면 송수신 컴퓨터가 정상적으로 연결을 닫음

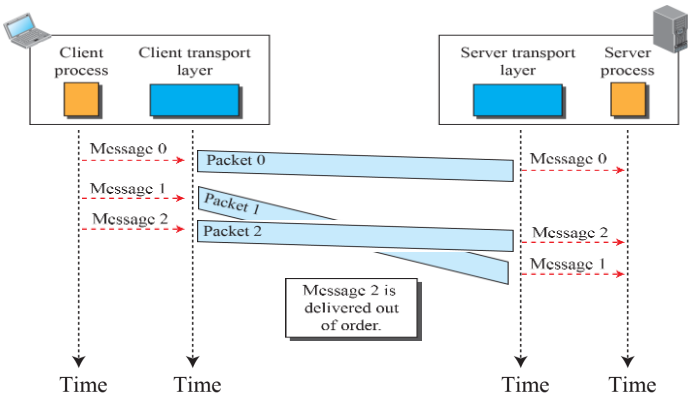


# 6.2 전송 계층의 개념

TCP/IP 교과서

» 연결 지향 및 비연결 프로토콜

- **비연결 프로토콜:** 사전 연결 설정 과정 없이 데이터를 전송.
  수신 측은 데이터를 수신한 후 상태 정보를 송신 측으로 전송하지 않음

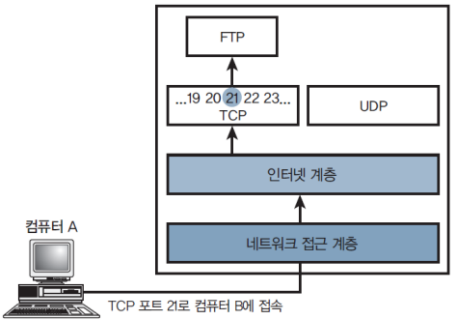



# 6.2 전송 계층의 개념

TCP/IP 교과서

» 포트와 소켓(socket)

- 전송 계층의 애플리케이션에 특화된 주소 지정 체계를 자세히 살펴보면 TCP와 UDP 데이터는 실제로 소켓으로 주소가 지정되어 있음을 알 수 있음
- 소켓은 인터넷 응용이 네트워크 서비스를 이용할 수 있게 하는 창구 역할을 수행하는 API
- 소켓 주소는 전송계층 프로토콜, 소스 IP 주소/포트 번호, 목적지 IP 주소/포트 번호로 구성됨



# 6.3 TCP와 UDP 이해하기


» TCP와 UDP 이해하기

- ❖ **UDP**
  - ➢ **비신뢰적이며 단순한 전송프로토콜**
    - ✓ **오류제어 / 흐름제어 기능이 없음**
- ❖ **TCP**
  - ➢ **신뢰적인 연결지향 전송프로토콜**
    - ✓ **신뢰성이 중요한 응용에서 사용됨**

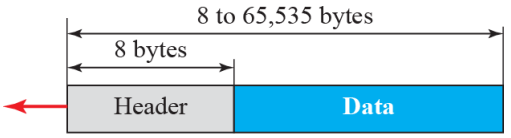
# 6.3 TCP와 UDP 이해하기


» UDP: 비연결 전송 프로토콜

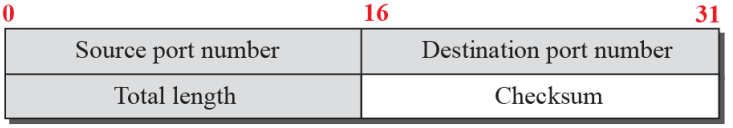
- Connectionless (비연결형)
  - 송신자와 수신자간 no handshaking
  - 각 데이터그램은 다른 것과 독립적으로 처리 - 데이터그램에 번호가 부여되지 않음
- 비신뢰적인 전송 프로토콜
  - 메시지가 손실되거나 중복될 수 있음
  - 내부적으로 흐름/오류제어 메커니즘을 갖는 프로세스에 적합
- 매우 간단한 프로토콜
  - 흐름제어 기능이 없음
  - 혼잡제어 기능이 없음
- 멀티캐스트 지원
- 메시지 지향적 프로토콜

# 6.3 TCP와 UDP 이해하기

» UDP 데이터그램 포맷



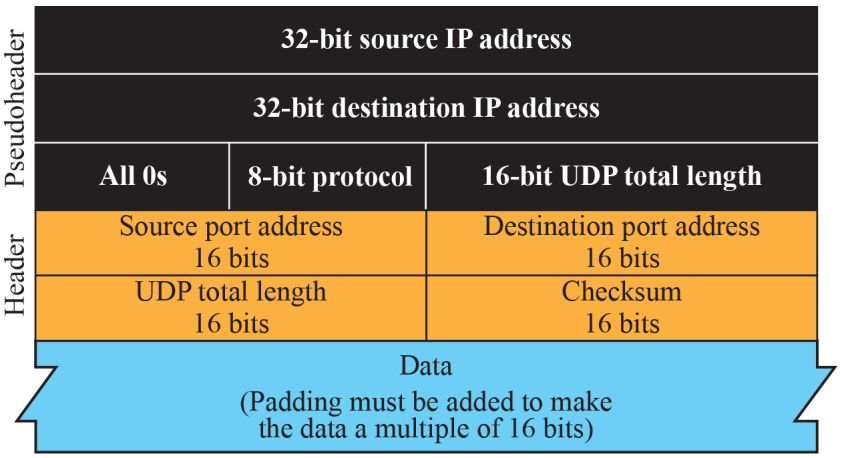

a. UDP user datagram




b. Header format

# 6.3 TCP와 UDP 이해하기

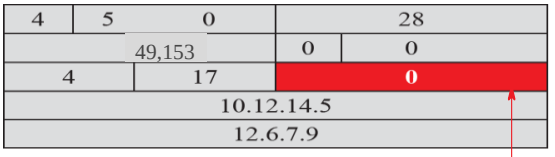
» UDP 체크섬(checksum)



# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

» 체크섬 계산 예




| | | | | | |
|---|---|---|---|---|---|
| 4, 5, and 0 | → | 4 | 5 | 0 | 0 |
| 28 | → | 0 | 0 | 1 | C |
| 49,153 | → | C | 0 | 0 | 1 |
| 0 and 0 | → | 0 | 0 | 0 | 0 |
| 4 and 17 | → | 0 | 4 | 1 | 1 |
| 0 | → | 0 | 0 | 0 | 0 |
| 10.12 | → | 0 | A | 0 | C |
| 14.5 | → | 0 | E | 0 | 5 |
| 12.6 | → | 0 | C | 0 | 6 |
| 7.9 | → | 0 | 7 | 0 | 9 |
| Sum | → | 1 3 | 4 | 4 | E |
| Wrapped sum | → | 3 | 4 | 4 | F |
| **Checksum** | → | **C** | **B** | **B** | **0** |

# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

» UDP: 비연결 전송 프로토콜

**UDP와 브로드캐스트**

- UDP의 간결하고 비연결형 디자인은 네트워크 브로드캐스트 상황에 적합한 프로토콜
- 브로드캐스트는 서브넷에서 모든 컴퓨터가 수신하고 처리하는 단일 메시지
- 만약 소스 컴퓨터가 단일 브로드캐스트를 보내기 위해 서브넷의 모든 컴퓨터에 동시다발적으로 TCP 스타일의 연결을 연다면 네트워크 성능이 현저히 저하될 수 있음

# 6.3 TCP와 UDP 이해하기


» 왜 UDP를 사용하는가?

- 연결 설정이 없다
  - ✓ 연결 설정에 따른 지연 감소
- 구현이 간단하다
  - ✓ 송신 및 수신 측에서 연결 상태를 관리할 필요가 없음
- 헤더의 크기가 작다
  - ✓ 작은 오버헤드
- 오류/혼잡제어 기능을 지원하지 않는다
  - ✓ 원하는 대로 데이터를 빨리 전송할 수 있음
- 신뢰성이 요구되지 않는 작은 메시지 전송에 적합

❁ 인터넷 전화, 멀티미디어 스트리밍 서비스, DNS, SNMP, RIP, TFTP 등의 응용에서 이용

# 6.3 TCP와 UDP 이해하기

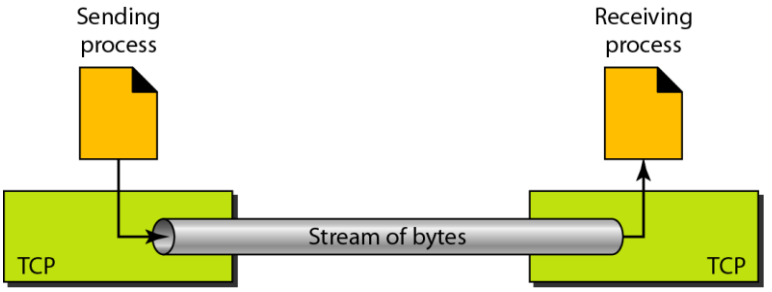

» TCP: 연결 지향 전송 프로토콜

- Connection-oriented (연결지향 )
  - ✓ 데이터 송수신전 송신자와 수신자간 양방향 전이중 통신 가능한 논리적 연결 설정
  - ✓ 데이터 송수신 완료 후 연결 해제
- 신뢰적인 전송 프로토콜
  - ✓ 데이터 전달 및 전달 순서의 보장
  - ✓ ACK와 재전송
- Stream-oriented protocol – 스트림 전달
- Flow control (흐름제어)
  - ✓ 송신 호스트의 데이터 전송 속도 조절
- Congestion Control (혼잡제어)
  - ✓ 네트워크 혼잡 시 데이터 전송속도 조절

# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

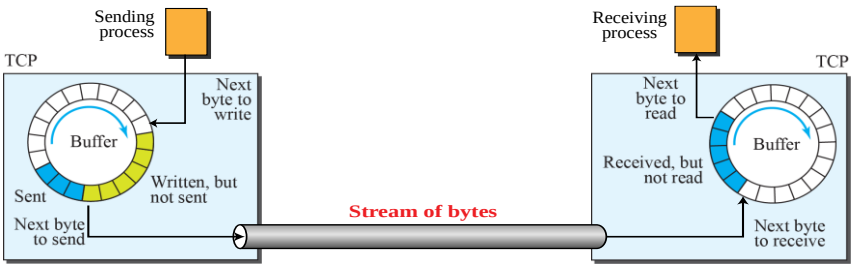
- TCP: 연결 지향 전송 프로토콜
  - 스트림 지향 처리 - 송신 프로세스와 수신 프로세스가 연속된 바이트들의 흐름을 송수신함
    - 데이터 패키징의 제한이 없음
    - 수신자가 데이터를 읽어가는 단위는 송신된 데이터 크기와 독립적




# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

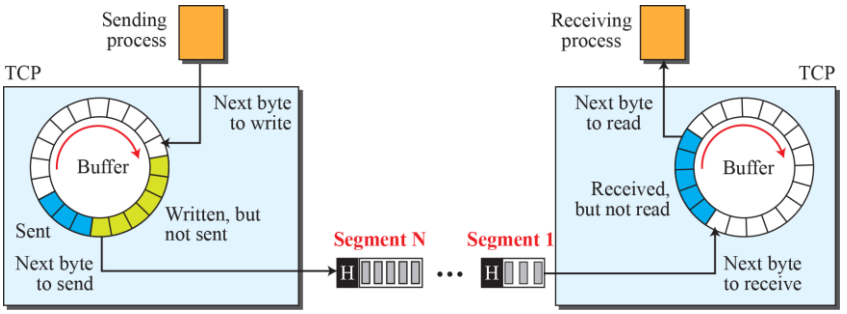
- TCP: 연결 지향 전송 프로토콜
  - 각 프로세스는 2개의 버퍼를 사용
    - 송신 버퍼와 수신 버퍼
    - 흐름제어 및 오류제어에 이용



# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

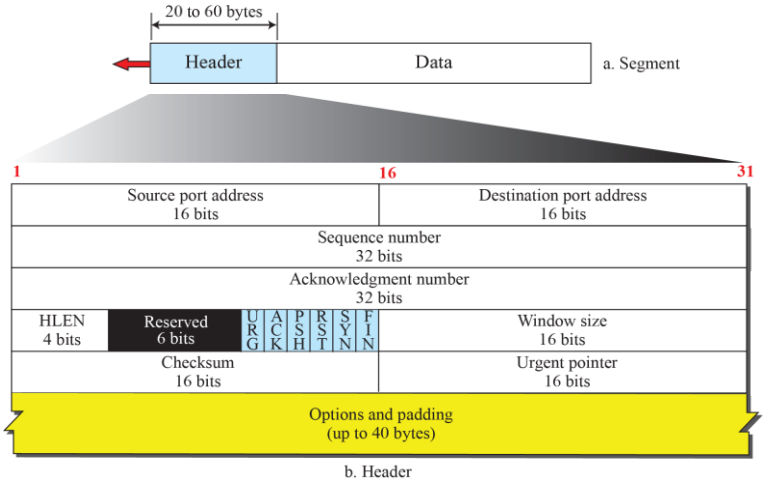
- TCP: 연결 지향 전송 프로토콜
  - 세그먼트(segment) – TCP 패킷에 대한 명칭
  - 세그먼트의 크기는 가변적임




# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

- TCP 세그먼트 포맷



# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

## TCP 세그먼트 포맷

- Sequence Number (순서번호)
  - 해당 세그먼트에 포함된 데이터의 첫 번째 바이트에 할당된 번호
  - 초기 순서번호는 TCP 연결설정 시 0 ~ $2^{32}$-1 사이의 임의의 값으로 할당되며,
  - 이후 전송된 바이트들의 개수에 따라 증가됨
- Acknowledgement Number (응답번호)
  - Receiver가 상대방으로 수신할 것으로 기대하고 있는 바이트 번호
  - 예 - 상대방으로부터 순서번호 x를 가지며, 데이터의 길이가 y인 세그먼트를 정상적으로 수신하였다면, ACK 번호는 x+y 임
- Header Length – 헤더의 길이, 4-byte words의 개수

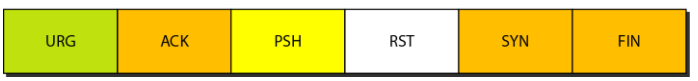
# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

## TCP 세그먼트 포맷

- 제어 필드

URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection



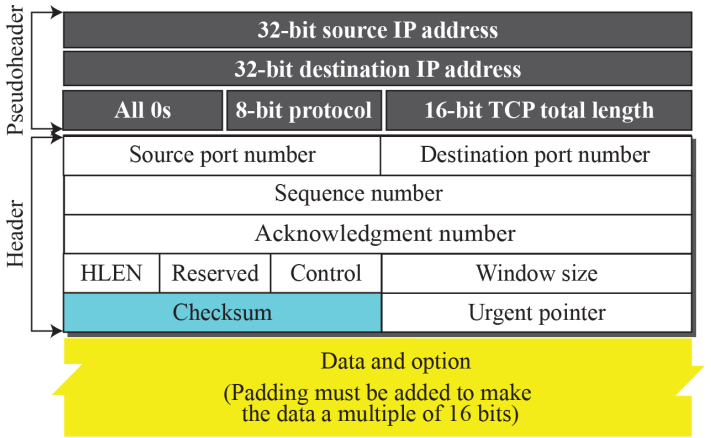

- 윈도우 크기
  - 수신윈도우의 크기 – 수신자의 가용한 버퍼 크기
  - 0 ~ 65,535 bytes
- 체크섬 (검사합) – 반드시 포함되어야 함
- 긴급 포인터 – 긴급데이터의 끝을 알기 위해 순서번호에 더해져야 하는 수
- 선택사항 – 최대 40 bytes

# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

## TCP 세그먼트 포맷

- 검사합




# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

## TCP 연결

- TCP를 이용한 2개의 프로세스간 데이터 통신 과정
  1. 두 TCP 사이에 연결을 설정
  2. 양방향으로 데이터 교환
  3. 연결을 해제
- TCP 연결은 물리적인 연결이 아닌 가상 (virtual) 연결임
  - IP 데이터그램에 캡슐화된 TCP 세그먼트는 실제적으로 서로 다른 경로를 통해 목적지 노드로 전달될 수 있으며,
  - 손실되거나 순서가 틀리거나 손상될 수 있음
  - 목적지 노드의 TCP는 재전송을 통한 손실 및 손상 복구와 세그먼트 순서 정렬을 수행함

# 6.3 TCP와 UDP 이해하기


## TCP 연결

- TCP는 다음 두 가지 연결 상태를 지원
  - **수동 개방(passive open):** 서버 측에서 클라이언트로부터의 연결 요청을 받아 처리할 수 있는 상태

    주어진 애플리케이션 프로세스는 TCP에 TCP 포트를 통해 들어오는 연결 요청을 받을 준비가 되었음을 알림
  - **능동 개방(active open):** 클라이언트에서 서버로 TCP 연결을 요청할 수 있는 상태

# 6.3 TCP와 UDP 이해하기


## 연결 설정

- 시퀀스/확인 시스템이 제대로 작동하려면 컴퓨터가 반드시 자신의 시퀀스 번호를 동기화해야 함
- 컴퓨터 B는 컴퓨터 A가 시퀀스를 시작할 때 사용하는 ISN을 알아야 함
- 컴퓨터 A는 컴퓨터 B가 전송할 모든 데이터에 대한 시퀀스를 시작하기 위해 사용할 ISN을 알아야 함
- 이러한 시퀀스 번호의 동기화를 **3방향 핸드셰이크(3-way handshake)**라고 함

# 6.3 TCP와 UDP 이해하기


## 연결 설정

- 3방향 핸드셰이크의 세 단계는 다음과 같음

**1.** 컴퓨터 A가 세그먼트를 다음과 같이 보냄

SYN = 1

ACK = 0

시퀀스 번호 = X(X는 컴퓨터 A의 ISN)

수동 연결 컴퓨터(컴퓨터 A)는 SYN 플래그가 1, ACK 플래그가 1로 설정된 세그먼트를 보냄

SYN은 동기화를 뜻함

해당 플래그는 연결하기 위한 시도를 나타냄

또한, 첫 세그먼트 헤더는 컴퓨터 A가 전송하는 데이터의 시퀀스 번호의 시작을 표시하는

초기 시퀀스 번호(ISN)를 가짐

컴퓨터 B로 전송된 첫 바이트는 시퀀스 번호 ISN + 1을 가지게 됨

# 6.3 TCP와 UDP 이해하기


## 연결 설정

**2.** 컴퓨터 B는 컴퓨터 A의 세그먼트를 받고 다음과 함께 세그먼트를 반환

SYN = 1 (여전히 동기화 단계)

ACK = 1 (확인 번호 필드는 하나의 값을 가지게 됨)

시퀀스 번호 = Y(Y는 컴퓨터 B의 ISN)

확인 번호 = M + 1(M은 컴퓨터 A로부터 받은 마지막 시퀀스 번호)

**3.** 컴퓨터 A는 ISN의 수신을 확인한 컴퓨터 B에 세그먼트를 보냄

SYN = 0

ACK = 1

시퀀스 번호 = 일련 번호의 다음 시퀀스 번호(M + 1)

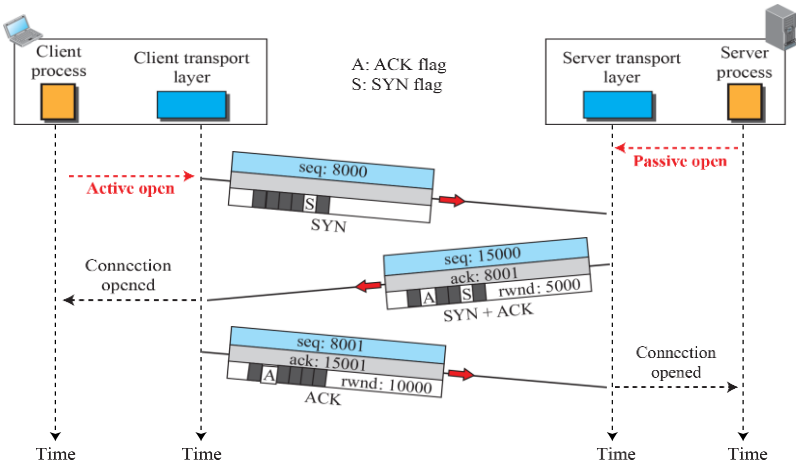
승인 번호 = N + 1(N은 컴퓨터 B에서 받은 마지막 시퀀스 번호)

- 3방향 핸드셰이크 이후 연결이 열리고, TCP 모듈은 시퀀스와 승인 체계를 사용해 데이터를 송수신

# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

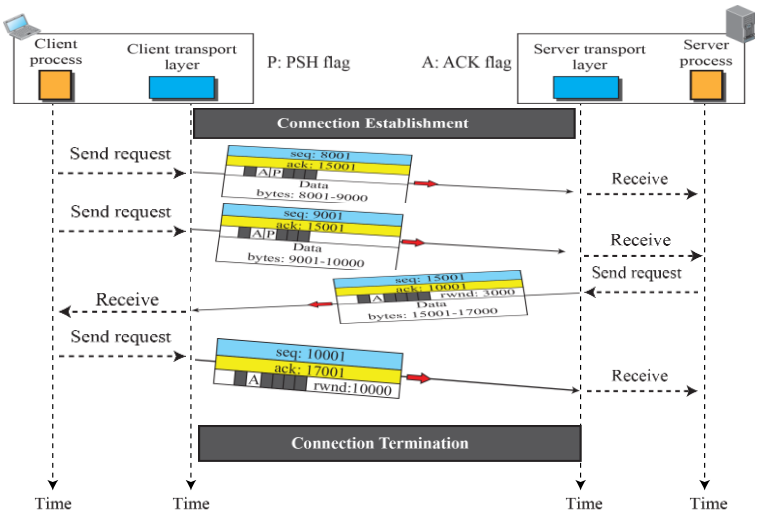
» TCP 연결 설정 과정




# 6.3 TCP와 UDP 이해하기

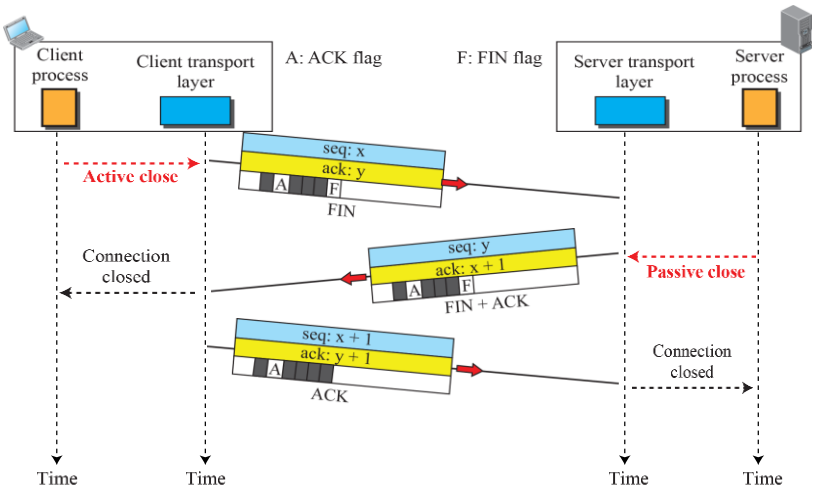
TCP/IP 교과서

» TCP 데이터 전달 과정



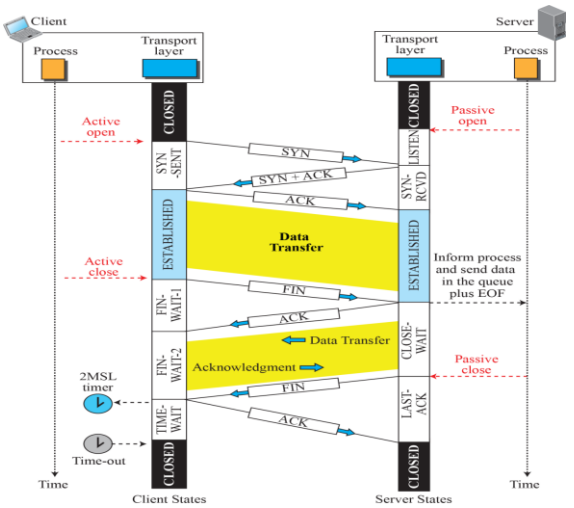
# 6.3 TCP와 UDP 이해하기

» 연결 종료 과정




# 6.3 TCP와 UDP 이해하기

» TCP의 일반적 신호 흐름도



# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

## TCP 흐름 제어

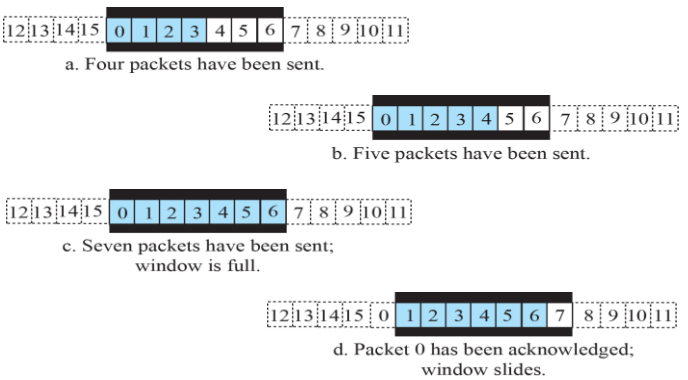
- TCP 헤더의 윈도우 필드는 연결을 위한 흐름 제어 메커니즘을 제공
- 윈도우 필드의 목적은 송신 컴퓨터가 너무 많은 데이터를 너무 빨리 보내지 않도록 하는 것
- 이는 수신 컴퓨터가 수신 중인 세그먼트를 송신할 수 있을 만큼 빨리 처리할 수 없기 때문에 데이터가 손실되는 상황이 생길 수 있음
- TCP가 사용하는 흐름 제어 방식을 **슬라이딩 윈도우**(sliding window) 방식이 라고 함
- 수신 컴퓨터는 윈도우 필드를 사용해서 송신 컴퓨터가 전달할 수 있는 마지막으로 확인된 시퀀스 번호를 초과하는 시퀀스 번호의 윈도우를 정의
- 송신 컴퓨터는 다음 확인 응답을 받을 때까지 해당 윈도우를 초과해 송신할 수 없음

# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

## 슬라이딩 윈도우(sliding window)

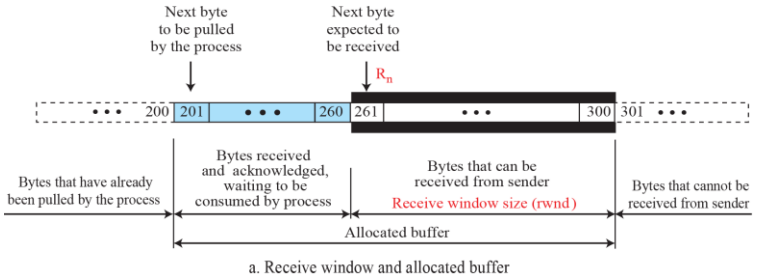
- 송신 윈도우 - 응답확인 수신 전 송신할 수 있는 순서번호의 범위
- 수신 윈도우 – 수신 버퍼에 저장할 수 있는 순서번호의 범위
- 송신 윈도우 크기가 7인 경우의 예



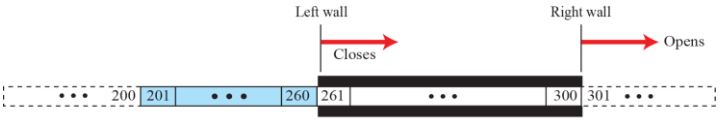
# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

» TCP 윈도우 – 수신 윈도우




a. Receive window and allocated buffer

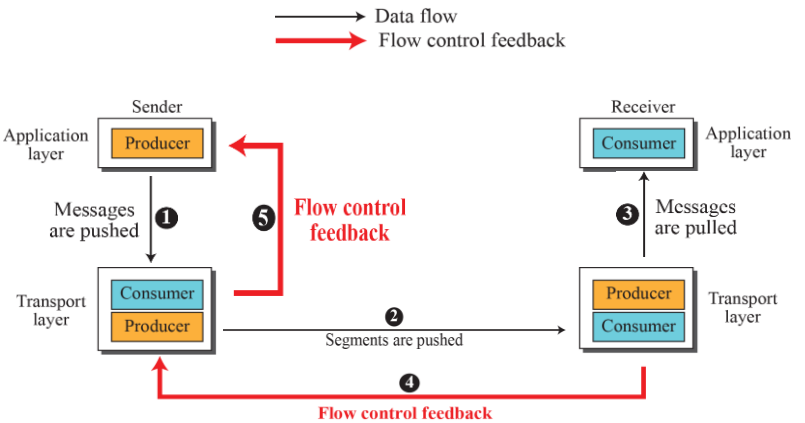



b. Opening and closing of receive window

# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

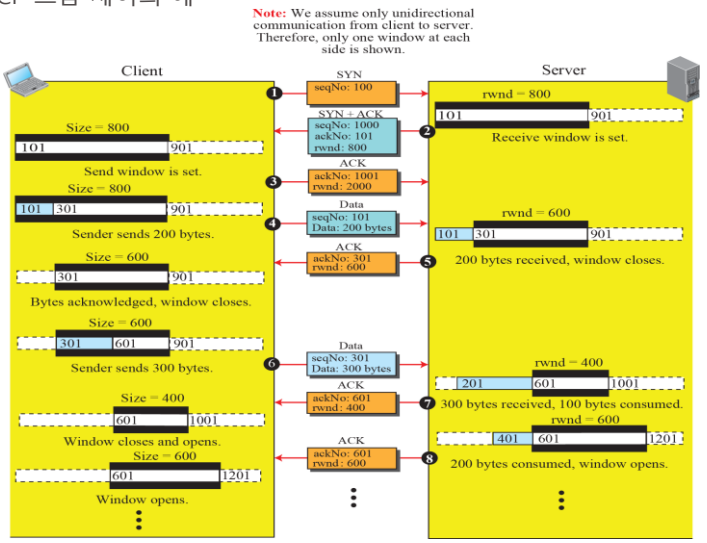
» TCP 흐름 제어



# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

» TCP 흐름 제어의 예




# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

» TCP 오류 제어

- ❒ 검사합
  - ❖ 의무사항
  - ❖ 손상된 세그먼트는 폐기하고 손실로 간주
- ❒ 확인응답(ACK: Acknowledgement)
  - ❖ 긍정 누적 확인응답
  - ❖ 확인응답의 생성
    1. 피기배킹(piggybacking)
    2. 지연된 ACK 전송 (500ms)
    3. 순서에 맞는 세그먼트 2개 쌓일 때 ACK 전송
    4. 순서에 맞지 않은 세그먼트 수신 시 ACK(수신하고자 하는 다음 순서번호 ) 즉시 전송
    5. 누락된 세그먼트 수신 시 ACK(수신하고자 하는 다음 순서번호 ) 즉시 전송
    6. 중복 세그먼트 수신 시 폐기하고 수신하고자 하는 다음 순서번호를 알리기 위한 ACK 전송

# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

» TCP 오류 제어

❒ 재전송

❖ 재전송 타임아웃(RTO) 시 재전송

- 송신 TCP는 각각의 연결을 위해 하나의 재전송 타이머 구동
- 타임아웃 시 순서번호가 가장 작은 세그먼트를 재전송하고 타이머 재구동

❖ 빠른 재전송

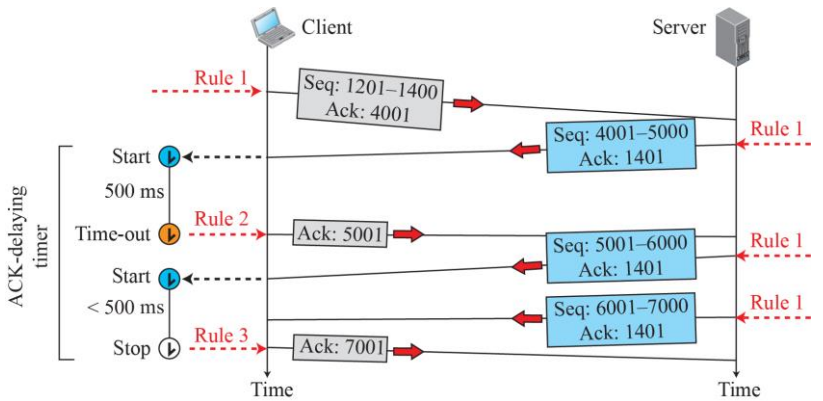
- 3개의 중복 ACK 수신 시 RTO 전에 즉시 재전송

# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

» TCP 오류 제어

- 정상 동작

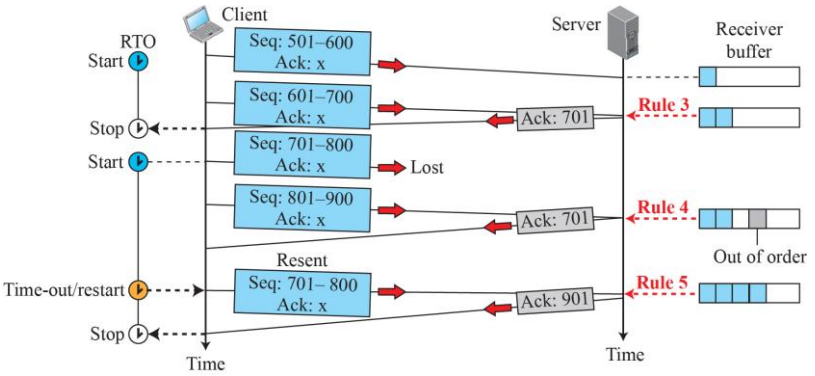


# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

» TCP 오류 제어

- 세그먼트 손실 시

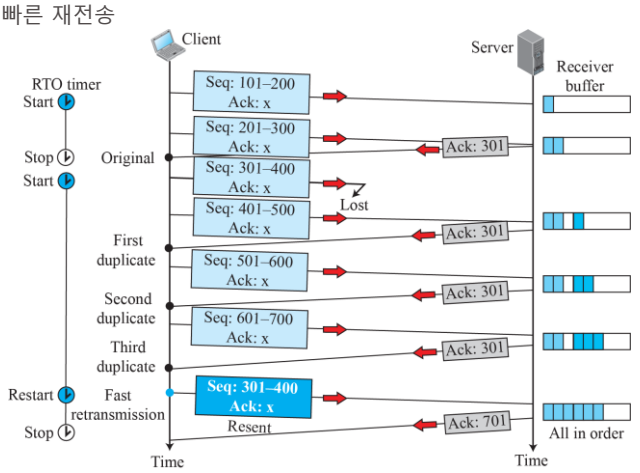



6.3 TCP와 UDP 이해하기

TCP/IP 교과서

» TCP 오류 제어

- 빠른 재전송

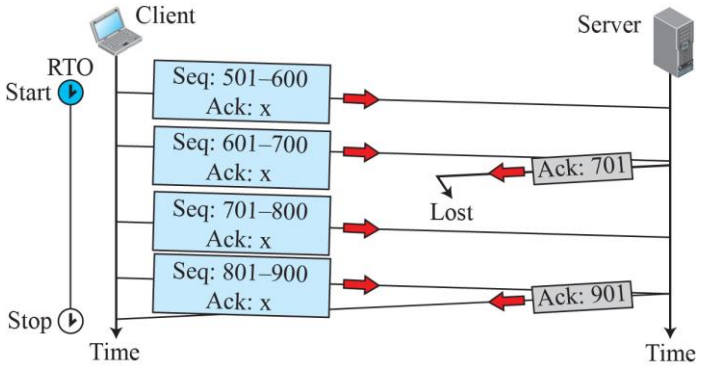


# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

» TCP 오류제어

- ACK 손실I

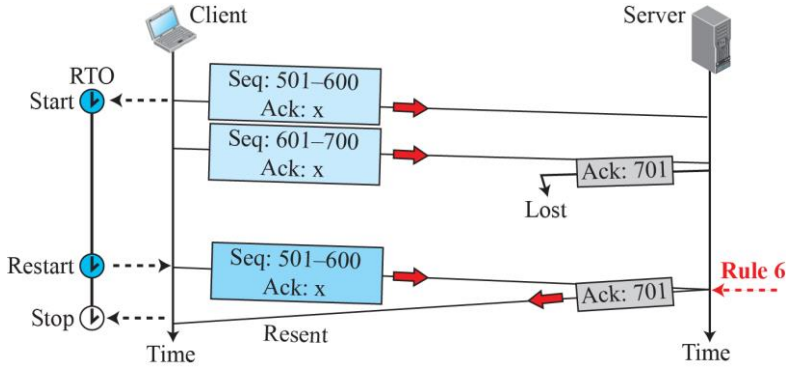



# 6.3 TCP와 UDP 이해하기

TCP/IP 교과서

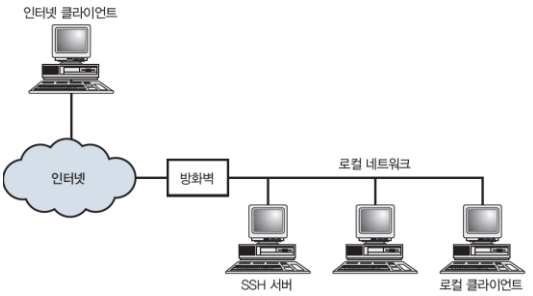
» TCP 오류 제어

- ACK 손실II



## 6.4 방화벽과 포트

- **방화벽**(firewall)은 인터넷에서 권한이 없는 사용자가 LAN에 접근하려는 공격으로부터 로컬 네트워크를 보호하는 시스템
- 특정 TCP와 UDP 포트의 접근을 차단하는 방화벽의 기능이 바로 중요한 특징
- 예를 들어, 서버에 보안 셸(SSH, Secure Shell) 세션을 시작하려면 클라이언트 기기가 SSH의 잘 알려진 포트 주소인 TCP 포트 22로 요청을 보내야 함
- 그림과 같이 방화벽을 설치하고 TCP 포트 22의 접근을 못하도록 설정하는 것




## 6.5 요약

- 이 장에서는 TCP/IP 전송 계층의 주요 기능들을 알아봤고 연결 지향과 비연결 프로토콜, 다중화와 비다중화, 그리고 포트와 소켓을 살펴봤음
- TCP/IP의 전송 계층 프로토콜인 TCP와 UDP를 소개하면서 중요한 TCP 및 UDP의 기능들을 알아봤음
- TCP 데이터 형식, 흐름 제어, 그리고 오류 복구에 대해 배웠고, 3방향 핸드셰이크는 TCP 연결에서 사용됨
- 마지막으로 UDP 헤더의 형식도 배웠음
- 방화벽은 네트워크 내에서 외부 사용자가 서비스에 접근하는 것을 막으면서도, 내부 사용자가 네트워크 외부의 서비스에 접근하는 것도 막을 수 있음

# 6.8 핵심 용어


- **ACK:** TCP 헤더의 응답 확인 번호 필드가 중요하다는 것을 알리는 제어 플래그
- **확인 응답 필드:** 컴퓨터가 받을 다음 시퀀스 번호를 알려주는 TCP 헤더의 필드
- **확인 응답 번호:** 확인 응답 번호에 지정된 바이트 이전의 모든 시퀀스 바이트 수신을 확인
- **능동 개방:** TCP가 연결을 시도하려는 상태
- **연결 지향 프로토콜:** 통신 컴퓨터 간 연결을 설정을 통해 통신을 관리하는 프로토콜
- **비연결 프로토콜:** 원격 컴퓨터와 연결을 설정하지 않고 데이터를 전송하는 프로토콜
- **제어 플래그:** TCP 세그먼트에 대한 특별한 정보를 가진 1비트 플래그
- **FIN:** TCP 연결 종료하는 과정에서 사용되는 제어 플래그
- **ISN(초기 시퀀스 번호):** 컴퓨터가 TCP를 통해 전송된 바이트를 시퀀싱하기 위해 사용할 숫자 범위의 시작을 나타내는 숫자
- **수동 개방:** TCP 포트(보통 서버 애플리케이션)에 들어오는 연결을 수신할 준비가 된 상태
- **포트번호:** 애플리케이션에서 전송 계층 프로토콜에 대한 인터페이스를 제공하는 내부 주소

# 6.8 핵심 용어


- **순서 번호:** TCP를 통해 전송된 바이트와 관련된 고유 번호
- **슬라이딩 윈도우:** 수신 컴퓨터가 송신 컴퓨터가 송신할 수 있도록 권한 부여한 순서 번호창. 슬라이딩 윈도우 흐름 제어 방법은 TCP에서 사용되는 방법
- **소켓:** 특정 컴퓨터의 특정 애플리케이션을 위한 네트워크 주소. 애플리케이션의 포트 번호 뒤에 컴퓨터의 IP 주소로 구성되어 있음
- **스트림 지향 처리:** 이미 정의된 데이터 블록의 입력이 아닌 연속(바이트 단위) 입력
- **SYN:** 시퀀스 번호 동기화 중임을 알리는 제어 플래그.S YN 플래그는 3방향 핸드셰이크의 단계의 일부로 TCP 연결의 시작점에서 사용
- **3방향 핸드셰이크:** 시퀀스 번호를 동기화하고 TCP 연결을 시작하는 3단계 절차
- **잘 알려진 포트:** 일반적인 애플리케이션을 위해 사전 정의된 표준 포트 번호