표제

목차

[네트워크 3](#_Toc118358107)

[ TCP(Transmission Control Protocol) 3](#_Toc118358108)

[신뢰성 3](#_Toc118358109)

# 네트워크

## TCP(Transmission Control Protocol)

TCP/IP 프로토콜중 하나

OSI 계층모델 관점에서 4계층에 해당

양 종단 호스트 내 프로세스 상호 간에 신뢰적인 연결지향성 서비스를 제공

* IP의 비신뢰적인 최선형 서비스에다가, 신뢰적인 연결지향성 서비스를 추가 제공하게 됨.

### 신뢰성

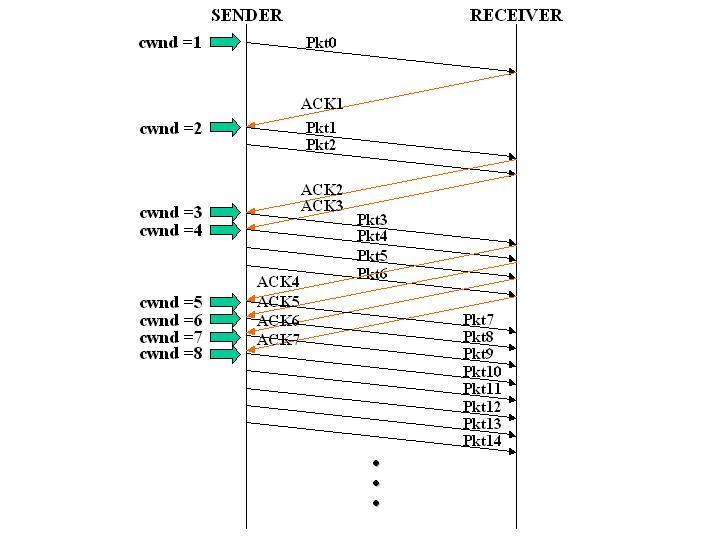
* + 흐름제어: 송신측과 수신측 사이의 데이터 처리 속도 차이(흐름)을 제어하기 위한 기법. 데이터처리 속도를 조절하여 수신자 버퍼 오버플로를 방지

1. Stop And Wait(정지 – 대기): 매번 전송한 패킷에 대한 응답(ACK)을 받아야 다음 패킷을 전송가능 하다.
2. Sliding Window(슬라이딩 윈도우): 1번의 방식은 패킷이 하나씩만 전송되므로 비효율. 그렇기에 2번을 씀. 수신측에서 설정한 윈도우 크기만큼 송신쪽에서 확인없이 전송함. 전송받은게 확인이 되면 다음 윈도우 크기로 확장되는 형태
   * 오류제어: 오류를 검출하여 재전송 ARQ(Auto Repeat Request)를 사용해 프레임이 손상되거나 손실되면, 재전송을 통해 오류를 복구. 흐름기법과 연관이 있음.
3. Stop And Wait ARQ: 흐름제어에서의 ACK나 NAK(Negative)를 받고, NAK면 다시 보내는 것. 다만 ACK NAK 둘 다 안 오면 일정 시간 뒤에 타임아웃으로 데이터를 재전송 함.
4. Go-Back-N ARQ: 슬라이딩 윈도우 방법에서의 신뢰성 보장인 것 같음. 패킷 손실이 난 이후의 모든 프레임을 폐기하고 싹 다 재전송함.

테이블이(가) 표시된 사진

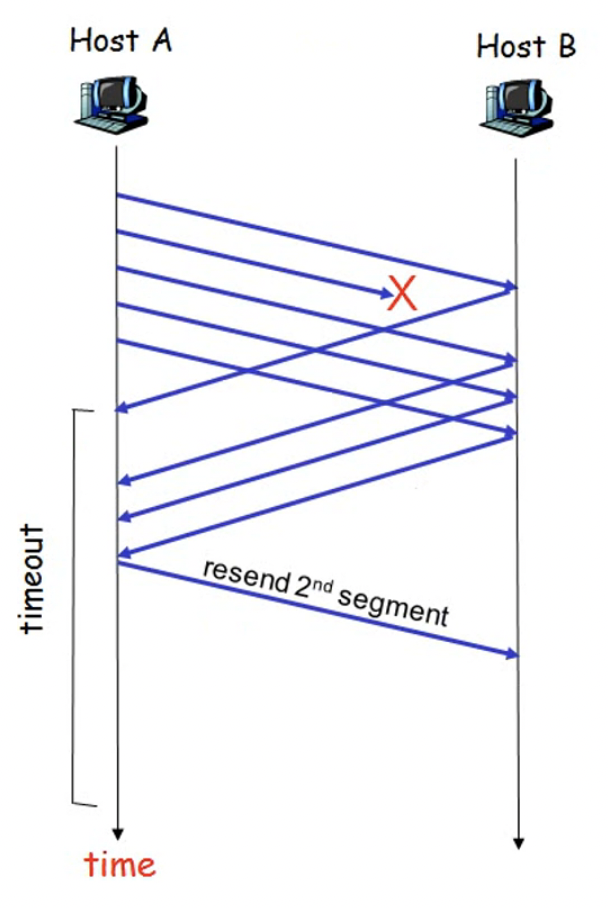
자동 생성된 설명

1. SR(Selectvie-Rejct) ARQ: GBn의 폐기하는 단점을 보완. 손실된것만 보냄. 그대신 구조가 복잡하고 프레임 순서 꼬인거 재배열하기 위한 버퍼가 필요함.
   * 혼잡제어: 송신측의 데이터 전달과 네트워크 데이터 처리속도를 해결하기 위한 기법 네트워크 혼잡을 피하기 위해 송신측에서 전송 속도를 제어하는 것.
2. AIMD(Additive Increase Multicative Decrease): 합 증가/곱 감소 알고리즘. 패킷이 잘 도착되면 Window size를 1씩 증가시켜버림 유실이 되면 window size를 반으로 줄임.



단점으로는 네트워크가 혼잡해지고 나서야 대역폭을 줄이는 방식임. 미리 대응을 못함.

1. Slow Start: AIMD가 네트워크 수용량에선 효율적이지만, 초반 전송속도 올리는 것이 느림. 얘도 1부터 시작하긴 하지만 2배로 늘려줌. 혼잡현상이 발생하면 윈도우 크기를 1로 떨어트림
2. Fast Retransmit(빠른 재전송): 보통 다음 패킷의 순번을 ACK패킷에 실어 보내는데, 이때 순서가 맞지 않는 패킷을 3개 받으면 유실된 패킷의 Time Out 시간이 지나지 않았어도 해당 패킷을 재전송 시켜주게 된다. 아래 사진에서는 2번 패킷의 TimeOut시간이 되지 않았음에도 빠른 재전송 기법에 의해 재전송 되는 것을 보여줌.



1. Fast Recovery(빠른 회복): 혼잡상태시 윈도우 크기를 1로 줄이지 않고 반으로 줄이고 지수형태로 늘어나던 것을 선형 증가하는 방식. 이걸 적용하면 혼잡을 한번 겪으면 AIMD 방식으로 동작.