2021/02/12, 2021 보안 기초 세미나

TCP/IP 완벽 가이드

- 2-8부 TCP/IP 전송 제어 프로토콜 -

박 재 형(jaehyoung@pel.sejong.ac.kr)

세종대학교 프로토콜공학연구실

목 차

TCP 개요

- TCP 개요
  - 이전에 NCP(Network Control Protocol)을 사용하였지만, 인터네트워크의 성장으로 새로운 프로토콜 개발 필요
  - 1974년 12월, TCP는 PFC 675 문서에 “Transmission Control Program”으로 공식화됨
  - 1981년 9월, PFC 793 문서에 “Transmission Control Protocol”으로 정의됨
    - 지속적인 개선으로 버전 4로 명명됨
- TCP(Transmission Control Protocol) 정의
  - 서버와 클라이언트 간의 데이터를 신뢰성 있게 전달하기 위한 전송 계층 프로토콜

TCP 개요

- 특징

특징	설명
연결형	통신하기전에 연결을 수립
양방향	데이터를 양방향으로 송·수신
다중 연결과 종단 식별	연결된 두 장비가 사용하는 소켓 쌍은 TCP 연결의 종단을 식별
신뢰성	데이터가 목적지에 도착할 수 있도록 송·수신한 데이터를 추적
승인	데이터 전송에 대한 승인 메시지를 보냄으로써, 신뢰성 제공
스트림 기반 전송	애플리케이션이 블록 형태의 데이터가 아닌, 연속적인 데이터 스트림을 송신할 수 있도록 함

TCP 개요

- TCP 기능(1/2)
  - 주소지정 및 다중화
    - 포트 번호를 통해 상위계층 애플리케이션 식별
    - 여러 프로세스에서 온 데이터를 다중화하여, 네트워크 계층으로 송신
  - 연결 수립, 유지, 종료
    - 데이터를 이동 시키기 위해 연결을 협상
  - 데이터 처리와 패키징
    - 데이터를 패키징하여 목적지 TCP 소프트웨어로 전송
    - 목적지 TCP 소프트웨어는 패키징을 풀고 애플리케이션에 전송

TCP 개요

- TCP 기능(2/2)
  - 신뢰성과 전송품질 서비스
    - TCP 애플리케이션으로 송신된 데이터가 목적지에 도착하지 않거나 잘못된 순서로 도착하는 것을 회피함
  - 흐름제어와 혼잡 회피
    - TCP는 두 장비 간의 데이터의 흐름을 관리
      - 수신측 보다 송신측의 데이터 처리 속도가 빨라 데이터 손실이 발생하는 것을 회피
    - 통신에서 발생하는 혼잡을 처리
      - 송신자가 네트워크가 감당하기에 너무 많은 패킷을 빠르게 보낼때, 발생하는 데이터 손실이나 오버플로우 처리

TCP 개요

- TCP가 수행하지 않는 기능
  - 애플리케이션 사용 명시
    - TCP는 전송 프로토콜만 정의함
    - TCP 사용 방식은 애플리케이션 프로토콜이 결정
  - 보안 제공
    - TCP는 인증이나 프라이버시를 보장하지 않음
    - IPsec과 같은 보안 프로토콜 사용
  - 통신 보장
    - TCP는 재전송을 시도할 뿐, 모든 데이터에 대한 흐름 문제를 해결할 수 없음

TCP 개요

- 견고성의 원칙
  - TCP는 견고성의 원칙을 따른다고 표준에서 설명함

1. TCP 구현은 다른 장비의 TCP 계층에 문제를 일으킬 수 있는 일을 하지 않도록 노력해야함

2. TCP 구현은 다른 TCP 구현이 만들 수 있는 문제를 예상하고 이를 해결하려고 노력해야함

3. TCP 동작에 있어 비정상적인 상황에 대한 추가적인 보호를 제공할 수 있어야함

TCP 원리와 일반 동작

- TCP 데이터 취급과 처리
  - TCP 스트림 기반 동작
    - TCP는 애플리케이션에서 오는 데이터를 스트림으로 간주하여 데이터를 처리함
  - TCP 데이터 패키징
    - TCP는 애플리케이션에서 오는 스트림을 네트워크 계층(IP)으로 보내기 위해 분리된 메시지로 나눔
      - IP는 메시지 중심 프로토콜
      - 분리된 메시지 = 세그먼트
        - 전송계층(4계층)에서 사용되는 메시지
    - 최대 세그먼트 크기(MSS, Maximum Segment Size) 협상
      - 불필요한 단편화를 방지하기 위해 세그먼트에 대한 크기를 제한함
      - 연결 수립 과정에서 결정됨

TCP 원리와 일반 동작

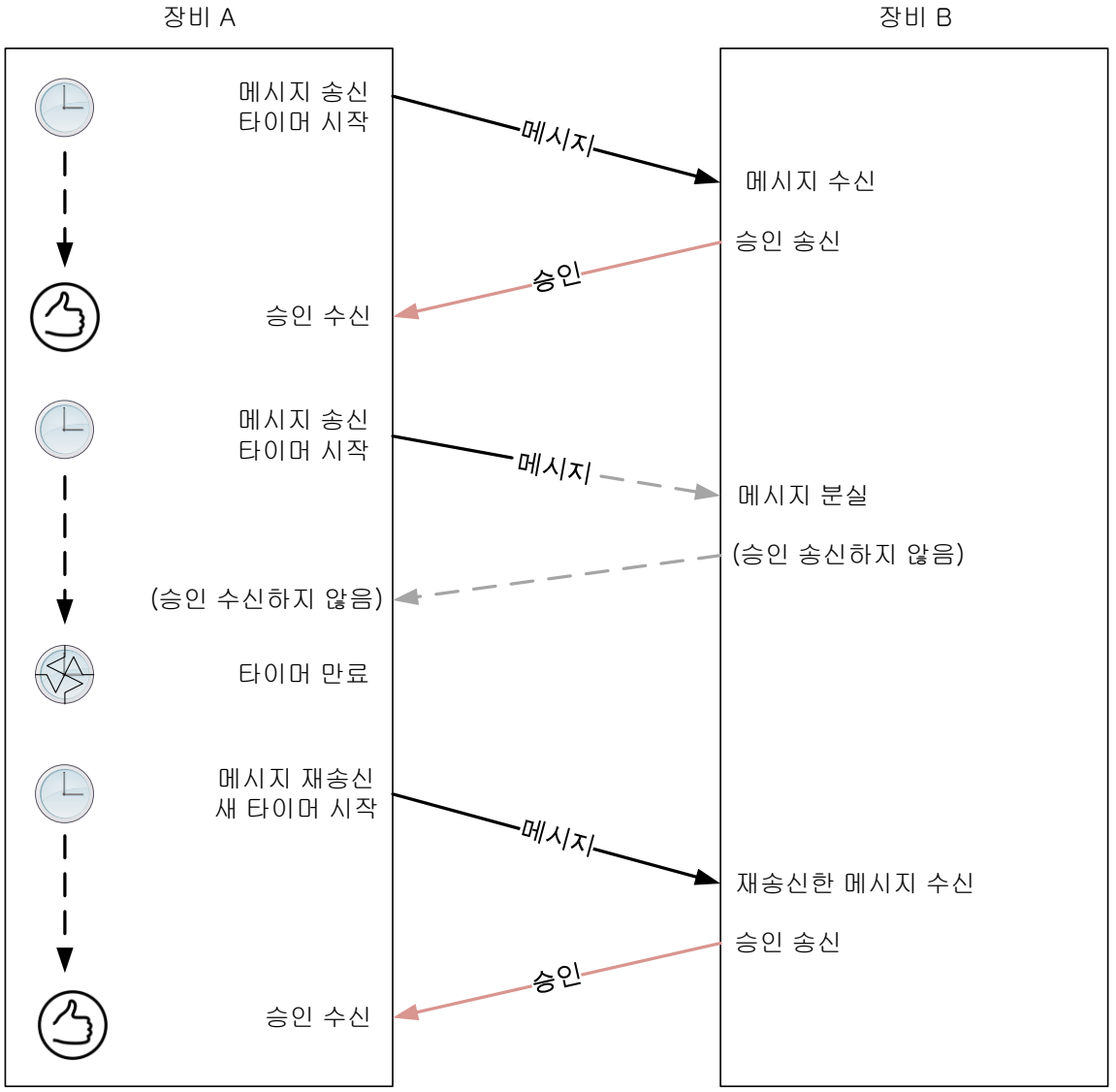
- TCP 데이터 취급과 처리
  - TCP 데이터 식별, 순서 번호
    - 애플리케이션에서 오는 데이터를 옥텟 단위 스트림으로 간주하여 데이터 식별
      - 스트림 기반이므로 모든 바이트에 대한 데이터 식별 필요
    - 데이터에 순서 번호를 할당하여 송신, 수신, 승인 과정에서 데이터 추적
    - 수신 장비는 순서 번호를 통해 세그먼트를 원본 데이터 스트림으로 재조합
  - TCP 애플리케이션의 데이터 구분
    - 애플리케이션은 TCP로 부터 구조화 되지 않은 데이터 스트림을 수신 받음
    - 데이터 구분을 위해 애플리케이션에서 직접 구현 되어야함

TCP 원리와 일반 동작

- TCP 슬라이딩 윈도우
  - 신뢰성과 데이터 흐름 제어를 제공하기 위해 사용되는 승인 체계
  - 프로토콜을 신뢰하기 위해서는 데이터에 대한 피드백 필요
  - 재전송을 사용하는 긍정 승인 (PAR, Positive Acknowledgment with Retransmission)
    - 승인 체계를 이용하여 피드백 제공
      - 전송에 대한 응답이 오기까지 특정 타이머 구동
      - 타이머가 만료되면 재전송을 함

TCP 원리와 일반 동작

- TCP 슬라이딩 윈도우 승인 체계
  - PAR 동작 과정

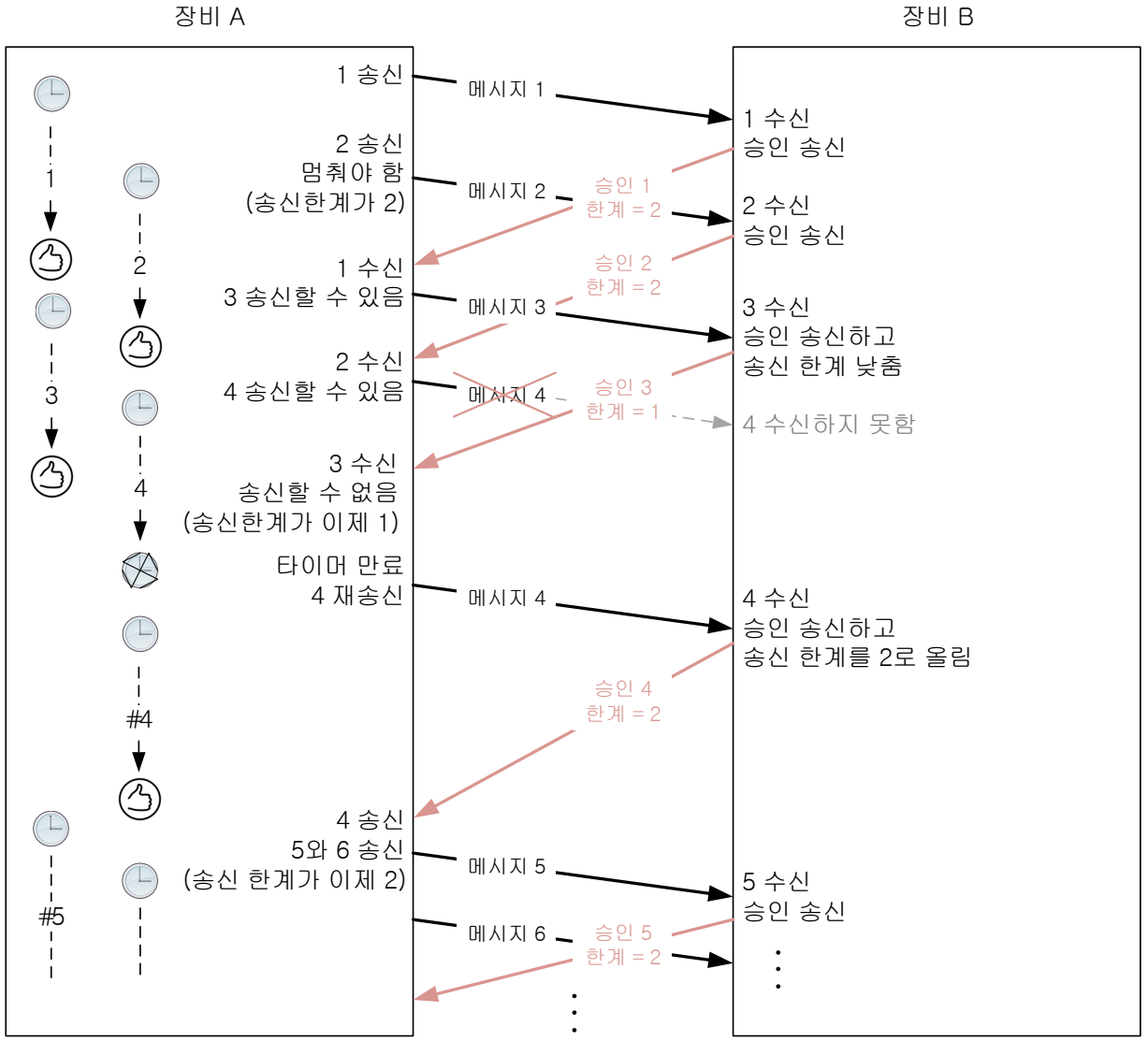


TCP 원리와 일반 동작

- TCP 슬라이딩 윈도우 승인 체계
  - PAR 개선
    - 기존 PAR 속도 문제 개선
      - 승인 받기 전까지 메시지를 보내지 못함
    - 메시지와 승인을 식별하기 위한 방법 제공
      - 순서번호를 사용하여 메시지를 식별하고 개별 승인
    - 승인 메시지에 송신 메시지 제한 인자 추가
      - 데이터 흐름제어 가능
    - TCP는 바이트를 개별적으로 보내지 않고 세그먼트로 묶어서 보냄
      - 속도 저하 문제 개선

TCP 원리와 일반 동작

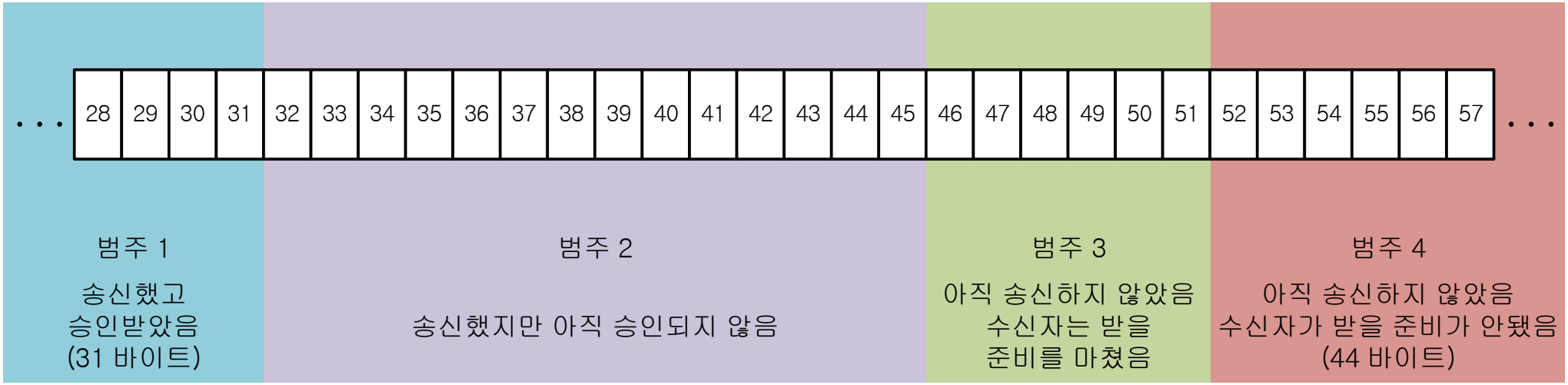
- TCP 슬라이딩 윈도우 승인 체계
  - 개선된 PAR 동작과정



TCP 원리와 일반 동작

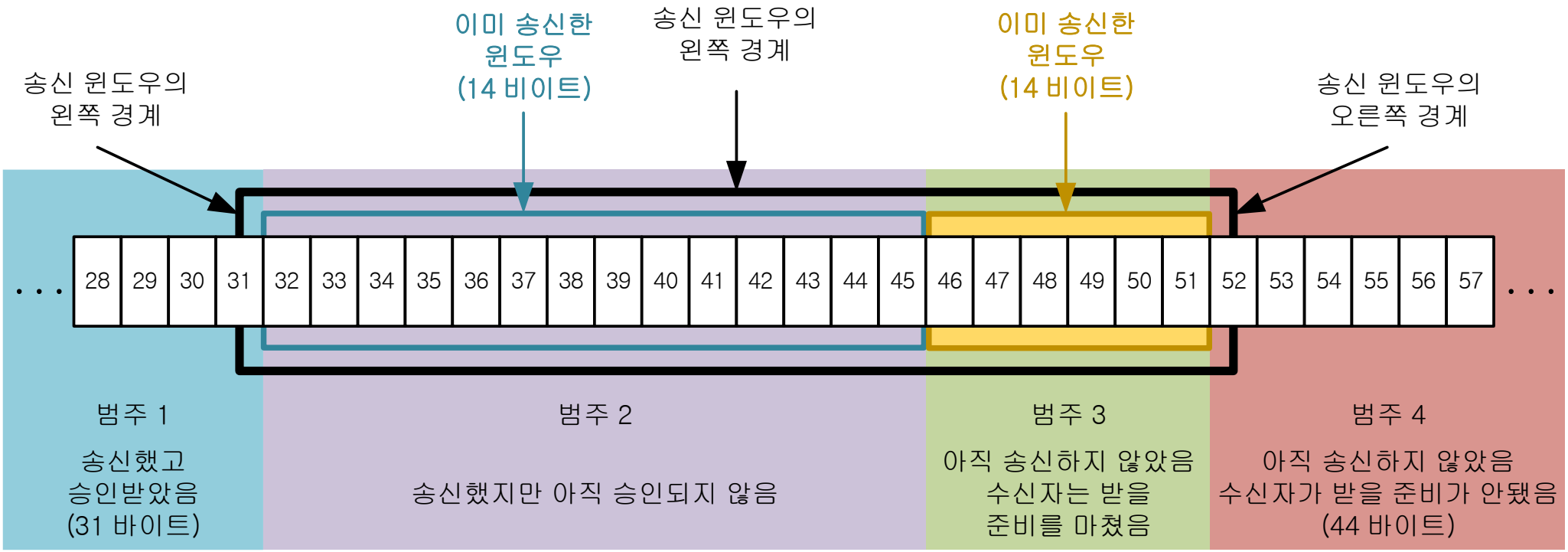
- TCP 슬라이딩 윈도우 승인 체계
  - 스트림 중심 슬라이딩 윈도우 승인 체계
    - TCP 전송 스트림의 개념적 구분

구분	설명
범주 1	스트림의 맨 처음 부분으로 이미 송신되어 승인 받은 바이트
범주 2	장비가 송신 했지만 아직 승인을 받지 못한 바이트
범주 3	송신 장비가 아직 보내지 않았지만, 최근 통신 과정을 보았을 때 수신장비가 충분히 받을 수 있는 바이트
범주 4	스트림의 맨 마지막 부분으로 수신 장비가 아직 받을 준비가 되지 않아 송신 장비도 보내지 않는 바이트



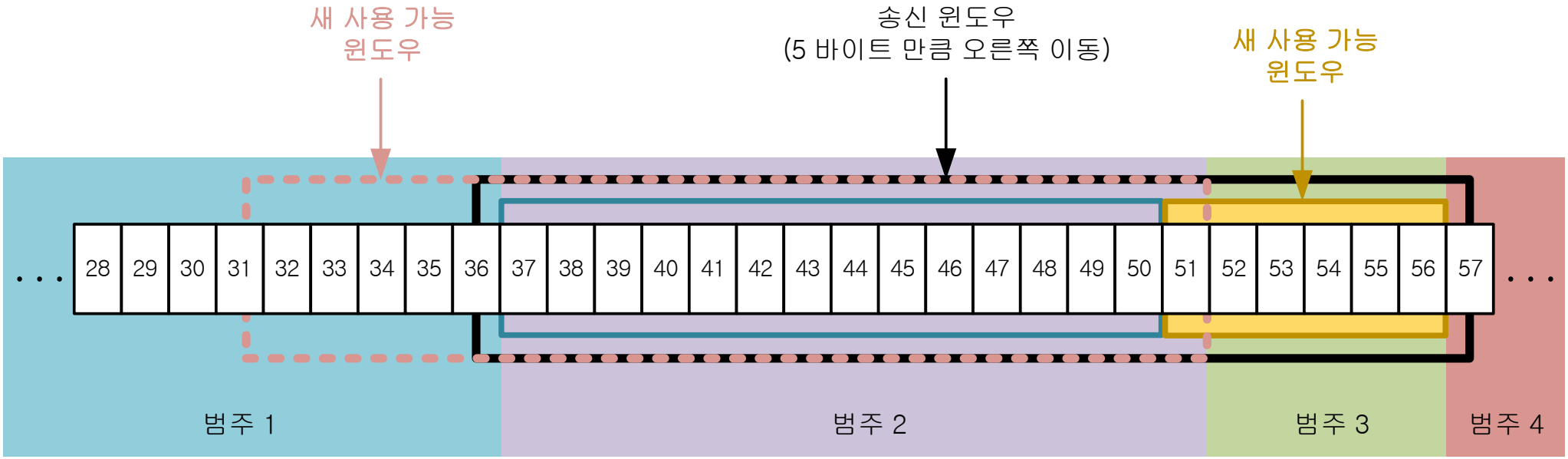
TCP 원리와 일반 동작

- TCP 슬라이딩 윈도우 승인 체계
  - 스트림 중심 슬라이딩 윈도우 승인 체계
    - TCP 전송 스트림 범주와 송신 윈도우 용어



TCP 원리와 일반 동작

- TCP 슬라이딩 윈도우 승인 체계
  - 스트림 중심 슬라이딩 윈도우 승인 체계
    - TCP 송신 윈도우



TCP 원리와 일반 동작

- TCP 슬라이딩 윈도우 승인 체계
  - 스트림 중심 슬라이딩 윈도우 승인 체계
    - 빠진 승인 처리
      - 세그먼트 각각에 대한 승인이 아니기 때문에 중간에 승인이 빠질 수 있음
      - 이미 수신 받은 세그먼트도 재전송 해야 할 수 있음

TCP 원리와 일반 동작

- TCP 포트, 연결, 연결 식별
  - TCP는 데이터를 프로세스로 보내며, 연결에 속한 데이터 중복이나 혼란을 관리함
  - TCP는 연결을 식별하기 위해 두 종단에 해당하는 소켓 쌍 이용
    - 소켓은 각 프로세스의 IP 주소와 포트 조합을 의미
    - 소켓 쌍은 출발지 주소, 출발지 포트, 목적지 주소, 목적지 포트 정보를 포함

TCP 연결 수립, 관리 종료

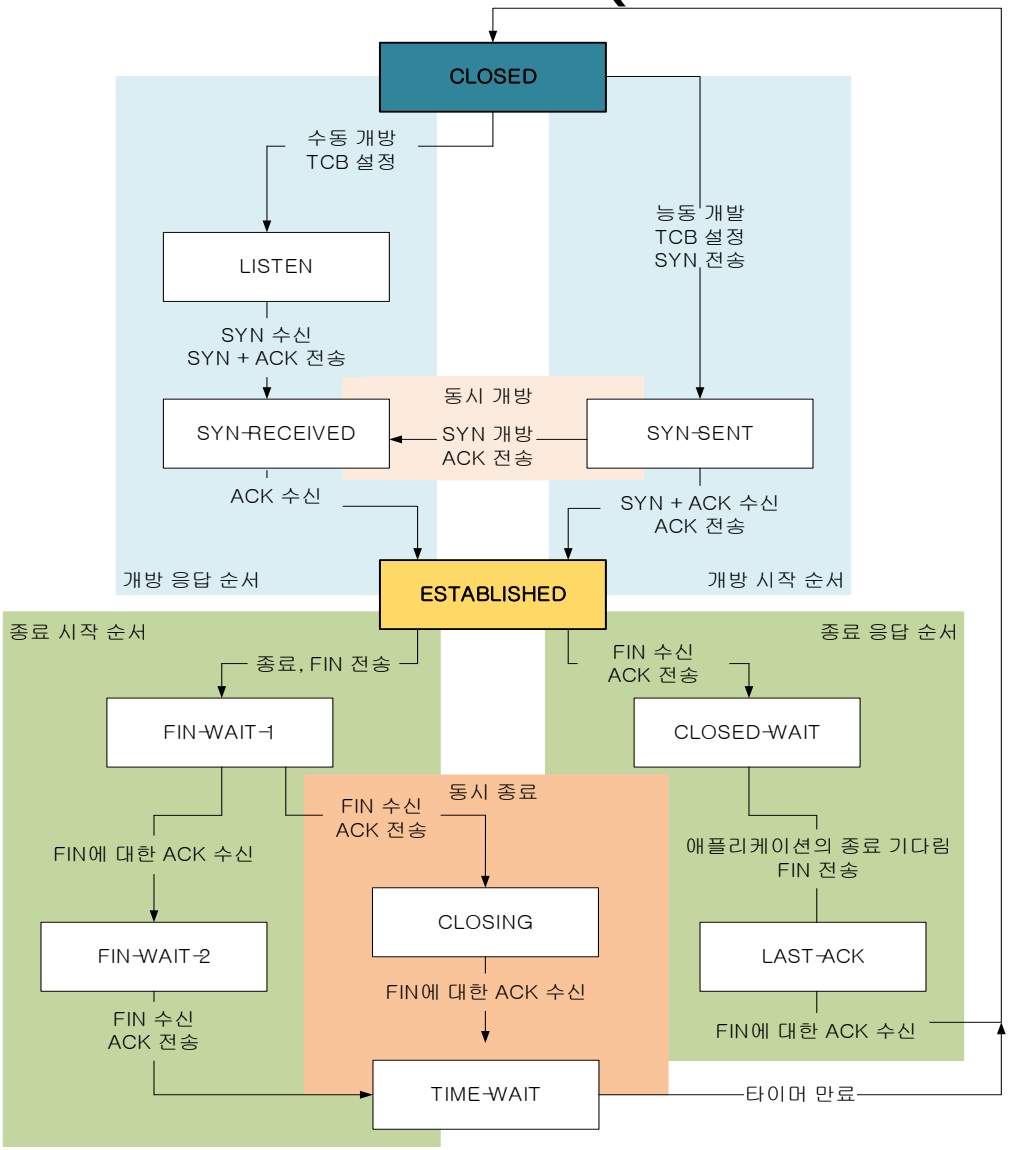
- TCP 동작과 유한 상태 머신(FSM, Finite State Machine)
  - 기본 FSM 개념
    - 상태
      - 특정 시간에 프로토콜 소프트웨어 상황
    - 전이
      - 한 상태에서 다른 상태로 움직이는 행위
    - 이벤트
      - 상태 간에 전이하게 만든 어떤 일
    - 행동
      - 장비가 이벤트에 대한 반응으로 다른 상태로 전이하기 전에 하는 일

TCP 연결 수립, 관리 종료

- TCP 동작과 유한 상태 머신(FSM, Finite State Machine)
  - 주요 이벤트
    - SYN(Synchronize) 메시지
      - 연결을 초기화하고 수립함
      - 장비 간의 순서 번호를 동기화 시킴
    - FIN(Finish) 메시지
      - TCP 세그먼트에 FIN 비트가 설정된 메시지
      - 장비가 연결을 종료하고 싶다는 것을 알림
    - ACK(Acknowledgment) 메시지
      - SYN이나 FIN과 같은 메시지를 받았다는 것을 알림

TCP 연결 수립, 관리 종료

- TCP 동작과 유한 상태 머신(FSM, Finite State Machine)
  - TCP FSM 동작 과정



TCP 연결 수립, 관리 종료

- TCP 연결 준비
  - TCP/IP는 연결 수립 시, 클라이언트와 서버는 개방 동작을 실행하여 연결 준비
    - 클라이언트와 서버는 연결에 대한 TCB 생성
    - 서버는 소켓 번호를 명시하지 않은 TCB를 생성하고 클라이언트의 능동 개방을 기다림
    - TCB는 연결이 완전히 종료되고 CLOSED 상태가 되면 없어짐
  - 전송 제어 블록(TCB, Transmission Control Block)
    - 연결 정보 저장
      - 연결 식별을 위한 두 소켓 번호
      - 송/수신 데이터를 가지고 있는 버퍼 포인트
      - 승인에 대한 정보, 승인하지 못한 순서 번호, 현재 위도우 크기 등을 추적하는 변수 저장

TCP 연결 수립, 관리 종료

- TCP 연결 준비
  - 능동과 수동 개방
    - 능동 개방
      - SYN 메시지를 보내 연결을 시장
    - 수동 개방
      - TCP를 사용하는 서버가 특정 클라이언트로부터 연결이 오도록 명시하거나, 모든 클라이언트의 연결을 기다림

TCP 연결 수립, 관리 종료

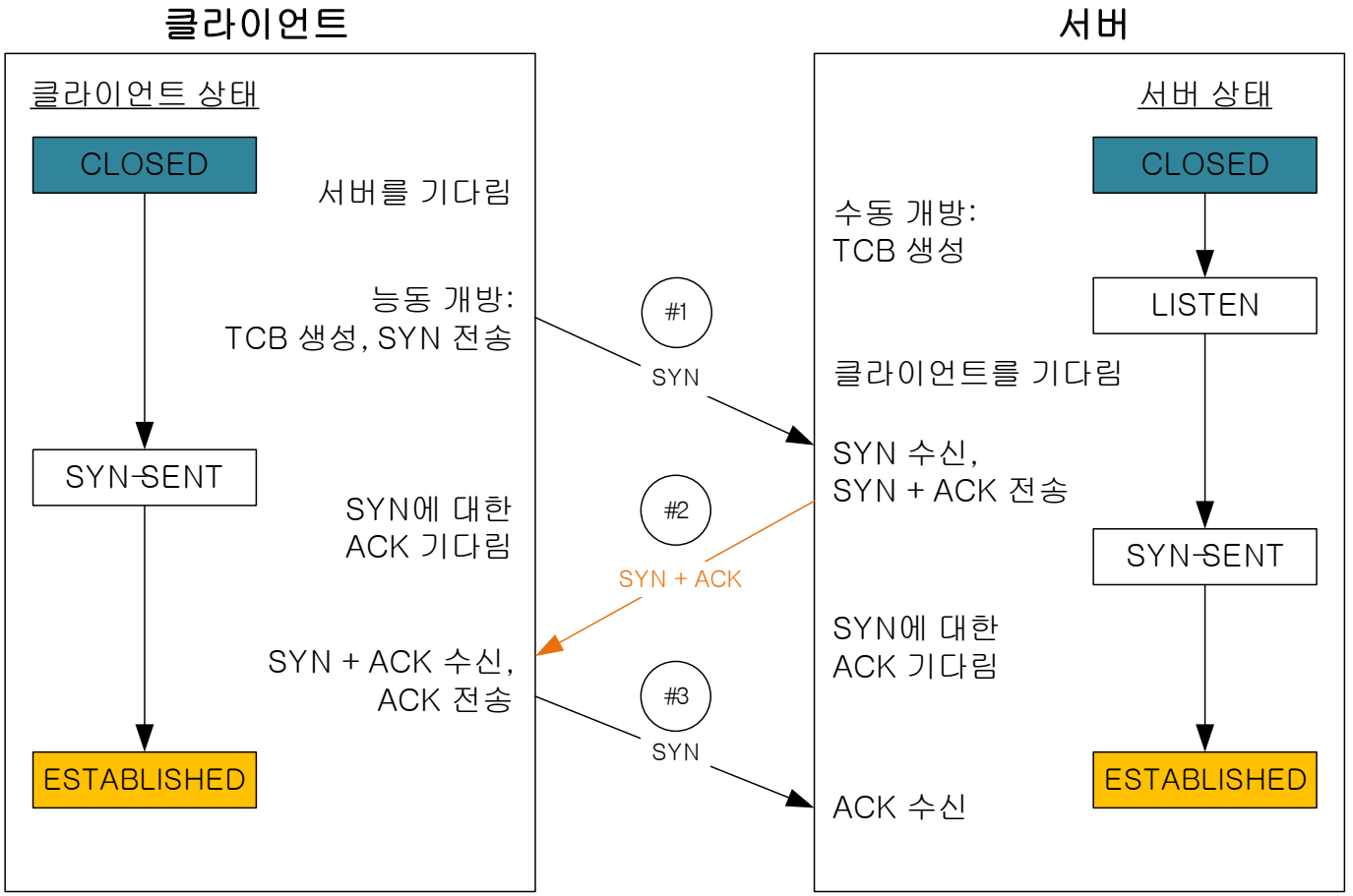
- TCP 연결 수립 과정
  - 두 장비 간의 연결이 수립되면서 초기 접속 상태(CLOSED)에서 정장 동작 상태(ESTABLISHED)로 전이함
  - 연결 수립 기능
    - 접속과 통신
      - 클라이언트와 서버는 접촉해 메시지를 전송하여 통신 시작
      - 서버는 연결을 수립할 때, 클라이언트 발견
    - 순서 번호 동기화
      - 장비는 첫 번째 통신에서 사용할 초기 순서 번호를 알림
    - 인자 교환
      - TCP 연결의 동작을 제어하기 위한 인자를 교환

TCP 연결 수립, 관리 종료

- TCP 연결 수립 과정
  - 제어 메시지
    - SYN
      - 연결을 초기화하는 데 사용하는 세그먼트라는 것을 알림
      - 연결 수립 과정에서 순서 번호를 동기화
    - ACK
      - 세그먼트를 전송하는 장비에게 메시지를 잘 받았다고 알리는 승인 메시지

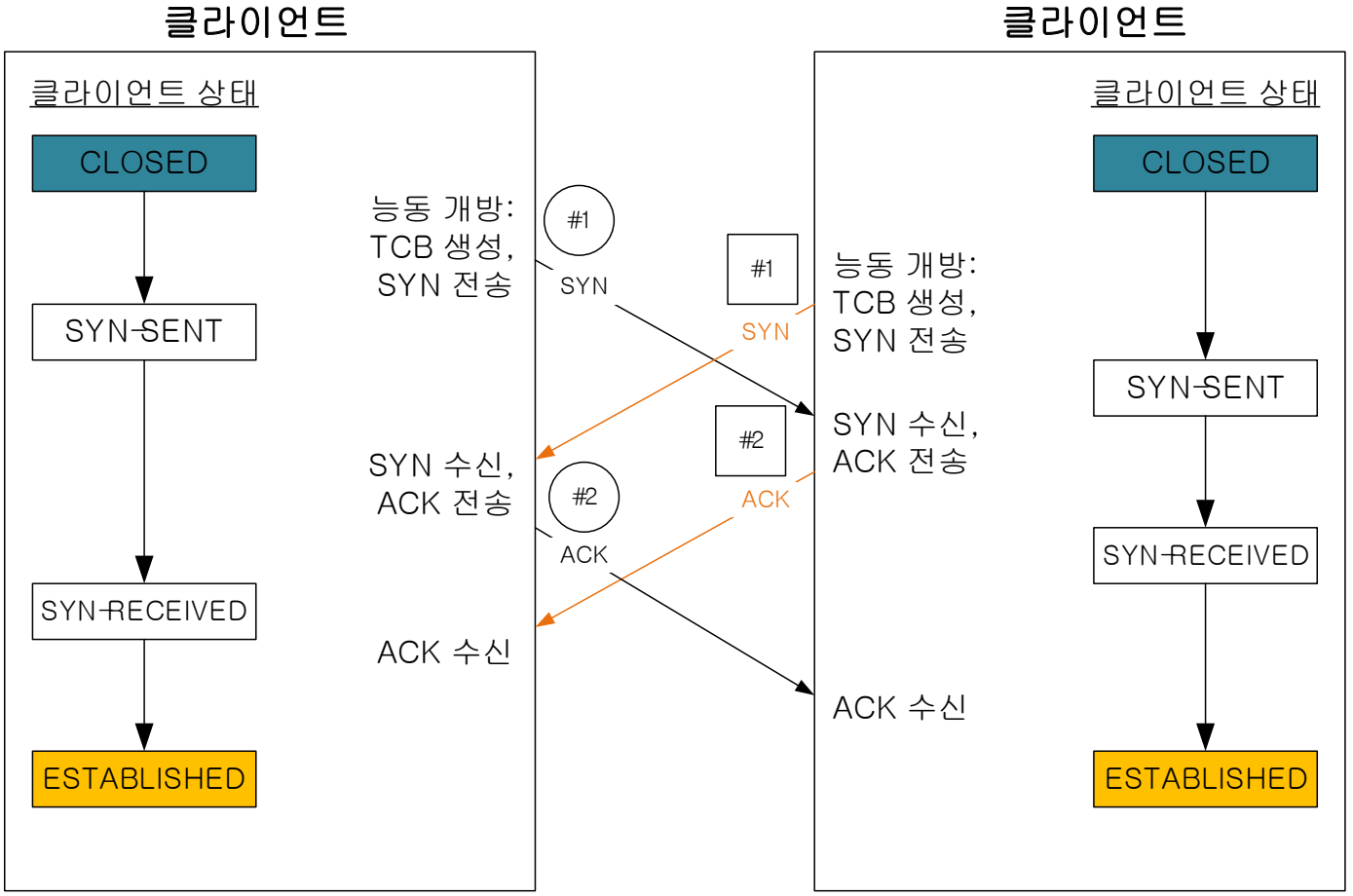
TCP 연결 수립, 관리 종료

- TCP 연결 수립 과정
  - 정상 연결 수립
    - 쓰리 웨이 핸드셰이크(Three-way handshake) 사용
      - 동작 과정



TCP 연결 수립, 관리 종료

- TCP 연결 수립 과정
  - 동시 개방 연결 수립
    - TCP는 두 장비가 동시에 능동 개방을 하는 경우도 처리 가능

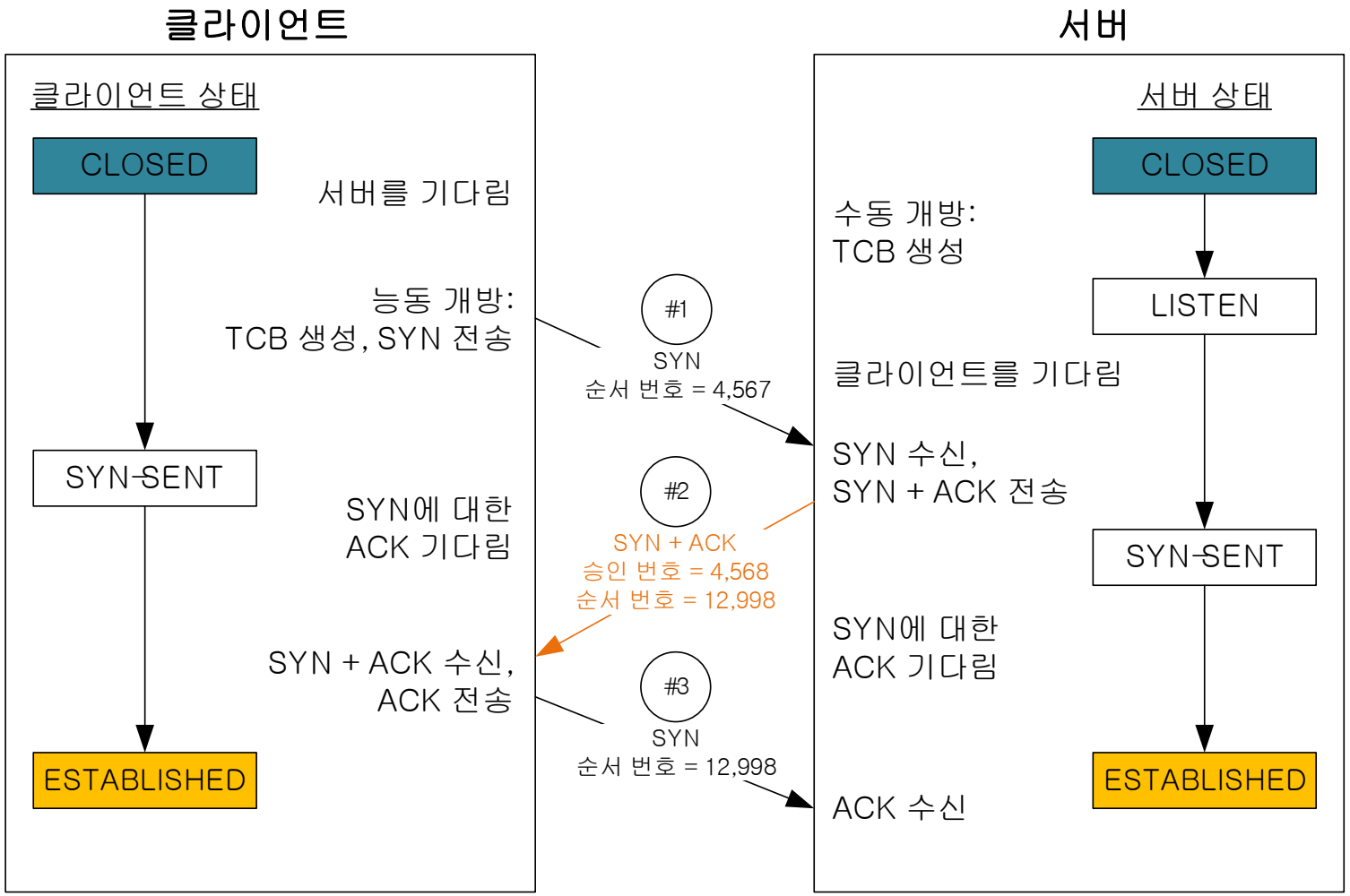


TCP 연결 수립, 관리 종료

- TCP 연결 수립 순서 번호 동기화와 인자 교환
  - TCP 초기 순서 번호
    - 다른 연결에서 온 세그먼트와 섞이지 않도록 하는 순서 번호
      - 기본적으로 초기 순서 번호(ISN, Initial Sequence Number)은 $4\mu s$마다 증가하는 카운터 사용
      - 악의적인 사람의 ISN 예측을 방지하기 위해 최근 무작위 ISN을 사용
  - TCP 인자 교환
    - 윈도우 크기 인자
      - 16bits로 주어진 윈도우 크기 보다 더 큰 값을 사용할 수 있도록 함
    - 선택적 승인 허용
      - 장비가 잃어버린 세그먼트만을 재전송할 수 있도록 함
    - 대체 체크섬 방식
      - 표준 TCP 체크섬이 아닌 체크섬 방법을 사용할 수 있음

TCP 연결 수립, 관리 종료

- TCP 연결 수립 순서 번호 동기화와 인자 교환
  - TCP 순서 번호 동기화 과정



TCP 연결 수립, 관리 종료

- TCP 연결 관리와 문제 처리
  - 슬라이딩 윈도우를 사용하면 두 장비 모두 무한정 ESTABLISHED 상태가 됨
  - ESTABLISHED 상태에서 벗어나게 되는 환경
    - 연결 종료
      - 두 장비 중 하나가 연결을 종료하기로 결정
    - 연결 방해
      - 문제가 방생하여 연결을 방해

TCP 연결 수립, 관리 종료

- TCP 연결 관리와 문제 처리
  - TCP 초기화 기능
    - 한 장비는 ESTABLISHED 상태, 다른 장비는 CLOSED 상태나 다른 임시 상태가 된, 반 개방 연결 상태를 RST 플래그로 초기화
      - RST
        - TCP의 리셋 플래그
    - 연결을 초기화 하는 상황
      - 세그먼트를 보낸 장비와 연결을 맺고 있지 않을 때
      - 잘못됐거나 부정확한 순서 번호나 승인 번호 필드를 가지는 메시지를 수신한 경우
      - 연결을 기다리는 프로세스가 없는 포트로 SYN 메시지를 받은 경우

TCP 연결 수립, 관리 종료

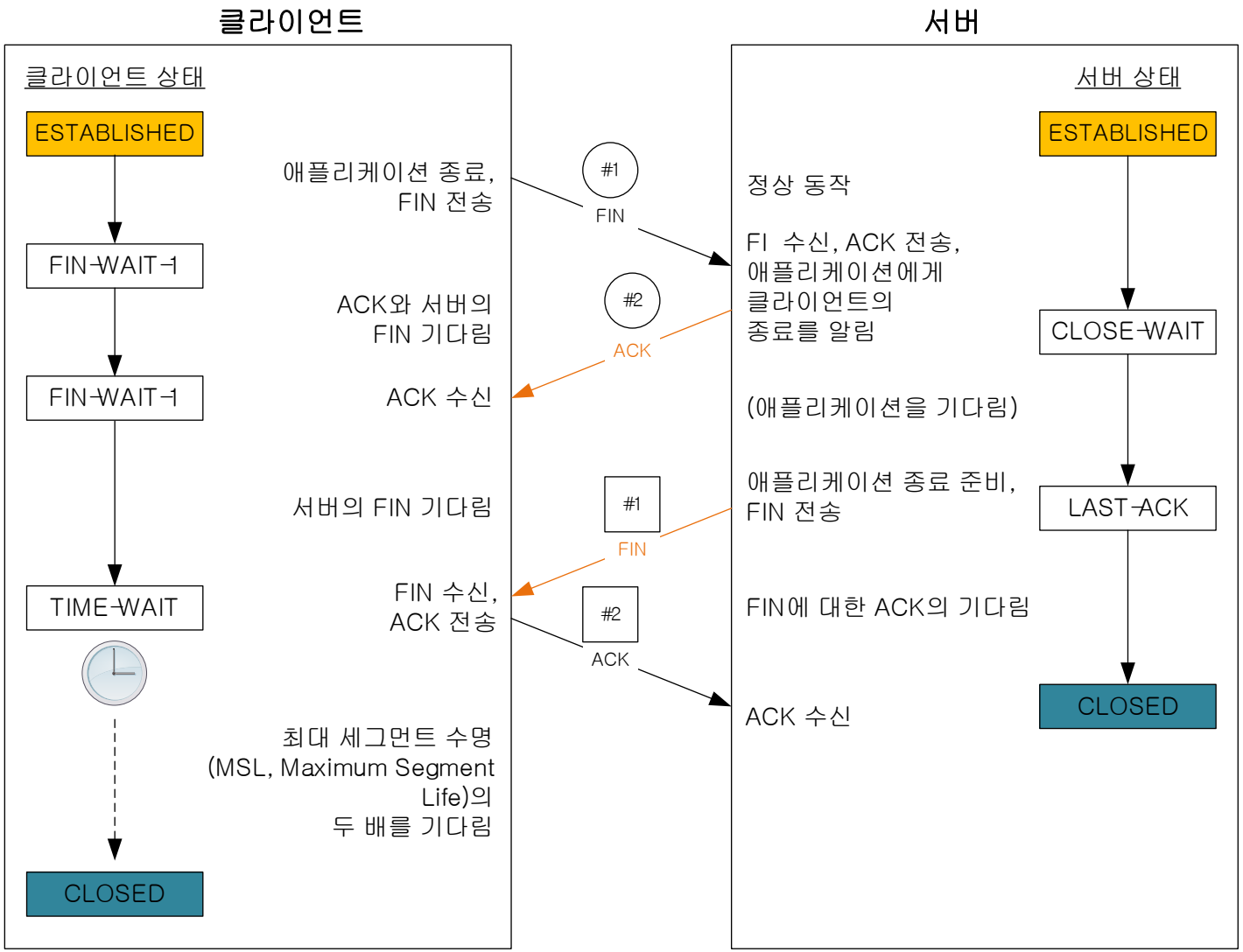
- TCP 연결 관리와 문제 처리
  - 초기화 세그먼트 처리
    - RST 비트가 설정된 세그먼트를 받으면 장비는 연결을 초기화 하여 연결을 재개방 할 수 있음
      - 장비가 LISTEN 상태에 있었다면 초기화 메시지를 무시함
      - SYN-RECEIVED 상태에 있었지만 이전에 LISTEN 상태에 있었다면 LISTEN 상태로 되돌아감
      - 이외의 상황에서 초기화 메시지를 받으면 CLOSED 상태로 돌아감
        - 상위 계층 프로세스에게 TCP의 연결이 끊김을 알림
  - 킵얼라이브 메시지
    - TCP 연결 관리 문제 중 유휴(Idle) 연결을 처리하는 방법
      - 실제 사용하지 않는 연결
    - TCP 소프트웨어에서 선택적으로 킵얼라이브 메시지를 구현할 수 있음

TCP 연결 수립, 관리 종료

- TCP 연결 종료
  - 연결 종료의 요구사항
    - 두 장비 모두 연결이 종료 되어야함
    - 연결이 종료되기 전에 보내야 할 데이터를 모두 방출해야 함
  - TIME-WAIT 상태
    - 상대 장비가 ACK를 받았다는 것을 확신하기 위해 충분한 시간을 둠
      - ACK을 분실하면 재전송함
    - 한 연결의 종료와 다음 연결 간에 일정 시간을 둠
      - 다른 연결에서 온 패킷이 뒤섞여 혼란을 일으키는 것을 방지

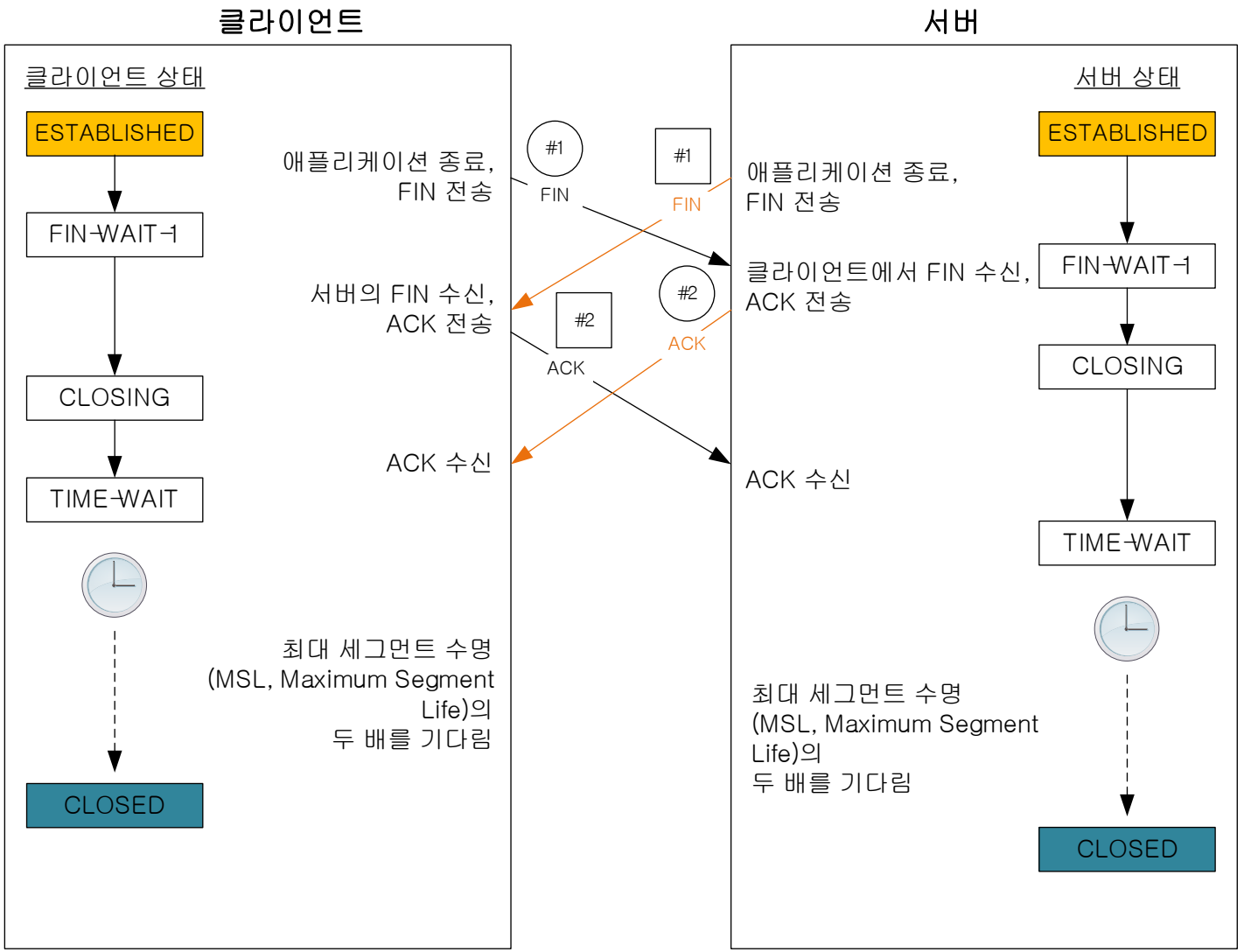
TCP 연결 수립, 관리 종료

- TCP 연결 종료
  - 정상 연결 종료 동작 과정



TCP 연결 수립, 관리 종료

- TCP 연결 종료
  - 동시 연결 종료 동작 과정



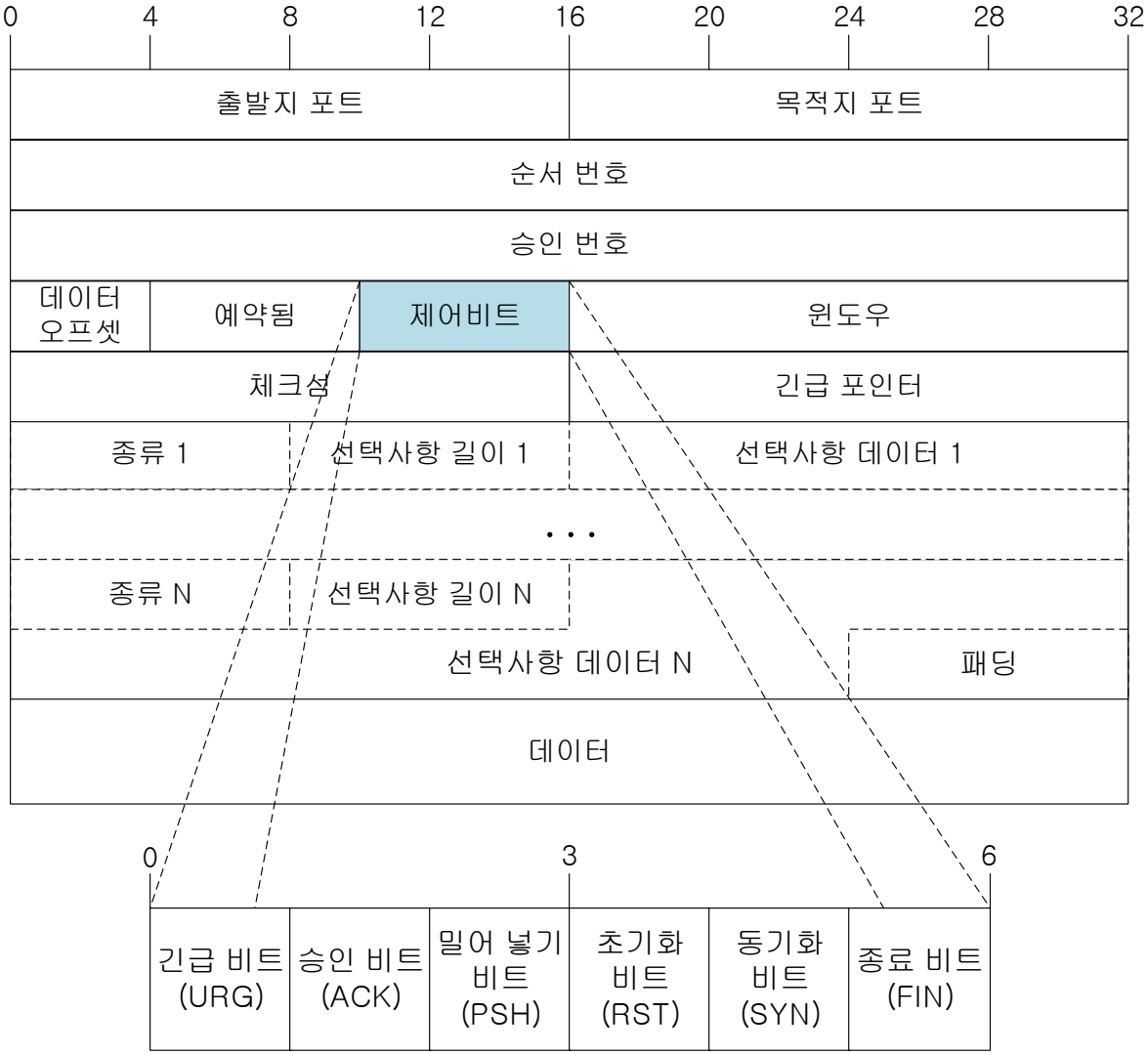
TCP 메시지 포맷과 데이터 송신

- TCP 세그먼트 포맷
  - TCP 헤더 필드의 목적
    - 프로세스 주소지정
      - 출발지와 목적지 장비에 있는 프로세스는 포트번호로 식별함
    - 슬라이딩 윈도우 시스템 구현
      - 순서 번호, 승인 번호, 윈도우 크기 필드로 TCP 슬라이딩 윈도우 시스템을 구현
    - 제어 비트와 필드 설정
      - 제어 기능을 구현하기 위한 비트와 이에 필요한 포인터나 다른 데이터를 저장하는 필드
    - 데이터 송신
    - 다양한 기능 구현
      - 데이터 보호를 위한 체크섬과 연결 수립을 위한 선택사항

TCP 메시지 포맷과 데이터 송신

- TCP 세그먼트 포맷

하위 필드명	크기 (비트)	설명
URG	1/8(1비트)	세그먼트에 우선순위가 높은 데이터가 있다는 뜻 (필드 값: 1)
ACK	1/8(1비트)	세그먼트가 승인을 포함한다는 뜻 (필드 값: 1)
PSH	1/8(1비트)	송신 장비가 TCP 밀어 넣기 기능을 사용했으므로 세그먼트를 받는 즉시 애플리케이션으로 송신하라는 뜻
RST	1/8(1비트)	송신 장비에 문제가 생겨 연결을 초기화해야 한다는 뜻
SYN	1/8(1비트)	순서 번호를 동기화하고 연결 수립을 요청하는 세그먼트를 의미
FIN	1/8(1비트)	세그먼트의 송신 장비가 연결 종료를 요청한다는 뜻



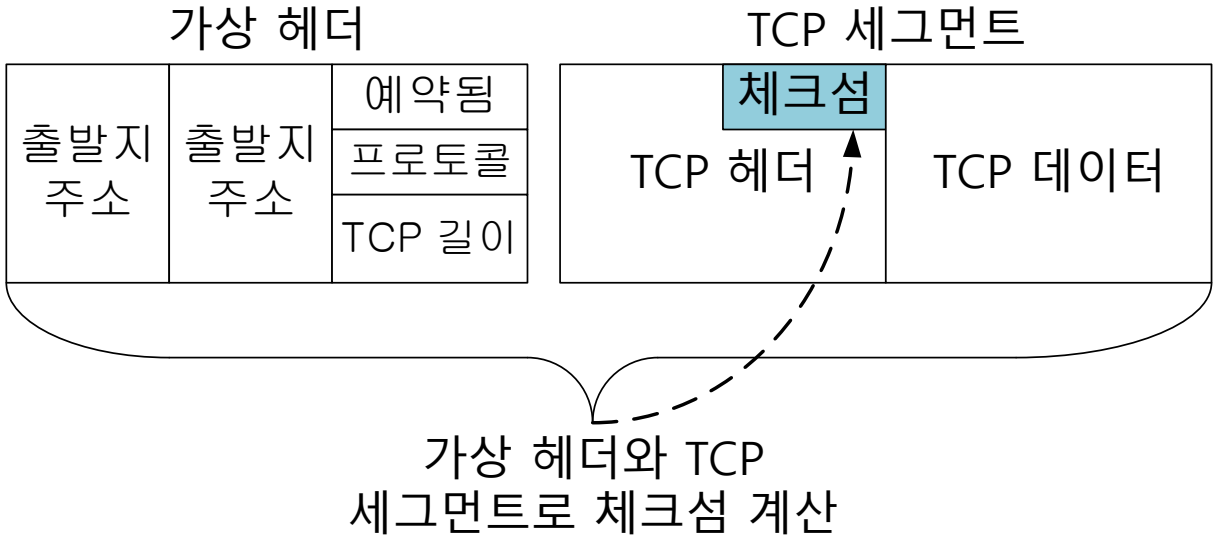
TCP 메시지 포맷과 데이터 송신

- TCP 체크섬 계산과 TCP 가상 헤더
  - TCP는 체크섬 계산에 가상헤더를 포함
    - 체크섬으로 송신 에러를 검출할 수 있음
    - 가상 헤더에는 IP 필드에서 가져온 정보다 들어있음
    - 가상 헤더는 체크섬을 계산할 때 TCP 세그먼트 앞에 덧붙음
      - 체크섬의 값을 체크섬 필드에 저장하고 나면 가상 헤더는 버림

TCP 메시지 포맷과 데이터 송신

- TCP 체크섬 계산과 TCP 가상 헤더
  - 체크섬 계산을 위한 TCP 가상 헤더 포맷

  - TCP 헤더 체크섬 계산



TCP 메시지 포맷과 데이터 송신

- TCP 체크섬 계산과 TCP 가상 헤더
  - 가상 헤더 사용의 장점
    - TCP 세그먼트에서 발생한 문제를 방지
      - 잘못된 세그먼트 송신
        - 목적지 장비가 수신한 목적지 주소가 일치 하지 않으면 체크섬이 일치 하지 않음
      - 잘못된 프로토콜
        - 패킷이 다른 이유로 다른 프로토콜을 통해 목적 TCP로 송신됐다면 검출됨
      - 잘못된 세그먼트 길이
        - TCP 세그머트의 일부가 빠지면 길이 값이 일치하지 않아 검출 가능

TCP 메시지 포맷과 데이터 송신

- TCP 최대 세그먼트 크기(MSS)
  - 각 메시지에 애플리케이션에서 받은 데이터를 얼마나 많은 바이트를 넣을 것인지 결정
  - TCP 장비는 현재 윈도우 크기와 세그먼트 크기의 한계가 있음
    - 윈도우가 아무리 크더라도 MSS를 넘을 수 없음
  - MSS 선택
    - 과부하 관리
      - MSS가 너무 작으면 대역폭을 비효율적으로 사용하게 됨
    - IP 단편화
      - 하위 네트워크의 최대 송신 단위(MTU, Maximum Transmission Unit) 이상은 송신 불가

TCP 메시지 포맷과 데이터 송신

- TCP 최대 세그먼트 크기(MSS)
  - 기본 MSS
    - 기본 MSS 계산은 IP 네트워크의 최소 MTU인 576바이트에서 시작함
      - TCP 헤더 20바이트, IP 헤더 20바이트를 사용해야 하므로 MSS는 536 바이트
  - MSS 값 명시
    - 536보다 크거나 작은 MSS를 사용할 수 있으나 반드시 명시
      - e.g., IPsec을 사용하면 헤더의 크기가 커지기 때문에 더 작은 MSS를 사용할 수 있음

TCP 메시지 포맷과 데이터 송신

- TCP 슬라이딩 윈도우 데이터 송신과 승인 방식
  - 전송 카테고리

전송 카테고리	의미
1	전송했고 승인 받음
2	전송했지만 아직 승인을 받지 못함
3	수신자는 준비됐지만 아직 전송하지 못한 바이트
4	수신자가 준비되지 않았고 전송하지 못한 바이트

  - 수신 카테고리

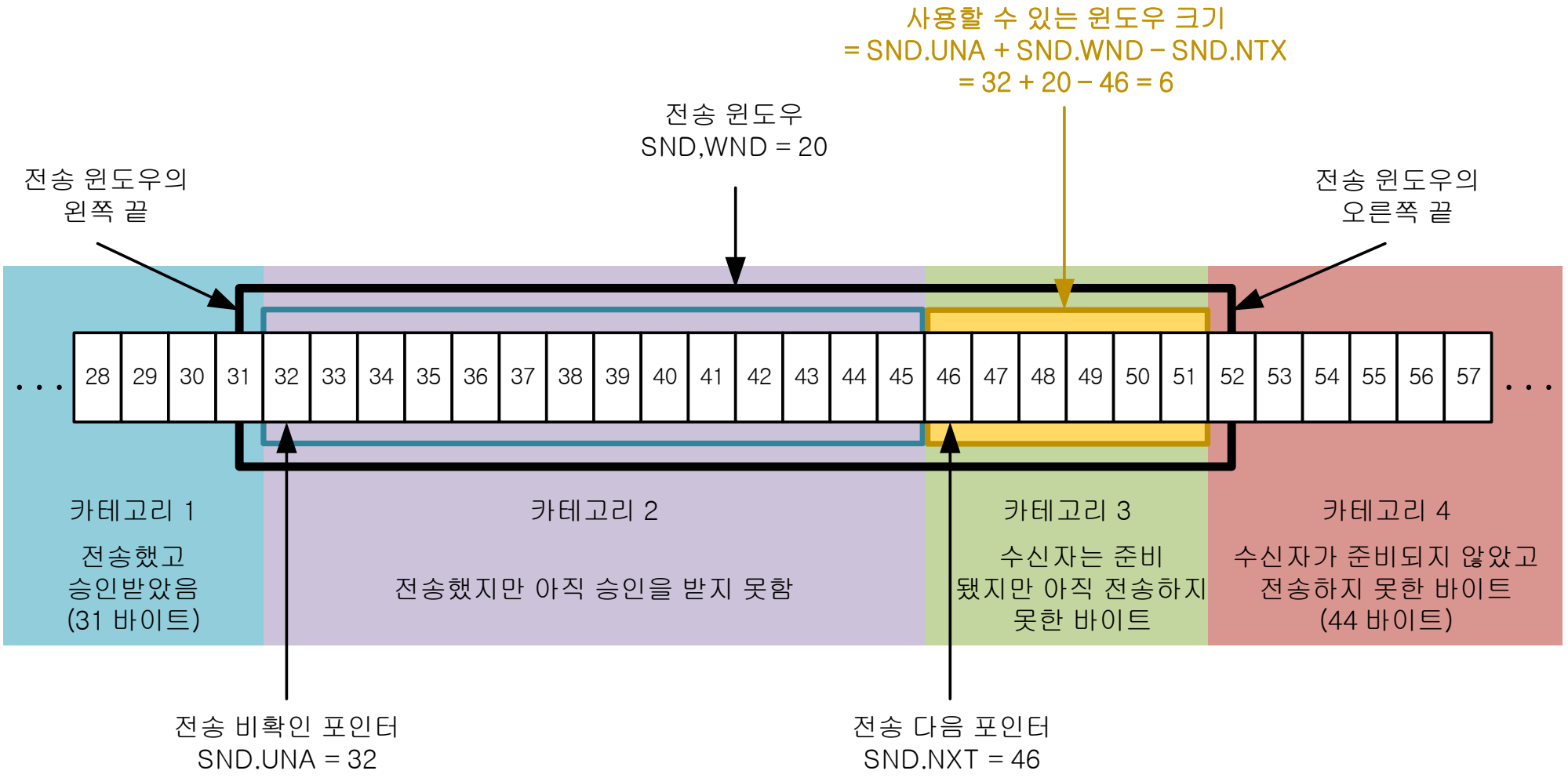
수신 카테고리	의미
1 + 2	수신했고 승인 받음
2	• 수신하지 않았지만 송신자가 보내도 되는 바이트 • 전송 카테고리 3에 대응됨
3	• 수신하지 못했고 송신자가 보내지도 않았음 바이트 • 전송 카테고리 4에 대응됨

TCP 메시지 포맷과 데이터 송신

- TCP 슬라이딩 윈도우 데이터 송신과 승인 방식
  - 송신(SND)과 수신(RCV) 포인터
    - 송신 비확인(SND.UNA)
      - 송신했지만 아직 승인되지 않은 첫 번째 데이터의 순서번호
      - 전송 카테고리 2의 첫 번째 바이트를 가리킴
    - 송신 다음(SND.NXT)
      - 다른 장비에게 보내야 할 다음 바이트의 순서번호
      - 전송 카테고리 3의 첫 번째 바이트를 가리킴
    - 송신 윈도우(SND.WND)
      - 송신 윈도우의 크기
      - 특정 시점에 승인 없이 보낼 수 있는 바이트의 수
      - SND.UNA + SND.WND = 전송 카테고리 4의 첫 바이트

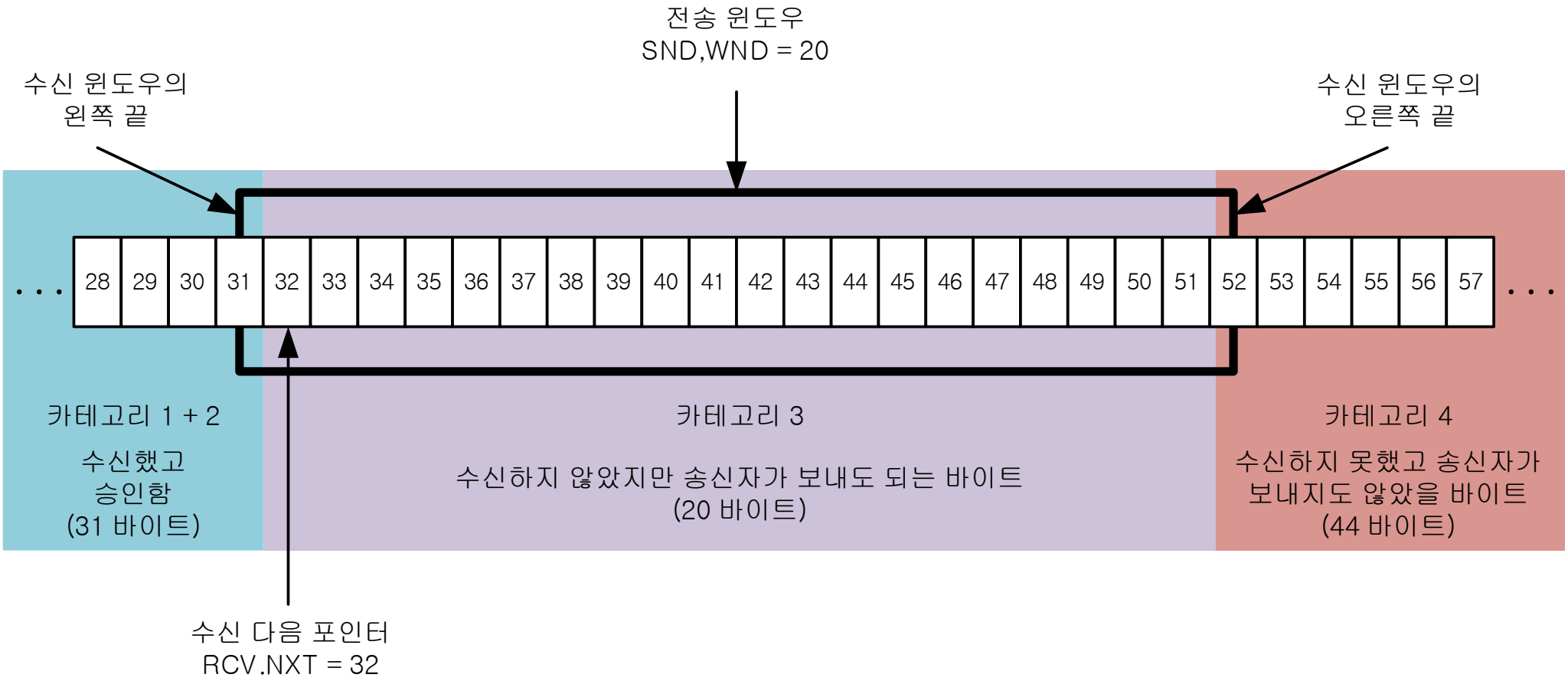
TCP 메시지 포맷과 데이터 송신

- TCP 슬라이딩 윈도우 데이터 송신과 승인 방식
  - TCP 전송 카테고리, 송신 윈도우, 포인터



TCP 메시지 포맷과 데이터 송신

- TCP 슬라이딩 윈도우 데이터 송신과 승인 방식
  - TCP 수신 카테고리, 수신 윈도우, 포인터



TCP 메시지 포맷과 데이터 송신

- TCP 슬라이딩 윈도우 데이터 송신과 승인 방식
  - 슬라이딩 윈도우 방식의 실제 복잡도
    - 중복 송신
      - 클라이언트는 서버에서 수신한 세그먼트를 새로운 요청을 보내면서 승인함
    - 다중 세그먼트 승인
      - 장비는 두 세그먼트를 받고 난 뒤 한 ACK를 보내낼 수도 있음
    - 흐름 제어를 이해 윈도우 크기 조절
      - 윈도우 크기를 조절하여, 흐름 제어를 구현할 수 있음

TCP 메시지 포맷과 데이터 송신

- TCP 슬라이딩 윈도우 데이터 송신과 승인 방식
  - 슬라이딩 윈도우 방식의 실제 복잡도
    - 송신 실패
      - 송신 실패 시 TCP의 재송신 방식에 따라 재송신이 처리됨
    - 작은 윈도우 문제 회피
      - 작은 윈도우 사용시 바보 윈도우 증후군이 생길 수 있음
    - 혼잡 처리와 회피
      - 기본 슬라이딩 윈도우 방식을 수정하여 인터네트워크에서 혼잡을 피할 수 있음

TCP 메시지 포맷과 데이터 송신

- TCP 밀어넣기 기능
  - 애플리케이션이 데이터를 즉시 보내야 하는 경우 TCP로 데이터를 보낸 후 TCP의 밀어넣기 명령 사용
    - PSH 제어 비트를 1로 설정하면 목적지 장비의 TCP 소프트웨어는 즉각 애플리케이션으로 데이터를 보냄
- TCP 긴급 기능
  - TCP는 긴급 기능을 통해 데이터 송신에 우선 순위를 둘 수 있음
    - URG 제어 비트를 1로 설정하면 수신 장비는 세그먼트 내의 데이터 중 어디까지가 긴급한 것인지 결정

TCP 신뢰성과 흐름 제어

- TCP 세그먼트 재전송 타이머와 재전송 큐
  - 재전송 큐를 사용하여 재전송 관리
    - 재전송 큐에 배치, 타이머 시작
      - 세그먼트가 전송되면 TCP는 세그먼트의 복사본을 재전송 큐라는 데이터 구조에 삽입
      - 재전송 타이머는 세그먼트를 큐에 삽입할 때 시작
    - 승인처리
      - 타이머 만료 전에 승인이 온다면 TCP는 세그먼트를 재전송 큐에서 제거
    - 재전송 시간 만료
      - 타이머 만료 전에 승인이 오지 않는다면 재전송 시간 만료가 일어나 세그먼트는 자동적으로 재전송됨

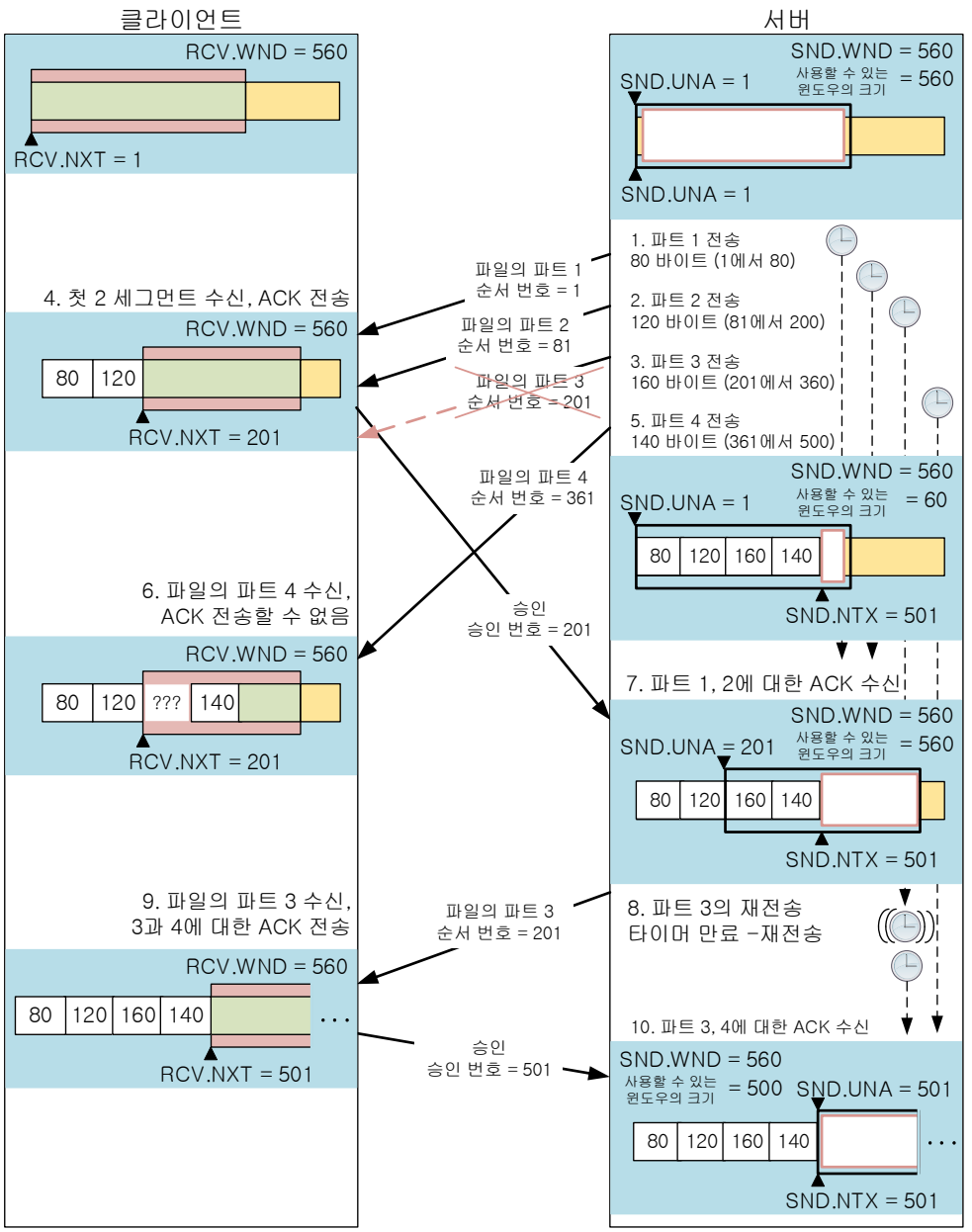
TCP 신뢰성과 흐름 제어

- TCP 세그먼트 재전송 타이머와 재전송 큐
  - 세그먼트가 완전히 승인된 시간
    - 누적 순서 번호를 사용
    - 승인 메시지를 보낼 때 수신 받은 마지막 순서번호+1을 전송

세그먼트	순서번호 필드	세그먼트 길이	세그먼트의 마지막 순서번호
1	1	80	80
2	81	120	200
3	201	160	360
4	361	140	500

TCP 신뢰성과 흐름 제어

- TCP 세그먼트 재전송 타이머와 재전송 큐
  - TCP 재전송 동작과정



TCP 신뢰성과 흐름 제어

- TCP 비연속적 승인 처리와 선택적 승인(SACK, Selective Acknowledgment)
  - 고속 네트워크거나 신뢰할 수 없는 물리 네트워크를 사용할 경우 누적 승인으로 인해 성능 저하가 발생할 수 있음
    - 세그먼트가 전송 중에 사라지면 후속 세그먼트는 사라진 세그먼트가 재전송되어 성공적으로 받아지기 전에는 승인 될 수 없음
  - 비연속적 승인 처리
    - 시간 만료된 세그먼트만 재전송
      - 많은 세그먼트가 사라졌을 경우 기다려야 하는 시간이 길어짐
    - 승인 받지 못한 모든 세그먼트 재전송
      - 시간 만료가 되면 사라진 세그먼트와 승인 받지 못한 모든 세그먼트를 재전송하게 되어 불필요한 재전송을 하게 됨

TCP 신뢰성과 흐름 제어

- TCP 비연속적 승인 처리와 선택적 승인
  - 선택적 승인(SACK)
    - 연결을 수립할 때 선택적 승인 허용 선택사항을 사용해서 협상해야 함
    - 특정 세그먼트를 재전송할 때 그 뒤에 있는 세그먼트 중 SACK 비트가 1이 아닌 모든 세그먼트를 재전송함

TCP 신뢰성과 흐름 제어

- TCP 적응형 재전송과 재전송 타이머 계산
  - 재전송 타이머의 값은 왕복 시간(RTT, Round-Trip Time)보다 약간 크면 이상적
    - 왕복 시간 : 클라이언트에서 출발한 세그먼트가 서버에 도착하여 승인을 클라이언트에게 되돌려 보내는 데 드는 시간
    - 일반적인 RTT가 존재하지 않는 이유
      - 연결 거리의 차이
      - 일시적인 지연시간과 변동성
  - RTT 계산에 기반한 적응형 재전송
    - 평균적인 지연시간을 계산
      - 새 RTT= α ×예전 RTT +(1−α×가장 최근에 측정한 RTT)
      - α는 0~1 값을 가짐

TCP 신뢰성과 흐름 제어

- TCP 적응형 재전송과 재전송 타이머 계산
  - 모호한 승인(Acknowledgment ambiguity)
    - 승인이 왔을 때 승인이 원래 세그먼트에 대한 것인지 재전송된 세그먼트에 대한 것인지 알 수 없음
  - RTT 계산 수정과 칸의 알고리즘
    - 필 칸(Phil Karn)의 이름을 딴 알고리즘
      - 재전송 세그먼트에 대해서 타이머 백오프(Backoff) 방식을 도입
        - 세그먼트를 재전송할 때 타이머를 초기 전송에서 사용하는 값으로 설정하지 않고 백오프를 통해 몇 배 증가시켜 재전송 되는 시간을 잡음
        - 타이머 값은 재전송이 성공할 때까지 증가하지만 일정 한도가 있음
        - 재전송에 타이머 계산과 평균 RTT 계산을 분리

TCP 신뢰성과 흐름 제어

- TCP 윈도우 크기 조절과 흐름 제어
  - 클라이언트와 서버가 자신이 한 번에 받을 수 있는 바이트 수를 알림
    - 그 값은 서버의 송신 윈도우와 클라이언트의 송신 윈도우 크기가 됨
    - 윈도우 크기를 줄이거나 증가시키면서 상대 장비가 데이터를 보낼 때 수신 장비가 처리할 수 있을 만큼만 보냄

TCP 신뢰성과 흐름 제어

- TCP 윈도우 관리 문제
  - TCP 윈도우 크기 감소와 관련된 문제
    - 장비가 바빠졌을 때 얼마나 빨리 자신의 수신 윈도우를 줄이느냐에 달려있음
      - 서버의 부하가 심해 윈도우 크기를 줄여야 한다면 클라이언트에게 윈도우 크기 감소(Shrinking the window) 정보를 전송해야 함
      - 클라이언트는 서버의 윈도우 크기 감소 정보를 받기 전에 세그먼트를 보내는 경우 데이터 손실이 생길 수 있고 재전송 해야 하는 비효율이 생김

TCP 신뢰성과 흐름 제어

- TCP 윈도우 관리 문제
  - 수신 윈도우 닫기 문제
    - 서버의 수신 윈도우가 닫히면 클라이언트는 서버의 윈도우 개방 세그먼트를 받을 때 까지 기다려야 함
    - 클라이언트는 주기적으로 탐사(Probe) 세그먼트를 서버에 전송
      - 탐사 세그먼트를 받은 서버는 현재 윈도우 크기를 갖는 세그먼트를 전송
      - 탐사 세그먼트에는 데이터가 없을 수도 있고 한 바이트를 포함할 수 있음
  - 클라이언트는 윈도우가 다시 열릴 때까지 주기적으로 탐사 세그먼트를 전송

TCP 신뢰성과 흐름 제어

- TCP 바보 윈도우 증후군(SWS, Silly Window Syndrome)
  - 바보 윈도우 증후군의 원인
    - 최소 세그먼트의 크기를 지정하지 않아서 생기는 문제
    - 수신되는 세그먼트를 처리하는 속도가 늦는 경우 윈도우가 닫히게 되고 윈도우를 1바이트 크기로 개방하게 되어 TCP의 효율을 급격히 떨어트림

TCP 신뢰성과 흐름 제어

- TCP 바보 윈도우 증후군(SWS, Silly Window Syndrome)
  - 바보 윈도우 증후군 회피 알고리즘
    - 수신자 SWS 회피
      - 수신자는 윈도우의 오른쪽 끝을 조금씩 움직이지 않아야 함
        - 오른쪽 끝이 움직이는 최소 단위는 MSS 파라미터 값이나 버퍼 크기의 절반 중에서 작은 것으로 결정
    - 송신자 SWS 회피와 네이글의 알고리즘
      - 네이글의 알고리즘 동작 방식
        - 승인 받지 못한 데이터가 없다면 애플리케이션이 원하는 대로 데이터를 즉각 전송
        - 승인 받지 못한 데이터가 있는 경우 모두 승인되거나 충분한 크기의 세그먼트를 만들 만큼 데이터가 모이지 않은 한 후속 데이터를 보내지 않음
        - 사용자가 밀어넣기를 요청하더라도 이 방식을 동일하게 적용

TCP 신뢰성과 흐름 제어

- TCP 혼잡 처리와 혼잡 회피 알고리즘
  - TCP는 4계층이기 때문에 데이터가 중간에 어떻게 송신되는지는 3계층 IP에게 위임
  - 인터네트워크가 매우 바빠지게 되면 세그먼트를 송신하는 속도가 느려지거나 버려지기도 함
    - 계속적인 재송신으로 인해 혼잡 붕괴(Congestion collapse) 현상이 발생할 수 있음

TCP 신뢰성과 흐름 제어

- TCP 혼잡 처리와 혼잡 회피 알고리즘
  - TCP 혼잡 처리 방식

TCP 혼잡 처리 방식	설명
느린 시작 (Slow Start)	• TCP 장비는 초기에 세그먼트를 보내는 속도를 제한함 • 송신 장비는 MSS의 세그먼트를 하나만을 송신한 후 송신 윈도우 크기만큼 보내거나 네트워크에 혼잡이 있다는 걸 알게 될 때까지 보냄 • 혼잡이 발생한 경우 혼잡 회피 기능이 사용됨
혼잡 회피 (Congestion Avoidance)	• TCP 링크에 혼잡이 발생한 것 같으면 세그먼트를 보내는 속도를 억제함 • 장비는 다시 혼잡이 일어나지 않도록 느린 시작을 통해 다시 송신 속도를 증가 시킴 • 혼잡 회피 메커니즘 - (TCP Tahoe, TCP Reno, TCP New-Reno, 선택적 승인(SACK))

TCP 신뢰성과 흐름 제어

- TCP 혼잡 처리와 혼잡 회피 알고리즘
  - TCP 혼잡 회피 알고리즘
    - 빠른 재송신(Fast Retransmit)
      - 중복 승인 메시지를 3번 이상 받으면 정상적인 재송신 큐 과정을 생략하고 사라진 세그먼트를 재송신함
      - 타임 만료를 기다리지 않아 TCP 성능을 향상 시킴
      - 동작과정
        1. 재전송한 이후 느린 시작 한계(ssthresh)의 값은 W(loss) / 2로 설정됨
        2. 재전송된 후에는 다시 윈도우의 크키를 1로 설정하여 반으로 줄어든 한계점까지는 느린 시작 상태로 증가
        3. 빠른 재전송에 의한 복구가 불가능한 경우라도 느린 시작이 다시 시작됨
           - ssthresh = W(loss) / 2

TCP 신뢰성과 흐름 제어

- TCP 혼잡 처리와 혼잡 회피 알고리즘
  - TCP 혼잡 회피 알고리즘
    - 빠른 회복(Fast Recovery)
      - 빠른 재전송 과정을 거침
      - Pipe에 남아 있는 패킷들을 통해 현재 상태를 알 수 있음
      - 다시 느린 시작 상태로 가지 않고 혼잡 회피 상태가 계속됨
        - 느린 시작 한계의 값으로 설정된 윈도우로 지속됨

TCP 신뢰성과 흐름 제어

- TCP 혼잡 처리와 혼잡 회피 알고리즘
  - 혼잡 회피 메커니즘
    - TCP Tahoe
      - 빠른 재전송 알고리즘을 사용
        - 중복 수신이 3회 발생
        - 느린 시작 한계를 현재 혼잡 윈도우 크기의 반으로 설정
        - 재전송에 대한 승인 메시지를 받으면 느린 시작 상태에서 윈도우 크기 증가
        - 느린 시작 한계 값은 두 배로 증가 함
        - 윈도우 크기가 ssthresh 값에 도달하면 혼잡 회피 상태로 변경

TCP 신뢰성과 흐름 제어

- TCP 혼잡 처리와 혼잡 회피 알고리즘
  - 혼잡 회피 메커니즘
    - TCP Reno
      - 빠른 재전송, 빠른 회복 알고리즘을 사용
        - 빠른 재전송에 의해서 재전송될 때까지의 동작은 Tahoe와 동일함
        - ssthresh 값 = (현재 윈도우의 크기 / 2) + 중복 승인 메시지 수
        - 이후 빠른 회복 동안 송신원은 중복 승인 패킷이 수신될 때 마다 윈도우의 크기를 1씩 증가시킴
        - 승인 메시지를 받으면 빠른 회복을 종료하고 혼잡 회피 상태가 됨
          - ssthresh 값 = 빠른 재전송에서 설정한 윈도우의 크기 / 2
      - 단점
        - 다수의 패킷이 동시에 손실되는 경우 재전송에 의해 복구가 불가능
          - TCO New-Reno또는 SACK 선택 옵션으로 해결 가능

TCP 신뢰성과 흐름 제어

- TCP 혼잡 처리와 혼잡 회피 알고리즘
  - 혼잡 회피 메커니즘
    - TCP New-Reno
      - 빠른 재전송에 의해서 패킷의 순서번호를 recover라는 변수에 저장
      - 이보다 낮은 패킷이 승인되었다고 메시지가 오면 패킷이 손실되었다고 판단
        - 부분 승인(Partial Acknowledgement)을 함
      - 빠른 재전송 때 설정된 ssthresh 값과 동일하게 설정
      - 변수에 저장된 수보다 뒷 순서의 패킷이 승인되었다고 메시지가 오면 빠른 회복 상태를 종료 되고 혼잡 회피 상태가 됨

Thanks!

박 재 형 (jaehyoung@pel.sejong.ac.kr)