2024/09/11, 2024 네트워크 세미나

TCP/IP 프로토콜

-15장 TCP-

이 정 민(jeongmin@pel.sejong.ac.kr)

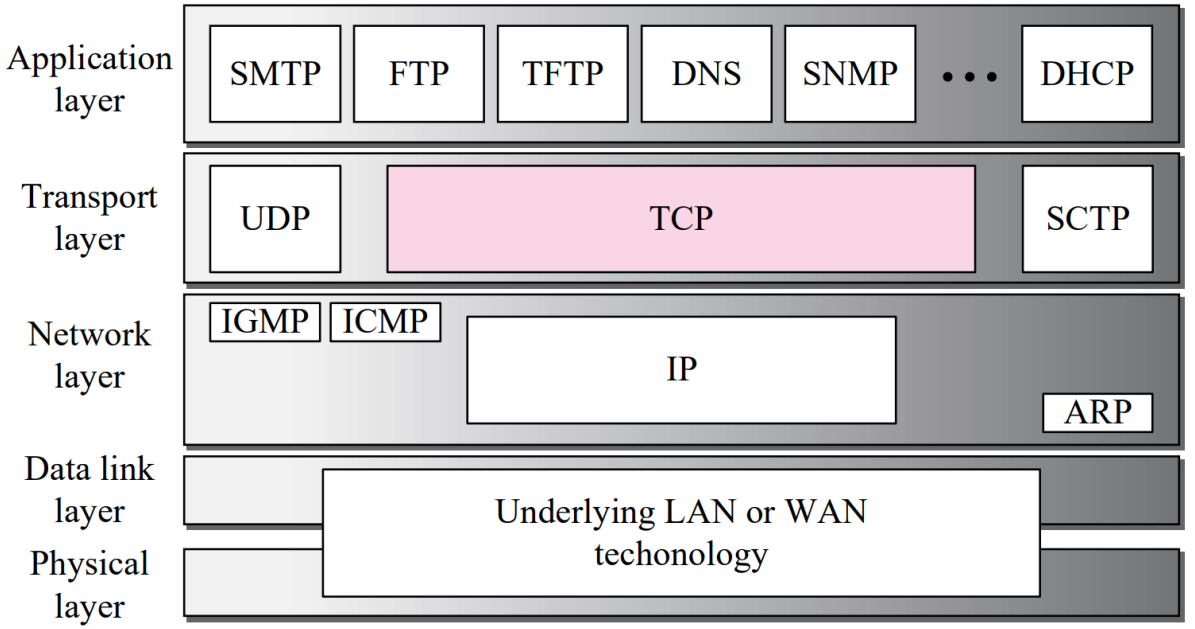
세종대학교 프로토콜공학연구실

목 차

목 차

TCP 서비스와 특징 (1/9)

- TCP (Transmission Control Protocol)
  - 전송 계층 (Transport Layer)에서 데이터가 정확하고 순서대로 전달되도록 보장하는 프로토콜
  - 응용 프로그램과 네트워크 동작 사이의 매개체로서 사용됨



TCP 서비스와 특징 (2/9)

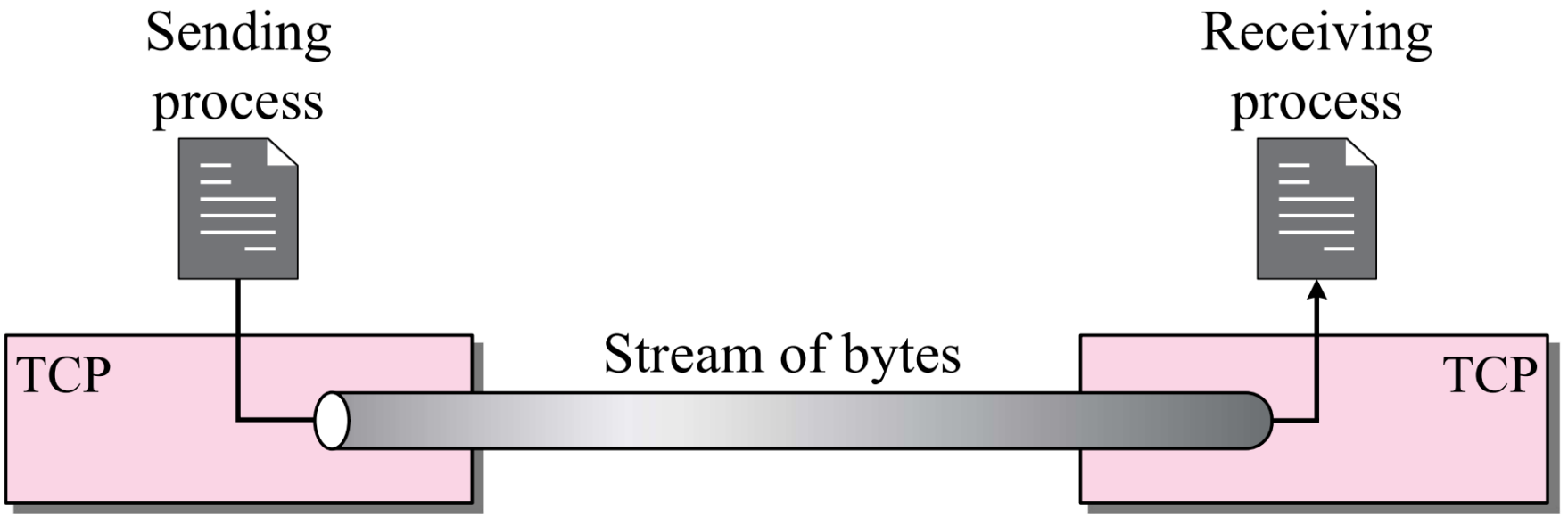
- TCP 서비스 (1/5)
  - 프로세스 대 프로세스 통신
    - 포트 번호를 이용하여 프로세스 대 프로세스 통신을 제공함
    - UDP에서도 제공하는 서비스임

Port	Protocol	Description
7	Echo	수신한 데이터그램을 송신자에게 되돌려보냄
9	Discard	수신한 데이터그램을 폐기함
11	Users	활성 사용자들을 보여줌
13	Daytime	현재 날씨와 시간을 반환함
17	Quote	오늘의 명언을 반환함
19	Chargen	문자열을 반환함
20, 21	FTP	파일 전송 프로토콜임

Port	Protocol	Description
22	SSH	암호화를 통한 터미널 네트워크 연결을 제공함
23	Telnet	터미널 네트워크 연결을 제공함
25	SMTP	이메일을 전송함
53	DNS	도메인 이름을 IP 주소로 변경함
79	Finger	사용자 정보를 조회함
80	HTTP	웹 페이지를 전송함
443	HTTPS	웹 페이지를 암호화하여 전송함

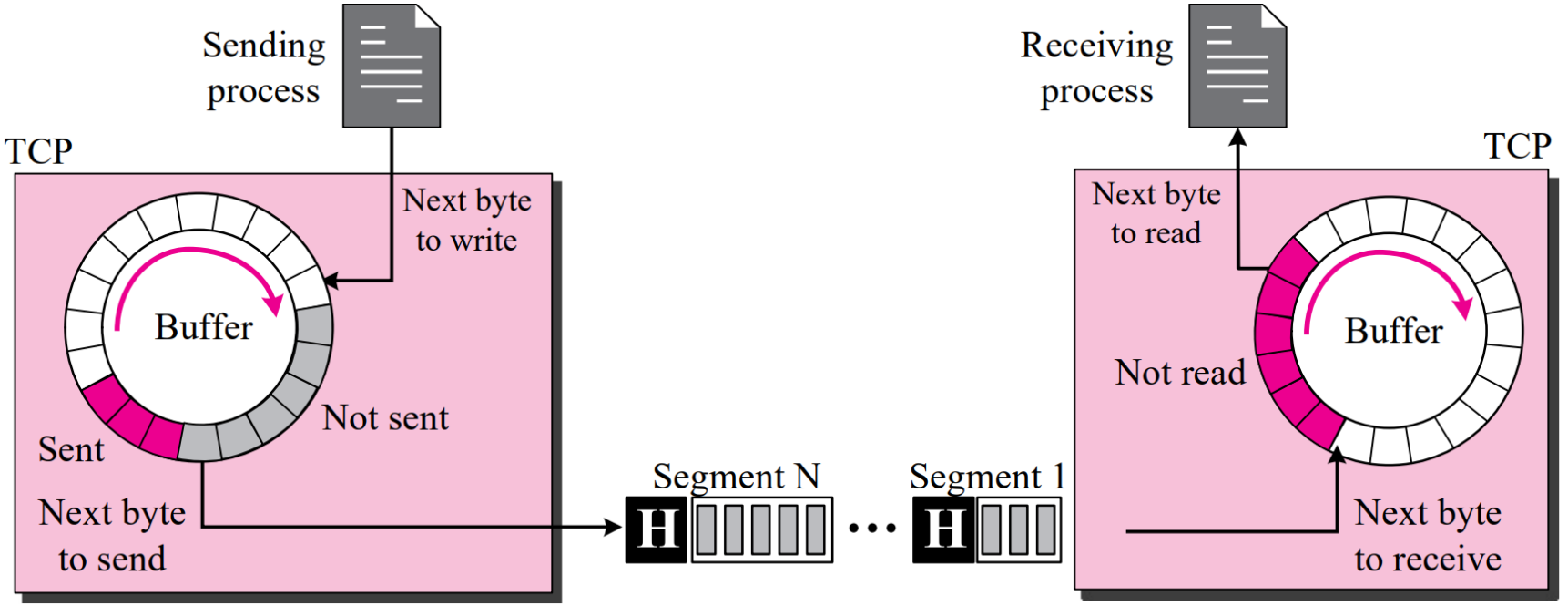
TCP 서비스와 특징 (3/9)

- TCP 서비스 (2/5)
  - 스트림 전송 서비스 (1/2)
    - 데이터를 바이트 단위의 스트림 형태로 송수신하는 기능임
    - TCP는 스트림 지향 프로토콜로 분류됨
    - 송수신 속도가 동일하지 않을 수 있으므로, 버퍼가 요구됨



TCP 서비스와 특징 (4/9)

- TCP 서비스 (3/5)
  - 스트림 전송 서비스 (2/2)
    - 송신 버퍼와 수신 버퍼
      - TCP 소켓 각각에 존재함
        - 소켓 생성 시 자동으로 생성됨
      - 일반적으로, 링 버퍼의 형태로 구현됨

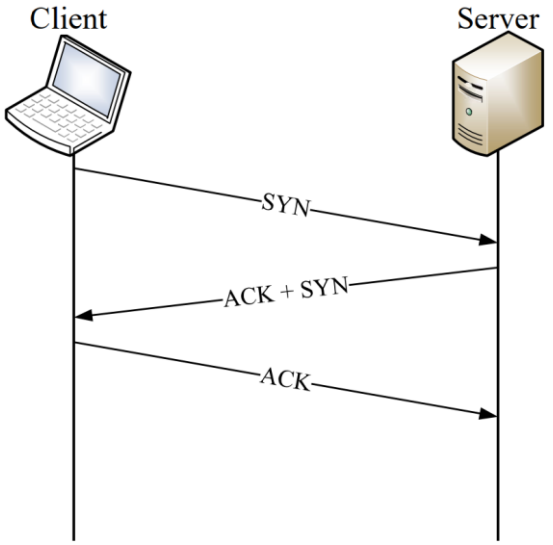


TCP 서비스와 특징 (5/9)

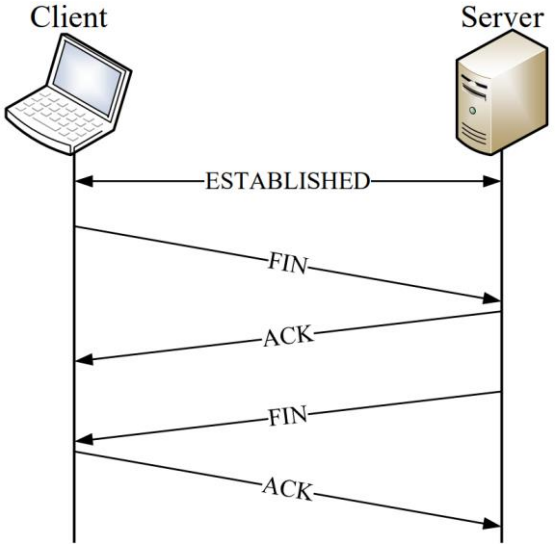
- TCP 서비스 (4/5)
  - 전 이중 (Full-duplex) 통신
    - 데이터를 동시에 양방향으로 전송 가능함
  - 다중화와 역다중화
    - 송신자는 패킷을 보내는 프로세스의 포트 번호를 이용하여 패킷을 다중화함 (TCP/IP L5 → L4)
    - 수신자는 패킷을 수신하는 프로세스의 포트 번호를 이용하여 역다중화함 (TCP/IP L4 → L5)

TCP 서비스와 특징 (6/9)

- TCP 서비스 (5/5)
  - 연결 지향 서비스
    - TCP는 연결 지향 프로토콜로 분류됨
    - 가상 연결을 수립하여 송수신자 간 통신 준비가 되었는지 확인함
    - 데이터 전송이 끝나면 연결을 명시적으로 종료함
  - 신뢰성 서비스
    - 데이터가 안전하고 오류 없이 잘 도착하였는지를 확인하기 위해 확인응답 메커니즘을 이용함



a. TCP 3-way-handshake



b. TCP 4-way-handshake

TCP 서비스와 특징 (7/9)

- TCP 특징 (1/3)
  - 번호화 시스템 (1/2)
    - 바이트 번호 (Byte Number)
      - 한 연결 내에서 전송되는 모든 데이터 바이트에 매겨지는 번호임
      - 각 방향에서 서로 독립적으로 매겨짐
      - 0에서 $2^{32}-1$사이의 임의의 값을 선택하여 초기 바이트 번호를 설정함
    - 순서 번호 (Sequence Number)
      - 전송하고자 하는 세그먼트에 할당되는 번호임
      - 각 세그먼트의 순서 번호는 세그먼트 첫 번째 바이트의 바이트 번호임
    - 확인응답 번호 (Acknowledgement Number)
      - 바이트를 수신하였음을 확인하기 위해 사용되는 번호임
      - 수신하기를 기대하는 다음 바이트의 번호를 나타냄
        - e.g., 5642의 순서 번호를 수신하면, 다음 바이트인 5643의 확인응답 번호를 전송함

TCP 서비스와 특징 (8/9)

- TCP 특징 (2/3)
  - 번호화 시스템 (2/2)
    - 예제 15.1

TCP 연결이 5000 바이트의 파일을 전송할 때, 첫 번째 바이트는 10001의 번호를 가지고 있다. 각각이 1000 바이트를 가지는 5개 세그먼트에 의해 데이터가 전달된다면, 각 세그먼트의 순서 번호는 어떻게 되는가?

- 풀이
  - 세그먼트 1: 순서 번호 10001을 가짐 (10001 ~ 11000)
  - 세그먼트 2: 순서 번호 11001을 가짐 (11001 ~ 12000)
  - 세그먼트 3: 순서 번호 12001을 가짐 (12001 ~ 13000)
  - 세그먼트 4: 순서 번호 13001을 가짐 (13001 ~ 14000)
  - 세그먼트 5: 순서 번호 14001을 가짐 (14001 ~ 15000)

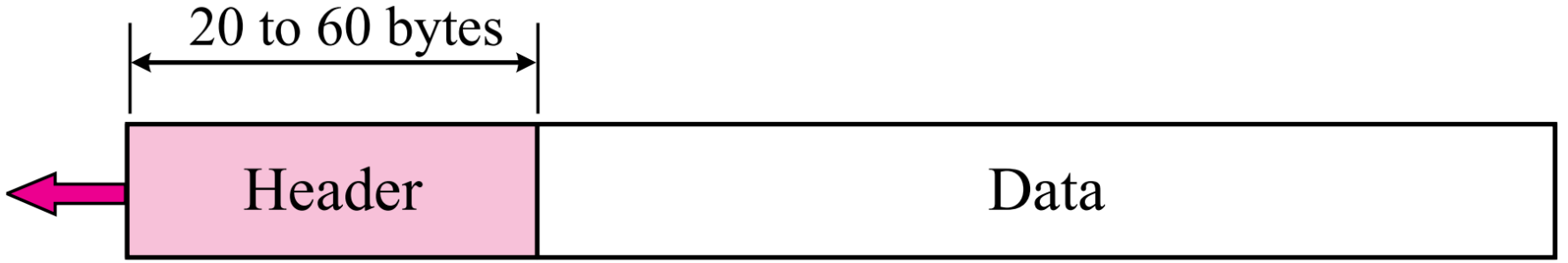
TCP 서비스와 특징 (9/9)

- TCP 특징 (3/3)
  - 흐름 제어
    - 송신 TCP는 송신 프로세스로부터 수신되는 데이터 양을 조절함
    - 수신 TCP는 송신 TCP로부터 전송되는 데이터의 양을 조절함
    - 바이트 단위의 흐름 제어를 위하여 번호화 시스템을 이용함
  - 오류 제어
    - 신뢰성 있는 서비스를 제공하기 위해 오류제어 메커니즘을 구현함
    - 오류 감지는 세그먼트 단위로 동작하나, 오류 제어는 바이트 단위로 동작함
  - 혼잡 제어
    - 송신 측에서 전송되는 데이터 양은 수신 측에 의해 조절될 뿐만 아니라 (흐름 제어), 망의 혼잡 정도에 의해서도 결정됨

목 차

TCP 세그먼트 (1/5)

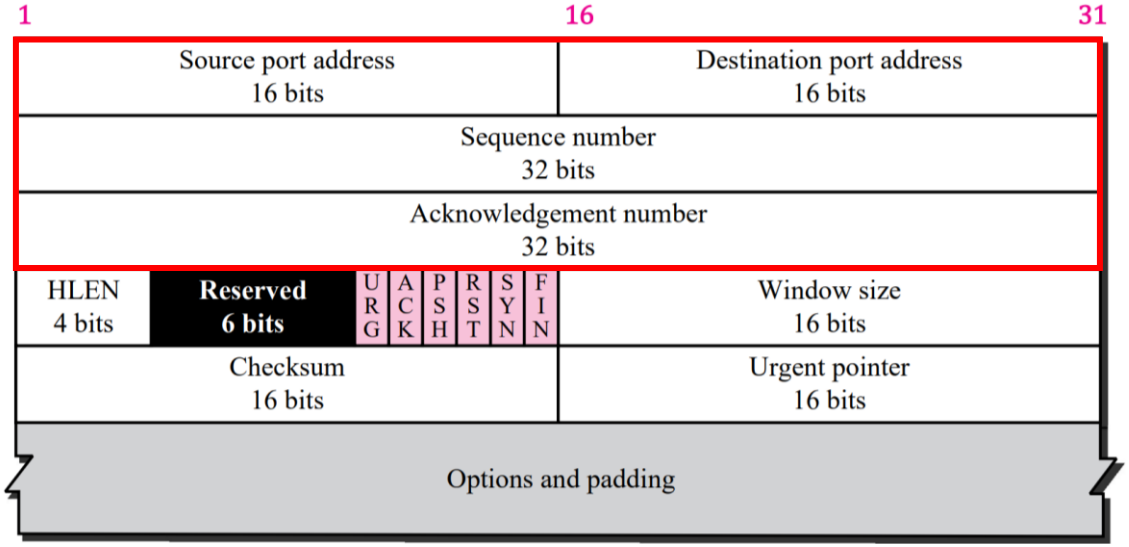
- 정의
  - TCP 세션으로 연결된 종단 간에 교환, 전송되는 데이터 단위
- 특징
  - 헤더와 데이터로 구성됨
    - 20~60 바이트의 헤더를 가짐
    - 데이터는 응용 프로그램으로부터 생성됨
  - MSS (Maximum Segment Size) 값을 가짐
    - 한 개의 TCP 세그먼트로 보낼 수 있는 최대 데이터 크기를 정의함 (TCP 헤더 길이는 제외됨)



TCP 세그먼트 (2/5)

- 헤더 형식 (1/3)

필드명	크기	설명
발신지 포트 주소 (Source port address)	16	• 발신지 응용 프로그램의 포트 번호를 정의함
목적지 포트 주소 (Destination port address)	16	• 수신지 응용 프로그램의 포트 번호를 지정함
순서 번호 (Sequence number)	32	• 세그먼트의 첫 번째 데이터 바이트에 부여된 번호를 가짐 • 연결 설정 단계 동안 난수 발생기를 통해 초기 순서 번호 (ISN, Initial Sequence Number)을 만듦
확인응답 번호 (Acknowledgement number)	32	• 수신 노드가 상대편 노드로부터 수신하고자 하는 바이트 번호를 정의함 • 확인응답과 데이터는 함께 피기백 (Piggyback)* 될 수 있음

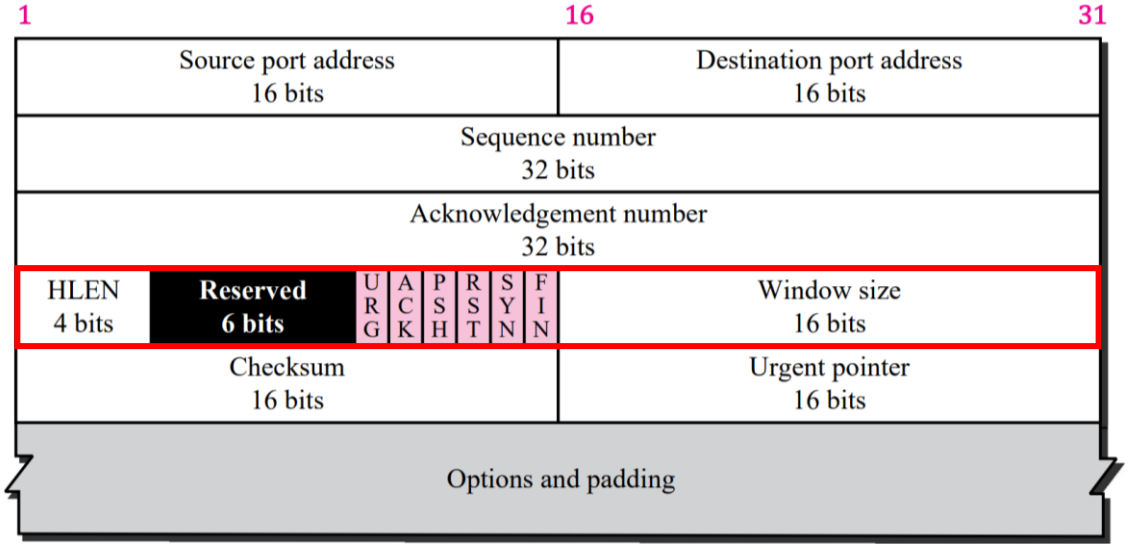


*피기백: 한 작업에 다른 작업을 동시에 실어 함께 처리하는 기법

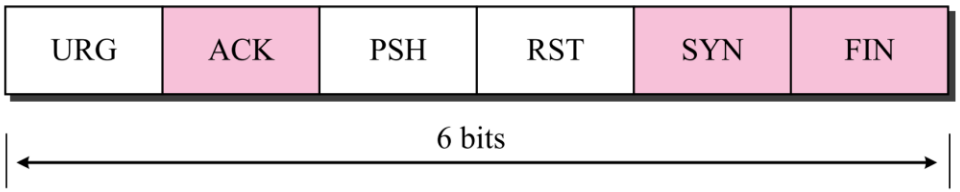
TCP 세그먼트 (3/5)

- 헤더 형식 (2/3)

필드명	크기	설명
헤더 길이 (HLEN)	5	• TCP 헤더 길이를 4바이트 단위로 나타냄
예약 (Reserved)	6	• 차후 사용을 위해서 예약되어 있음 • 0 값으로 채워짐
제어	6	• 6개의 서로 다른 제어 비트를 가짐 • 동시에 여러 비트가 1로 설정될 수 있음
윈도우 크기 (Window size)	16	• TCP 흐름제어를 위해 수신 버퍼 여유 용량 크기를 통보하는데에 사용됨 • 수신 윈도우 (rwnd, receiving window)로도 불림



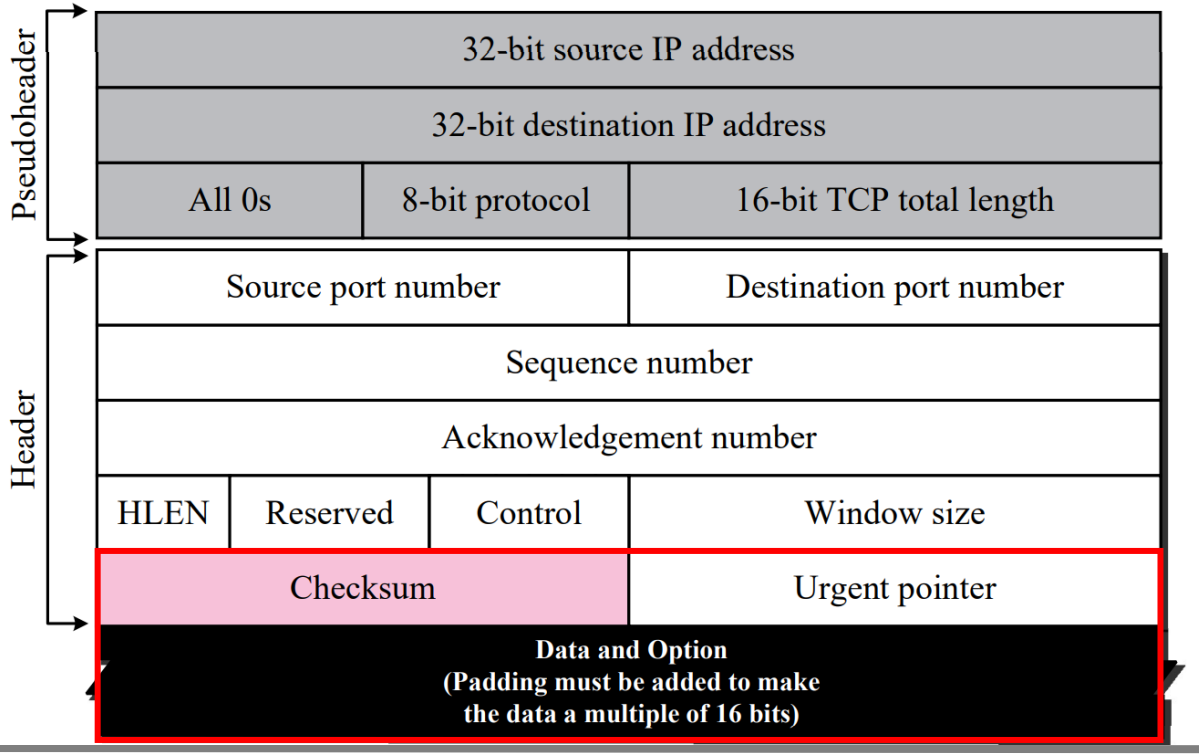
URG: Urgent Pointer is valid
ACK: Acknowledgement is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection



TCP 세그먼트 (4/5)

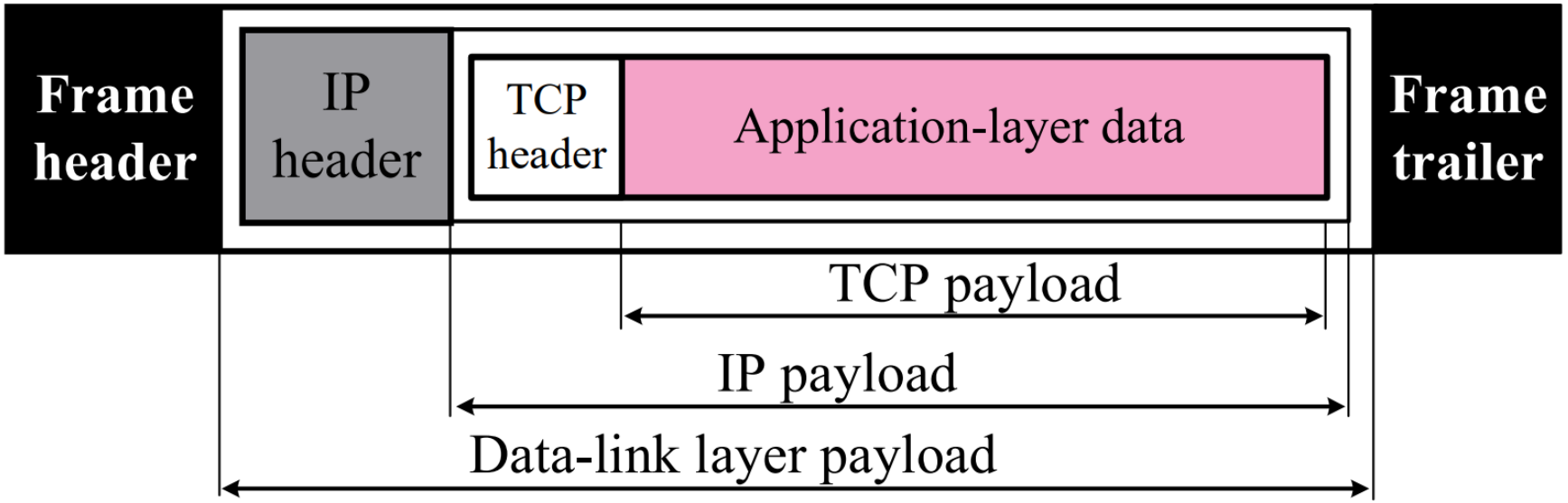
- 헤더 형식 (3/3)

필드명	크기	설명
검사합 (Checksum)	16	• 필수적으로 검사합을 포함함 (UDP는 선택 사항임) • 검사합 계산을 위해 의사 헤더가 추가됨
긴급 포인터 (Urgent pointer)	16	• 긴급 플래그 (URG)가 1로 설정되어있을 경우에 유효함
옵션 (Opiton)	Variable	• 최대 40 바이트의 옵션 정보를 가짐



TCP 세그먼트 (5/5)

- 캡슐화
  - 응용 계층으로부터 전달받은 데이터를 캡슐화 함
  - TCP 세그먼트는 IP 데이터그램에 캡슐화 됨
  - IP 데이이터그램은 데이터링크 계층 프레임에 캡슐화 됨



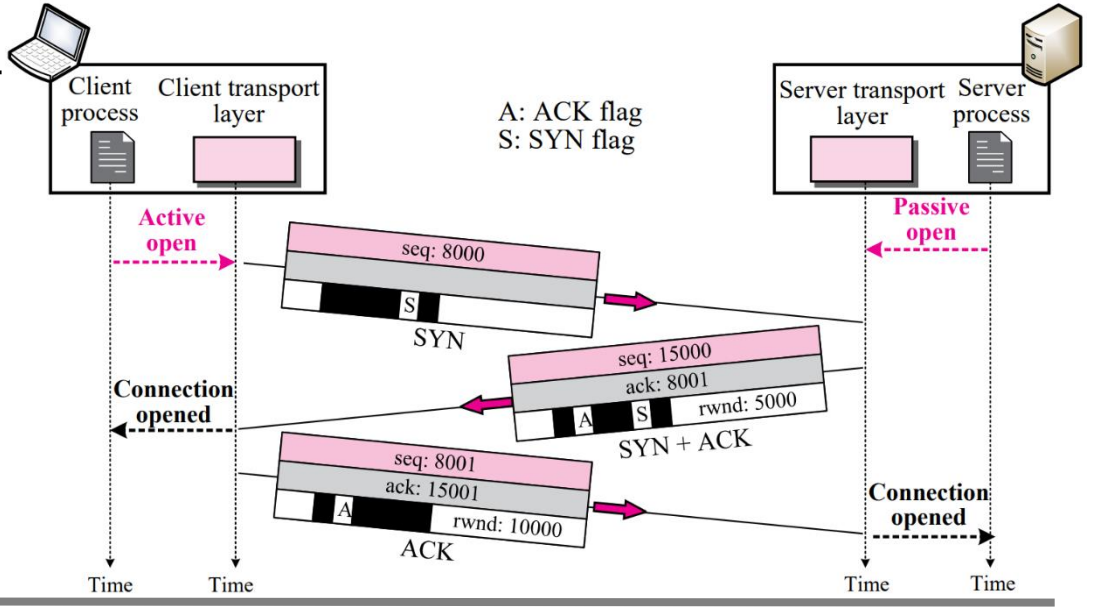
목 차

TCP 연결 (1/26)

- 연결 설정 (1/6)
  - 정의
    - 데이터 교환이 이루어지기 전에, 통신을 개시하고 통신 개시 요구에 대한 승인이 이루어지는 단계
  - 3단계 핸드셰이크 (Three-way handshaking)
    - 수동 개방 (Passive open)과 능동 개방 (Active open)
      - 서버 프로그램은 자신의 TCP에 연결을 수락할 준비가 되었음을 알림
        - 수동 개방이라고 함
        - 다른 시스템으로부터 연결을 수락할 수 있지만, 먼저 연결을 개설할 수는 없음
      - 클라이언트 프로그램은 능동 개방을 위한 요청을 수행함
        - 자신의 TCP에게 특정 서버와 연결할 것임을 알림
        - 이후, 3단계 절차가 수행됨

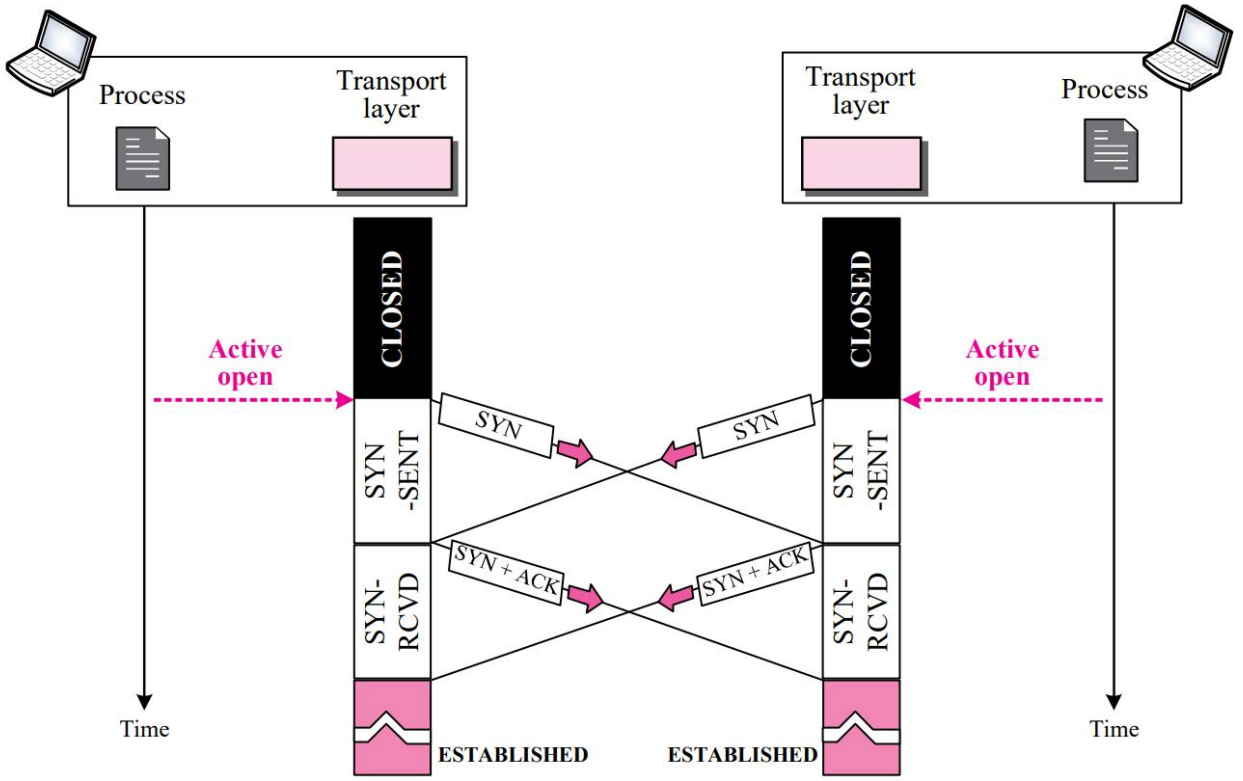
TCP 연결 (2/26)

- 연결 설정 (2/6)
  - 3단계 핸드셰이크
    - 절차
      1. 클라이언트는 순서 번호 동기화를 목적으로, SYN 플래그가 설정된 세그먼트를 전송함
         - 데이터를 전송하지 않지만, 순서번호를 1 소비함
      2. 서버는 SYN과 ACK 플래그가 설정된 세그먼트를 전송함
         - 서버에서 클라이언트로 전송되는 순서 번호를 초기화하고, 클라이언트의 SYN 세그먼트를 확인응답함
         - 수신 윈도우 크기인 rwnd를 포함함
      3. 클라이언트는 ACK 세그먼트를 전송하여 두 번째 세그먼트의 수신을 확인함
         - 데이터를 전송하지 않고, 순서 번호를 소비하지 않음



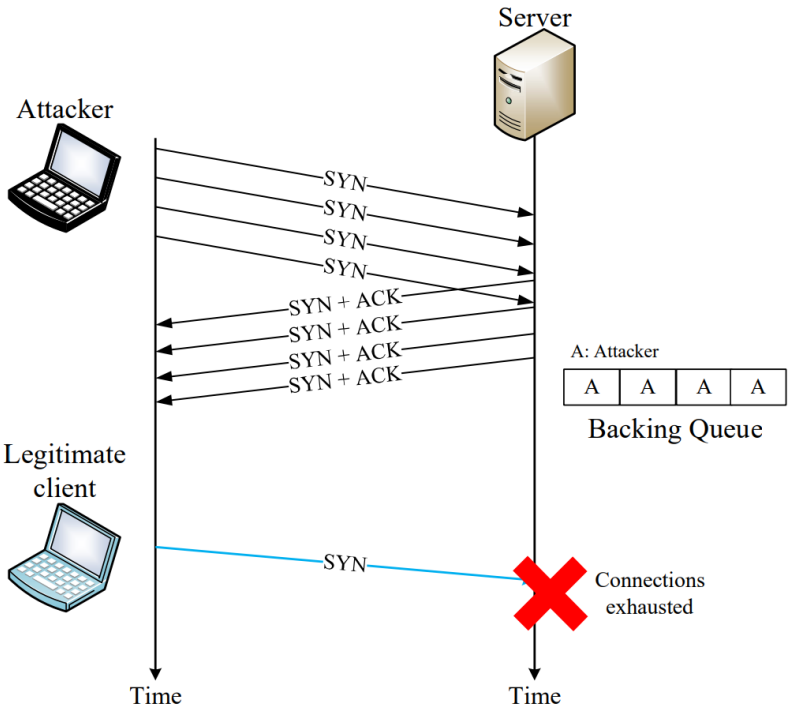
TCP 연결 (3/26)

- 연결 설정 (3/6)
  - 동시 개방 (Simultaneous open)
    - 송, 수신 프로세스가 서로 능동 개방을 요구하는 경우임
    - 양측의 TCP는 서로에게 SYN + ACK 세그먼트를 전송함으로써 단일 연결이 두 TCP 사이에 수립됨



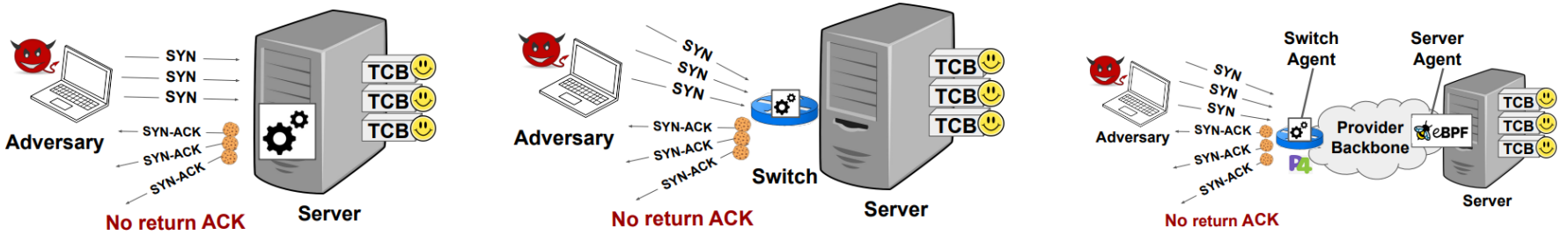
TCP 연결 (4/26)

- 연결 설정 (4/6)
  - SYN 플러딩 (Flooding) 공격 (1/3)
    - 정의
      - 다수의 SYN 세그먼트를 하나의 서버에 전송하는 서비스 거부 공격 (Denial of Service)
    - 영향
      - 서버는 SYN 패킷을 수신하고 SYN + ACK 세그먼트를 전송함
      - 클라이언트의 ACK 패킷을 기다리는 동안 자원이 사용되지 않고 할당되어 있음
        - 가용성 (Availability)이 침해됨



TCP 연결 (5/26)

- 연결 설정 (5/6)
  - SYN 플러딩 공격 (2/3)
    - 대응 방안 (1/2)
      - SYN Cookie
        - 클라이언트에서 연결 요청이 있을 경우, SYN + ACK 패킷에 쿠키 값을 포함함
        - ACK에서 쿠키 값을 검증하여 연결을 형성함
        - 자원의 할당을 연기함으로써 공격을 방지할 수 있음
        - e.g., 리눅스에서 `sysctl -w net.ipv4.tcp_syncookies=1` 명령어로 설정 가능함



(a) Server-based defense. (b) Switch-based defense. (c) SMARTCOOKIE: a split defense.

(ref: S. Yoo *et al.*, "SmartCookie: Blocking Large-Scale SYN Floods with a Split-Proxy Defense on Programmable Data Planes," in *USENIX Security*, 2024.)

TCP 연결 (6/26)

- 연결 설정 (6/6)
  - SYN 플러딩 공격 (3/3)
    - 대응 방안 (2/2)
      - 백 로그 큐 사이즈 증가
        - 백 로그 큐의 크기를 증가시켜 접속 요구 허용량을 늘림
        - 근본적인 해결 방법이 아니므로 다른 방법과 병행되어야 함
        - e.g., 리눅스에서 `sysctl -w net.ipv4.tcp_max_syn_backlog=1024` 명령어로 설정할 수 있음
      - 필터링
        - 의심 패킷에 대한 차단 정책을 수행함
        - iptables 등이 사용될 수 있음
        - e.g., 80번 포트에 대해 초당 10개 까지의 SYN 패킷을 허용하는 경우

```
# iptables -A INPUT -p TCP --dport 80 --syn -m limit 10/second -j ACCEPT
# iptables -A INPUT -p TCP --dport 80 --syn -j DROP
```

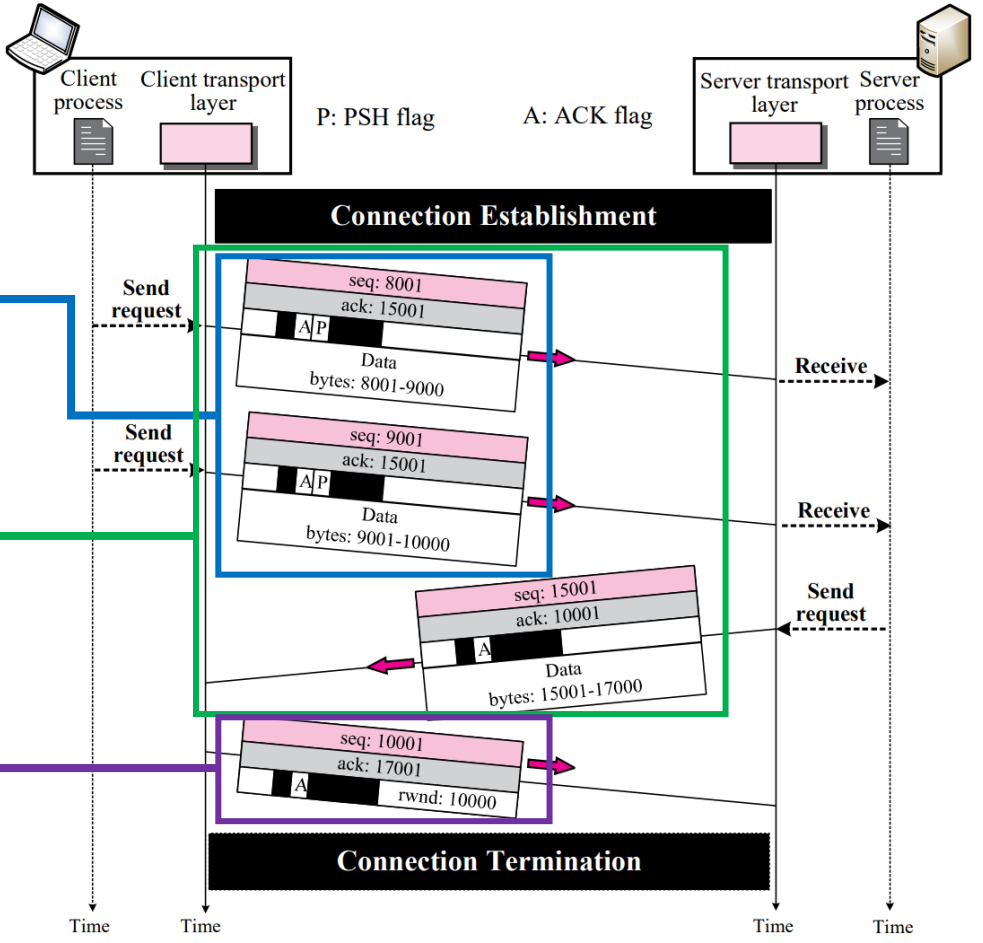
TCP 연결 (7/26)

- 데이터 전송 (1/3)
  - 정의
    - 연결이 설정된 후, 양 방향으로 데이터와 확인응답이 전송되는 단계

PSH 플래그를 설정하여, 서버에 데이터가 도착하는 대로 서버 프로세스에 전송되도록 함

데이터와 확인 응답이 동시에 전달 됨

더 이상 보낼 데이터가 없으므로 확인응답만을 전송함



TCP 연결 (8/26)

- 데이터 전송 (2/3)
  - 푸쉬 데이터
    - 정의
      - PSH 플래그가 1로 설정되어 버퍼링 없이 즉시 처리되는 데이터
    - 동작
      - 송신 프로세스에서 송신 TCP에 푸쉬 동작을 요구하면, 송신 TCP는 세그먼트를 만들어서 즉시 전송해야 함
      - 수신 TCP는 PSH 플래그가 1로 설정되었음을 확인하고 즉각적으로 응용 프로그램에 데이터를 전달함
    - e.g., Telnet과 같은 대화형 응용 프로그램에서 사용됨

TCP 연결 (9/26)

- 데이터 전송 (3/3)
  - 긴급 데이터
    - 정의
      - URG 플래그가 1로 설정되어, 수신측에서 특별한 처리가 수행되는 데이터
    - 동작
      - 일반적으로 긴급 포인터가 긴급 데이터의 마지막 위치를 가리키고 긴급 데이터가 다른 일반 데이터에 비해 우선적으로 처리됨
    - e.g., Telnet에서 인터럽트 유형 명령을 보내는데에 사용됨

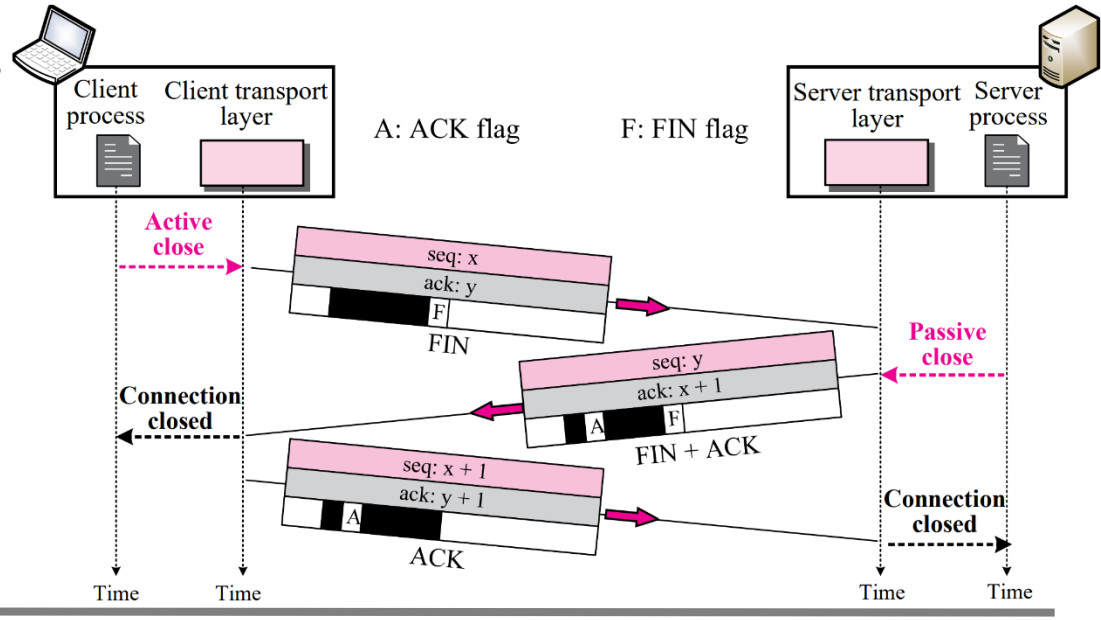
> **5. Advice to New Applications Employing TCP**
>
> As a result of the issues discussed in Section 3.2 and Section 3.4, new applications SHOULD NOT employ the TCP urgent mechanism. However, TCP implementations MUST still include support for the urgent mechanism such that existing applications can still use it.

TCP 연결 (10/26)

- 연결 종료 (1/4)
  - 정의
    - 더 이상 전송할 데이터가 없음을 확인한 후, 연결을 종료하기 위한 신호가 교환되는 단계
  - 특징
    - 일반적으로 클라이언트에서 종료를 시작함
    - 연결 종료를 위해 3단계 핸드셰이크와 반-닫기 (half-close) 옵션을 가진 4단계 핸드셰이크가 사용됨

TCP 연결 (11/26)

- 연결 종료 (2/4)
  - 3 단계 핸드셰이크
    - 절차
      - 클라이언트 프로세스로부터 close 명령을 수신한 클라이언트 TCP는 FIN 플래그를 설정한 세그먼트를 전송함
        - 데이터를 포함하지 않는 경우, 순서 번호 1을 소비함
      - 서버 TCP는 클라이언트 TCP로부터의 수신을 확인함과 동시에 연결 종료를 알리기 위해 FIN + ACK를 전송함
        - 데이터를 포함하지 않는 경우, 순서 번호 1을 소비함
      - 클라이언트 TCP는 FIN 수신을 확인하기 위해 ACK 세그먼트를 전송함
        - 데이터를 전달하지 않으며, 순서 번호를 소비하지 않음



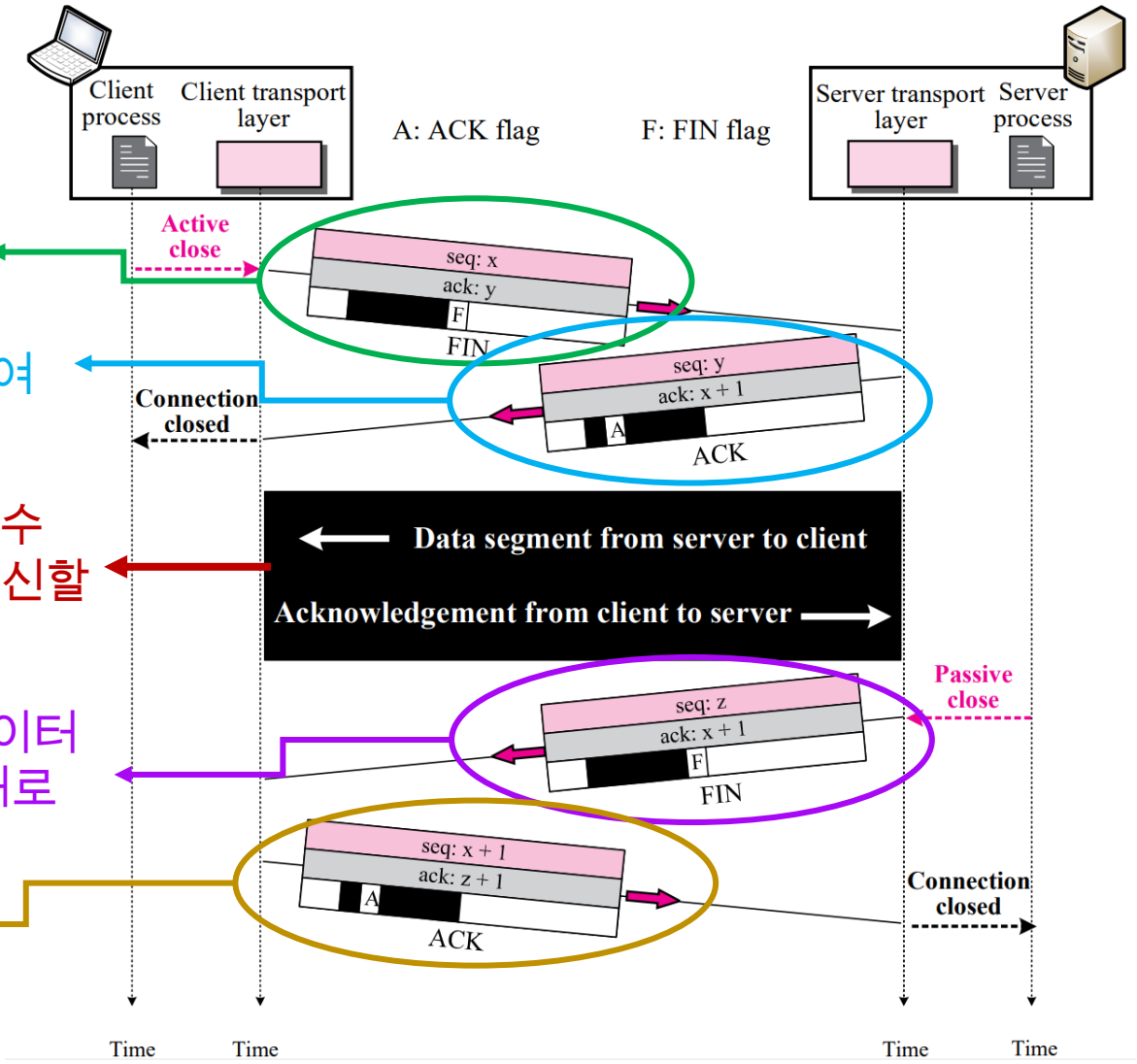
TCP 연결 (12/26)

- 연결 종료 (3/4)
  - 반-닫기 (1/2)
    - 정의
      - 양 측이 동시에 회선을 종료하지 않고, 한 쪽의 연결이 열린 채로 종료하는 기법
    - 특징
      - 서버나 클라이언트 모두 반-닫기 요청을 시작할 수 있음
      - 4 단계 핸드셰이킹을 사용함
      - 불필요한 자원을 미리 해제할 수 있음
    - e.g., 클라이언트가 정렬하기 원하는 데이터를 서버에 전송하는 경우

TCP 연결 (13/26)

- 연결 종료 (4/4)
  - 반-닫기 (2/2)
    - 절차

1. 클라이언트는 FIN 세그먼트를 전송하여 연결을 반-닫기 함
2. 서버는 ACK 세그먼트를 전송하여 반-닫기를 수락함
3. 서버는 여전히 데이터를 전송할 수 있고, 클라이언트는 데이터를 수신할 수 있는 상태임
4. 클라이언트는 서버에 전송한 데이터가 없기 때문에 확인 번호는 그대로 x + 1임
5. 양 측의 종료가 수행됨



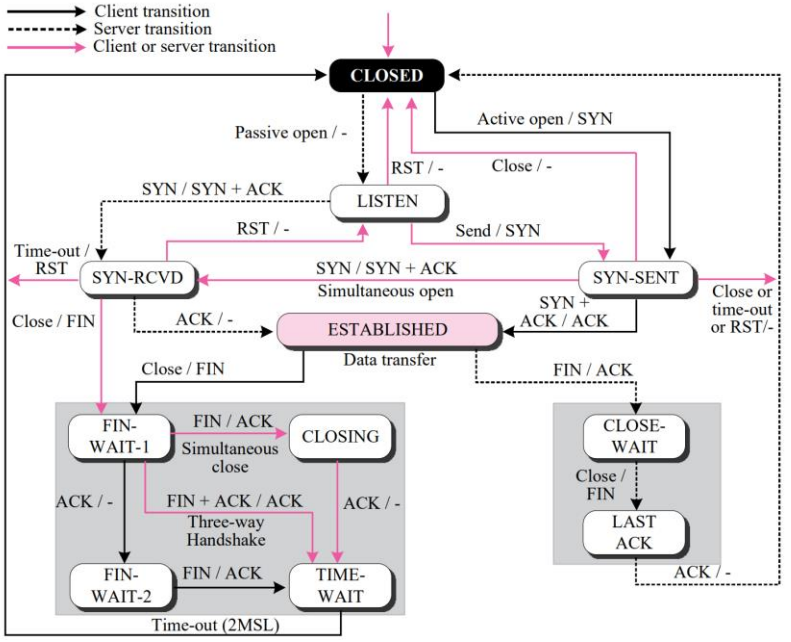
TCP 연결 (14/26)

- 연결 리셋
  - 연결 거절
    - 연결 요청을 거절하기 위해 RST 비트를 설정한 세그먼트를 전송할 수 있음
  - 연결 중단
    - 비정상적인 연결을 중단하기 위해 RST 세그먼트를 전송할 수 있음
  - 휴지 연결의 종료
    - 다른 편의 TCP가 오랜 기간 휴지 상태에 있음을 확인하고 연결을 파기하기 위해 RST 세그먼트를 전송할 수 있음

TCP 연결 (15/26)

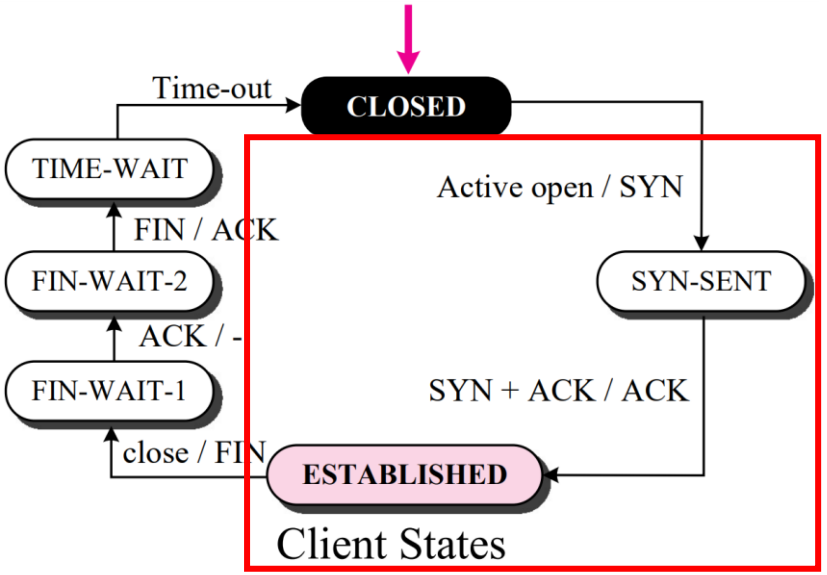
- 상태 천이 다이어그램 (State Transition Diagram) (1/12)
  - 구성 요소
    - 타원은 상태를 나타냄
    - 지시선은 한 상태에서 다른 상태로의 천이를 나타냄
      - 선 유형에 따라 서버가 겪는 천이, 클라이언트가 겪는 천이, 특수한 상황을 나타냄
    - 문자열은 수신된 입력 / 전송하는 출력의 형태를 가짐

State	Description
CLOSED	No connection exists
LISTEN	Passive open received; waiting for SYN
SYN-SENT	SYN sent; waiting for ACK
SYN-RCVD	SYN + ACK sent; waiting for ACK
ESTABLISHED	Connection established; data transfer in progress
FIN-WAIT-1	First FIN sent; waiting for ACK
FIN-WAIT-2	ACK to first FIN received; waiting for second FIN
CLOSE-WAIT	First FIN received, ACK sent; waiting for application to close
TIME-WAIT	Second FIN received, ACK sent; waiting for 2MSL time-out
LAST-ACK	Second FIN sent; waiting for ACK
CLOSING	Both sides decided to close simultaneously



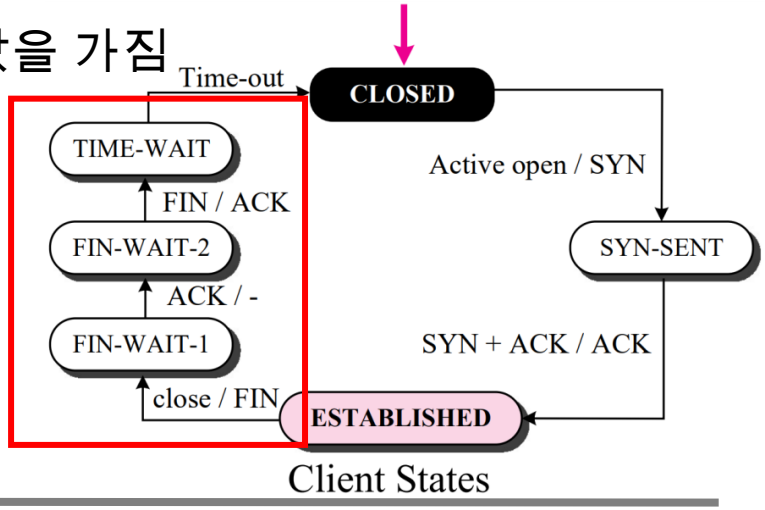
TCP 연결 (16/26)

- 상태 천이 다이어그램 (2/12)
  - 시나리오: 연결 설정 및 반-닫기 종료 (1/5)
    - 클라이언트 상태 (1/3)
      1. TCP는 프로세스로부터 능동 개방을 수신하고 SYN 세그먼트를 전송 후, SYN-SENT로 상태 변경을 수행함
      2. SYN + ACK을 수신한 TCP는 ACK을 전송하고 ESTABLISHED 상태가 됨
         - 이 상태에서는 양방향으로 데이터를 교환하며 확인응답 함



TCP 연결 (17/26)

- 상태 천이 다이어그램 (3/12)
  - 시나리오: 연결 설정 및 반-닫기 종료 (2/5)
    - 클라이언트 상태 (2/3)
      3. 더 이상 전송할 데이터가 없는 경우, 능동 닫기 명령을 수신한 TCP는 FIN 세그먼트를 전송하고 FIN-WAIT-1 상태로 천이됨
      4. FIN에 대한 ACK를 수신한 클라이언트 TCP는 FIN-WAIT-2 상태가 되고 서버로부터의 FIN을 기다림
      5. FIN 세그먼트를 수신한 TCP는 ACK를 전송하고 TIME-WAIT 상태에서 MSL (Maximum Segment Lifetime)의 2배의 시간동안 대기함
         - MSL은 일반적으로 30초에서 1분 사이의 값을 가짐

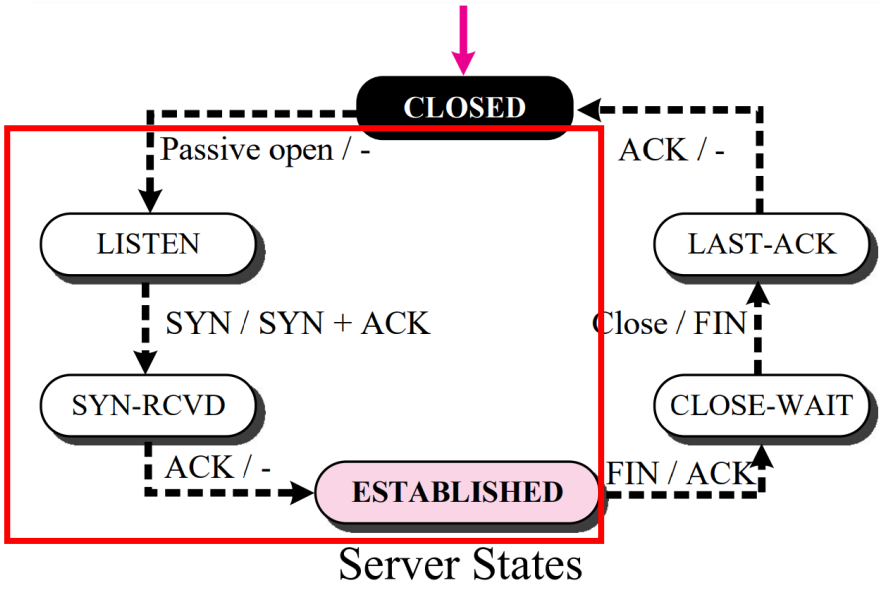


TCP 연결 (18/26)

- 상태 천이 다이어그램 (4/12)
  - 시나리오: 연결 설정 및 반-닫기 종료 (3/5)
    - 클라이언트 상태 (3/3)
      - TIME-WAIT 상태와 2MSL 타이머가 사용되는 이유
        - ACK 누락에 따른 서버 종료 보장
          - 서버의 FIN에 대한 응답으로 ACK 세그먼트가 손실되는 경우, 서버는 FIN 세그먼트를 재전송을 수행함
          - 클라이언트가 TIME-WAIT 없이 CLOSED 상태에 들어가면 재전송된 FIN을 수신하지 못하고 서버는 연결을 종료할 수 없음
          - 하나의 FIN 세그먼트가 없어지고, 또 다른 FIN이 도착하는데 충분한 시간으로써 2MSL 만큼의 타이머를 사용함
        - 한 연결에서의 중복 세그먼트 해결
          - 클라이언트와 서버가 연결이 종료되고 동일한 소켓으로 새로운 연결을 개방할 때, 충분한 시간 간격이 없다면 이전 연결에서의 세그먼트가 새 연결의 세그먼트로 간주될 수 있음
          - TCP에서는 2MSL 만큼의 시간이 지나지 않으면, 이전 연결과 동일한 소켓 주소를 갖는 새로운 연결이 설정되지 않도록 함

TCP 연결 (19/26)

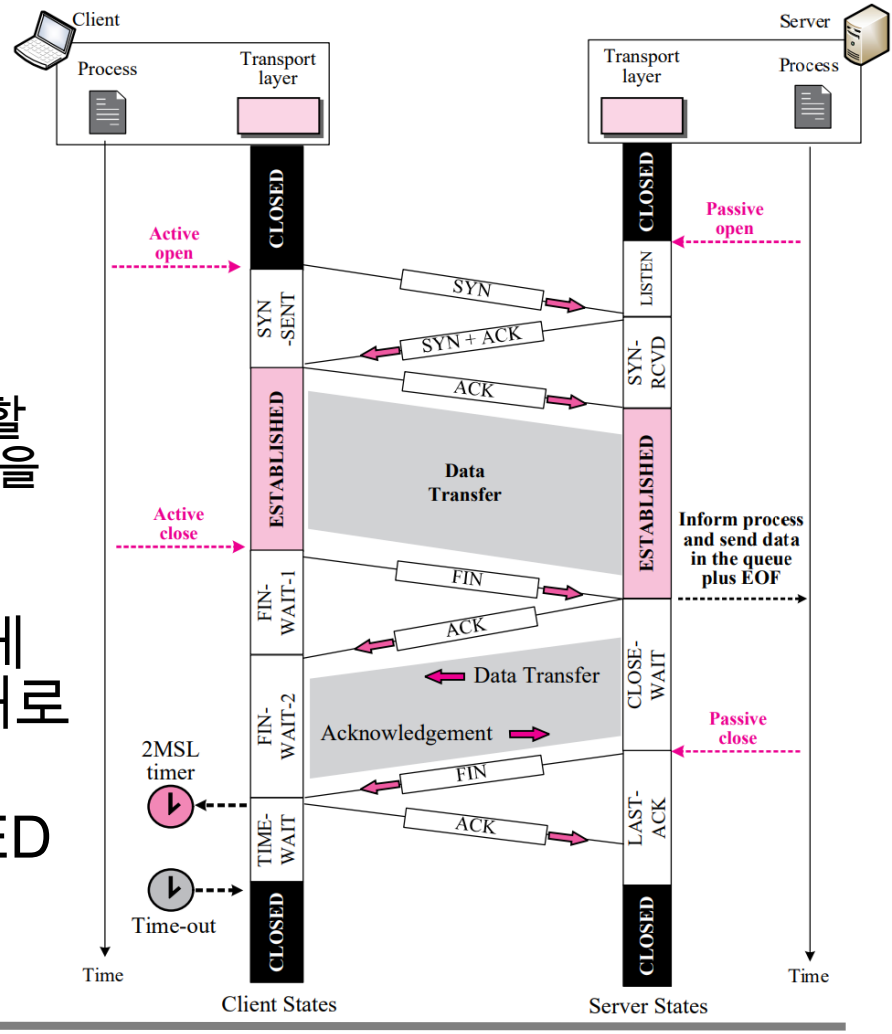
- 상태 천이 다이어그램 (5/12)
  - 시나리오: 연결 설정 및 반-닫기 종료 (4/5)
    - 서버 상태 (1/2)
      1. 서버 프로세스로부터 open 명령을 수신하고 SYN 세그먼트를 수신하기 전까지 LISTEN 상태에 머무름
      2. SYN 세그먼트를 수신한 서버 TCP는 SYN + ACK를 전송하고 SYN-RCVD 상태에서 클라이언트가 ACK를 전송하기를 기다림
      3. ACK를 수신한 서버 TCP는 상태를 ESTABISHED로 변경하고 클라이언트와 데이터를 교환함



TCP 연결 (20/26)

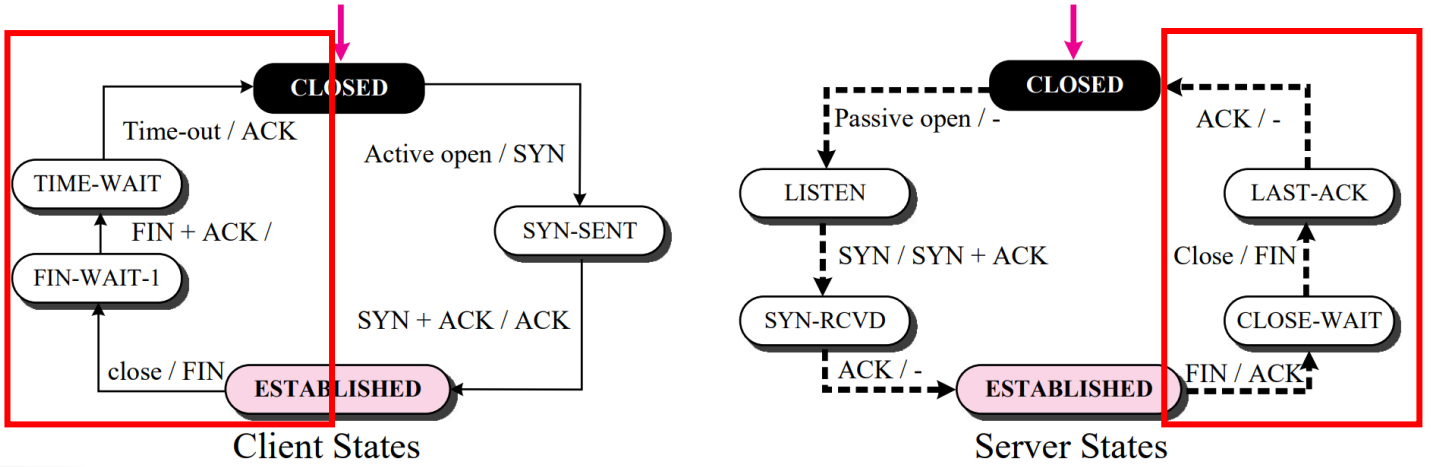
- 상태 천이 다이어그램 (6/12)
  - 시나리오: 연결 설정 및 반-닫기 종료 (5/5)
    - 서버 상태 (2/2)

4. FIN 세그먼트를 수신한 서버 TCP는 클라이언트에게 ACK를 전송하고 CLOSE-WAIT 상태로 변경함
   - 반-닫기 방식을 사용하므로 서버는 클라이언트에게 데이터를 계속 전송할 수 있고, 클라이언트로부터 확인응답을 수신할 수 있음
5. 응용 프로그램으로부터 close 명령을 수신하면 클라이언트에게 FIN을 전송하고 LAST-ACK 상태로 변경됨
6. 마지막 ACK를 수신하면 CLOSED 상태로 돌아감



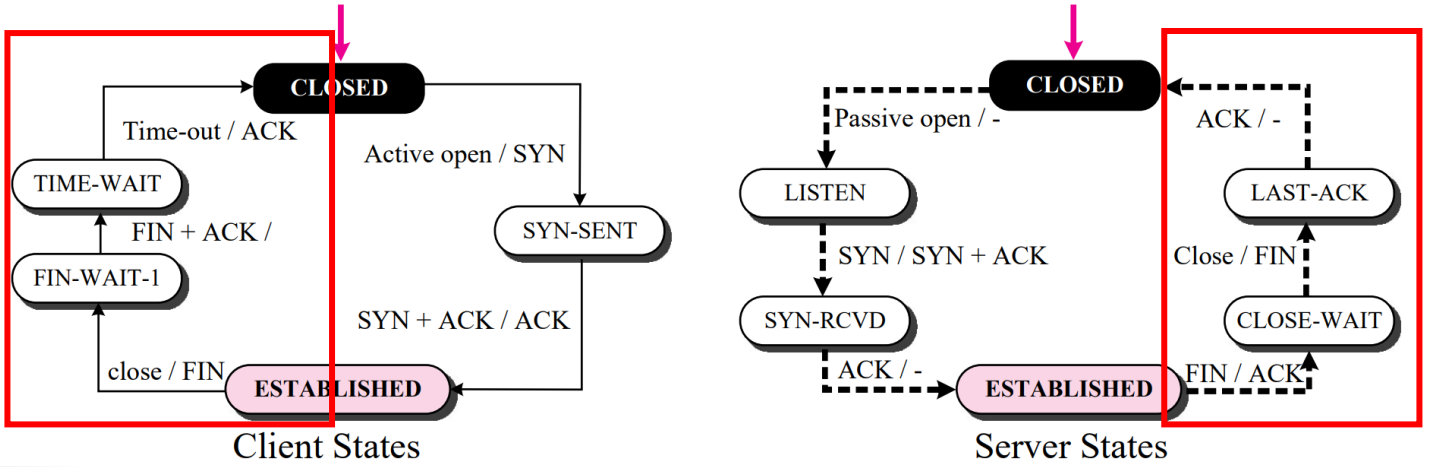
TCP 연결 (21/26)

- 상태 천이 다이어그램 (7/12)
  - 시나리오: 3 단계 핸드셰이크를 통한 연결 종료 (1/2)
    1. 클라이언트 TCP는 FIN 세그먼트를 전송하고 FIN-WAIT-1 상태로 진행함
    2. FIN 세그먼트를 수신한 서버 TCP는 CLOSE-WAIT 상태에서 프로세스로부터 close 명령을 수신하기까지 확인응답 전송을 보류함
    3. close 명령을 수신한 후, FIN + ACK 메시지를 전송하고 LAST-ACK 상태가 됨



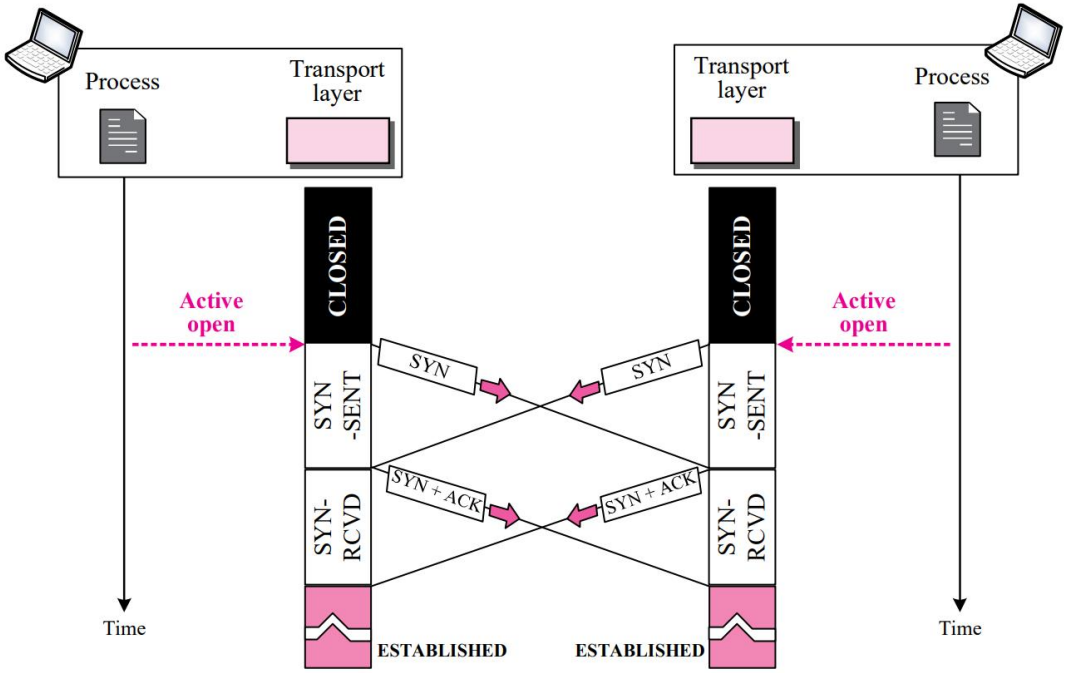
TCP 연결 (22/26)

- 상태 천이 다이어그램 (8/12)
  - 시나리오: 3 단계 핸드셰이크를 통한 연결 종료 (2/2)
    4. 클라이언트는 FIN-WAIT-2 상태를 무시하고 바로 TIME-WAIT로 이동함
    5. 이후 과정은 4 단계 핸드셰이크와 동일함



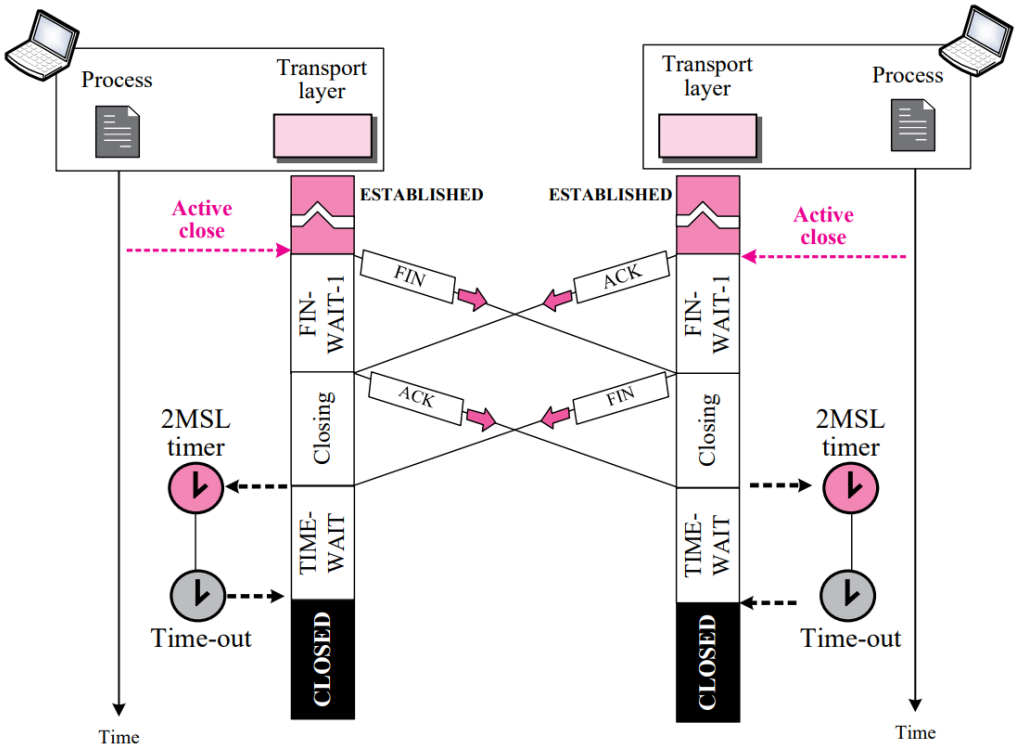
TCP 연결 (23/26)

- 상태 천이 다이어그램 (9/12)
  - 시나리오: 기타 연결 및 종료 방식 (1/4)
    - 동시 개방
      - 클라이언트, 서버가 아닌 서로의 지역 및 포트를 알고 있는 두 피어 사이에 일어남
      - 두 TCP 모두 SYN-SENT 및 SYN-RCVD 상태를 거쳐 ESTABLISHED 됨



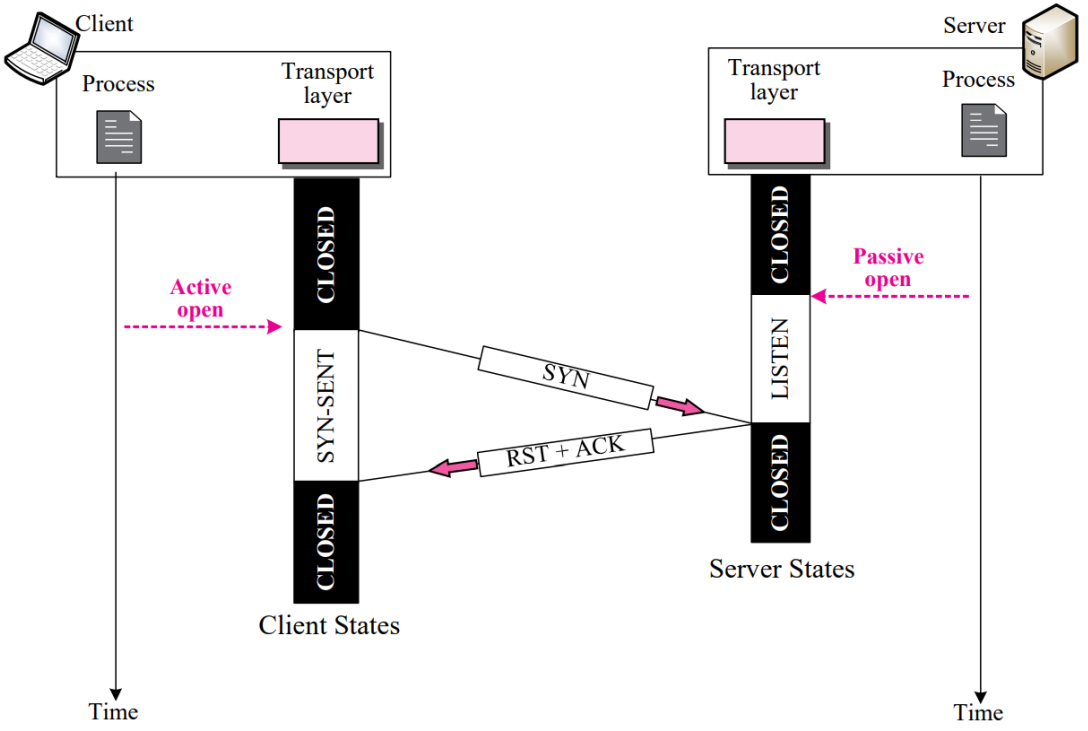
TCP 연결 (24/26)

- 상태 천이 다이어그램 (10/12)
  - 시나리오: 기타 연결 및 종료 방식 (2/4)
    - 동시 닫기
      - 양측의 TCP는 동시에 FIN 세그먼트를 전송하고 FIN-WAIT-1 상태로 들어감
      - FIN 세그먼트를 수신한 후 양측은 ACK 세그먼트를 전송하고 CLOSING 상태로 들어감



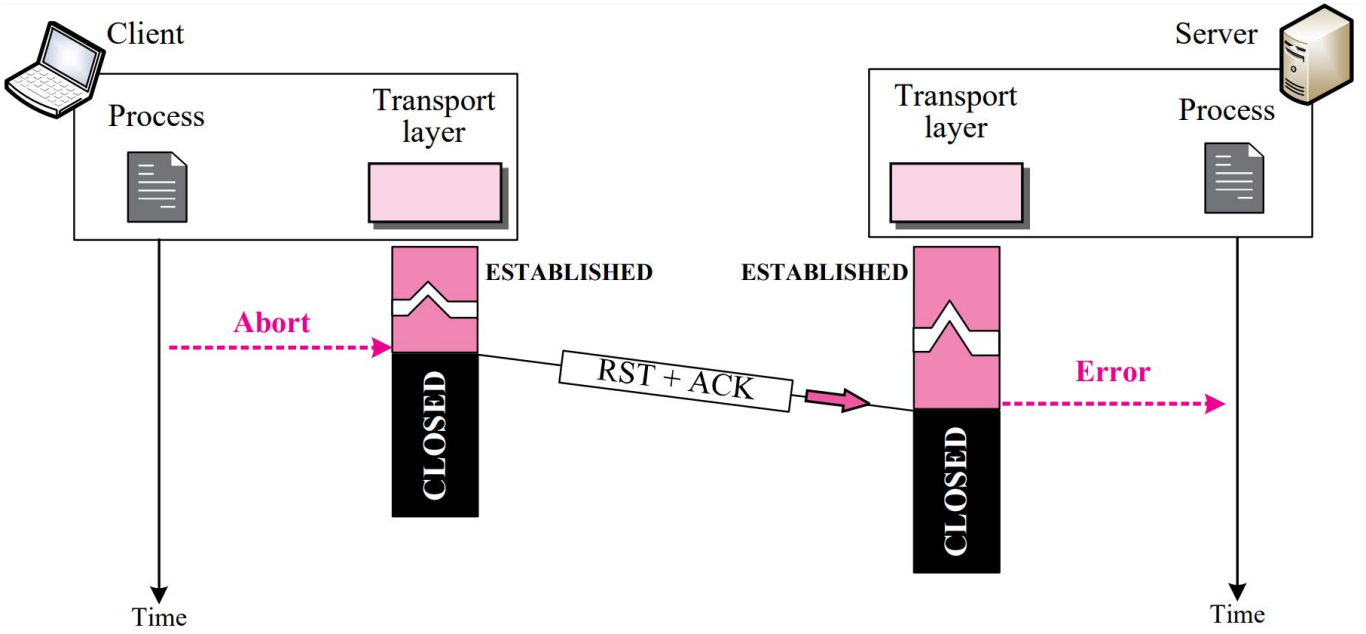
TCP 연결 (25/26)

- 상태 천이 다이어그램 (11/12)
  - 시나리오: 기타 연결 및 종료 방식 (3/4)
    - 연결 거절
      - 일반적으로, SYN 세그먼트에 있는 목적지 포트 번호가 LISTEN 상태에 있지 않은 경우 사용됨
      - SYN 세그먼트를 수신한 서버 TCP는 SYN을 확인응답하며 동시에 연결을 거절하기 위해 RST + ACK 세그먼트를 전송함
      - RST + ACK를 수신한 클라이언트는 CLOSED 상태가 됨



TCP 연결 (26/26)

- 상태 천이 다이어그램 (12/12)
  - 시나리오: 기타 연결 및 종료 방식 (4/4)
    - 연결 중단
      - 시스템에서 Abort와 같은 인터럽트가 발생하면 RST + ACK를 송신한 후, 송수신 버퍼에 있는 모든 데이터를 버림
      - RST + ACK를 수신한 주체 또한 버퍼에 있는 모든 데이터를 버리고 시스템에 에러가 발생했음을 알림

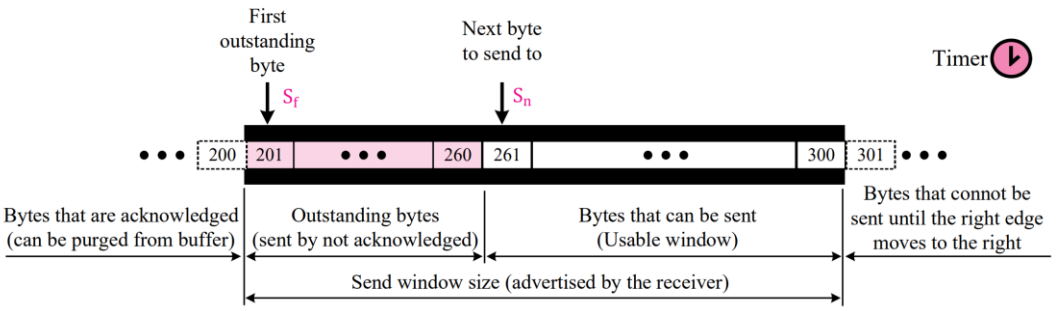


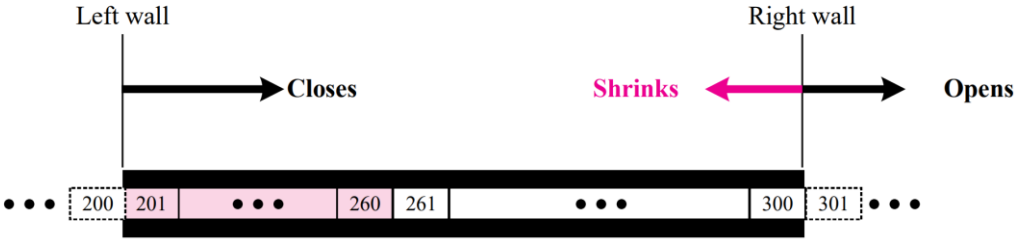
목 차

- TCP 서비스와 특징
- TCP 세그먼트
- TCP 연결
- TCP 윈도우

TCP 윈도우 (1/2)

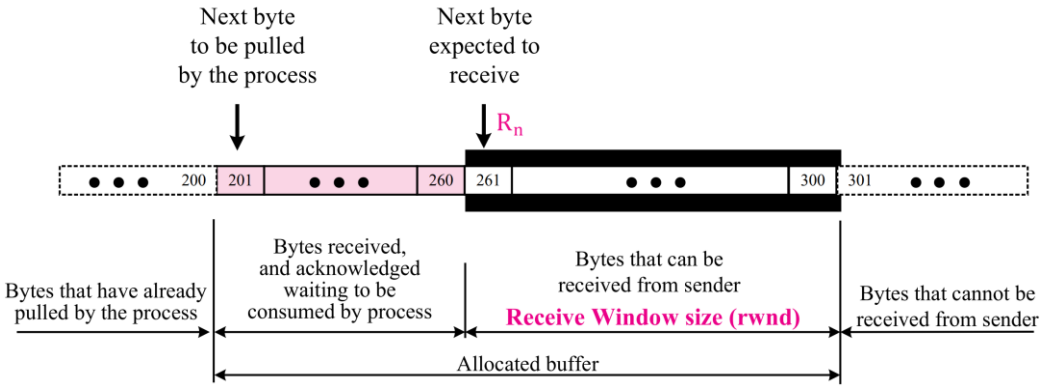
- 송신 윈도우
  - 특징
    - 선택적 반복 (Selective-Repeat) 프로토콜에서의 송신 윈도우와 유사함
      - 선택적 반복은 패킷 단위로 윈도우가 사용되나, TCP는 바이트 단위로 사용됨
      - 선택적 반복은 전송되는 각각의 패킷마다 타이머를 사용하지만, TCP는 하나의 타이머만 사용함
    - 수신자, 네트워크 혼잡도에 따라 열림 (Open), 닫힘 (Close), 축소 (Shrink)가 이루어질 수 있음

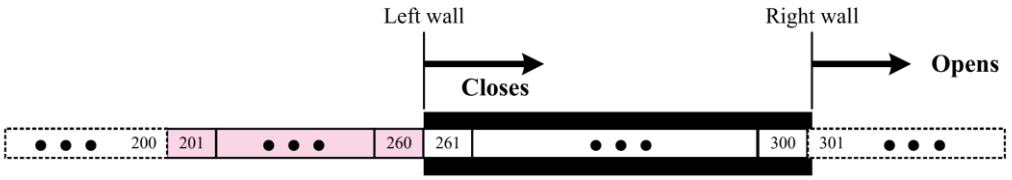




TCP 윈도우 (2/2)

- 수신 윈도우
  - 특징
    - 선택적 반복 프로토콜에서의 수신 윈도우와 유사함
      - TCP 수신 윈도우는 응용 프로세스가 자신의 속도로 데이터를 읽을 수 있음
        - rwnd = 버퍼 크기 – (수신 프로세스로부터) 읽히기를 기다리는 바이트의 수로 결정될 수 있음
      - 선택적 반복에서 확인응답은 선택적이었으나, TCP에서는 수신하기를 기대하는 바이트 번호를 전송하는 누적 확인응답 방법이 사용됨
    - 수신 윈도우가 축소되는 경우는 일어나지 않음





Thanks!

이 정 민(jeongmin@pel.sejong.ac.kr)