2026/10/06, 2026 네트워크 기초 세미나

TCP/IP 프로토콜

- 15장 TCP -

김 민 식 (minsik@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

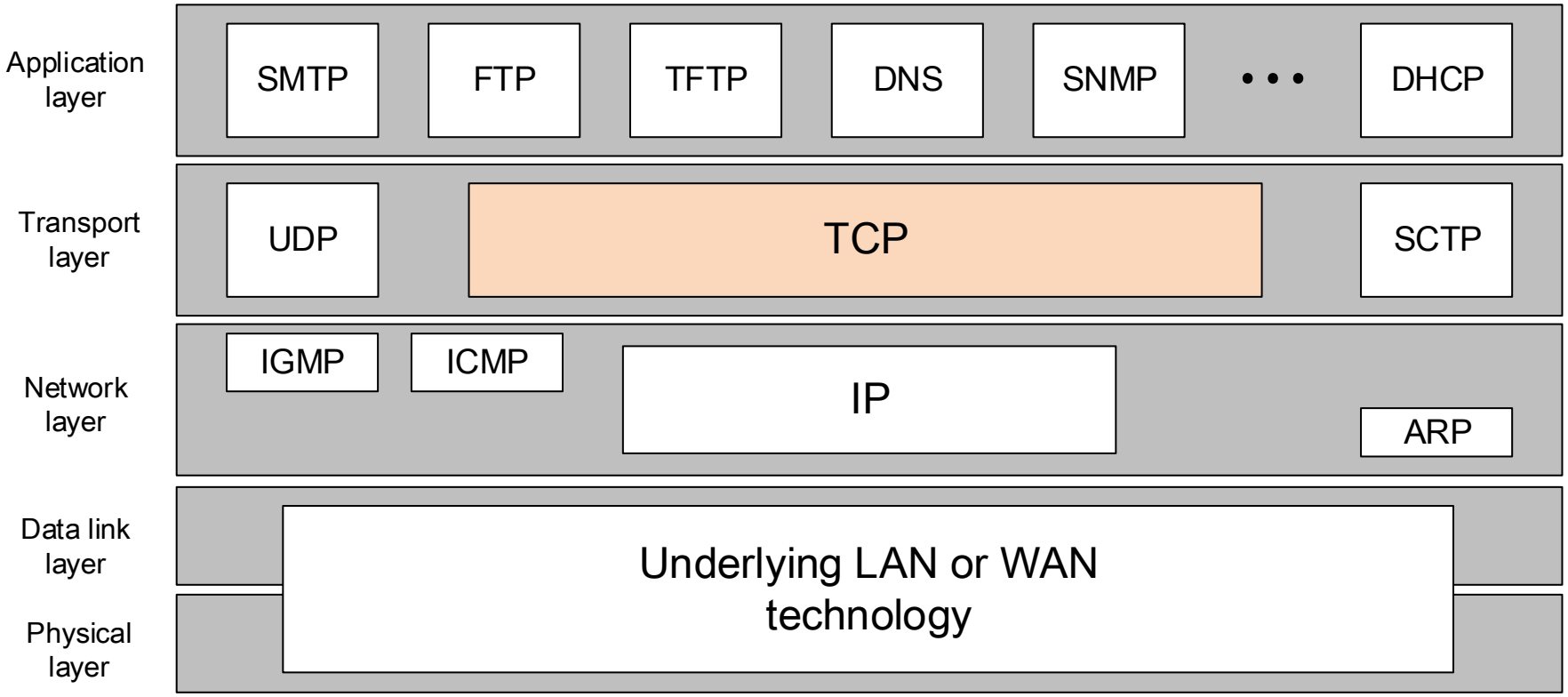
- TCP 개요
- TCP 연결 및 전송
- TCP 제어
- TCP 동작 지원 및 구현

목 차

- TCP 개요
- TCP 연결 및 전송
- TCP 제어
- TCP 동작 지원 및 구현

TCP 개요

- TCP(Transmission Control Protocol)
  - 정의
    - 연결을 설정한 후 신뢰성 있는 바이트 스트림 전달 서비스를 제공하는 전송 계층 프로토콜



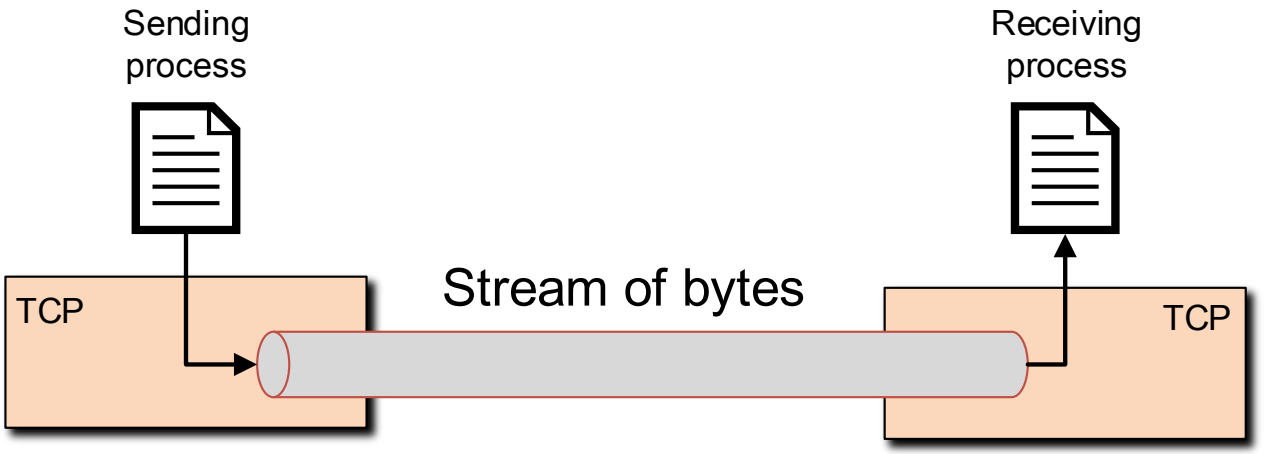
TCP 개요

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

포트 번호	프로토콜	설명
7	Echo	받은 데이터그램을 그대로 송신자에게 반환
9	Discard	받은 데이터그램을 폐기
13	Daytime	날짜와 시간을 반환
17	Quote	일일 격언 반환
19	Chargen	무작위 문자 스트림 반환
20, 21	FTP	파일 전송 프로토콜(데이터, 제어)
23	TELNET	원격 터미널 접속 프로토콜
25	SMTP	이메일 송신 프로토콜
53	DNS	도메인 이름과 IP 주소 간 변환
79	Finger	특정 사용자 정보 조회
80	HTTP	웹 문서 전송

TCP 개요

- TCP 서비스 (2/5)
  - 스트림 배달 서비스 (1/2)
    - 응용 프로그램의 데이터를 연속된 바이트 스트림 형태로 송수신함
    - 송신 프로세스는 바이트 스트림을 생성하고, 수신 프로세스는 전달된 바이트 스트림을 읽음



TCP 개요

- TCP 서비스 (3/5)
  - 스트림 배달 서비스 (2/2)
    - 송신 / 수신 버퍼(Buffer)
      - 송신 프로세스와 수신 프로세스의 데이터 처리 속도가 서로 다를 수 있으므로 데이터를 임시 저장하기 위한 버퍼를 사용함
      - 송신 버퍼는 전송할 데이터와 전송 후 아직 확인 응답되지 않은 데이터를 저장함
      - 수신 버퍼는 수신한 데이터를 응용 프로그램이 읽기 전까지 저장함
    - 세그먼트(Segment)
      - TCP는 바이트 스트림을 여러 바이트 단위로 묶어 세그먼트 형태로 전송함
      - 각 세그먼트에는 제어 정보를 위한 TCP 헤더가 추가됨
      - 세그먼트의 크기는 항상 동일하지 않음

*버퍼(Buffer)
데이터를 바로 처리하지 못할 때를 대비해 일시적으로 저장해두는 메모리 공간

TCP 개요

- TCP 서비스 (4/5)
  - 전이중 통신(Full-Duplex)
    - 송신과 수신이 동시에 가능하여 데이터를 양방향으로 전송할 수 있음
    - 각 TCP는 송신 버퍼와 수신 버퍼를 가짐
  - 다중화(Multiplexing) 및 역다중화(Demultiplexing)
    - 송신 측에서 여러 프로세스의 데이터를 모아 전송하고, 수신 측에서 포트 번호를 기준으로 해당 프로세스에 전달함
    - 연결 지향 프로토콜이므로 프로세스 간 연결이 설정되어야 함

*다중화(Multiplexing)
여러 프로세스의 데이터를 하나의 전송 계층에서 모아 전송하는 과정
*역다중화(Demultiplexing)
수신한 데이터를 목적지 포트 번호에 따라 해당 프로세스로 전달하는 과정

TCP 개요

- TCP 서비스 (5/5)
  - 연결 지향 서비스(Connection-Oriented Service)
    - 데이터 전송 전에 송신 측과 수신 측 사이에 가상 연결을 설정해야 함
    - 연결 설정 → 데이터 전송 → 연결 종료의 과정으로 통신함
  - 신뢰성 서비스(Reliable Service)
    - 데이터가 오류 없이 정상적으로 도착했는지 확인하기 위해 확인응답(ACK) 메커니즘을 사용함

TCP 개요

- TCP 특징 (1/3)
  - 번호화 시스템(Numbering System)
    - TCP는 데이터의 각 바이트에 번호를 부여하여 전송 순서와 수신 여부를 관리함
    - 번호화에는 바이트 번호, 순서 번호(Sequence Number), 확인응답 번호(Acknowledgement Number)가 사용됨
      - 바이트 번호(Byte Number)
        - TCP는 전송되는 모든 데이터 바이트에 번호를 부여함
        - 첫 번째 바이트 번호는 $0 \sim 2^{32} - 1$ 사이에서 임의로 결정됨
      - 순서 번호(Sequence Number)
        - 세그먼트에 포함된 첫 번째 데이터 바이트의 번호를 의미함
      - 확인응답 번호(Acknowledgement Number)
        - 수신 측이 다음에 받기를 기대하는 바이트 번호를 의미함
        - 이전 번호까지의 바이트가 정상적으로 수신되었음을 나타내는 누적 확인응답 방식임

TCP 개요

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

첫 번째 바이트 번호가 10,001일 때, 3,000 바이트의 데이터를 1,000 바이트의 세그먼트 3개로 보내는 경우, 각 세그먼트의 순서 번호를 구하시오.

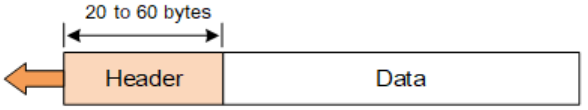
- 세그먼트 1: 순서 번호 10,001 / 데이터 범위: 10,001 ~ 11,000
- 세그먼트 2: 순서 번호 11,001 / 데이터 범위: 11,001 ~ 12,000
- 세그먼트 3: 순서 번호 12,001 / 데이터 범위: 12,001 ~ 13,000

TCP 개요

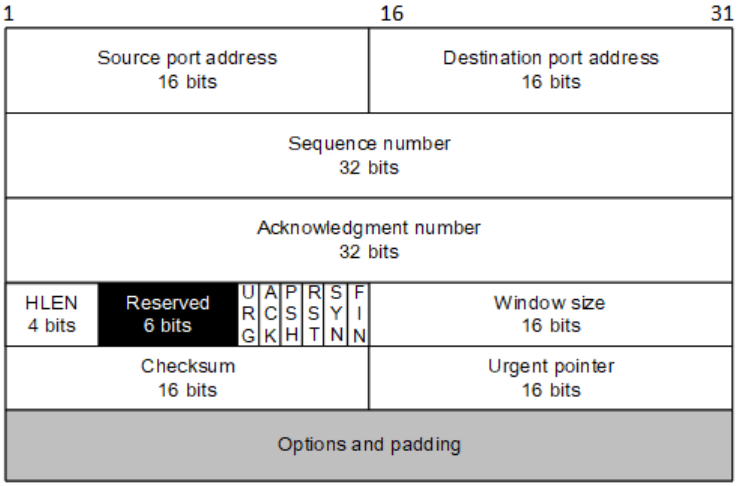
- TCP 특징 (3/3)
  - 흐름 제어(Flow Control)
    - 수신 측의 처리 능력을 초과하지 않도록 송신 데이터의 양을 조절하는 기능
  - 오류 제어(Error Control)
    - 데이터가 손실되거나 오류가 발생한 경우 이를 처리하여 신뢰성 있는 데이터 전송을 보장하는 기능
  - 혼잡 제어(Congestion Control)
    - 네트워크의 혼잡 상태에 따라 송신 측의 데이터 전송량을 조절하는 기능

TCP 개요

- 세그먼트 구조 (1/8)
  - TCP 세그먼트는 TCP 헤더 + 응용 프로그램 데이터로 구성됨
  - TCP 헤더의 길이는 20~60바이트임
  - 헤더에는 포트 번호, 순서 번호, 확인응답 번호, 제어 플래그, 윈도우 크기, 체크섬 등의 필드가 포함됨



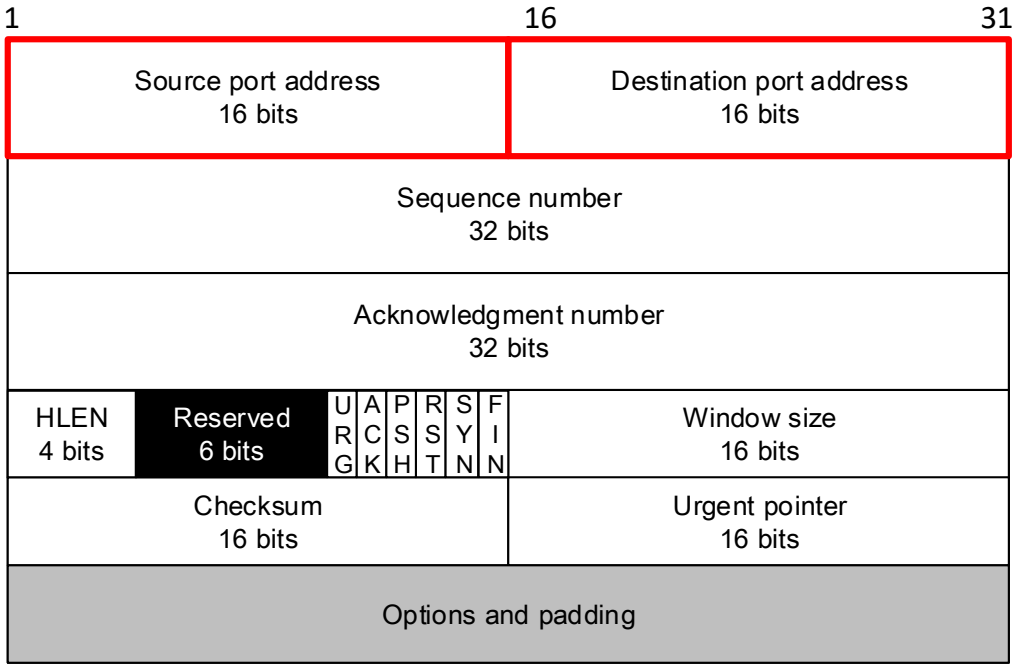
a. Segment



b. Header

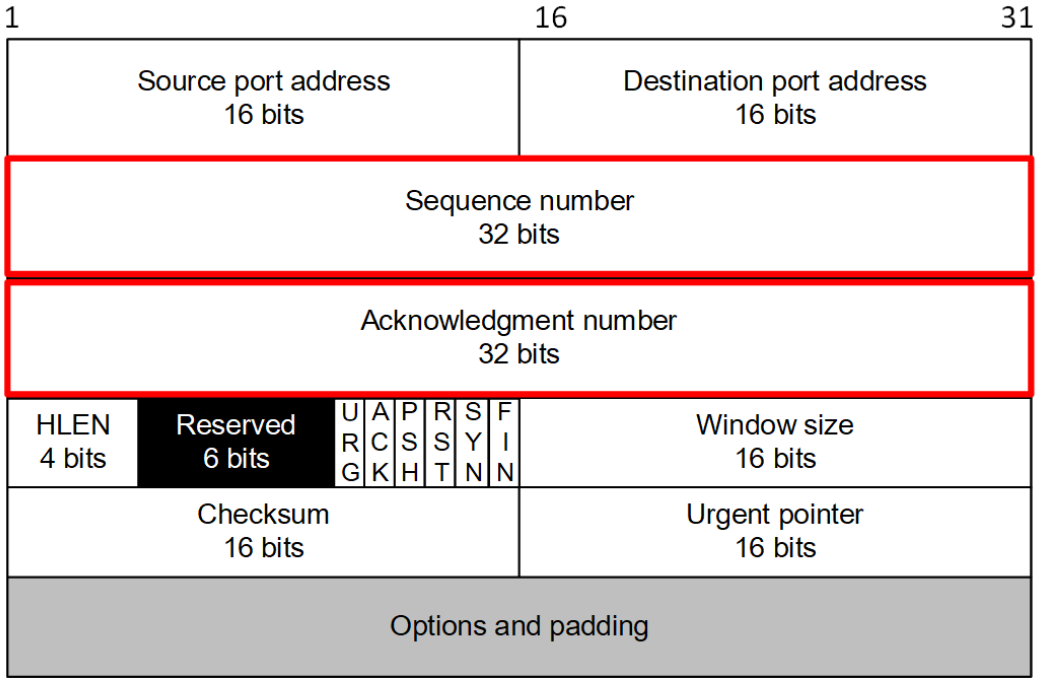
TCP 개요

- 세그먼트 구조 (2/8)
  - 발신지 포트 주소(Source Port Address)
    - 세그먼트를 전송하는 응용 프로그램의 포트 번호
  - 목적지 포트 주소(Destination Port Address)
    - 세그먼트를 수신하는 응용 프로그램의 포트 번호



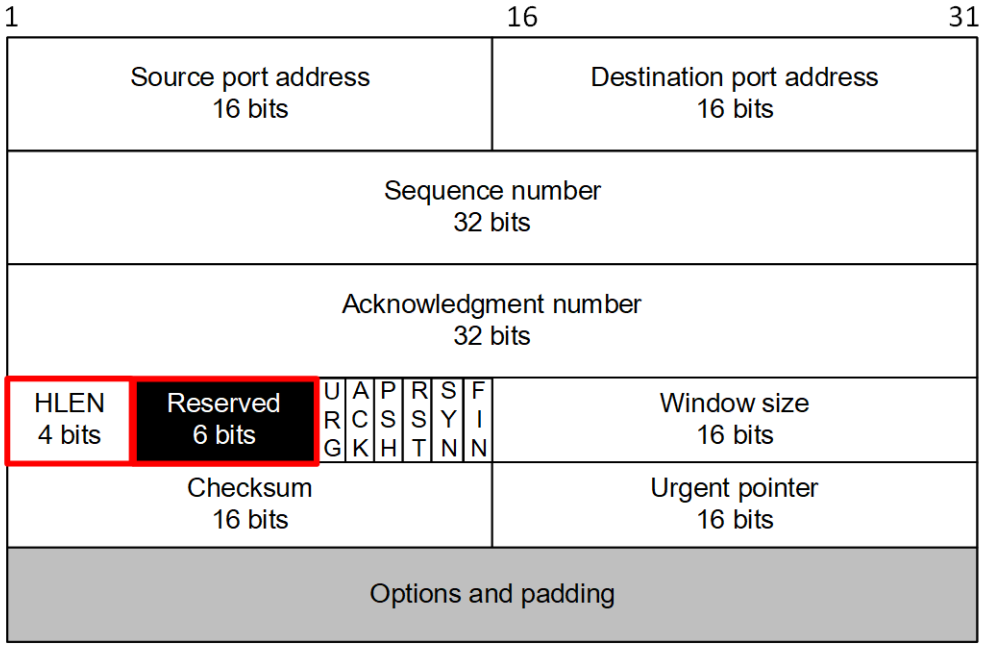
TCP 개요

- 세그먼트 구조 (3/8)
  - 순서 번호(Sequence Number)
    - 세그먼트에 포함된 데이터의 첫 번째 바이트 번호
  - 확인응답 번호(Acknowledgement Number)
    - 수신 측이 다음에 수신하기를 기대하는 바이트 번호



TCP 개요

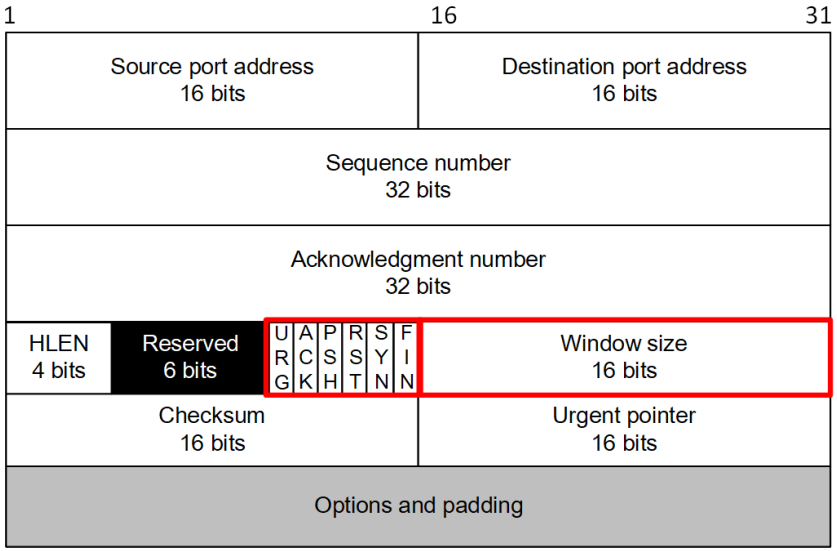
- 세그먼트 구조 (4/8)
  - 헤더 길이(HLEN, Header Length)
    - TCP 헤더의 길이를 나타내는 4비트 필드로, 4바이트 단위로 표현함
  - 예약(Reserved)
    - 향후 사용을 위해 예약된 6비트 필드



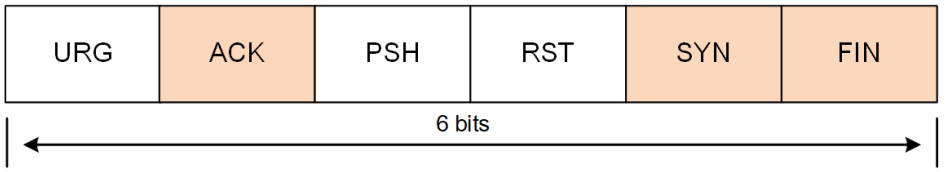
TCP 개요

- 세그먼트 구조 (5/8)
  - 제어(Control)
    - TCP 연결 및 데이터 전송을 제어하기 위해 사용하는 6개의 플래그(Flag) 비트
  - 윈도우 크기(Window Size)
    - 수신 측이 현재 수신할 수 있는 데이터의 양을 나타내며, 송신 측의 전송량을 조절하는 데 사용되는 필드

*플래그(Flag)
TCP가 지금 어떤 동작을 하려는지 알려주는 0/1 신호



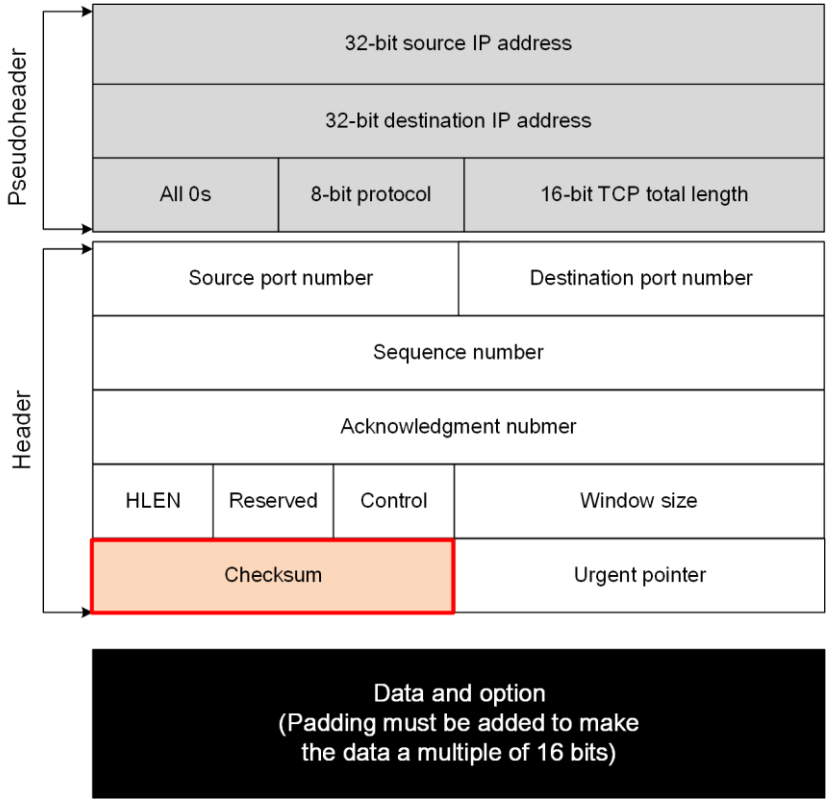
URG: Urgent pointer is valid
ACK: Acknowledgment is valid
PSH: Request for push
RST: Reset the connection
SYN: Synchronize sequence numbers
FIN: Terminate the connection



TCP 개요

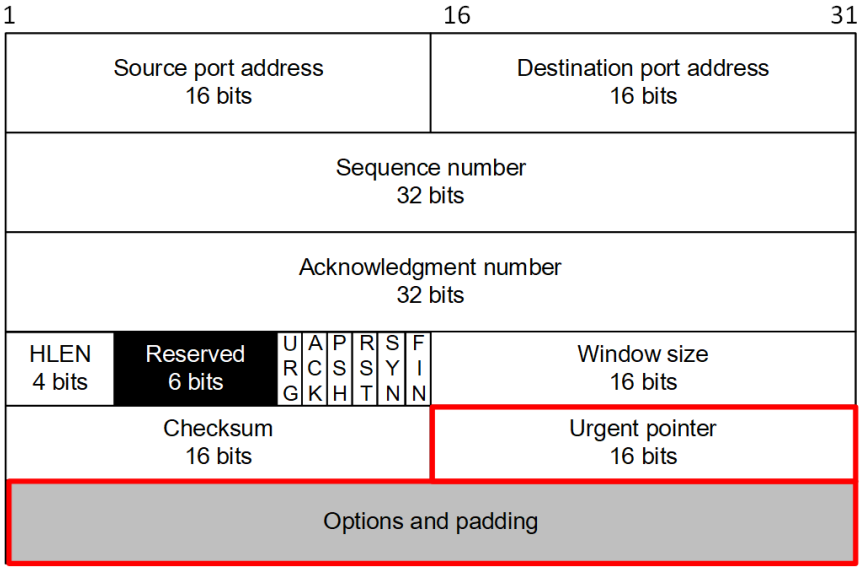
- 세그먼트 구조 (6/8)
  - 체크섬(Checksum)
    - TCP 세그먼트 전송 중 오류가 발생했는지 검사하기 위한 필드
    - 계산 시 IP 주소 등을 포함한 의사 헤더(Pseudo Header)도 함께 사용함

*의사 헤더(Pseudo Header)
TCP 데이터가 올바른 IP 주소 정보와 함께 전달됐는지 까지 확인하기 위해 체크섬 계산에만 임시로 사용하는 정보



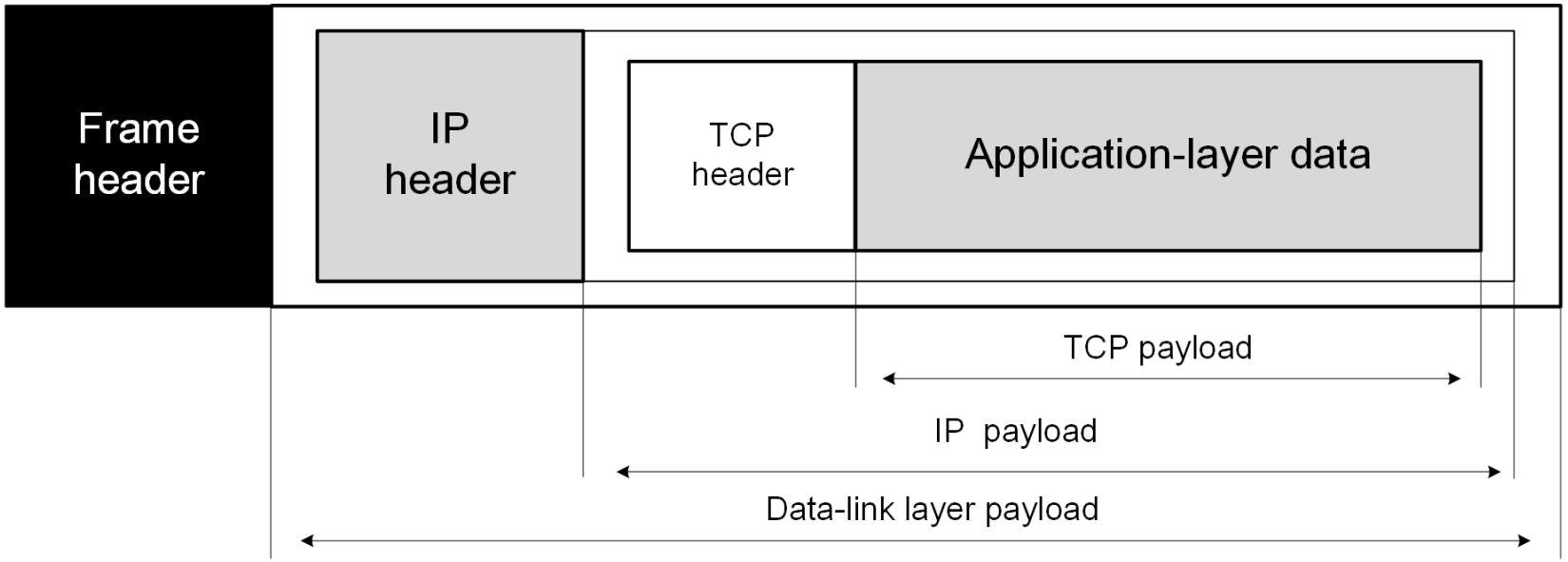
TCP 개요

- 세그먼트 구조 (7/8)
  - 긴급 포인터(Urgent Pointer)
    - 세그먼트 안에서 긴급 데이터의 끝 위치를 나타내는 16비트 필드로, URG 플래그가 설정된 경우에 사용됨
  - 옵션(Option)
    - TCP의 기본 헤더 외에 추가적인 제어 정보를 담는 영역으로, 최대 40바이트까지 사용할 수 있음



TCP 개요

- 세그먼트 구조 (8/8)
  - 캡슐화(Encapsulation)
    - TCP 세그먼트는 응용 계층에서 전달받은 데이터를 캡슐화함
    - 생성된 TCP 세그먼트는 IP 데이터그램에 캡슐화되고, IP 데이터그램은 데이터링크 계층의 프레임에 캡슐화됨



목 차

- TCP 개요
- TCP 연결 및 전송
- TCP 제어
- TCP 동작 지원 및 구현

TCP 연결 및 전송

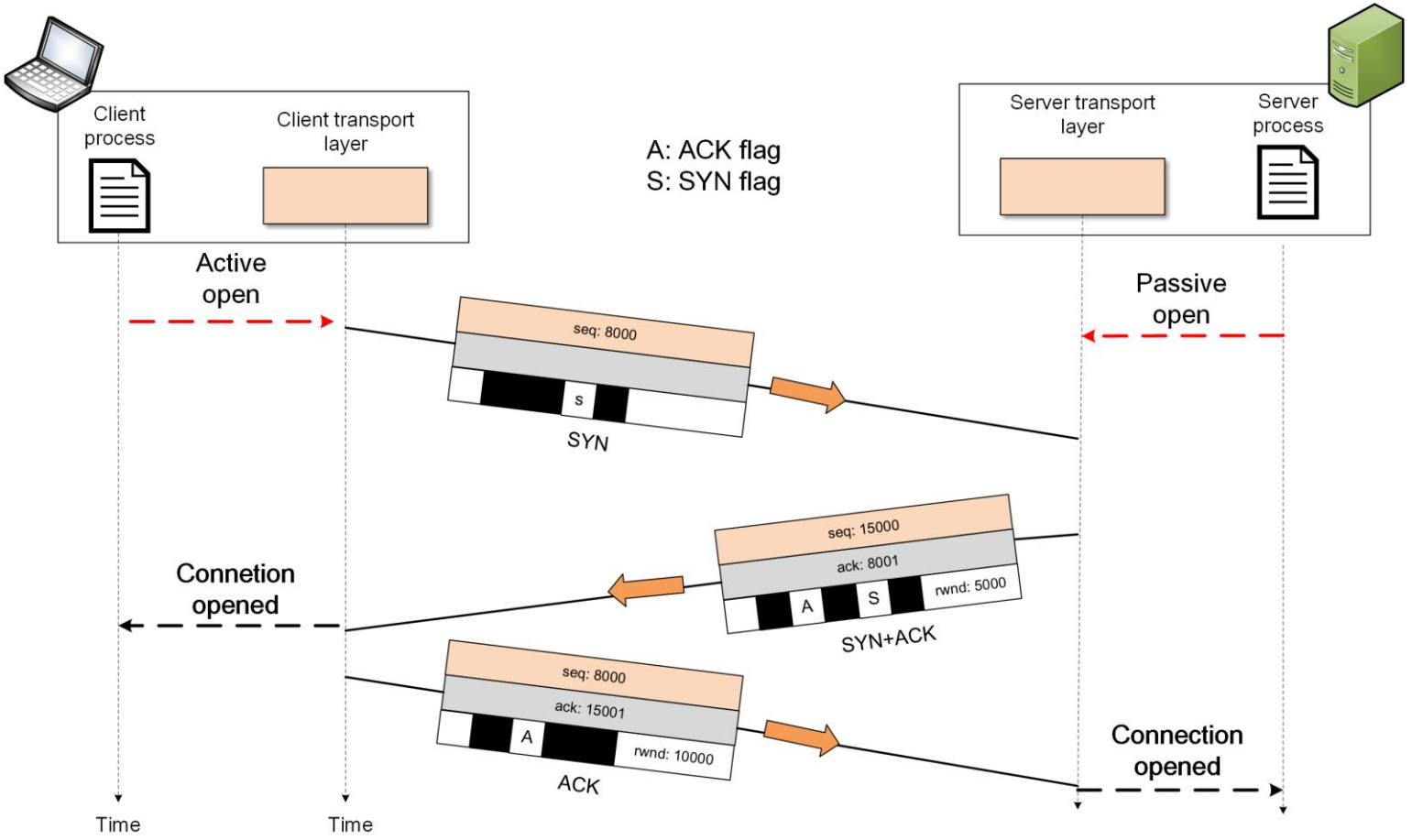
- 개요
  - TCP는 연결 지향 프로토콜로, 데이터 전송 전에 송신 측과 수신 측 사이에 가상 연결을 설정함
  - TCP 통신은 연결 설정 → 데이터 전송 → 연결 종료의 과정으로 이루어짐

TCP 연결 및 전송

- 연결 설정 (1/2)
  - 개방 요청
    - 서버 프로그램은 자신의 TCP에게 Passive Open을 요청하여 클라이언트의 연결 요청을 받을 준비가 되었음을 알림
    - 클라이언트 프로그램은 자신의 TCP에게 Active Open을 요청하여 특정 서버와 연결을 설정하고자 함을 알림
  - 3단계 핸드셰이크(Three-Way Handshake)
    1. 클라이언트는 순서 번호를 동기화하기 위해 초기 순서 번호(ISN, Initial Sequence Number)를 포함한 SYN 세그먼트를 서버에 전송함
    2. 서버는 자신의 ISN과 클라이언트의 SYN에 대한 확인응답 번호를 포함한 SYN+ACK 세그먼트를 전송함
    3. 클라이언트는 서버의 SYN에 대한 확인을 위해 ACK 세그먼트를 전송함

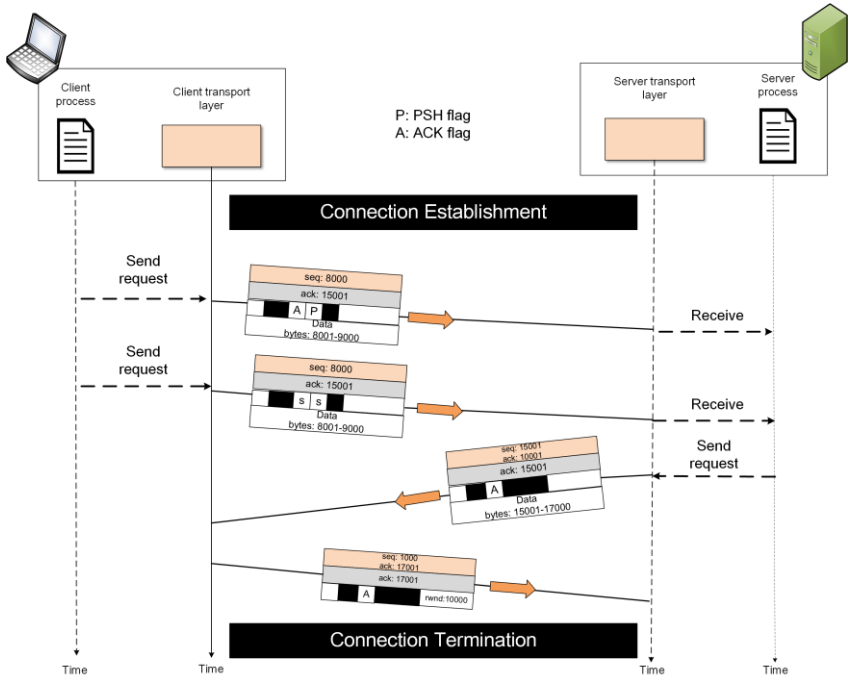
TCP 연결 및 전송

- 연결 설정 (2/2)
  - 3단계 핸드셰이크(Three-Way Handshake)



TCP 연결 및 전송

- 데이터 전송 (1/2)
  - TCP 연결이 설정되면 클라이언트와 서버는 양방향으로 데이터와 확인응답(ACK)을 전송할 수 있음
  - 데이터와 확인응답이 같은 방향으로 전송되는 경우, 하나의 세그먼트에 함께 포함하여 전송할 수 있음
  - 데이터에 확인응답을 함께 실어 보내는 방식을 피기배킹(Piggybacking)이라고 함



TCP 연결 및 전송

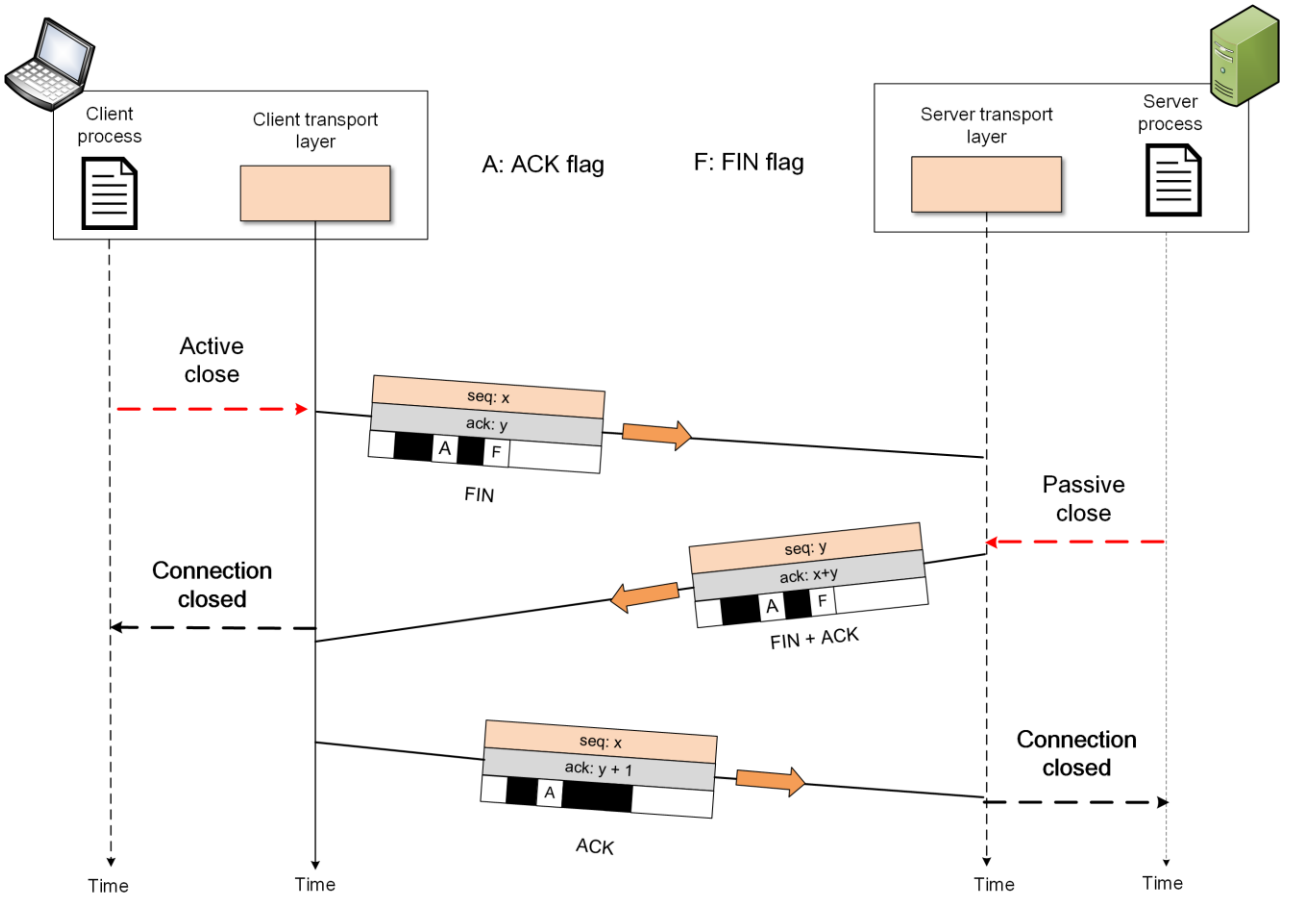
- 데이터 전송 (2/2)
  - 푸싱 데이터(Pushing Data)
    - 버퍼가 채워질 때까지 기다리지 않고 즉시 전송하며 PSH 플래그를 설정함
    - 수신 TCP는 해당 데이터를 가능한 한 빨리 응용 프로그램에 전달함
  - 긴급 데이터(Urgent Data)
    - 특별한 처리가 필요한 데이터에 URG 플래그를 설정하여 전송함
    - Urgent Pointer를 이용해 긴급 데이터의 끝 위치를 나타냄

TCP 연결 및 전송

- 연결 종료 (1/4)
  - 종료 요청
    - 클라이언트 프로그램은 자신의 TCP에게 Active Close를 요청하여 서버와의 연결을 종료할 것임을 알림
    - 서버 프로그램은 자신의 TCP에게 Passive Close를 요청하여 해당 클라이언트와의 연결을 종료할 것임을 알림
  - 3단계 핸드셰이크
    - 클라이언트 TCP가 FIN 세그먼트를 전송함
    - 서버는 FIN에 대한 확인응답과 반대 방향의 종료를 알리기 위해 FIN + ACK 세그먼트를 전송함
    - 클라이언트는 서버의 FIN에 대한 확인응답으로 ACK 세그먼트를 전송함

TCP 연결 및 전송

- 연결 종료 (2/4)
  - 3단계 핸드셰이크
    - 데이터를 포함하지 않는 FIN 세그먼트도 하나의 순서 번호를 소비함



TCP 연결 및 전송

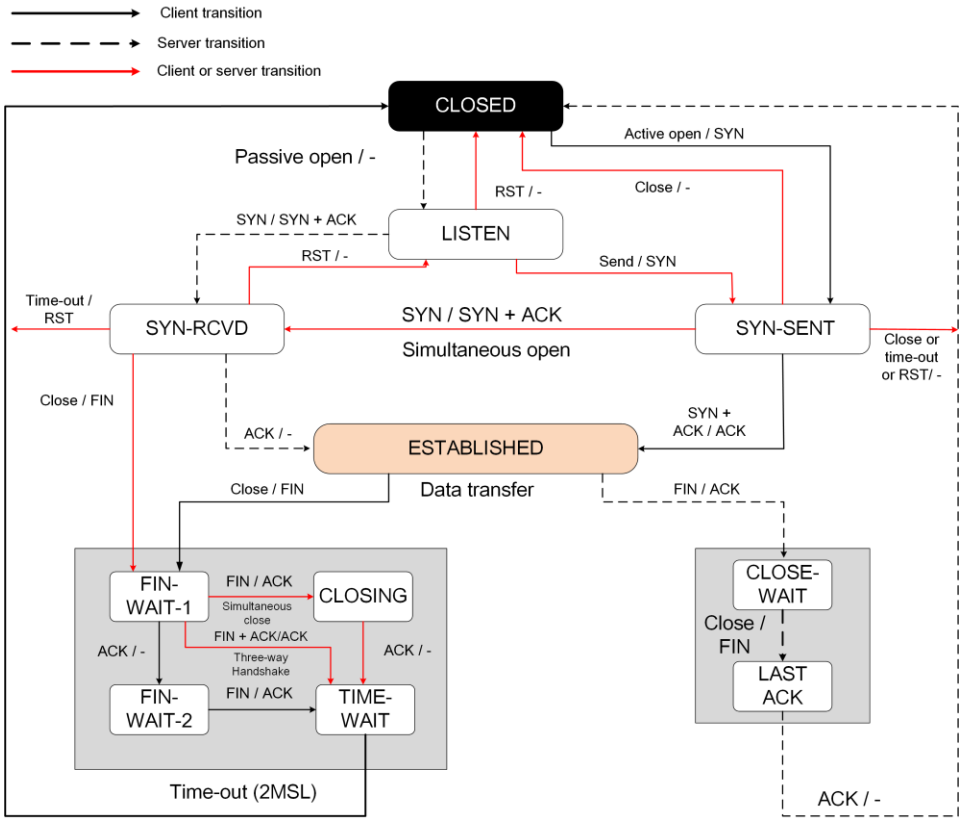
- 연결 종료 (3/4)
  - 반-닫기(Half-Close)
    - 한쪽 방향의 데이터 전송만 종료하고 반대 방향의 데이터 전송은 계속 유지함
    - 먼저 종료하는 측이 FIN을 전송하고, 상대 측은 필요한 데이터 전송을 완료한 후 FIN을 전송하여 연결을 완전히 종료함

TCP 연결 및 전송

- 연결 종료 (4/4)
  - 연결 리셋(Connection Reset)
    - 연결 거절(Denying a Connection)
      - 존재하지 않는 포트 등에 연결 요청이 들어온 경우 RST 세그먼트를 보내 연결 요청을 거절함
    - 연결 중단(Aborting a Connection)
      - 비정상적인 상황이 발생하면 RST 세그먼트를 보내 기존 연결을 즉시 중단함
    - 휴지 연결의 종료(Terminating an Idle Connection)
      - 상대 TCP가 오랫동안 동작하지 않는 상태로 유지되면 RST 세그먼트를 보내 연결을 종료함

TCP 연결 및 전송

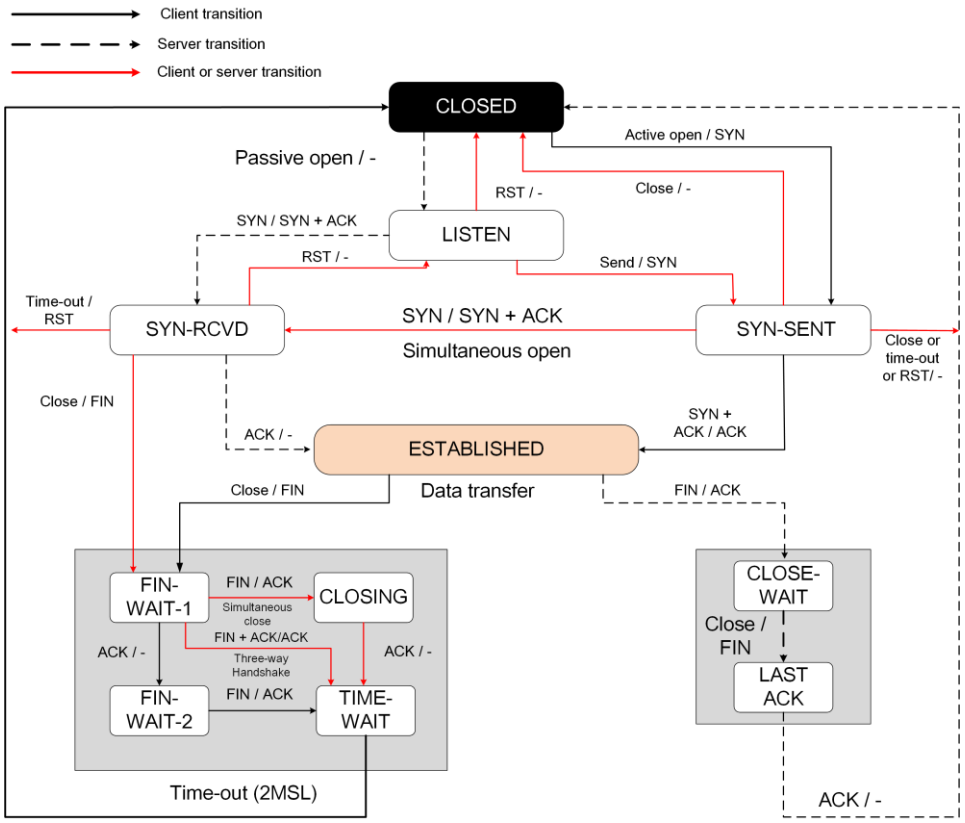
- 상태 전이도(State Transition Diagram) (1/2)
  - 정의
    - TCP의 연결 설정, 데이터 전송, 연결 종료 과정에서 발생하는 상태와 이벤트에 따른 상태 변화를 나타낸 유한 상태 머신



TCP 연결 및 전송

- 상태 전이도(State Transition Diagram) (2/2)
  - 표현
    - 도형 내 단어는 상태를 나타냄

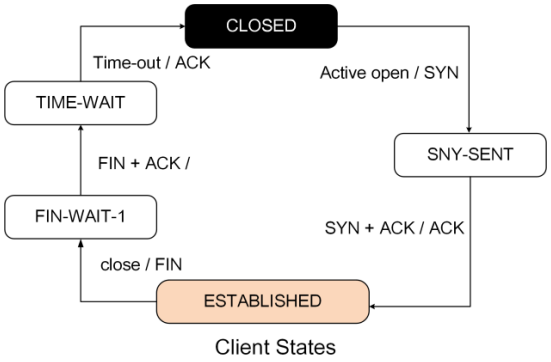
**최대 세그먼트 수명(MSL, Maximum Segment Lifetime)
네트워크 상에서 TCP 세그먼트가 살아남을 수 있는 최대 시간

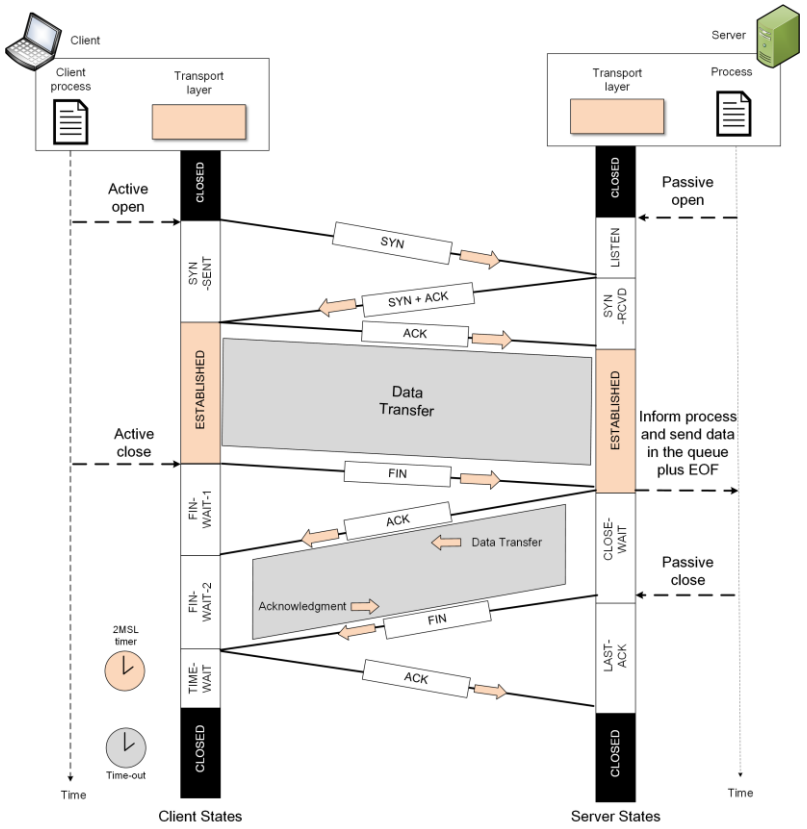


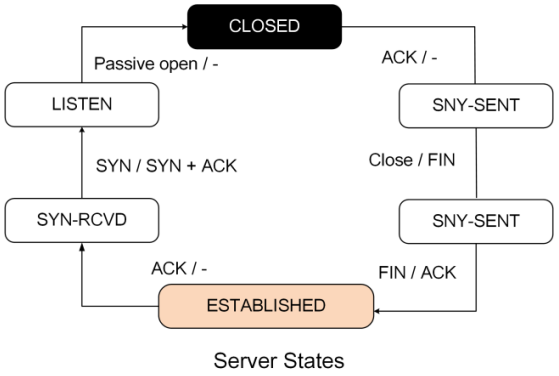
상태	설명
CLOSED	연결이 존재하지 않음
LISTEN	수동 개방 요청을 수신하고 SYN을 기다리고 있음
SYN-SENT	SYN을 전송하고 ACK를 기다리고 있음
SYN-RCVD	SYN+ACK을 전송하고 ACK를 기다리고 있음
ESTABLISHED	연결이 수립되고 데이터 전송 과정을 진행함
FIN-WAIT-1	첫 번째 FIN을 전송하고 ACK를 기다리고 있음
FIN-WAIT-2	첫 번째 FIN에 대한 ACK를 수신하고 두 번째 FIN을 기다리고 있음
CLOSE-WAIT	첫 번째 FIN을 수신받고 ACK를 전송한 후, 연결 종료를 기다리고 있음
TIME-WAIT	두 번째 FIN을 수신받고 ACK를 전송한 후, 2MSL 타임아웃을 기다리고 있음
LAST-ACK	두 번째 FIN을 전송하고 ACK를 기다리고 있음
CLOSING	양측이 동시에 닫기로 결정함

TCP 연결 및 전송

- 연결 및 전송 시나리오 (1/6)
  - 연결 설정 및 반-닫기
    - 한쪽이 먼저 데이터 전송을 종료하지만, 반대 방향의 데이터 전송은 계속 유지해야 하는 경우 발생함

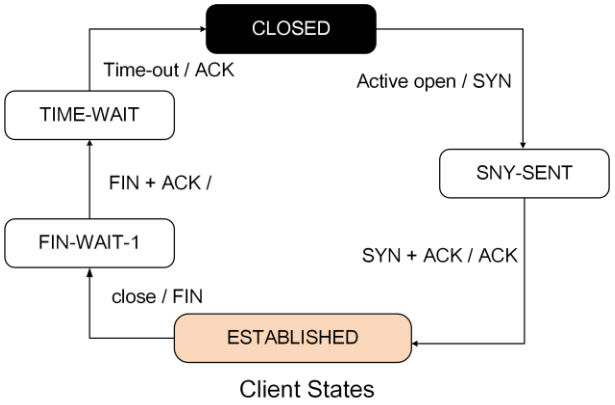


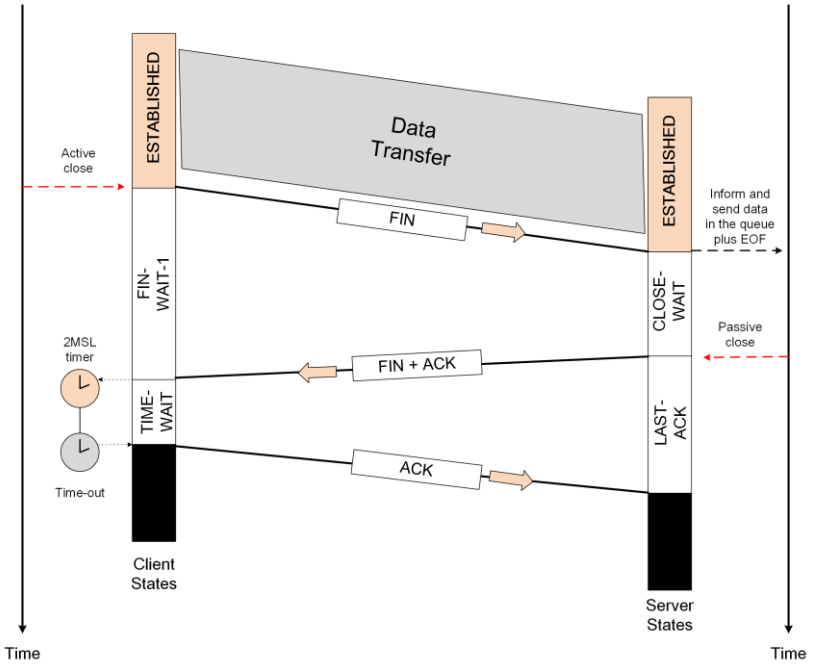


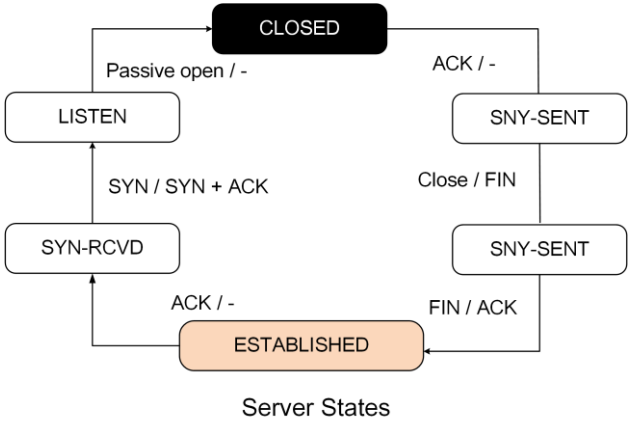


TCP 연결 및 전송

- 연결 및 전송 시나리오 (2/6)
  - 연결 설정 및 3단계 핸드셰이크를 이용한 연결 종료
    - 한쪽의 FIN에 대해 상대 측이 자신의 FIN과 ACK를 함께 전송하는 경우 발생함

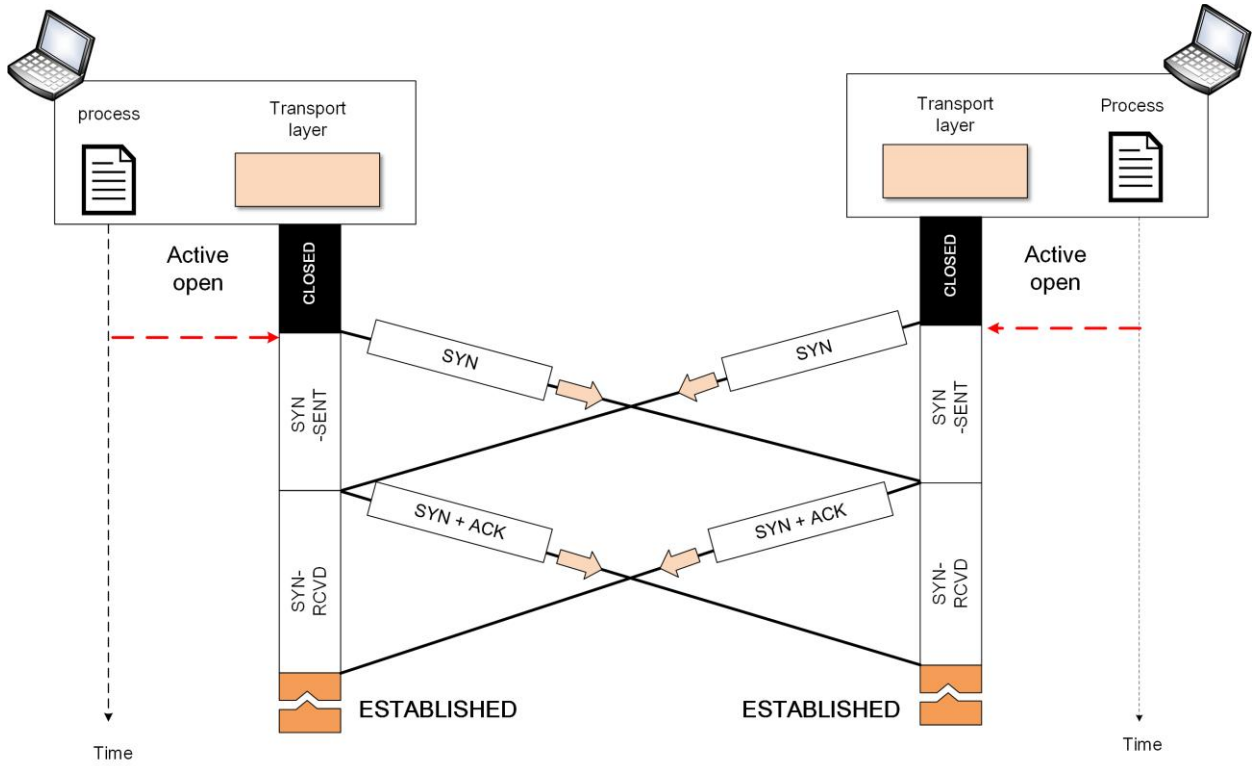






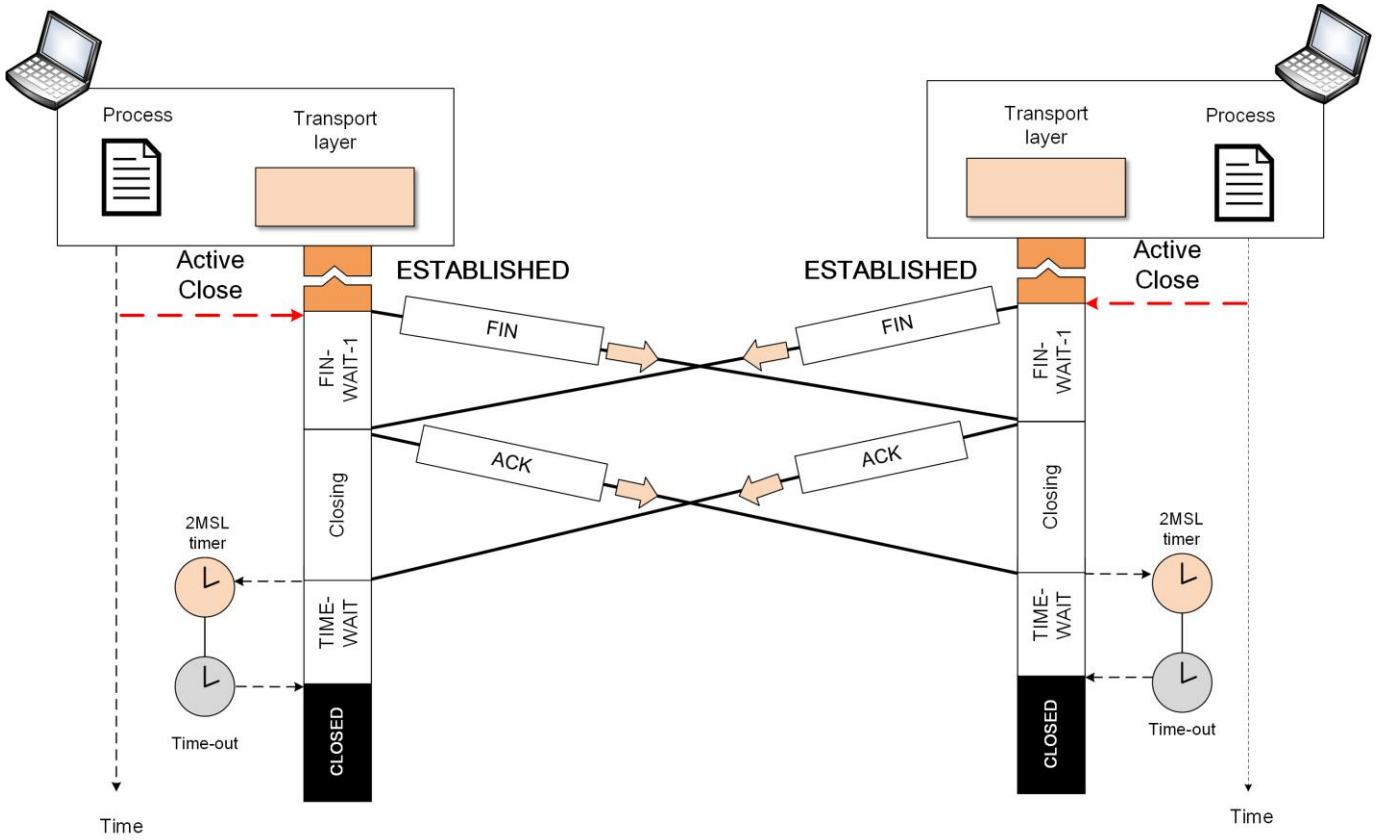
TCP 연결 및 전송

- 연결 및 전송 시나리오 (3/6)
  - 동시 개방
    - 두 호스트가 거의 동시에 Active Open을 수행하는 경우 발생함



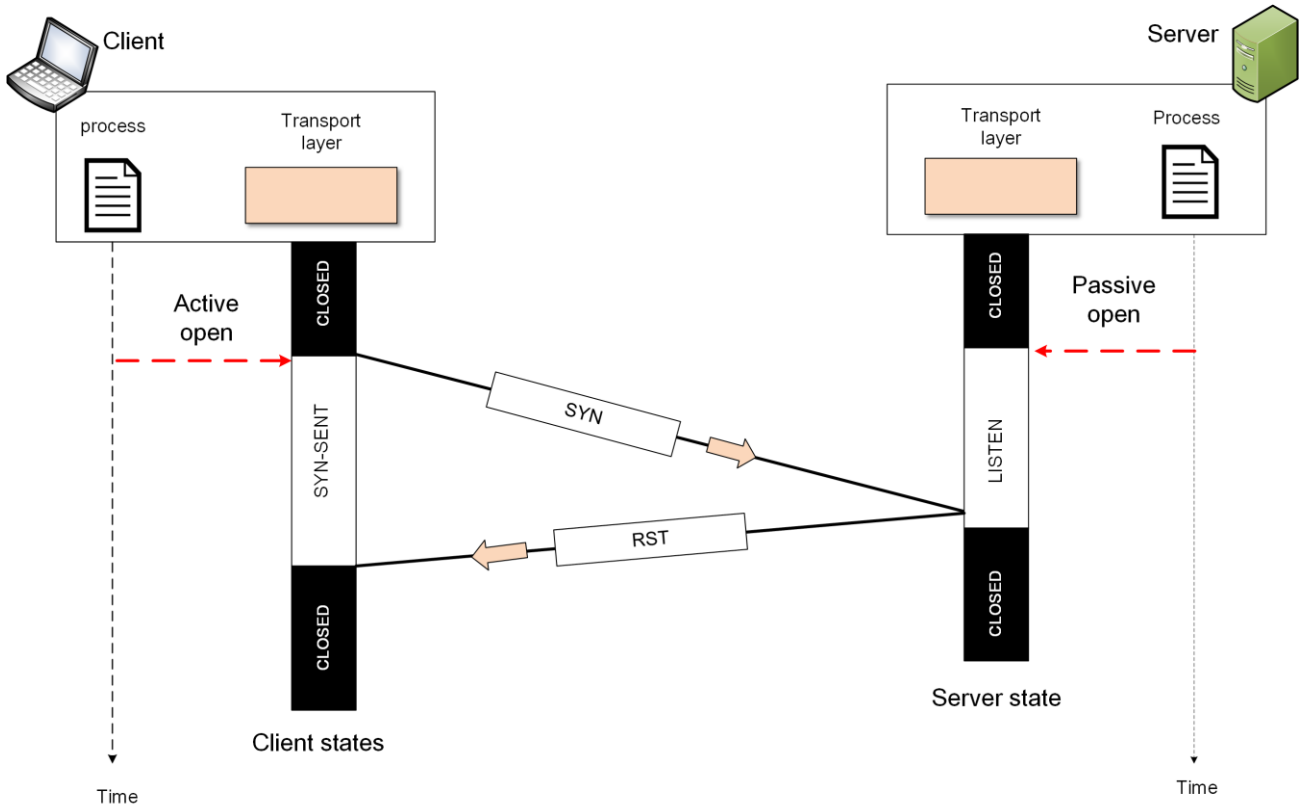
TCP 연결 및 전송

- 연결 및 전송 시나리오 (4/6)
  - 동시 닫기
    - 두 호스트가 동시에 연결 종료를 시도하는 경우 발생함



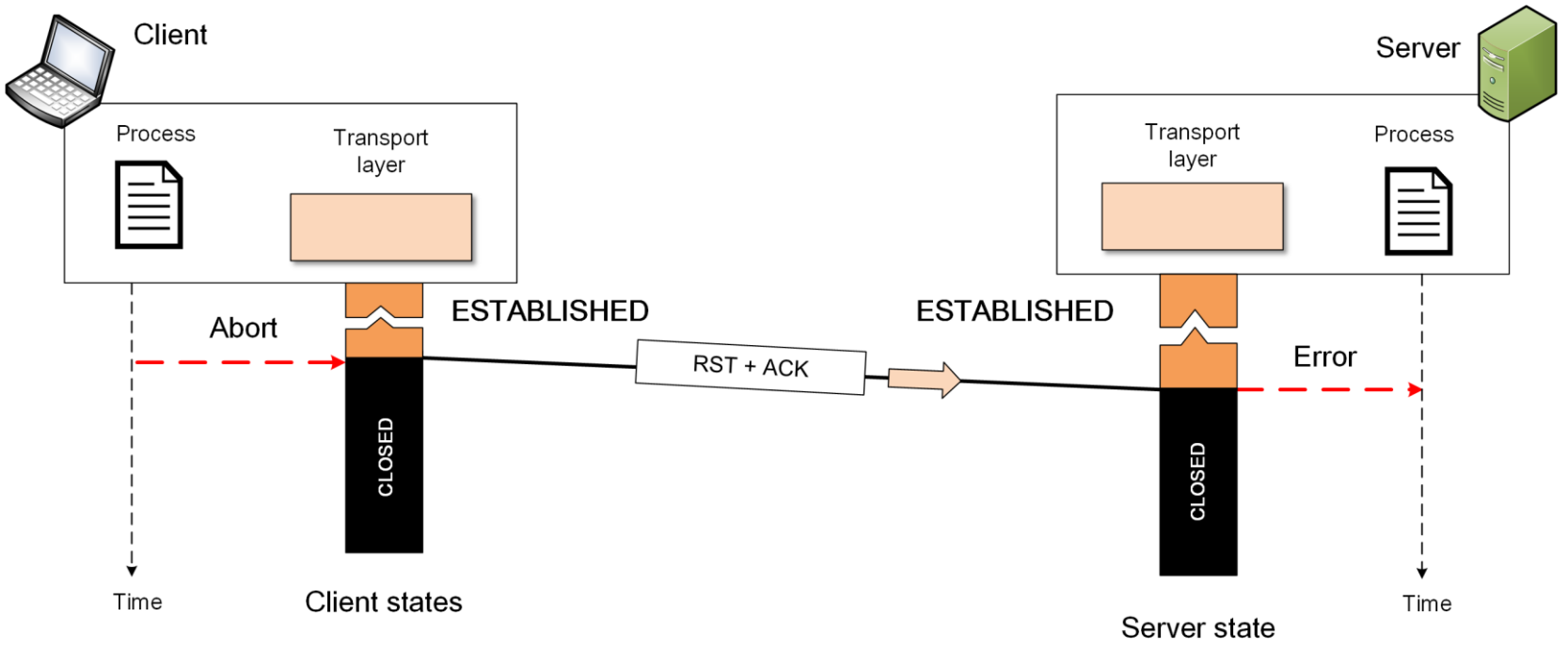
TCP 연결 및 전송

- 연결 및 전송 시나리오 (5/6)
  - 연결 거절
    - 서버 TCP가 클라이언트의 연결 요청을 받아들이지 않는 경우 발생함



TCP 연결 및 전송

- 연결 및 전송 시나리오 (6/6)
  - 연결 중단
    - 루프로 인한 프로세스 고장이나 잘못된 연결의 세그먼트 수신 등 비정상적인 상황에서 발생함



목 차

- TCP 개요
- TCP 연결 및 전송
- TCP 제어
- TCP 동작 지원 및 구현

TCP 연결 및 전송

- TCP 윈도우 (1/2)
  - 정의
    - 송·수신 측 버퍼에서 한 번에 전송하거나 수신할 수 있는 데이터의 범위
  - 특징
    - 클라이언트와 서버가 각각 송신 윈도우와 수신 윈도우를 가짐
    - 세그먼트 단위가 아니라 바이트 단위로 동작함
    - 흐름 제어와 혼잡 제어의 핵심 메커니즘으로 사용됨

TCP 연결 및 전송

- TCP 윈도우 (2/2)
  - 송신 윈도우
    - 송신 측이 현재 전송할 수 있는 바이트의 범위를 나타냄
    - 흐름 제어에 따라 윈도우가 열리거나(Open), 닫히거나(Close), 축소(Shrink)될 수 있음
    - 전송한 데이터의 타임아웃을 확인하기 위해 하나의 타이머를 사용함
  - 수신 윈도우
    - 수신 측이 현재 추가로 받을 수 있는 바이트의 범위를 나타냄
    - 송신 측의 혼란을 방지하기 위해 윈도우 축소는 허용하지 않음
    - 수신 버퍼가 넘치지 않도록, 응용 프로세스가 아직 읽지 않은 데이터 양을 고려하여 크기가 결정됨

Thanks!

김 민 식(minsik@pel.sejong.ac.kr)