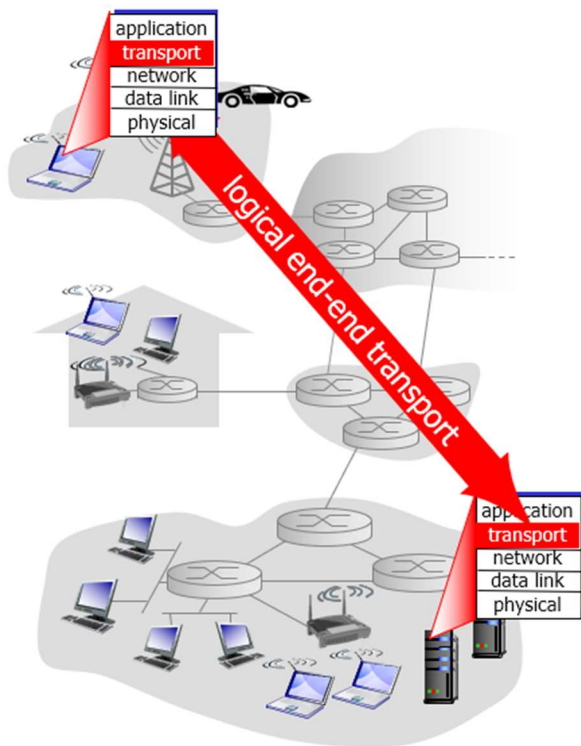


3장에서는 트랜스포트 계층에 대해서 배웠다.

➤ 트랜스포트 계층 서비스 및 개요



트랜스포트 계층 프로토콜은 각기 다른 호스트에서 동작하는 애플리케이션 프로세스간의 논리적 통신(logical communication)을 제공한다. 왼쪽의 그림과 같이 트랜스포트 계층 프로토콜은 네트워크 라우터가 아닌 종단 시스템에서 구현이 되는데 송신 측의 트랜스포트 계층은 송신 애플리케이션 프로세스로부터 수신한 메시지를 인터넷 용어로는 트랜스포트 계층 세그먼트(segment)라고 알려진 트랜스포트 계층 패킷으로 변환한다. 이후 수신 애플리케이션에서 세그먼트 내부의 데이터를 이용할 수 있도록 수신된 세그먼트를 처리한다.

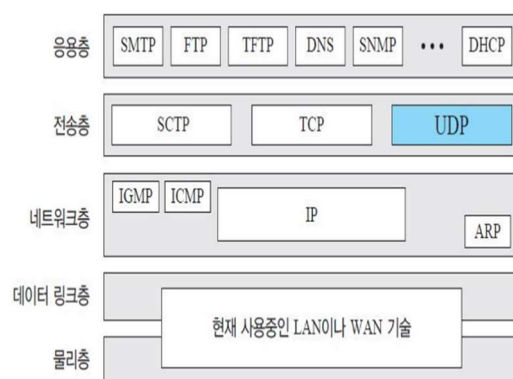
네트워크 애플리케이션에는 하나 이상의 트랜스포트 계층 프로토콜을 사용할 수 있다.

예를 들어, 인터넷은 TCP와 UDP라는 두 가지 프로토콜을 갖고 있다. 이러한 프로토콜은 서비스를 요청하는 애플리케이션에게 트랜스포트 계층에서 각기 다른 서비스 집합을 제공한다.

■ 트랜스포트 계층과 네트워크 계층 사이의 관계

트랜스포트 계층 프로토콜은 각기 다른 호스트에서 동작하는 프로세스들 사이의 논리적 통신을 제공하지만, 네트워크 계층 프로토콜은 호스트들 사이의 논리적 통신을 제공한다.

■ 인터넷 트랜스포트 계층의 개요



인터넷은 두 가지 구별되는 트랜스포트 계층 프로토콜을 제공하고 그 중에 하나는 비신뢰적이고 비연결형인 서비스를 요청한 애플리케이션에게 제공하는 UDP(User Datagram Protocol)이고, 다른 하나는 신뢰적이고 연결 지향형 서비스를 요청한 애플리케이션에게 제공하는 TCP(Transmission Control Protocol)이다. 또한 IP(Internet Protocol)이라는 프로토콜이 있는데 서비스 모델은 호스트들 간에

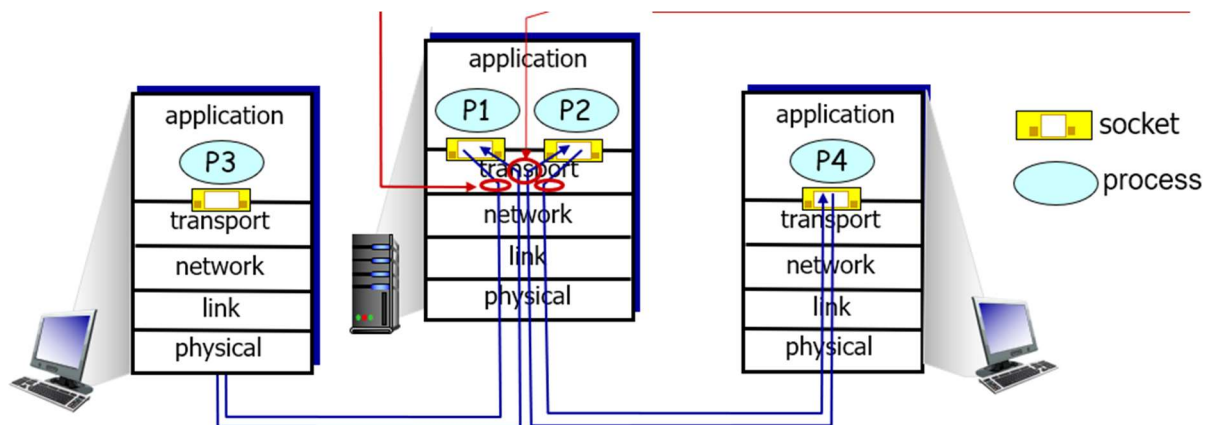
논리적 통신을 제공하는 최선형 전달 서비스이다. 이것은 IP가 통신하는 호스트들 간에

세그먼트를 전달하기 위해 최대한 노력하지만, 어떤 보장을 하지 않는다는 것을 의미, 이러한 이유로 비신뢰적인 서비스라고도 부른다.

UDP는 하나의 프로세스에 의해 전송된 데이터가 손상되지 않고 목적지 프로세스에 도착한다는 것을 보장하지 않는다. 반면에, TCP는 애플리케이션에 몇 가지 추가적인 서비스를 제공한다. 가장 먼저 TCP는 신뢰적인 데이터 전송을 제공하며, 흐름 제어, 순서 번호, 확인 응답, 타이머를 사용함으로써 TCP는 송신하는 프로세스로부터 수신하는 프로세스에게 데이터가 순서대로 정확하게 전달되도록 확실하게 한다.

또한 TCP는 혼잡 제어(congestion control)를 사용한다. 혼잡 제어는 인터넷에 대한 통상적인 서비스처럼 야기한 애플리케이션에게 제공되는 특정 서비스가 아니라, 전체를 위한 일반 서비스다. 한 TCP연결이 과도한 양의 트래픽으로 모든 통신하는 호스트들 사이의 스위치와 링크를 혼잡하게 하는 것을 방지하는 것이 TCP 혼잡 제어이다.

➤ 다중화와 역다중화



목적지 호스트에서의 트랜스포트 계층은 바로 아래의 네트워크 계층으로부터 세그먼트를 수신한다.

수신 측 호스트가 수신한 트랜스포트 계층 세그먼트를 어떻게 적절한 소켓으로 향하게 하는지에 대해서 트랜스포트 계층 세그먼트의 데이터를 올바른 소켓으로 전달하는 작업을 역다중화라고 한다. 출발지 호스트에서 소켓으로부터 데이터를 모으고, 이에 대한 세그먼트를 생성하기 위해 각 데이터에 헤더 정보로 캡슐화하고, 그 세그먼트들을 네트워크 계층으로 전달하는 작업을 다중화라고 한다.

➤ 비연결형 트랜스포트:UDP

■ UDP 세그먼트 구조

DNS에 대한 데이터 필드에는 질의 메시지나 응답 메시지를 포함하고, 헤더는 2바이트씩 구성된 단 4개의 필드만을 갖는다.

■ UDP 체크섬

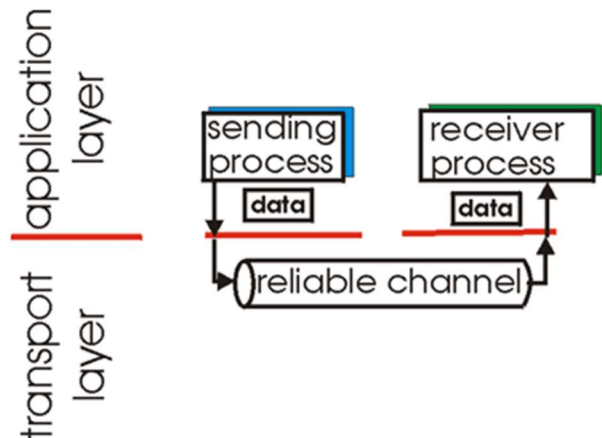
UDP 체크섬은 오류 검출을 위한 것이다. 즉 세그먼트가 출발지로부터 목적지로 이동했을 때, UDP 세그먼트 안의 비트에 대한 변경사항이 있는지 검사하는 것이다. UDP는 오류 검사를 제공하지만, 오류를 회복하기 위한 어떤 일도 하지 않는다. 일부 UDP 구현에서는 손상된 세그먼트를 그냥 버리기도 하고, 다른 구현에서는 손상된 세그먼트를 경고와 함께 애플리케이션에게 넘겨주기도 한다.

➤ 신뢰적인 데이터 전송의 원리

■ 신뢰적인 데이터 전송 프로토콜의 구축

긍정 확인응답(positive acknowledgment), 부정 확인응답(negative acknowledgment)

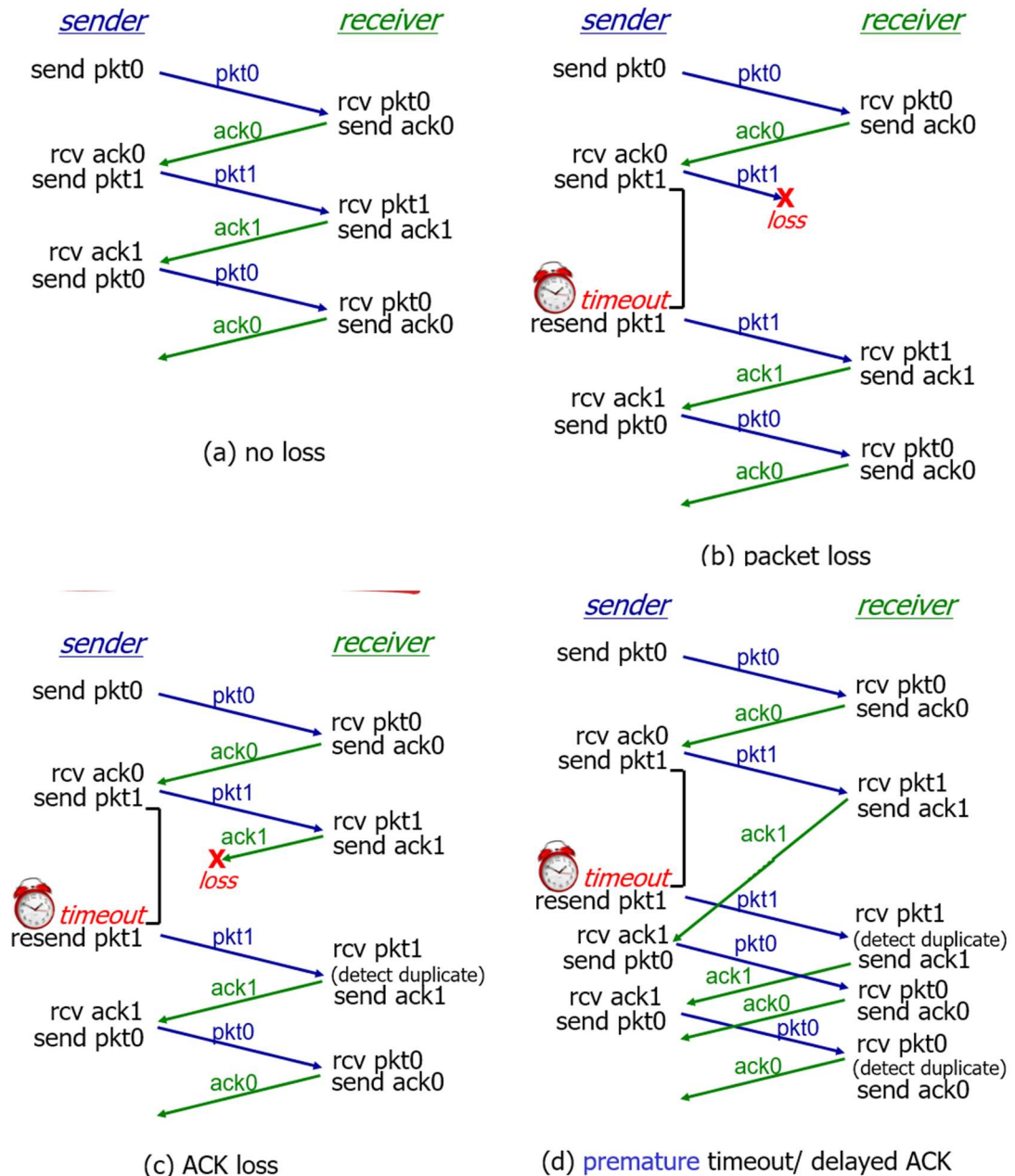
자동 재전송 요구(Automatic Repeat request)



오류 검출: 비트 오류가 발생했을 때 수신자가 검출할 수 있는 기능이 필요하다. 앞에서 설명한 UDP는 이러한 목적을 위해 인터넷 체크섬 필드를 사용한다. 수신자 피드백: 송신자와 수신자가 일반적으로 수천 킬로미터 떨어진 각기 다른 종단 시스템에서 동작하므로, 송신자가 수신자의 상태를 알기 위한 유일한 방법은 수신자가 송신자에게 피드백을 제공하는 것이다. 이에 사용되는 것의 위의 긍정 확인 응답과 부정 확인 응답 그리고 재

전송 요구이다.

■ 파이프라이닝된 신뢰적인 데이터 전송 프로토콜



이와 같이 패킷 손실과 timeout에 대해서 그림으로 보았듯이 패킷을 실제로 전송하는데 필요한 시간이 존재한다. 따라서 길지도 짧지도 않은 timeout을 설정해야 좋은 구성을 만들 수 있는 것이다. 위의 문제점인 전송 후 대기 방식으로 동작하는 대신에 송진자에게 확인 응답을 기다리지 않고 여러 패킷을 전송하도록 허용하는 것에 대해서 파이프라이닝(pipelining)이라고 한다.

■ GBN

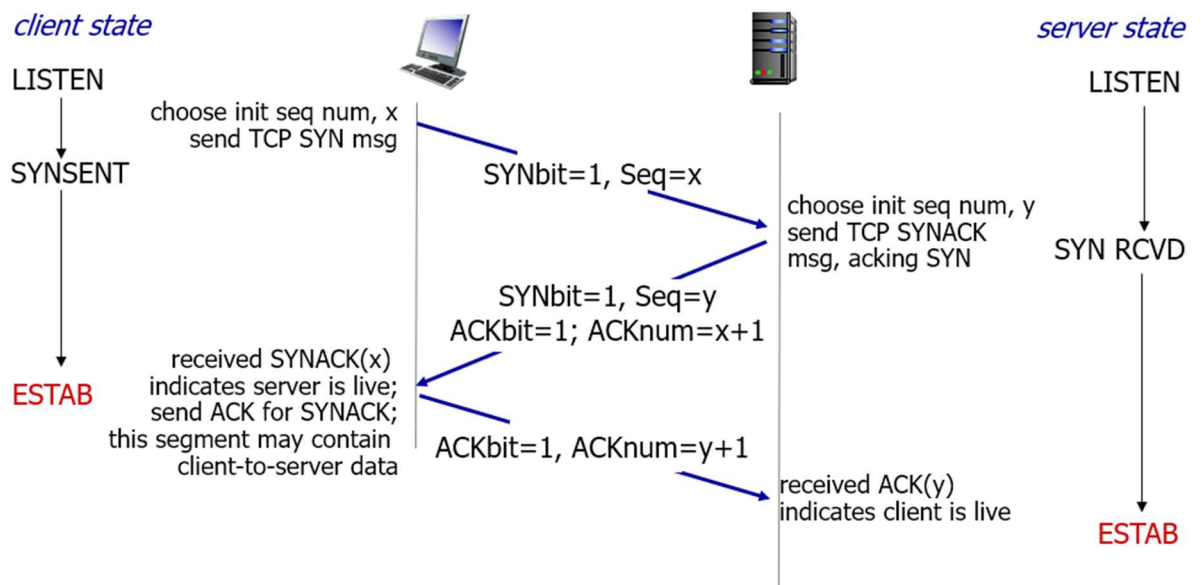
GBN(Go-Back-N)프로토콜에서 송신자는 확인응답을 기다리지 않고 여러 패킷을 전송할 수 있다. 그러나 파이프라인에서 확인응답이 안 된 패킷의 최대 허용 수 N보다 크지 말아야 한다.

■ SR

SR(Selective Repeat)프로토콜은 수신자에서 오류가 발생한 패킷을 수신했다고 의심되는 패킷만을 송신자가 다시 전송하므로 불필요한 재전송을 피한다. 따라서 상황에 맞게 GBN과 SR을 선택적으로 사용해야 한다는 것이다.

➤ 연결지향형 트랜스포트: TCP

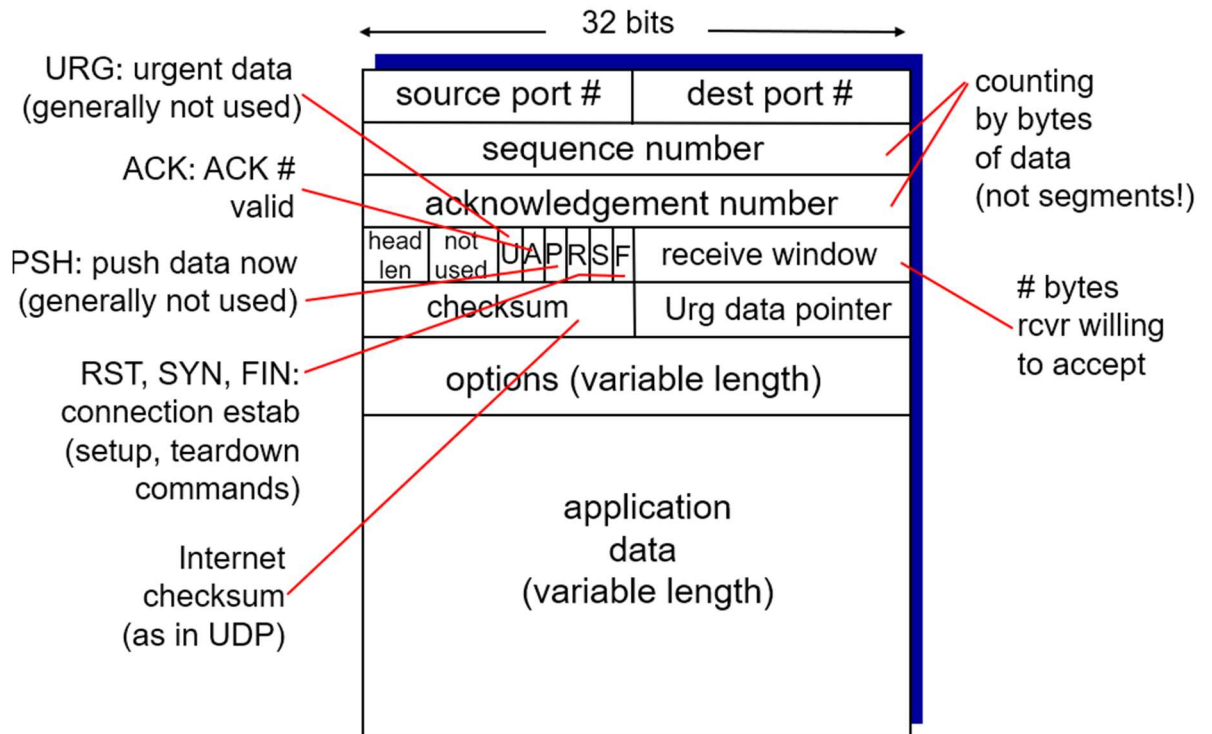
■ TCP 연결



TCP는 애플리케이션 프로세스가 데이터를 다른 프로세스에게 보내기 전에, 두 프로세스가 서로 '핸드셰이크'를 먼저 해야 하므로 연결지향형(connection-oriented)이다. 즉, 데이터 전송을 보장하는 파라미터들은 각자 설정하기 위한 어떤 사전 세그먼트들을 보내야 한다는 것이다.

TCP는 연결을 위해서 프로세스간의 세 방향 핸드셰이크(Three-way handshake)라 하는 연결 절차후 서로 데이터를 보낸다.

■ TCP 세그먼트 구조



32비트 순서 번호 필드와 32 비트 확인 응답 번호 필드가 존재하고, 16비트 수신 윈도우, 4 비트 헤더 길이 필드, 옵션 필드, 플래그 필드가 있다.

■ 왕복 시간(RTT) 예측과 타임아웃

손실 세그먼트를 발견하기 위해 타임아웃/재전송 메커니즘을 사용하는데 개념은 간단해 같지만, 실질적으로는 타임아웃은 전성된 시간부터 긍정 확인이 응답될 때까지의 시간인 연결의 왕복 시간 RTT보다 좀 커야 한다. 그렇지 않으면 불필요한 재전송이 발생할 것이고, 이를 미연에 방지하기 위해서는 얼마나 크게 구성을 할지 고민해야 하는 부분이다.

■ 흐름 제어

TCP 송신자가 수신자의 버퍼를 오버플로시키는 것을 방지하기 위해 애플리케이션에게 흐름 제어 서비스를 제공하는 것이다.

■ TCP 연결 관리

TCP연결이 어떻게 설정되고 해제되는지에 대해서 알아보는 부분

➤ 혼잡 제어의 원리

■ 혼잡의 원인과 비용

- 망에 입력되는 트래픽 양이 망이 처리할 수 있는 한도를 초과할 경우 발생 (그래프 제공)
- 인터넷 망에서의 트래픽의 흐름이 일정하지 않다.
- 해결책
 - 네트워크 망의 트래픽 처리용량을 최대 트래픽 처리량보다 크게 ➔ 현실적인 어려움(비용 등)
 - TCP에서 완전한 해결이 어렵다. 왜냐하면 하위 레벨인 네트워크 전반적인 문제이기 때문이다.

■ 혼잡 제어에 대한 접근법

혼잡 제어 방식

- **예방적 (Preventive)** 혼잡 제어
 - 사전에 네트워크 망에 입력되는 되는 트래픽 양을 조절
 - 두 단계
 - 사전에 전송할 데이터 양을 정한다. 망 사업자와 사용자 계약
 - 망 사업자는 사용자가 약속을 준수하는 지 감시
 - 라우터에서 패킷을 전송할 때, 사전에 약속된 계약에 따라 패킷의 전송 순위를 결정 (priority control)
- **반응적 (Reactive)** 혼잡 제어
 - 네트워크에서 체증이 발생할 때마다 트래픽 양을 감소
 - 패킷의 지연시간, 라우터의 버퍼의 길이 등을 계속 측정하여 혼잡 정도를 초기에 발견하고 제어

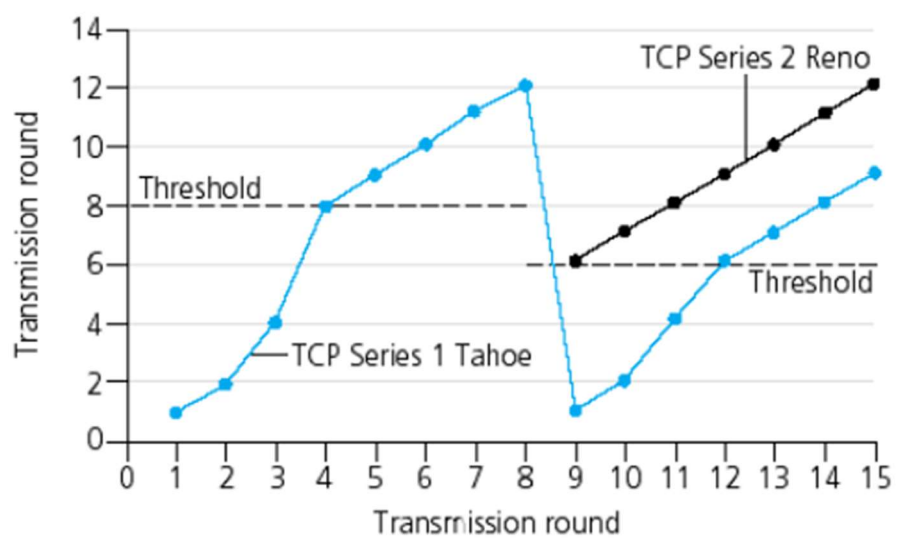
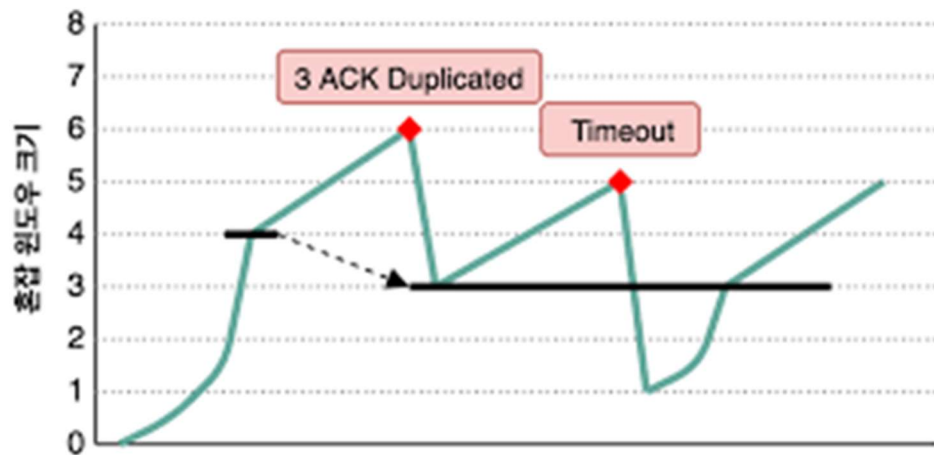
➤ TCP 혼잡 제어

■ 전통적인 TCP의 혼잡 제어

- ◆ 슬로 스타트
- ◆ 혼잡 회피
- ◆ 빠른 회복

방식이 있는데,

가법적 증가, 승법적 감소의 혼잡 제어 형식으로 부르기도 한다.



- 네트워크 지원 명시적 혼잡 알림과 지연 기반 혼잡 제어

명시적 혼잡 알림, 지연 기반 혼잡 제어가 있다.

- 공평성

공평성과 UDP, 공평성과 병렬 TCP 연결

- 트랜스포트 계층 기능의 발전

- QUIC: 빠른 UDP 인터넷 연결