TCP/IP 완벽 가이드

- II-8부 전송계층 TCP 프로토콜 -

명 세인(sein@pel.smuc.ac.kr)

상명대학교 프로토콜공학연구실

목 차

TCP 개요

- TCP 개요
  - RFC 793에서 TCP(Transmission Control Protocol)버전 4를 정의
    - 모든 세부 동작을 설명하지 않으며, 다른 문서에서 추가적으로 기술
  - 초기의 TCP는 Transmission Control Program 이라 불리고 RFC 675에서 공식화
    - 원래 TCP는 현재의 TCP와 IP기능을 모두 수행
    - RFC 793에서 TCP와 IP로 분리되면서 Program이 Protocol로 변경

TCP 개요

- TCP 개요
  - TCP에서 지원하는 기능
    - 주소지정, 다중화/다중 연결
      - 포트 번호와 소켓쌍 으로 다중연결을 식별
      - 여러 애플리케이션이 동시에 TCP를 사용할 수 있도록 포트번호로 구별하고 다중화
    - 연결 수립, 유지, 종료와 양방향성
    - 신뢰성, 승인 기능
      - 각 데이터에 대해 승인 기능을 제공(누적될 수도 있음)
    - 스트림 기반 전송
    - 데이터 흐름, 연결성 관리

TCP 개요

- TCP 개요
  - TCP기능의 한계
    - 보안
      - TCP는 어떠한 보안도 보장하지 않음
      - 다른 수단을 추가하여 보안
    - 메시지 경계
      - TCP는 연속 스트림으로 메시지 송신
      - 각 메시지의 경계는 애플리케이션에서 구현
    - 통신 보장
      - 여러 흐름제어, 혼잡 회피 기능을 제공하지만 이는 추가적으로 해결할 수 있는 옵션
      - TCP를 사용하면 100%의 신뢰도를 가질 수 있다는 개념이 아님

목 차

- TCP 개요
- TCP 원리와 일반 동작
- 연결의 수립, 관리 종료
- TCP 메시지 포맷과 데이터 송신
- TCP 신뢰성과 흐름제어

TCP 원리와 일반 동작

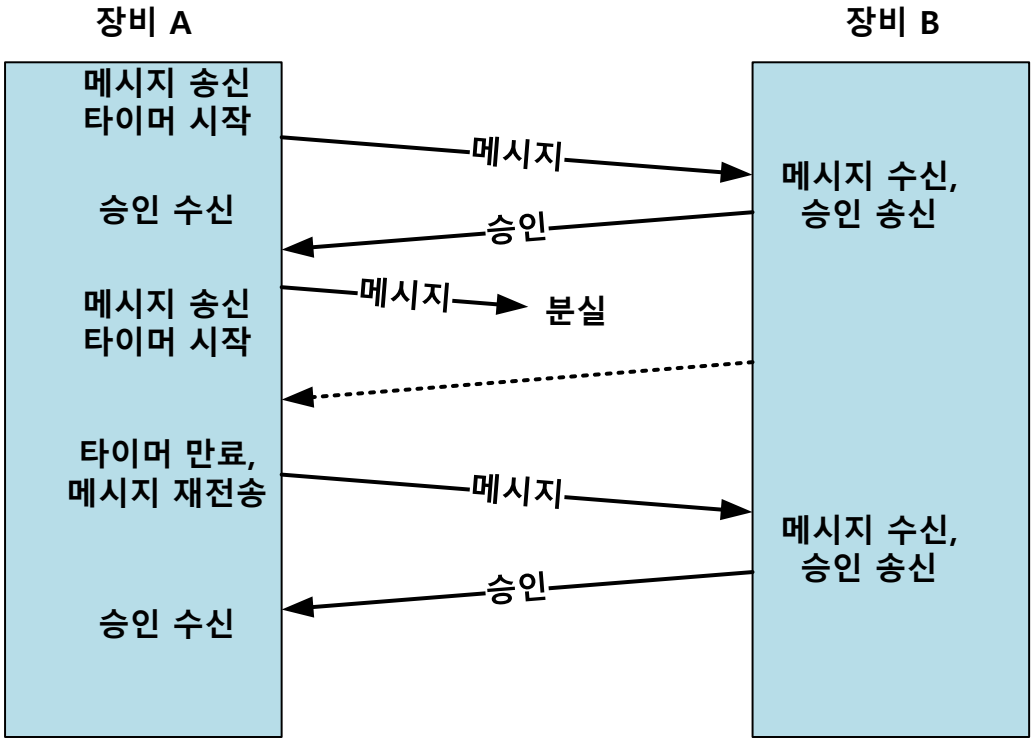
- TCP의 데이터관리와 처리
  - 스트림 기반
    - 애플리케이션이 전송을 원하는 모든 데이터는 TCP에서 옥텟 스트림으로 인식
  - TCP 데이터 패키징 (세그먼트)
    - 애플리케이션에서 받은 데이터는 IP를 이용해서 전송
    - IP를 사용할 수 있도록 TCP세그먼트 라는 분리된 메시지로 나눔
      - 세그먼트의 최대 크기(MSS: Maximum Segment Size)를 협상하여 사용
        - IP에서 불필요한 단편화를 막음
        - TCP를 사용하면서 전송의 효율을 높여줌

TCP 원리와 일반 동작

- TCP의 데이터관리와 처리
  - TCP 데이터 식별, 순서번호
    - 신뢰성이란 모든 데이터에 대해서 승인이 제공 되어야 함
    - TCP에서 각 바이트는 순서 번호를 할당 받고, 이 번호를 통해 목적 장비에서 데이터 순서, 모든 데이터 승인을 제공
  - TCP 데이터 구분
    - 애플리케이션수준에서 필요한 데이터의 의미에 대한 구분은 애플리케이션 에서 직접 구현 되어야 함

TCP 원리와 일반 동작

- TCP슬라이딩 윈도우 승인체계
  - 프로토콜을 신뢰할 수 있으려면 피드백(승인)이 필요
    - 재전송을 사용하는 긍정 승인 (PAR: Positive Acknowledgment with Retransmission)의 기본 동작
      - 전송에 대한 응답이 오기까지 특정 타이머를 가짐
      - 응답이 오면 타이머 초기화
      - 타이머가 만료되면 재전송



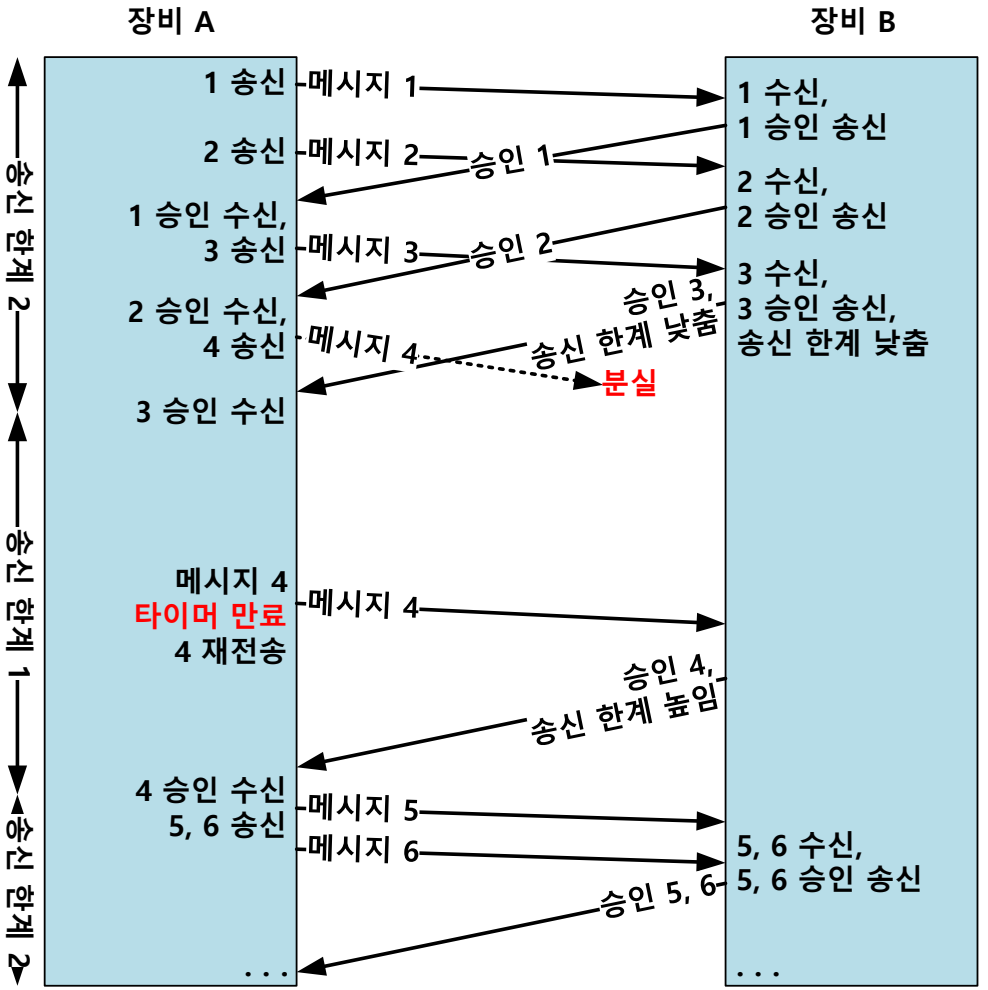
TCP 원리와 일반 동작

- TCP슬라이딩 윈도우 승인체계
  - 프로토콜을 신뢰할 수 있으려면 피드백(승인)이 필요
    - PAR 개선
      - 옥텟 단위로 보내지 않고 세그먼트 단위로 보냄
      - 각 세그먼트 의 마지막 옥텟의 순서번호를 이용해 개별 승인
      - 송신 한계라는 인자를 기준으로 승인 중인 최대 옥텟 수를 제한

      - TCP 에서는 슬라이딩 윈도우(Sliding Window) 기법을 사용

TCP 원리와 일반 동작

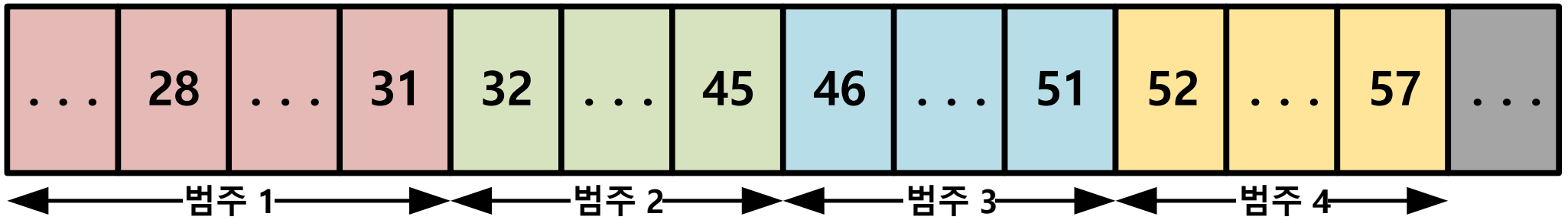
- TCP슬라이딩 윈도우 승인체계
  - 프로토콜을 신뢰할 수 있으려면 피드백(승인)이 필요
    - 개선된 PAR



TCP 원리와 일반 동작

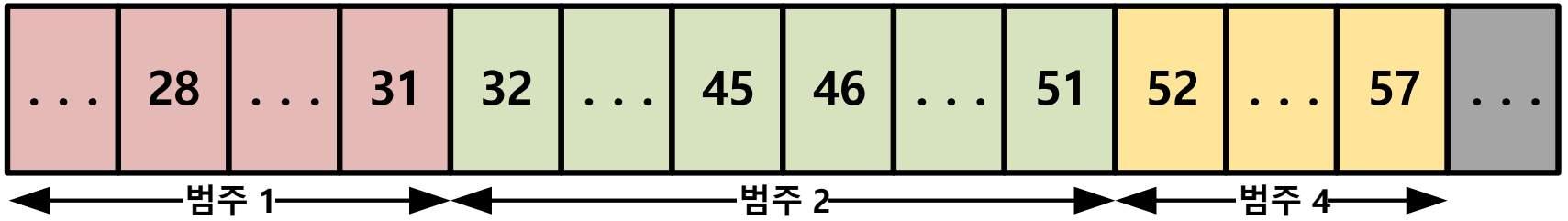
- TCP슬라이딩 윈도우 승인체계
  - 프로토콜을 신뢰할 수 있으려면 피드백(승인)이 필요
    - 슬라이딩 윈도우
      - TCP 전송 스트림의 개념적 구분
      - 슬라이딩 윈도우를 사용하기 위해 스트림 바이트에 할당된 순서번호를 동기화(Synchronization)해야 함
      - 송신자측 범주 구분 표와 그림
        - 범주 내에서 전송된 세그먼트 크기는 가변적

개념	송신	승인	수신자의 준비
범주 1	O	O	.
범주 2	O	X	.
범주 3	X	X	O
범주 4	X	X	X

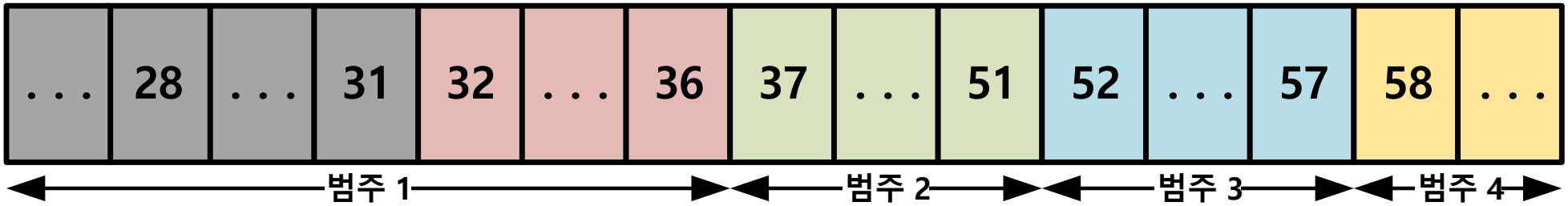


TCP 원리와 일반 동작

- TCP슬라이딩 윈도우 승인체계
  - 프로토콜을 신뢰할 수 있으려면 피드백(승인)이 필요
    - 슬라이딩 윈도우
      - 모든 송신 윈도우를 사용한 경우
        - 송신자가 전송하도록 허용된 최대 바이트수 = 범주 2 + 범주3

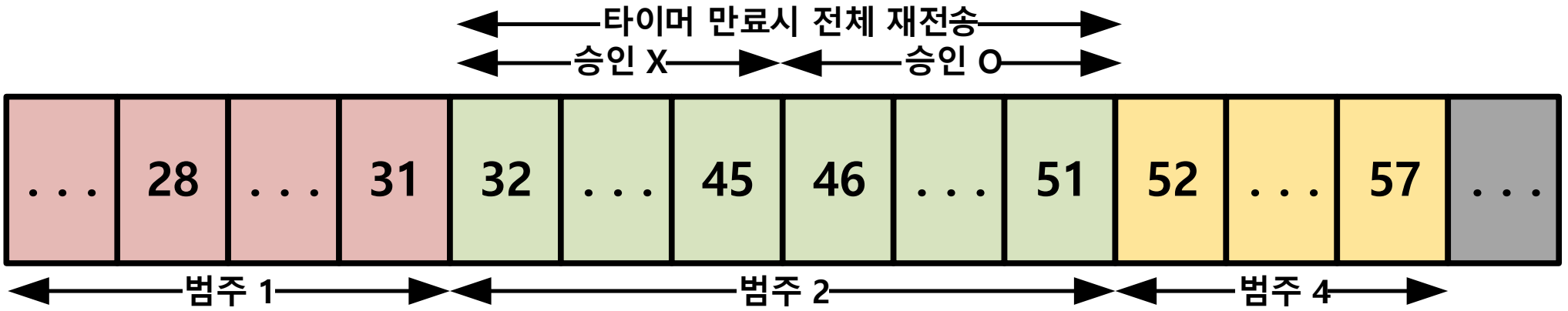


      - 승인이 수신되어 윈도우를 이동



TCP 원리와 일반 동작

- TCP슬라이딩 윈도우 승인체계
  - 프로토콜을 신뢰할 수 있으려면 피드백(승인)이 필요
    - 슬라이딩 윈도우
      - 빠진 승인 처리
        - 세그먼트 각각에 대한 승인이 아니기 때문에 중간 승인이 빠지면 이미 수신 받은 세그먼트도 재전송 해야될 수 있음



TCP 원리와 일반 동작

- TCP포트, 연결과 연결 식별
  - 소켓 개념
    - IP 주소와 Port번호 쌍으로 식별되는 연결
    - 소켓 개념을 사용하여 한 클라이언트에서 같은 서버로 다중 접속, 여러 클라이언트들이 서버로 다중 접속 하는 것을 모두 유일하게 식별하고 독립적으로 관리

목 차

연결의 수립, 관리, 종료

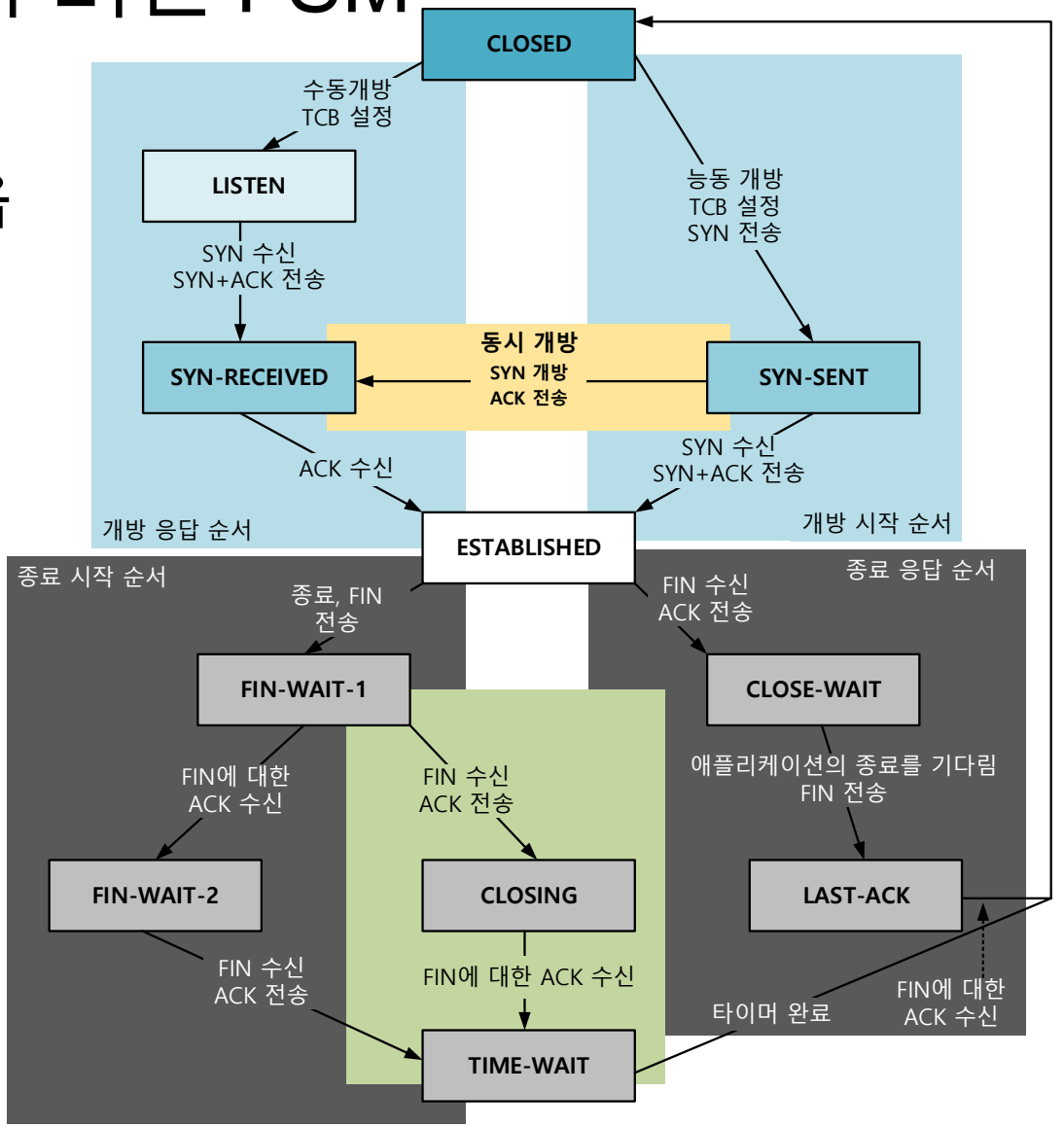
- TCP의 동작과 유한 상태 머신 FSM
  - 유한 상태 머신(FSM: Finite State Machine)이란 상태, 전이, 이벤트, 행동 등의 프로토콜이 처할 수 있는 모든 상태를 정의하고 설명하는 이론적 도구
    - 상태: 특정 시간에 프로토콜 소프트웨어의 상황
    - 전이: 상태에서 상태로 움직이는 행위
    - 이벤트: 전이를 발생시킨 사건
    - 행동: 이벤트에 대한 반응, 전이하기 전에 하는 행동

연결의 수립, 관리, 종료

- TCP의 동작과 유한 상태 머신 FSM
  - TCP의 FSM
    - SYN(Synchronize) 메시지: 연결을 초기화하고 수립, 장비간 옥텟 순서번호를 동기화
    - FIN(Finish) 메시지: TCP 세그먼트에 FIN 비트로 장비가 연결을 종료하고 싶다는것을 알림
    - ACK(Acknowledgment): 승인 메시지, SYN, FIN 메시지를 승인

연결의 수립, 관리, 종료

- TCP의 동작과 유한 상태 머신 FSM
  - 간단한 TCP FSM 그림
    - 대칭적으로 동작하지 않음
- TCP 연결 준비



연결의 수립, 관리, 종료

- TCP 연결 준비
  - 수립된 각 연결에 대한 데이터 저장
    - 전송 제어 블록(TCB: Transmission Control Block)
      - 연결을 식별하기 위한 두 소켓 번호,
      - 수신, 송신 데이터를 가지고 있는 버퍼 포인터,
      - 승인에 대한 정보, 승인하지 못한 순서번호, 현재 윈도우 크기 등을 추적하는 변수 저장
  - 연결 개방
    - 능동 개방
      - TCP를 사용하는 클라이언트가 능동 역할을 맡아 SYN메시지를 전송, 연결을 시작
    - 수동 개방
      - TCP를 사용하는 서버가 특정 클라이언트로부터 연결이 오도록 명시하거나, 모든 클라이언트의 연결을 기다림

연결의 수립, 관리, 종료

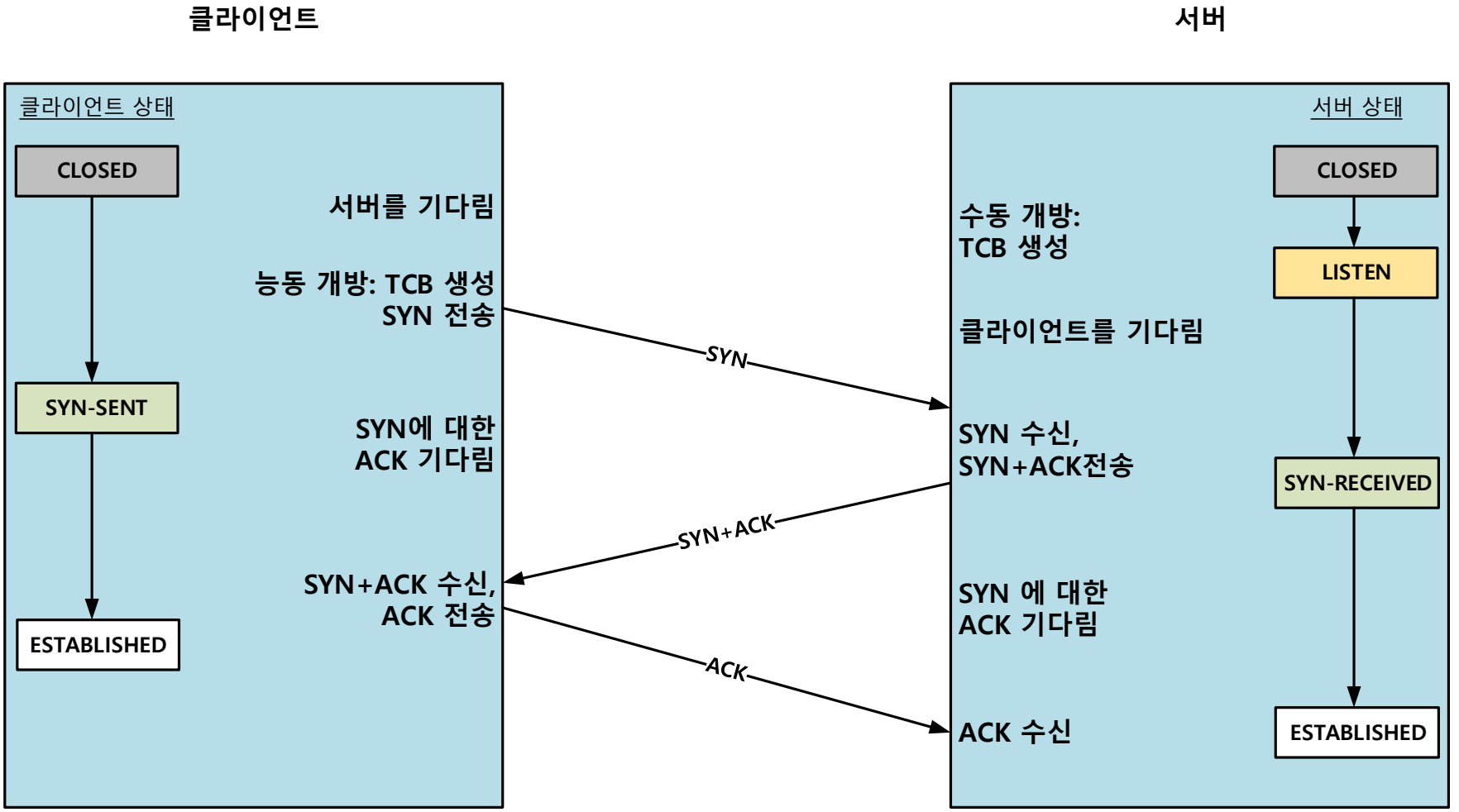
- TCP 연결 준비
  - 연결 준비
    - 클라이언트와 서버는 개방을 수행하는 동안 TCB를 생성, 각 연결을 고유하게 식별
    - 서버의 경우 소켓번호를 명시하지 않은 TCB를 생성하고 클라이언트의 능동 개방을 기다림
    - TCB는 연결이 완전히 종료되고 CLOSED상태가 되면 없어짐

연결의 수립, 관리, 종료

- TCP 연결 수립 과정: 쓰리 웨이 핸드셰이크
  - 연결 수립 기능
    - 접촉에 의한 통신 시작
    - 순서번호 동기화
    - TCP 연결의 동작을 제어하기 위한 인자 교환
  - 연결에 사용하는 제어 메시지
    - SYN: 연결을 초기화, 순서번호를 동기화
    - ACK: 세그먼트를 받았다는 승인 메시지

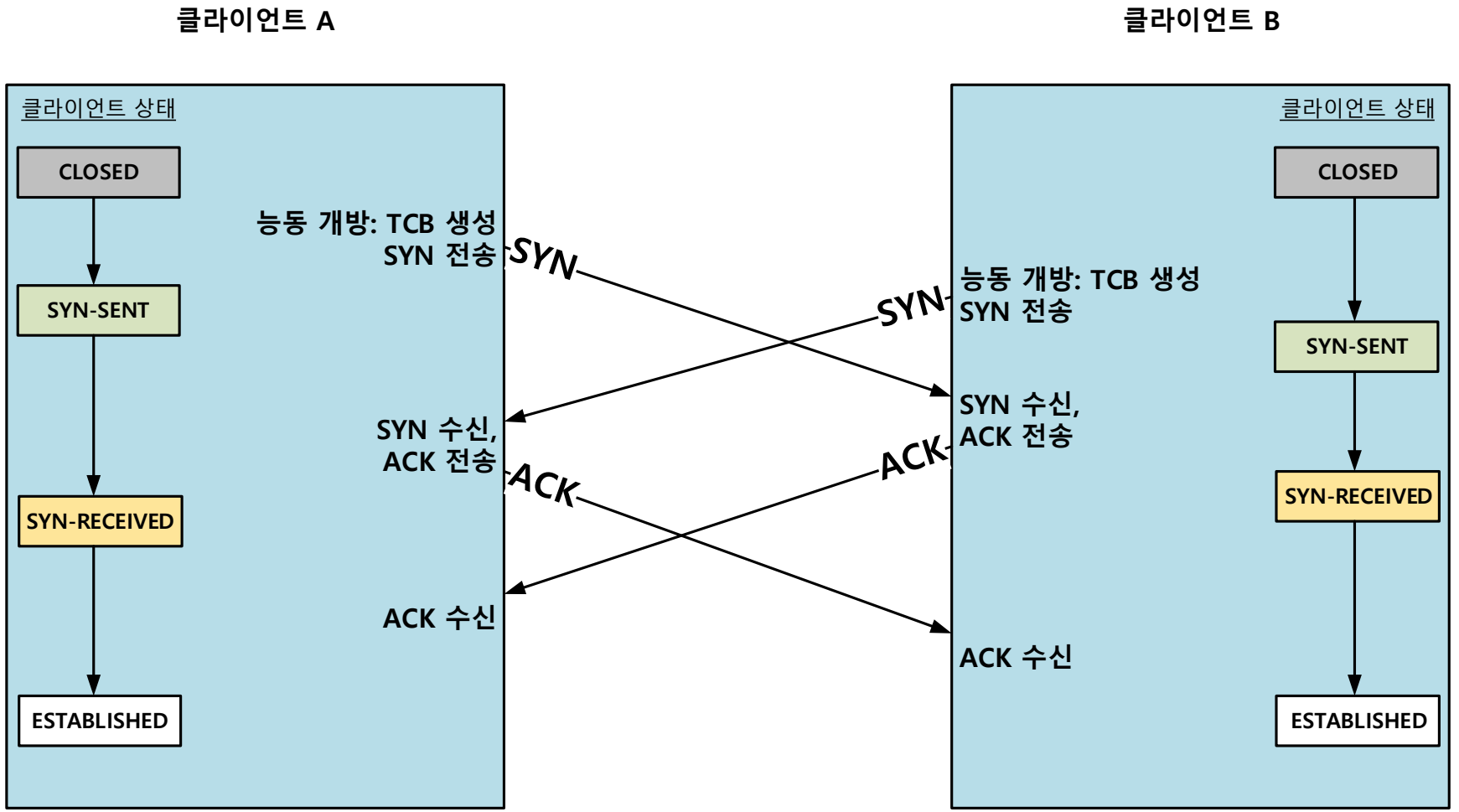
연결의 수립, 관리, 종료

- TCP 연결 수립 과정: 쓰리 웨이 핸드셰이크
  - 쓰리 웨이 핸드 셰이크(Three Way Handshake)



연결의 수립, 관리, 종료

- TCP 연결 수립 과정
  - 동시 개방 연결 수립 과정

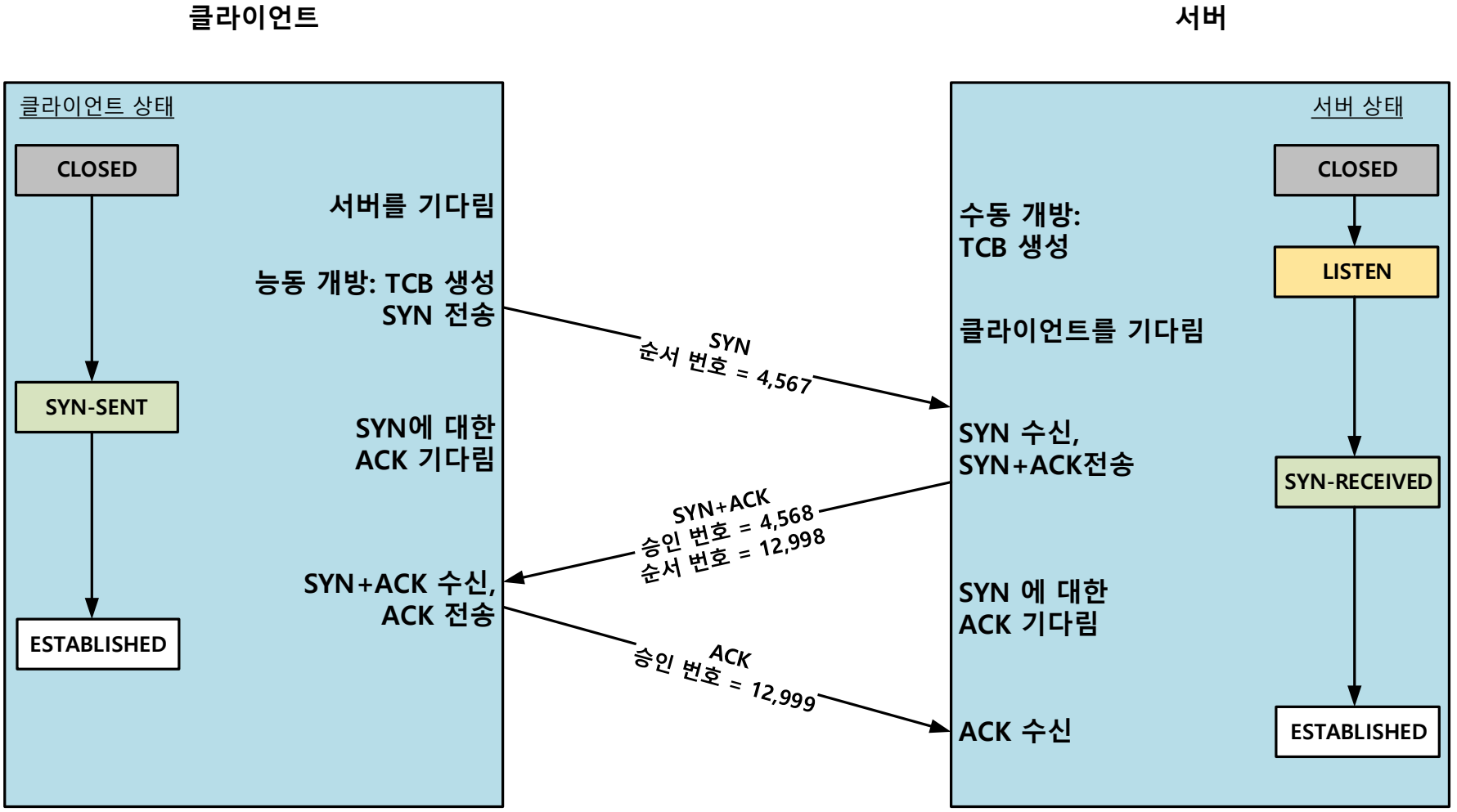


연결의 수립, 관리, 종료

- TCP 연결 수립, 순서번호 동기화와 인자교환
  - TCP 순서번호
    - 다른 연결에서 온 세그먼트와 섞이지 않도록 초기 순서번호를 잘 선택 해야 함
      - 전통적으로 초기순서번호(ISN: Initial Sequence Number)는 4μs 마다 증가하는 32비트 카운터로 ISN을 정함
      - 최근의 TCP는 무작위 ISN을 사용
  - TCP 인자 교환
    - 연결 단계에서 교환할 수 있는 인자
      - 윈도우 크기 인자
      - 선택적 승인 허용
      - 대체 체크섬 방식

연결의 수립, 관리, 종료

- TCP 연결 수립, 순서번호 동기화와 인자교환
  - TCP 순서 번호 동기화



연결의 수립, 관리, 종료

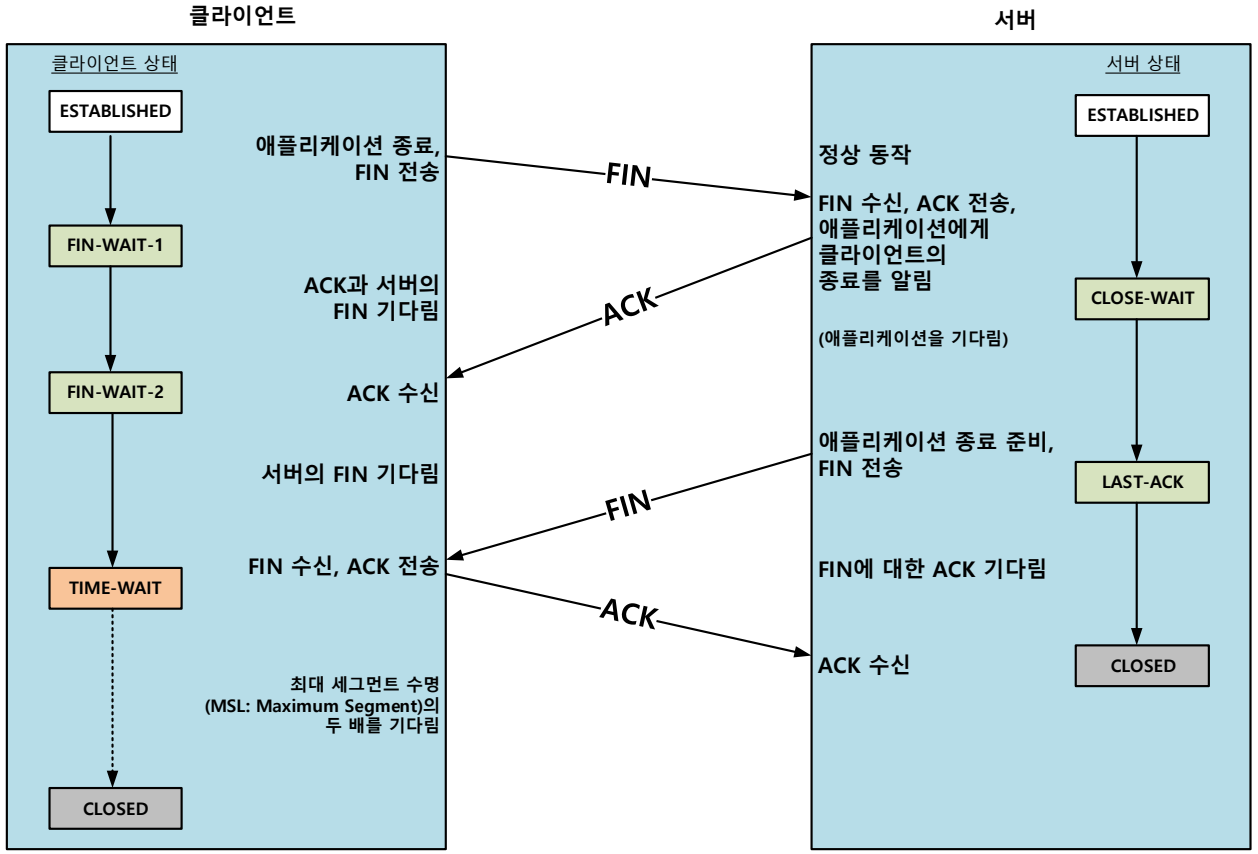
- TCP 연결 관리, 문제처리
  - 슬라이딩 윈도우를 사용하면 두 장비 모두 무한정 ESTABLISHED 상태
    - 연결을 종료 하려면 두 장비 중 하나가 연결 종료를 결정 하거나 문제가 발생하여 연결 방해
  - TCP 초기화 기능
    - 한 장비는 ESTABLISHED상태, 다른 장비는 CLOSED 상태나 다른 임시 상태 가 되어 반 개방 연결 상태를 RST(Reset)플래그로 초기화
      - 세그먼트를 보낸 장비와 연결을 맺고 있지 않는 경우
      - 잘못됐거나 부정확한 순서번호, 승인번호 필드를 가진 메시지를 수신한 경우
      - 연결을 기다리는 프로세스가 없는 포트로 SYN메시지를 받은 경우

연결의 수립, 관리, 종료

- TCP 연결 관리, 문제처리
  - 초기화 세그먼트 처리
    - RST 비트가 설정된 세그먼트를 받으면, 장비는 연결을 초기화하여 재 개방할 수 있도록 함
      - 장비가 LISTEN 상태 였다면 초기화 메시지를 무시
      - 현재 SYN-RECEVIED 상태지만 LISTEN 상태에 있었다면 LISTEN 상태로 되돌아감
      - 이외의 상황에서 초기화 메시지를 받으면 연결을 끊고 CLOSED 상태로 돌아감, 상위 계층에 연결이 끊김을 알림
  - TCP 킵얼라이브 메시지
    - TCP 소프트웨어에서 선택적으로 킵얼라이브 메시지를 구현할 수 있음
    - 표준에서는 아무것도 하지 않음

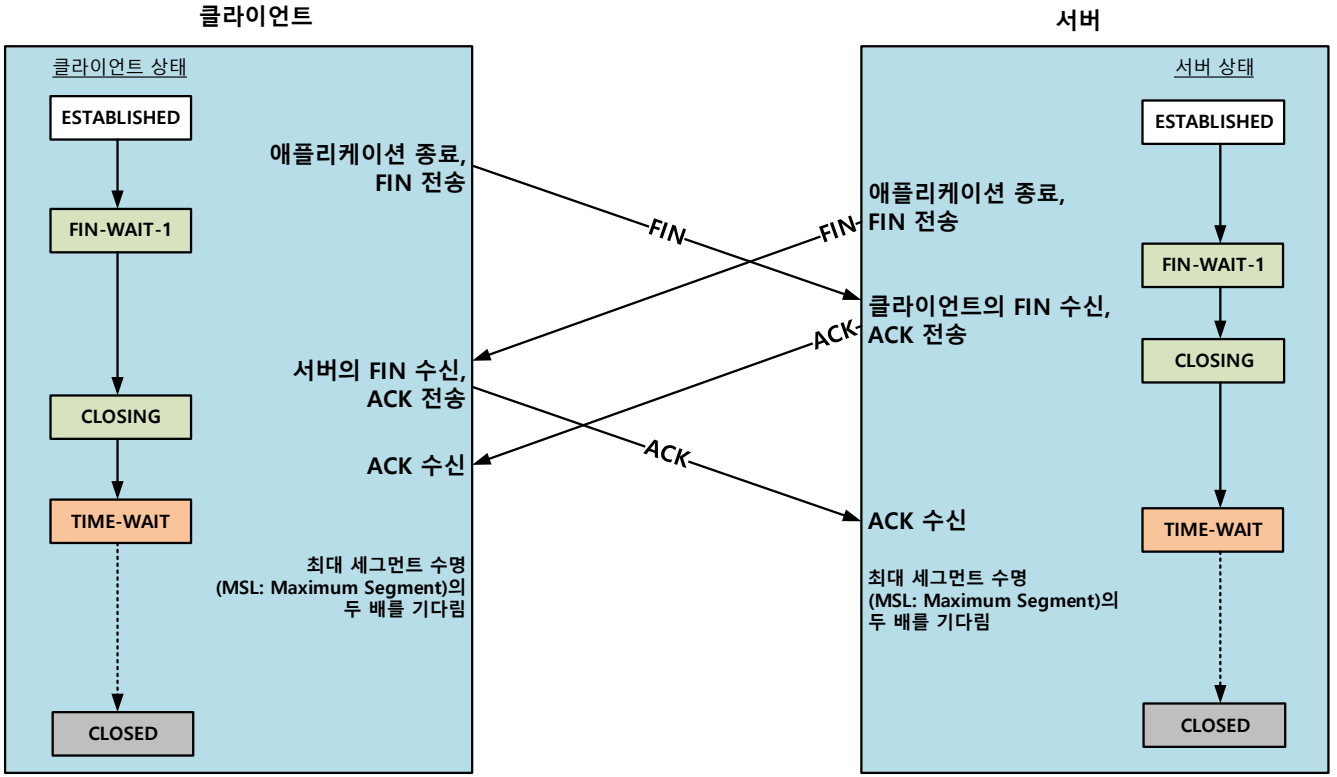
연결의 수립, 관리, 종료

- TCP 연결 종료
  - 연결 종료의 요구사항
    - 두 장비 모두 연결이 종료 되어야 함
    - 링크에 어떠한 데이터도 남지 않아야 함
  - 정상 연결 종료



연결의 수립, 관리, 종료

- TCP 연결 종료
  - TIME-WAIT
    - 상대 장비가 ACK를 받았다는 것을 확신하기 위해 충분한 시간을 둠, ACK가 사라졌다면 재전송
    - 한 연결의 종료와 다음 연결 시작에 간격을 둠
  - 동시 연결 종료



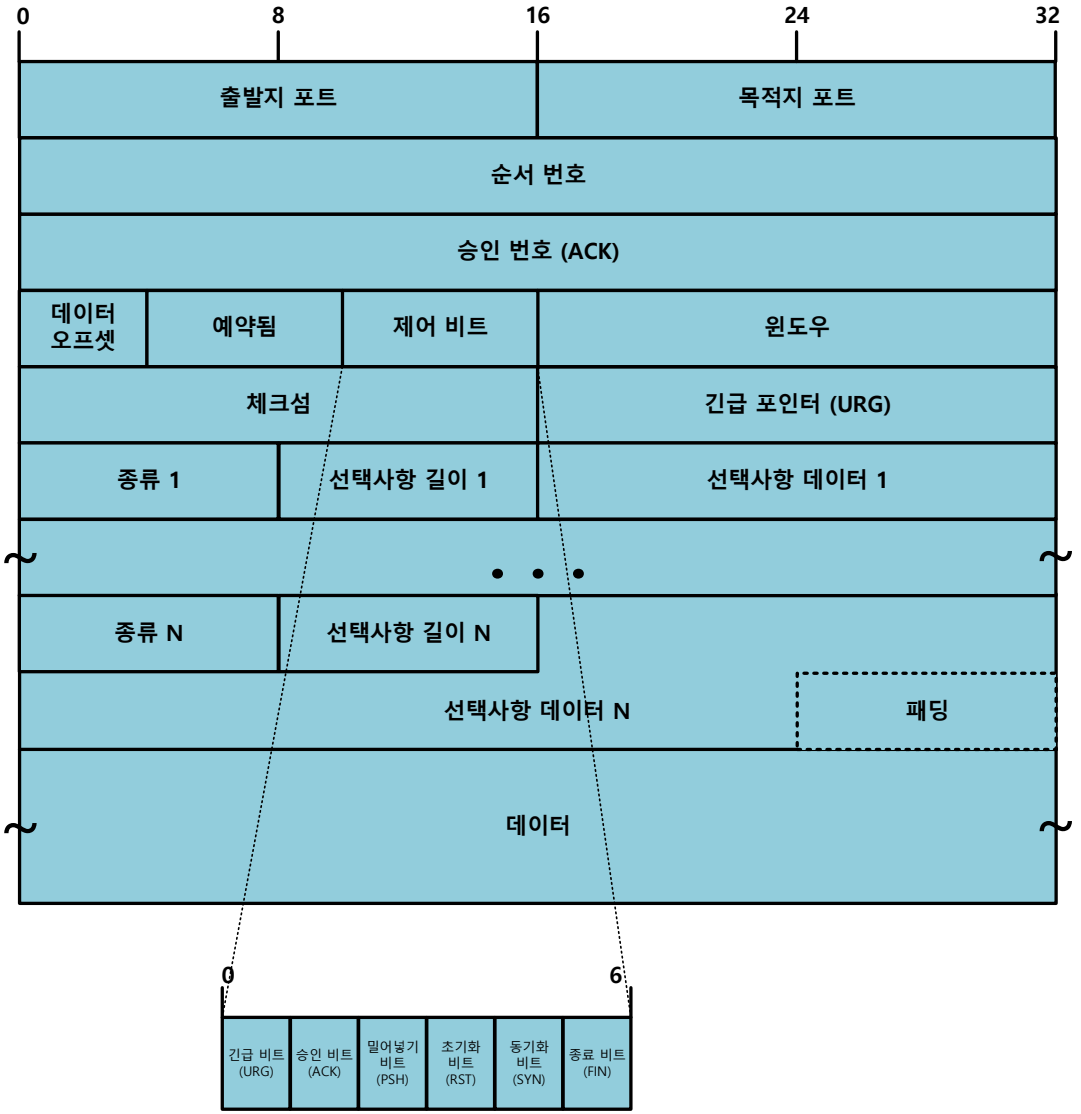
목 차

- TCP 개요
- TCP 원리와 일반 동작
- 연결의 수립, 관리 종료
- TCP 메시지 포맷과 데이터 송신
- TCP 신뢰성과 흐름제어

TCP 메시지 포맷, 데이터 송신

- TCP 메시지(세그먼트) 포맷

선택사항 종류	선택사항 길이	선택사항 데이터	의미
0	-	-	선택사항 목록의 끝, TCP 헤더의 끝과 맞지 않는 경우 사용
1	-	-	동작 없음, 선택사항 사이에 채워서 32비트 경계를 맞춤
2	4	최대 세그먼트 크기 값	연결요청(SYN)에서만 사용
3	3	윈도우 크기	매우 큰 윈도우를 사용하는 경우 이 데이터 값을 2의 지수로 사용
4	2	-	선택적 승인 허용
5	가변	선택적으로 승인하는 데이터 블록	
14	3	대체 체크섬 알고리즘	대체 체크섬 요청, 표준이 아닌 다른 알고리즘을 사용하도록 협상
15	가변	대체 체크섬	표준인 16비트로 표현할 수 없는 체크섬의 경우 이 필드를 사용



TCP 메시지 포맷, 데이터 송신

- TCP 체크섬 계산과 가상 헤더
  - TCP는 체크섬 계산에 가상헤더를 포함
    - 출발지, 목적지 주소가 일치해야 체크섬을 계산할 수 있음
    - 어떤 이유에 의해 다른 프로토콜을 통해 목적 TCP로 송신한 경우 검출
    - 세그먼트의 일부가 빠진다면 길이 값에 차이가 생겨 검출

TCP 메시지 포맷, 데이터 송신

- TCP 최대 세그먼트 길이 (MSS)
  - 한 세그먼트에 넣을 데이터의 크기는 현재 윈도우 상태를 고려 해야 함
  - 각 TCP 장비는 현재 윈도우 크기와 세그먼트 크기의 한계가 있음
  - MSS 선택
    - 과부하 관리: TCP 헤더 길이와 실제 데이터 의 길이 비율을 충분히 효율적으로 사용할 수 있는 길이로 설정
    - IP 단편화: TCP 세그먼트는 IP 데이터그램으로 묶이며, 최대 송신 단위(MTU)에 의한 추가적인 단편화가 없는 MSS값 선택

TCP 메시지 포맷, 데이터 송신

- TCP 최대 세그먼트 길이 (MSS)
  - TCP 기본 MSS
    - IP 네트워크 최소 MTU는 576바이트 이고, IP헤더 TCP헤더로 사용하는 바이트를 제외한 MSS 표준은 536 바이트
  - MSS 값 명시
    - 표준 MSS 크기가 모든 네트워크에 적합한 것은 아님
      - IPsec 등의 사용에 의한 헤더필드가 더 필요한 경우
      - 세그먼트가 지나는 네트워크의 MTU가 576 보다 큰 경우
    - 536보다 크거나 작은 MSS를 사용할 수 있고, 반드시 명시 해야 함

TCP 메시지 포맷, 데이터 송신

- TCP 슬라이딩 윈도우 송신과 승인
  - 전송 카테고리표

전송 카테고리	의미
1	전송했고 승인 받음
2	전송 했지만 승인 받지 못함
3	수신자는 준비 됨, 전송하지 못함
4	수신자가 준비되지 않음, 전송하지 못한 바이트

  - 수신 카테고리 표

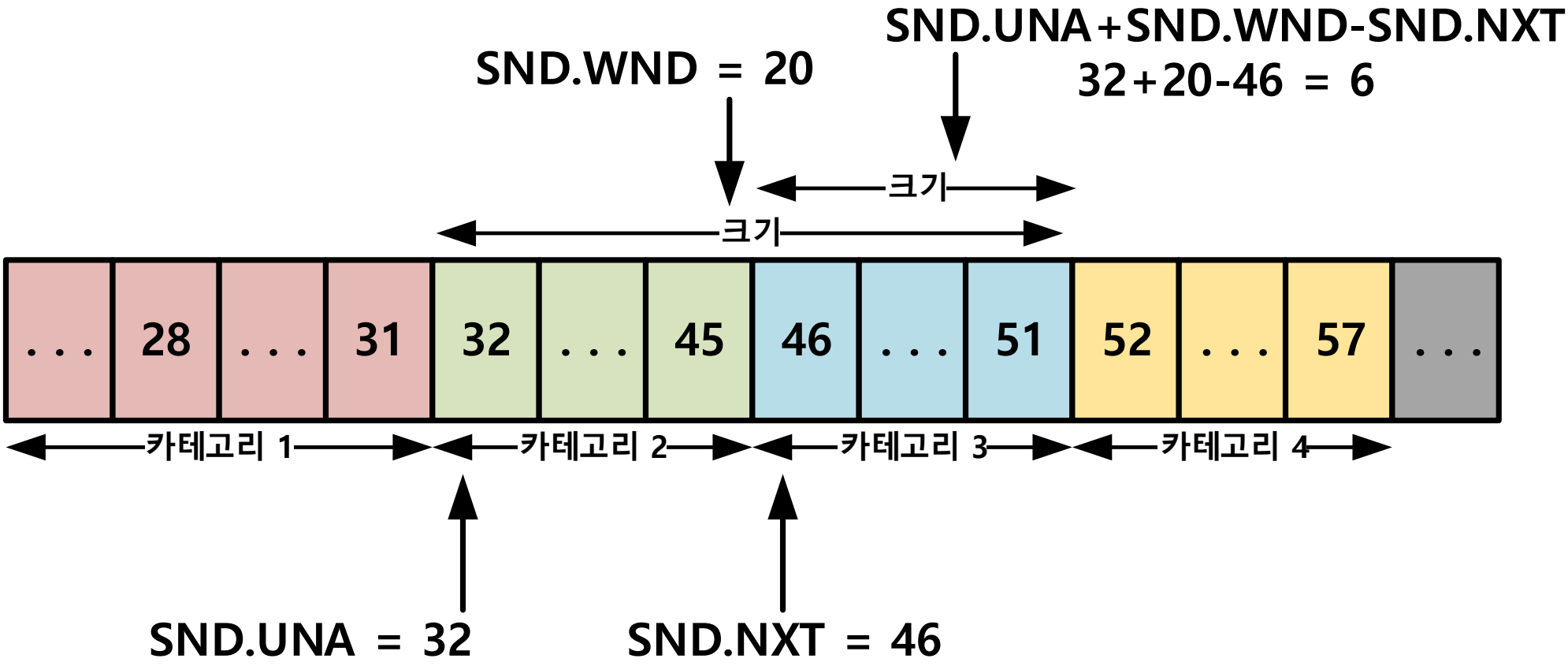
수신 카테고리	의미
1+2	수신했고 승인 받음
3	수신자는 준비됨, 아직 수신하지 못한 바이트
4	수신자가 준비되지 않았고 수신하지도 못한 바이트

TCP 메시지 포맷, 데이터 송신

- TCP 슬라이딩 윈도우 송신과 승인
  - 송신(SND) 포인터
    - 송신 비확인(SND.UNA)
      - 송신 했지만 아직 승인되지 않은 첫 번째 데이터의 순서번호
      - 전송 카테고리 2의 첫 번째 바이트를 가리킴
    - 송신 다음(SND.NXT)
      - 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

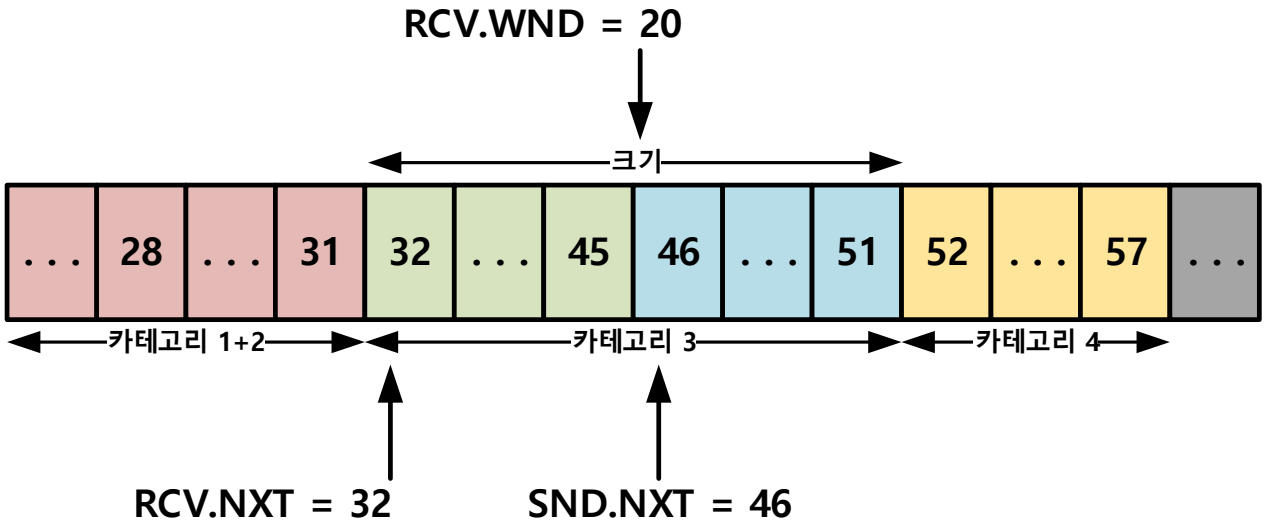
TCP 메시지 포맷, 데이터 송신

- TCP 슬라이딩 윈도우 송신과 승인
  - 송신(SND) 포인터



TCP 메시지 포맷, 데이터 송신

- TCP 슬라이딩 윈도우 송신과 승인
  - 수신(RCV) 포인터
    - 수신 다음(RCV.NXT)
      - 상대 장비에서 받으려고 하는 다음 바이트의 순서 번호
      - 수신 카테고리 3의 첫 번째 바이트를 가리킴
    - 수신 윈도우(RCV.WND)
      - 상대 장비에세 광고한 수신 윈도우의 크기
      - 대체로 이번 연결에서 사용하는 버퍼의 남은 양을 말함



TCP 메시지 포맷, 데이터 송신

- TCP 슬라이딩 윈도우 송신과 승인
  - 슬라이딩 윈도우 방식의 실제 복잡도
    - 중복 송신
      - 새로운 요청을 보내면서 승인
    - 다중 세그먼트 승인
      - 둘 이상의 세그먼트를 동시에 승인할 수 있음 (중복 승인)
    - 윈도우 크기 조절
      - 윈도우 크기 조절을 통한 흐름 제어 구현
    - 송신 실패
      - 송신된 세그먼트가 사라지는 경우 재송신

TCP 메시지 포맷, 데이터 송신

- TCP 슬라이딩 윈도우 송신과 승인
  - 슬라이딩 윈도우 방식의 실제 복잡도
    - 작은 윈도우 문제 회피
      - 너무 작은 세그먼트를 보내지 않기 위한 회피 기술이 필요
        - 작은 세그먼트를 보내면 성능이 떨어지고 바보 윈도우 증후군이 나타남
    - 혼잡 처리와 회피
      - 기본적인 슬라이딩 윈도우 방식을 수정하여 인터네트워크의 혼잡을 회피할 수 있도록 함

TCP 메시지 포맷, 데이터 송신

- TCP 밀어넣기 기능
  - 애플리케이션이 데이터를 즉시 보내야 하는 경우 TCP로 데이터를 보낸 후 TCP의 밀어넣기 명령을 사용
  - 밀어넣기 명령을 받은 TCP는 보낼수 있는 모든 데이터를 모아 세그먼트를 생성하고, 제어필드의 PSH비트를 설정후 전송

TCP 메시지 포맷, 데이터 송신

- TCP 긴급 기능
  - 전송중인 데이터에 문제가 있어 중지 하거나 우선순위의 차이를 주기 위한 기능
  - 긴급 기능을 사용하면 TCP는 특별한 세그먼트를 만들고 URG비트를 설정한 후 밀어넣기 기능도 같이 사용
    - 긴급 포인터를 이용해 세그먼트 내의 긴급데이터와 일반 데이터를 구분

목 차

- TCP 개요
- TCP 원리와 일반 동작
- 연결의 수립, 관리 종료
- TCP 메시지 포맷과 데이터 송신
- TCP 신뢰성과 흐름제어

TCP의 신뢰성과 흐름제어

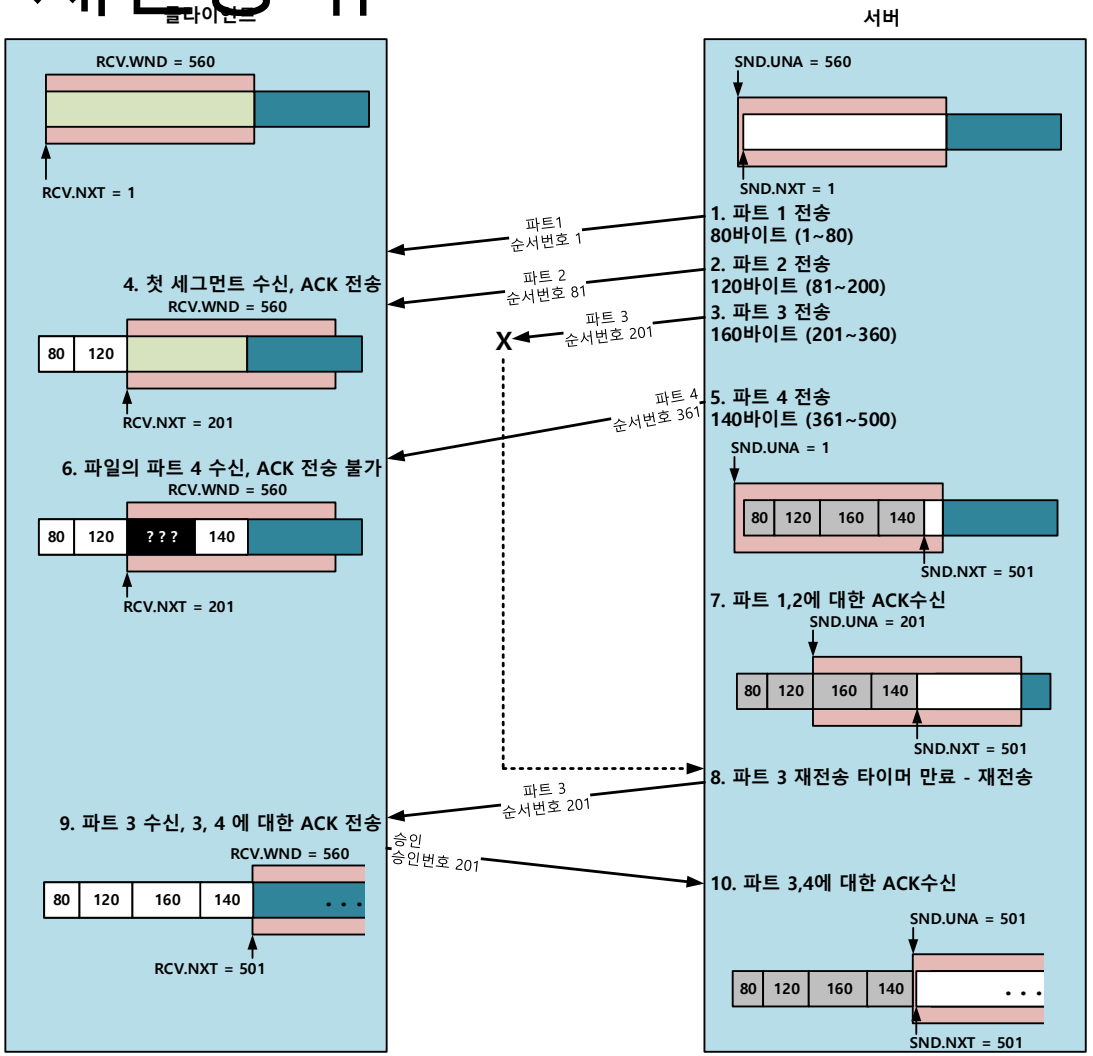
- TCP 세그먼트 재전송과 재전송 큐
  - 재전송 큐를 이용한 재전송 관리
    - 세그먼트를 전송하면 복사본을 재전송 큐에 삽입하고 타이머로 관리
    - 타이머 만료전에 승인이 온다면 해당 세그먼트를 재전송 큐에서 제거
    - 타이머가 만료되면 해당 세그먼트를 재전송

TCP의 신뢰성과 흐름제어

- TCP 세그먼트 재전송과 재전송 큐
  - 누적 승인을 이용한 재전송 관리
    - 승인 메시지를 보낼 때 수신 받은 마지막 순서번호+1 을 전송
    - 특정 세그먼트가 빠진 경우 승인을 보내지 않음

TCP의 신뢰성과 흐름제어

- TCP 세그먼트 재전송과 재전송 큐
  - 기본적인 TCP 재전송 그림



TCP의 신뢰성과 흐름제어

- TCP 비연속적 승인처리와 선택적 승인(SACK)
  - 고속 네트워크이거나 신뢰할 수 없는 물리네트워크의 경우 누적 승인으로 인한 문제가 발생
    - 누적승인은 승인번호필드 값보다 작은 모든 세그먼트를 승인하지만 세그먼트가 순서대로 오지 않을 수 있음
  - 비연속적 승인 처리
    - 타이머가 만료된 세그먼트만 재전송, 다른 세그먼트는 제대로 전송 되는 경우 효율적
    - 승인 받지 못한 모든 세그먼트 재전송, 재전송이 필요치 않은 경우 비효율적

TCP의 신뢰성과 흐름제어

- TCP 비연속적 승인처리와 선택적 승인(SACK)
  - 선택적 승인 (SACK: Selective Acknowledgment)
    - 연결 수립시 선택적 승인 허용 선택사항을 사용한 협상이 필요
    - 재전송 큐를 수정하여 세그먼트가 선택적으로 승인 되었을 경우 SACK비트를 1로 설정한 플래그를 가짐
    - 승인받지 못한 세그먼트의 타이머가 만료된 경우 SACK비트가 0인 세그먼트를 재전송

TCP의 신뢰성과 흐름제어

- TCP 적응형 재전송과 재전송 타이머
  - 재전송 타이머의 값은 왕복 시간(RTT: Round-Trip Time)보다 약간 크면 이상적
    - 연결 거리의 차이
      - 물리 네트워크의 신뢰도, 성능차이 실제 거리차이 등을 반영
    - 일시적인 지연시간과 변동성
      - 일시적인 인터네트워크 부하에 따른 지연시간 변화
  - RTT 계산에 기반한 적응형 재전송
    - 평균적인 지연시간을 계산
      - 새 $RTT = (\alpha \times$ 예전 $RTT) + ((1 - \alpha) \times$ 가장 최근에 측정한 $RTT)$
      - α는 부드럽게 하는 상수 1~0 값을 가짐

TCP의 신뢰성과 흐름제어

- TCP 적응형 재전송과 재전송 타이머
  - 모호한 승인
    - RTT의 계산의 개념은 세그먼트를 전송한 시간과 승인이 돌아온 시간의 차
    - 세그먼트를 전송하고, 타이머가 만료되어서 재 전송한 후 도착한 승인에 대한 RTT계산의 문제
  - RTT 계산 수정과 칸 알고리즘
    - Phil Karn 알고리즘
      - 재전송에 타이머 계산과 평균 RTT 계산을 분리 (모호한 승인 해결)
      - 세그먼트를 재전송 할 시 백오프를 통해 RTT를 두 배 증가시킨 시간값을 가짐
      - 재전송이 성공할 때까지 증가하지만 한계치가 있음
      - 재전송이 발생하지 않는 일반적인 상황이 되어 RTT를 계산할 때까지 백오프된 RTT 사용 (과부화된 상황 해결)

TCP의 신뢰성과 흐름제어

- TCP 윈도우 크기 조절과 흐름제어
  - 서버와 클라이언트는 서로 수신할 수 있는 윈도우 크기를 조절할 수 있음 (기본적인 흐름 제어)
    - 서버가 더 이상 수신하는 바이트를 처리하지 못할 경우 윈도우 크기를 0으로 만들 수 있음 (수신 윈도우 닫기)

TCP의 신뢰성과 흐름제어

- TCP 윈도우 관리 문제
  - 서버의 과부하에 의해 자체적인 윈도우 감소와 수신된 윈도우에 의한 실제적인 서버의 수신 윈도우 감소에 대한 정보를 클라이언트에게 전송
  - 클라이언트는 서버 자체적인 윈도우 감소 정보를 알려주는 메시지를 받기 전에 추가적인 세그먼트를 보내는 경우 데이터 손실이 생길 수 있고, 재전송 해야 함 (비효율)
  - 위와 같은 문제 때문에 TCP에서는 서버가 윈도우크기를 감소시키지 못하게 함

TCP의 신뢰성과 흐름제어

- 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 세그먼트를 서버에 보내 현재 윈도우 크기를 응답 받을 수 있음
      - 윈도우 개방 세그먼트가 클라이언트에게 전송된다고 보장할 수 없기 때문, 만약 전송되지 못하면 서버는 연결에 문제가 있다고 인식하고 연결 종료

TCP의 신뢰성과 흐름제어

- TCP 바보 윈도우 증후군 (SWS: Silly Window Syndrome)
  - 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

  - 바보 윈도우 증후군 회피 알고리즘
    - 수신자 SWS 회피
      - 윈도우의 끝을 움직이는 최소 단위를 지정
        - MSS파라미터 또는 버퍼의 절반중 작은 것으로 결정

TCP의 신뢰성과 흐름제어

- TCP 바보 윈도우 증후군 (SWS: Silly Window Syndrome)
  - 바보 윈도우 증후군 회피 알고리즘
    - 송신자 SWS회피와 네이클 알고리즘
      - 세그먼트를 적절한 크기가 되도록 기다리게 해서 송신 TCP에 제한을 두는 Nagle 알고리즘
        - 송신자가 보내고 승인 받지 못한 데이터가 없다면 애플리케이션이 원하는 방식으로 데이터를 전송(일반적이 상황)
        - 승인 받지 못한 데이터가 있는 경우 송신된 데이터가 모두 승인되거나 MSS크기 만큼 데이터가 모이지 않는다면 송신하지 않음

TCP의 신뢰성과 흐름제어

- TCP 혼잡 처리와 혼잡 회피 알고리즘
  - TCP는 3계층을 통해 실질적인 데이터를 전송
  - 인터네트워크에서 과부하가 생기면 세그먼트가 제대로 전송되지 않을수 있고, 승인받지 못한 데이터를 계속 재전송 하면 혼잡 붕괴(Congestion Collapse)현상이 발생

  - TCP 혼잡 처리 방식
    - 느린 시작
      - 두 장비가 연결이 수립되자 마자 데이터를 전송할 수 있음
      - 인터네트워크가 바쁜 상황이었다면 혼잡을 가중시키므로 현대 TCP에서 초기에 세그먼트를 보내는 속도를 제한
        - MSS크기의 세그먼트 하나만 송신하고 이후 송신 윈도우 크기만큼 보내거나 네트워크 혼잡을 알게 될 때까지 점점 더 많이 전송
        - 네트워크 혼잡을 알게 되면 혼잡 회피 기능을 사용

TCP의 신뢰성과 흐름제어

- TCP 혼잡 처리와 혼잡 회피 알고리즘
  - TCP 혼잡 처리 방식
    - 혼잡 회피
      - 혼잡이 발생한 경우 세그먼트를 보내는 속도를 재빨리 떨어트리는 알고리즘
      - 다시 혼잡이 일어나지 않도록 느린 시작 알고리즘을 이용해 송식속도를 증가
    - 빠른 재송신
      - 비연속적인 세그먼트 수신시 중간에 빠진 세그먼트를 알리는 메시지를 3번 이상 받으면 정상적인 재송신 큐 과정을 생략하고 사라진 세그먼트를 재전송

TCP의 신뢰성과 흐름제어

- TCP 혼잡 처리와 혼잡 회피 알고리즘
  - TCP 혼잡 처리 방식
    - 빠른 회복
      - 잃어버린 세그먼트를 재송신하기 위해 빠른 재송신을 사용할 때 장비는 혼잡 회피 방법을 이용해 전송율을 증가시킴
        - 특정 세그먼트만 도착하지 못했으므로 네트워크가 그다지 혼잡하지 않다고 판단할 수 있음

감사합니다!