

9장. TCP의 이해

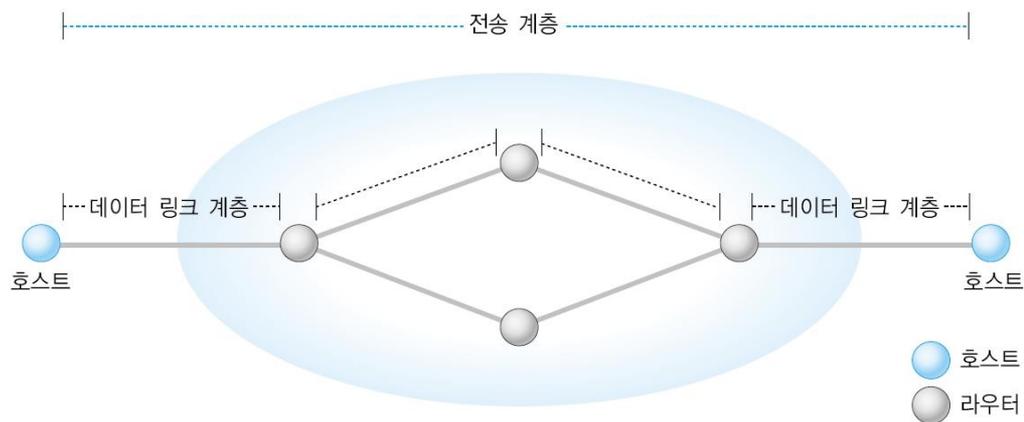
컴퓨터 네트워크

1. 전송 계층의 기능

- 학습 목표
 - 전송 계층 프로토콜의 기능 이해
 - 전송 계층 프로토콜 설계 과정에서 고려할 사항 이해

전송계층 프로토콜 개요

- 데이터 링크 계층과 유사
 - 오류 제어, 흐름 제어, 데이터 순서화 등 제공
- 데이터 링크 계층과의 차이점
 - 데이터 링크 계층은 물리적으로 1:1 연결된 호스트 사이의 전송
 - 전송 계층은 논리적으로 1:1 연결된 호스트 사이의 전송



[그림 9-1] 전송 계층과 데이터 링크 계층의 차이

전송계층의 주요 기능 (1)

- 흐름 제어
 - 기본 목적을 비롯해 유사한 점이 많지만 계층 2와는 다른 버퍼 관리 방법 필요
 - 수신자가 송신자의 전송 속도보다 느리게 수신 -> 버퍼 용량 초과로 데이터 분실 -> 타임아웃 기능을 통한 재전송 유발
 - 수신자가 슬라이딩 윈도우 프로토콜의 윈도우 하단 값을 조정
 - 송신자가 보낼 수 있는 패킷의 한계 설정
- 오류 제어
 - 데이터 변형, 데이터 분실 오류시 재전송 기능으로 복구
 - 수신자의 요구(NAK) 또는 송신자의 판단(타임 아웃)
 - 선로 오류보다 각 계층의 소프트웨어 동작 과정 중에 분실하는 경우가 대부분
 - 네트워크 계층의 기능적 한계
 - 잘못된 위치/경로 정보

전송계층의 주요 기능 (2)

- 분할과 병합
 - 상위계층에서 요구한 데이터 크기가 전송계층에서 처리할 수 있는 크기보다 큰 경우
 - 분할(Segmentation)
 - 병합(Reassembly)
- 서비스 프리미티브
 - 네트워크 계층의 경우 대부분 비연결형 서비스 프리미티브가 정의
 - 전송계층의 경우 비연결형과 신뢰성이 향상된 연결형 서비스 모두 제공

전송계층 설계 시 고려사항 (1)

- 주소 표현
 - 예 : 네트워크 계층의 호스트 IP 주소 + 전송 계층의 포트 번호 조합
 - TSAP(Transport Service Access Point)
 - 구조적 표현
 - 여러 개의 계층적 필드로 구분 (각 필드는 상하 계층 관계를 나타냄)
 - 예: 대한민국:서울:한국대학교:정보통신공학과:네트워크연구실:홍길동:50
 - 포트번호 : 하나의 컴퓨터에 다수의 포트 존재(통신 프로세스 구별)
 - 비구조적 표현
 - 값만 해석해서는 논리적인 위치 파악 불가
 - 값 자체에 위치가 아닌, 다른 중요한 정보를 담고 있는 경우
 - 초등학교의 반 번호, 일련 번호
 - IP 주소
 - 네트워크와 호스트의 계층적인 특성
 - 도메인 주소는 구조적 특징
 - www.korea.co.kr
 - 위치 정보와 관련해서는 비구조적 특징
 - 주소를 바탕으로 위치를 파악할 수 없음

전송계층 설계 시 고려사항 (2)

- 멀티플렉싱(Multiplexing)

- 개별적으로 설정된 TPDU(Transport Protocol Data Unity)의 주소가 동일한 경우 하나의 가상회선에 실어 보내는 것이 유리

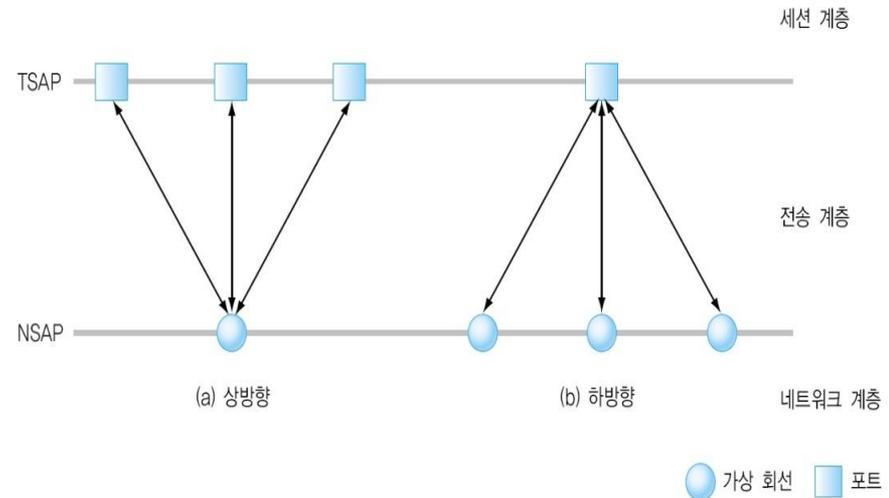
- 종류

- 상향

- 다수의 전송 계층 연결을 하나의 네트워크 계층에서 하나의 연결로 지원
- 네트워크 계층의 가상 회선 연결의 개수를 줄일 수 있음

- 하향

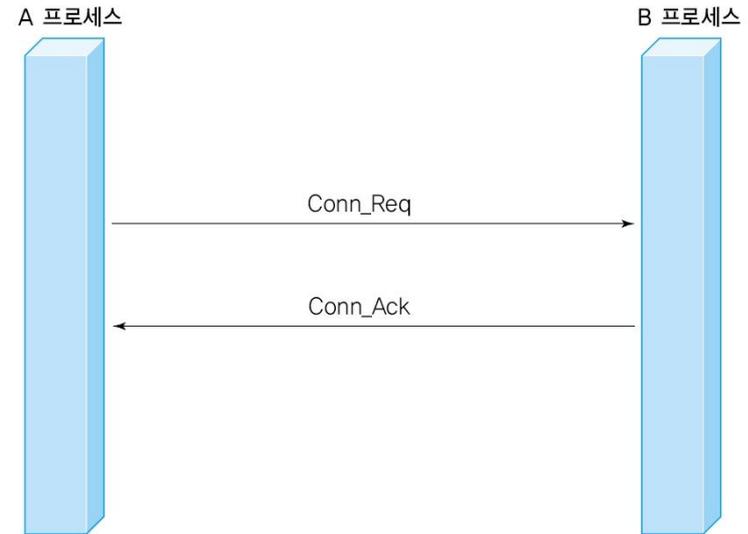
- 하나의 전송 연결에 대하여 네트워크 계층에서 다수의 가상회선을 지원
- 멀티미디어 전송에 유리: 음성, 영상, 자막 등을 별도의 가상 회선으로 처리



[그림 9-2] 멀티플렉싱

전송계층 설계 시 고려사항 (3)

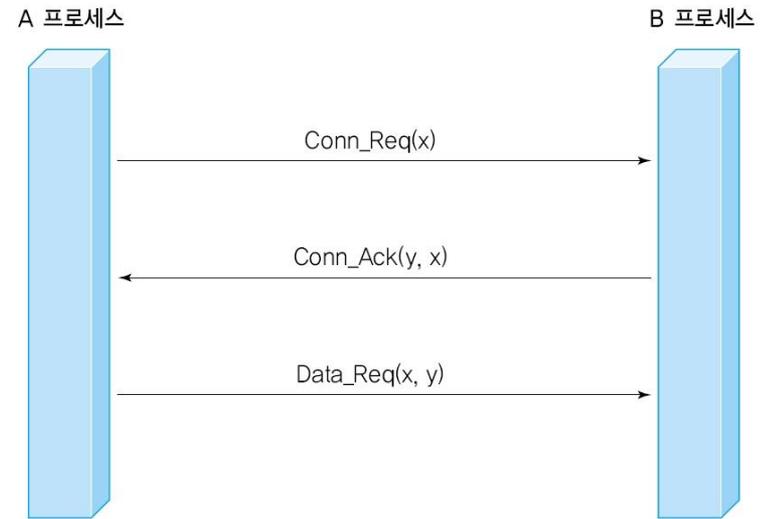
- 연결 설정
 - 개념적으로 양자 합의가 필요
 - 한쪽의 연결 설정 요구: Conn_Req
 - 상대방의 연결 수락 응답: Conn_Ack
 - 2단계 연결설정
 - 최소한의 단계
 - 수신단에서 거부 가능
 - 실제로 프리미티브 전달 과정에서의 분실, 변형, 복사 가능성이 있으므로 이러한 문제들을 고려해야 함



[그림 9-3] 개념적 연결 설정 절차 개념

전송계층 설계 시 고려사항 (4)

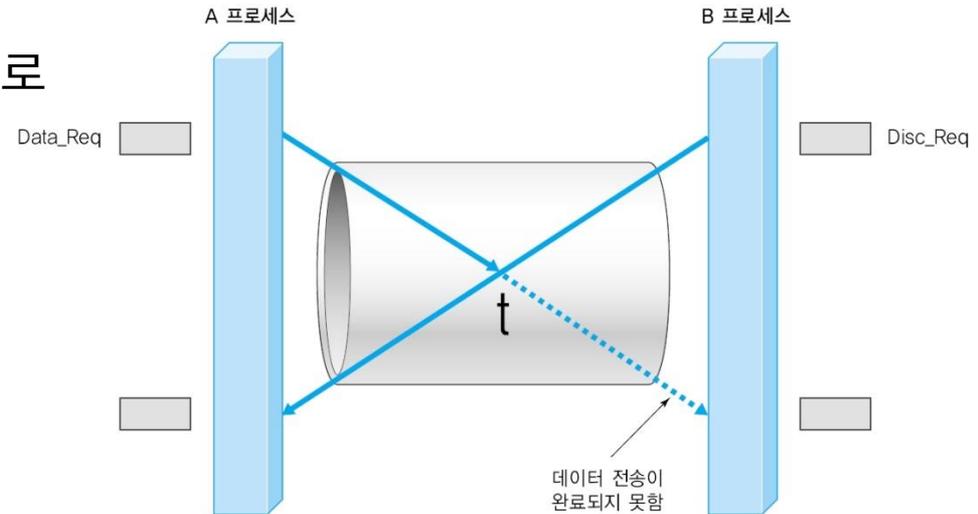
- 연결 설정 (계속)
 - 3단계 설정 (3-Way Handshake)
 - 세 번째의 Data_Req는 Conn_Ack에 대한 응답 기능도 수행
 - 보낼 데이터가 없는 경우 Conn_Ack(y,x)에 대한 응답을 따로 보내야 함



[그림 9-4] 3단계 설정 절차

전송계층 설계 시 고려사항 (5)

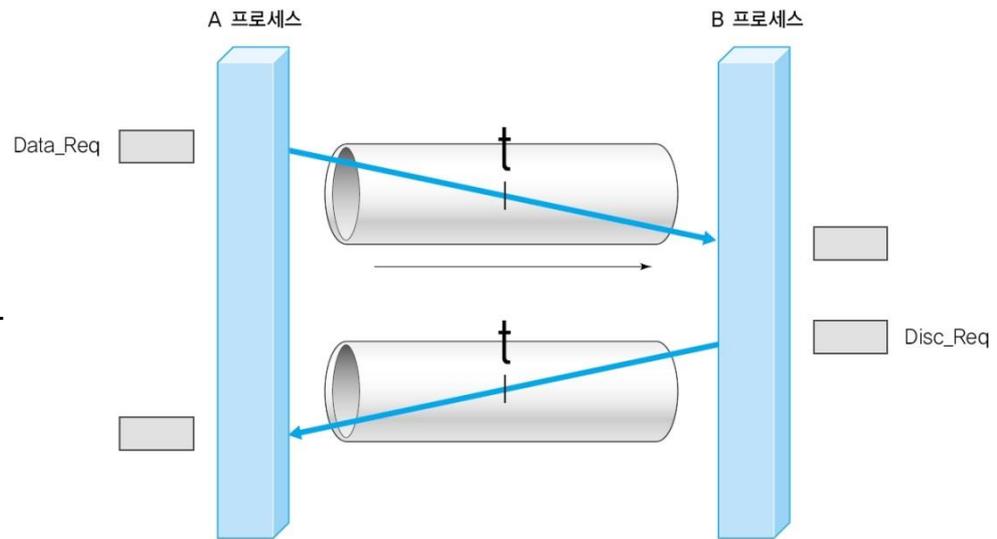
- 연결 해제
 - 일방적 연결 해제
 - 어느 한쪽의 Disc_req 로 전체 연결 해제
 - 상대방이 송신할 데이터가 남아있어도 종료



[그림 9-5] 일방적 연결 해제의 절차

전송계층 설계 시 고려사항 (6)

- 연결 해제 (계속)
 - 점진적 연결 해제
 - 논리적으로 2개의 단방향 연결을 지원
 - 어느 한쪽에서 해제 요청을 하는 경우 그 프로세스가 송신하는 연결만 해제
 - 양쪽 모두 Disc_req를 전송해야 해제가 됨



[그림 9-6] 점진적 연결 해제의 절차

2. TCP

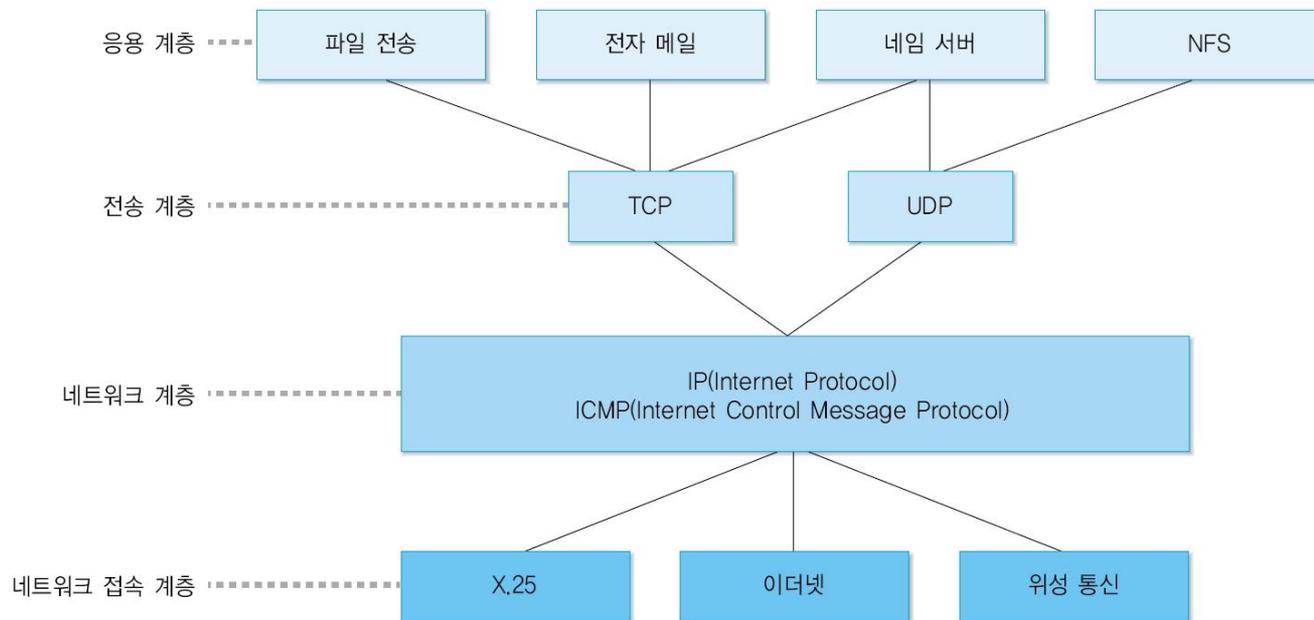
- 학습 목표
 - TCP 헤더에 정의된 필드의 역할 이해
 - Well-known 포트 개념 이해

TCP 개요 (1)

- Transport Control Protocol
- 주요 기능
 - 연결형 서비스 제공
 - 전이중(full-duplex) 방식의 양방향 가상회선 제공
 - 신뢰성 있는 데이터 전송 보장
- 일반적으로 전송계층 프로토콜은 운영체제 내부 기능으로 구현
- TCP에서는 데이터를 세그먼트(segment)라는 블록 단위로 분할하여 전송
 - 블록 크기는 네트워크 부하 정도, 윈도우 크기 등에 영향
 - 가변 크기
 - 세그먼트당 순서번호를 부여하지 않고, 세그먼트에 실어 보내는 데이터 바이트 수를 순서번호에 반영

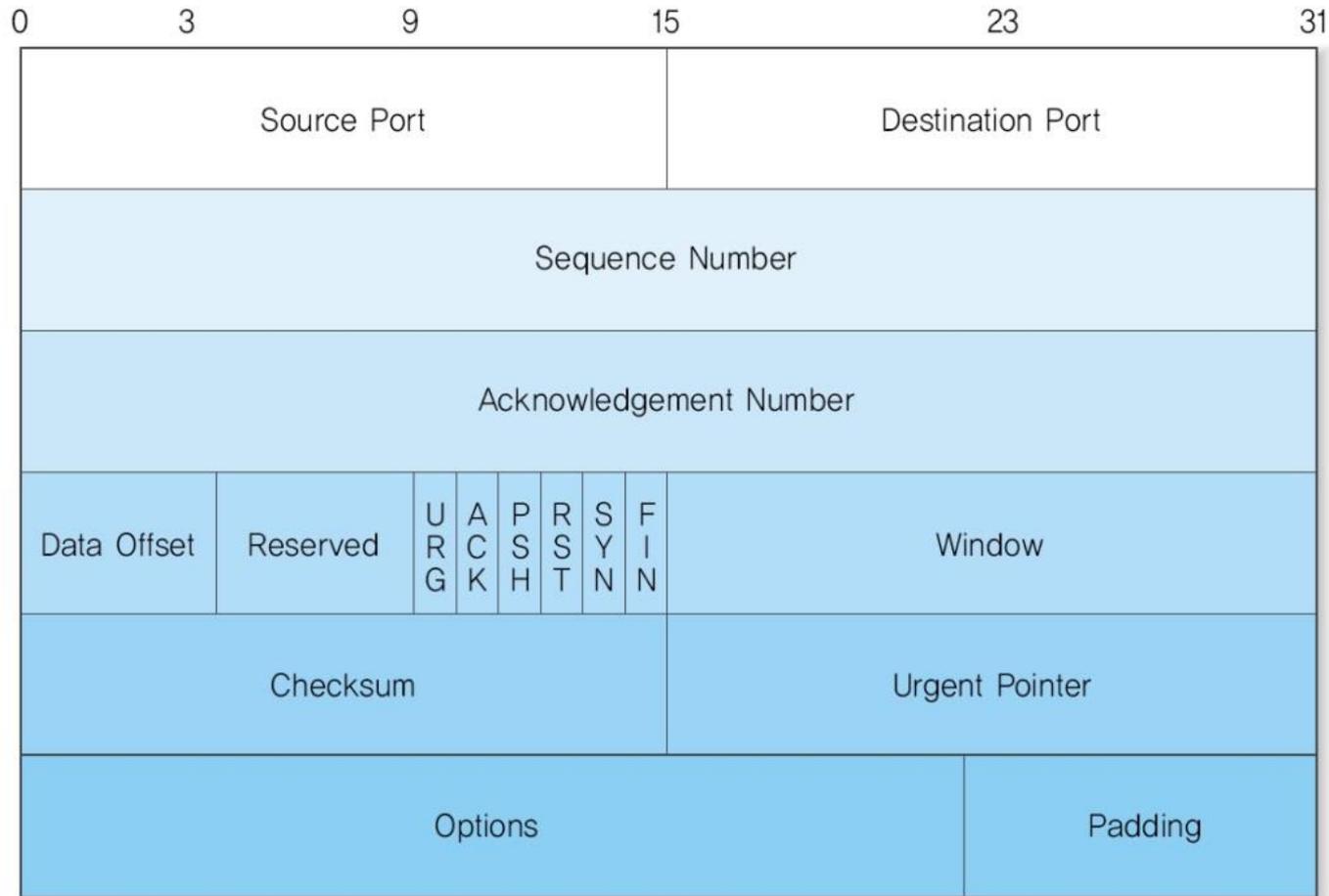
TCP 개요 (2)

- 상위계층에서는 연결형 서비스와 비연결형 서비스를 선택하여 전송 계층 프로토콜(TCP, UDP) 선택
 - 연결 유무, 신뢰성 외에도 해당 전송계층을 선택했을 때 각 응용 프로그램에서 수행해야 하는 기능에 대해서 이해 필요
- 데이터 링크 계층에는 다양한 네트워크 인터페이스 존재



[그림 9-7] TCP/IP 구조

TCP 헤더 (1)



[그림 9-8] TCP 헤더

TCP 헤더 (2)

- 헤더
 - 최소 20바이트
 - 최대 40바이트
- 헤더의 필드
 - Source Port / Destination Port
 - 송수신 포트 번호
 - Sequence Number
 - 순서 번호
 - 세그먼트 내의 바이트 수
 - 범위: 0 ~ 2³² - 1
 - 최초의 시작 순서번호는 임의로 설정 (연결 재설정 시 혼선 방지)
 - Acknowledgement Number
 - 응답 번호
 - ACK 플래그가 지정된 경우에 한해 유효
 - 다음에 수신하기를 원하는 데이터를 지정
 - 연결설정이나 연결해제와 같이 데이터가 없는 경우에도 1씩 증가

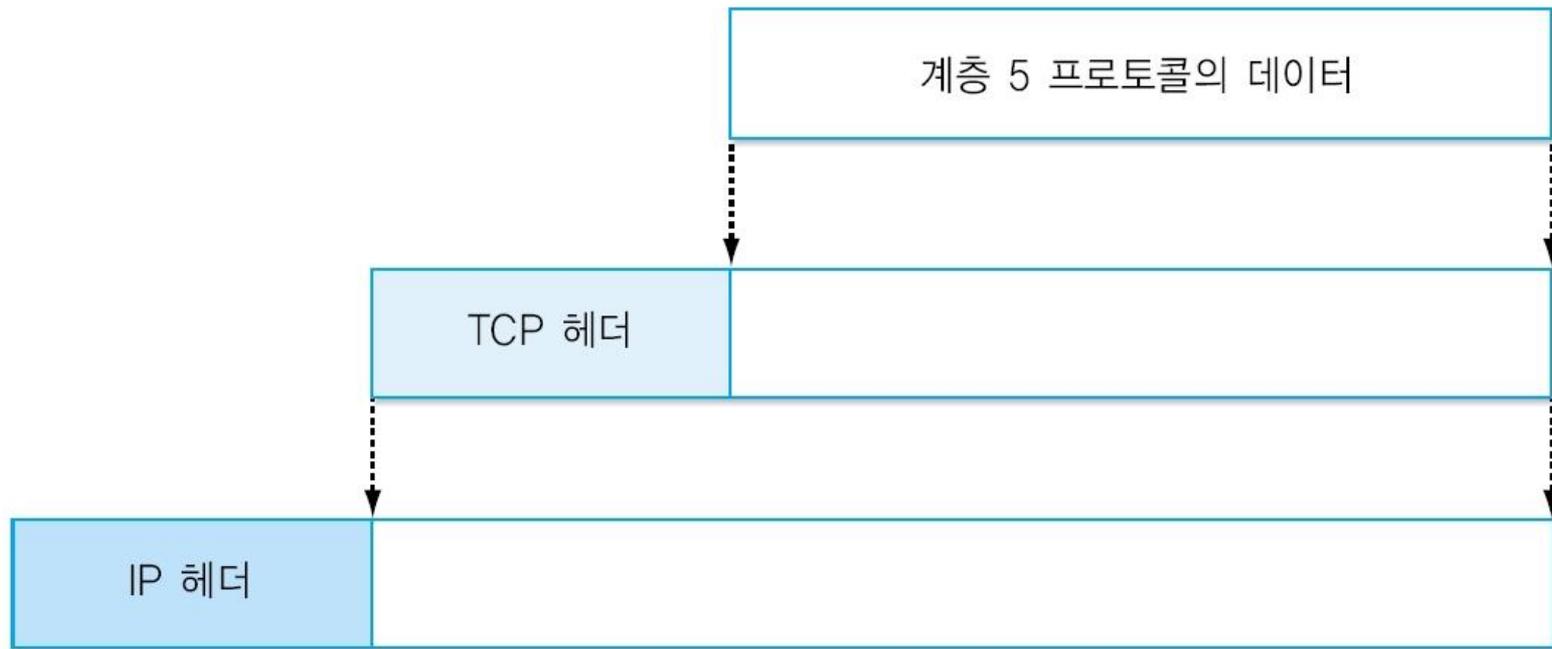
TCP 헤더 (3)

- 헤더의 필드 (계속)
 - Data Offset
 - 데이터가 시작하는 위치
 - TCP 헤더의 크기
 - Window
 - 수신 윈도우의 버퍼 크기 지정
 - 0이면 송신 프로세스의 전송 중지
 - Checksum
 - 헤더와 데이터에 대한 오류 검출
 - IP 프로토콜에서 사용하는 오류검출방식 이용
 - Urgent Pointer
 - 송신 프로세스가 긴급히 처리하기를 원하는 데이터의 처리
 - URG 플래그가 지정된 경우에 한해 유효
 - 플래그가 해제되면 정상 처리

TCP 헤더 (4)

- 헤더의 필드 (계속)
 - TCP 헤더의 플래그 비트
 - URG: Urgent Pointer 유효
 - ACK
 - Acknowledgement Number 유효
 - 정상적인 Piggybacking 환경에서는 연결 설정의 첫 번째 세그먼트를 제외한 모든 세그먼트에서 1로 설정
 - PSH
 - 현재 세그먼트의 데이터를 즉시 상위 계층에 전달하도록 지시
 - 이에 대한 응답이 도착하면 요청한 모든 데이터가 상위계층에 전달되었음을 의미
 - RST
 - 연결의 리셋이나 유효하지 않은 세그먼트에 대한 응답
 - 연결이 리셋된 후 앞서 전송했지만 정상 응답을 받지 못한 세그먼트는 재전송
 - SYN: 연결 설정 요청
 - FIN
 - 점진적 방식의 연결 종료 요청
 - 한 방향의 연결만 해제되므로 쌍방이 FIN을 보내야 모든 연결이 해제됨

TCP 캡슐화



[그림 9-9] TCP 세그먼트의 캡슐화

포트 번호 (1)

- TCP, UDP 프로토콜이 상위 계층에 제공하는 주소 표현 방식
- TCP, UDP가 독립적으로 관리
 - TCP, UDP 에서 동일한 번호를 가질 수 있으며, 각각 별개의 포트임
 - 하나의 서비스가 동일한 TCP 포트번호와 UDP포트번호를 가질 수도 있음
- Unix 계열의 경우 /etc/services에 저장
- 클라이언트-서버 방식에서 클라이언트는 서버의 IP주소와 포트 번호를 알아야 통신 가능
- Well-known 포트
 - 많이 사용하는 인터넷 서비스에 고정된 포트 번호 할당
- 서버의 포트번호는 Well-known 포트번호를 이용하고 클라이언트의 경우 시스템에서 임의의 포트를 자동 할당

포트 번호 (2)

[표 9-1] Well-known 포트

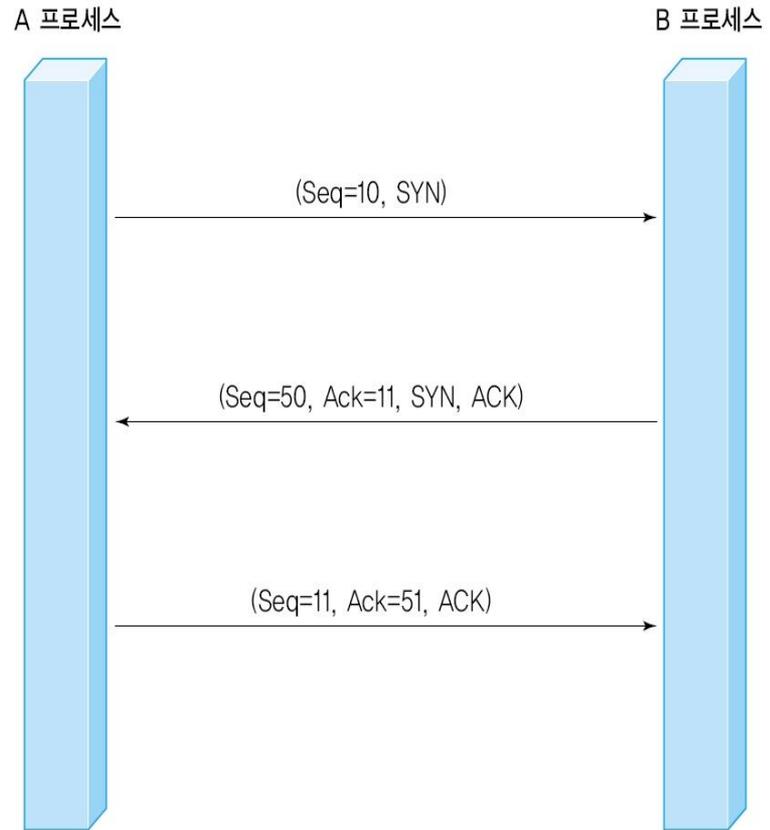
서비스	포트 번호
FTP(데이터 채널)	20
FTP(제어 채널)	21
Telnet(텔넷)	23
SMTP	25
DNS	53
HTTP	80
rlogin	513
rsh	514
portmap	111

3. TCP의 데이터 전송

- 학습 목표
 - TCP의 연결 설정, 데이터 전송, 연결 해제 과정 이해

TCP의 연결 설정

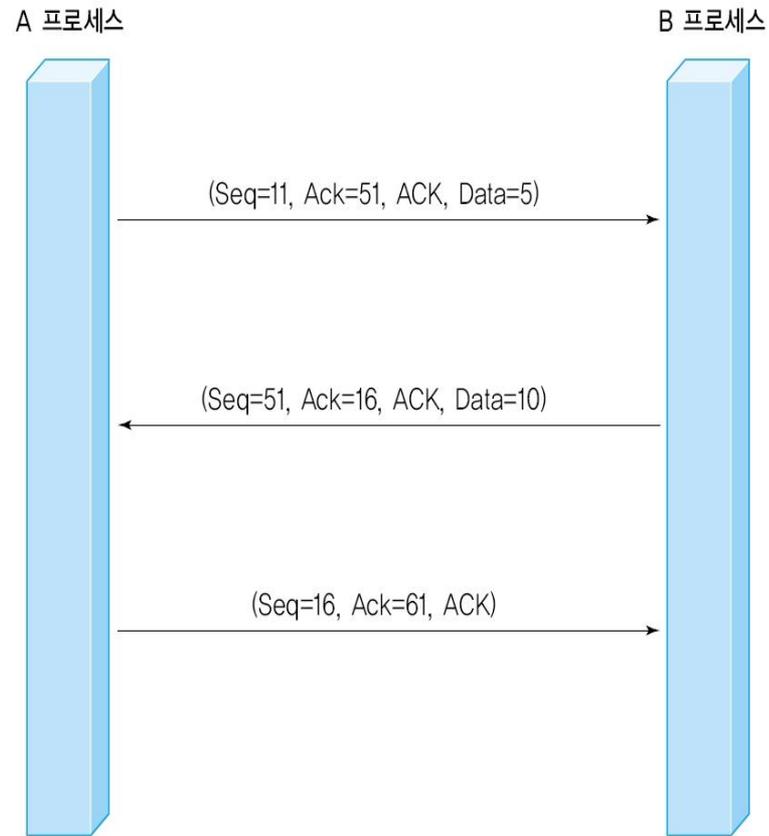
- 3 단계 설정 (3-way handshake)
 - A 프로세스의 연결설정 요청
 - 임의의 순서번호(10) 선택한 후 SYN(연결설정 요청) 전송
 - B 프로세스의 응답
 - 10에 대한 응답(ACK)과 임의로 선택한 순서번호(50)의 SYN 전송
 - A 프로세스가 현재 보낼 데이터가 없는 경우
 - 50에 대한 응답(ACK) 전송



[그림 9-10] TCP 연결 설정

TCP의 데이터 전송 (1)

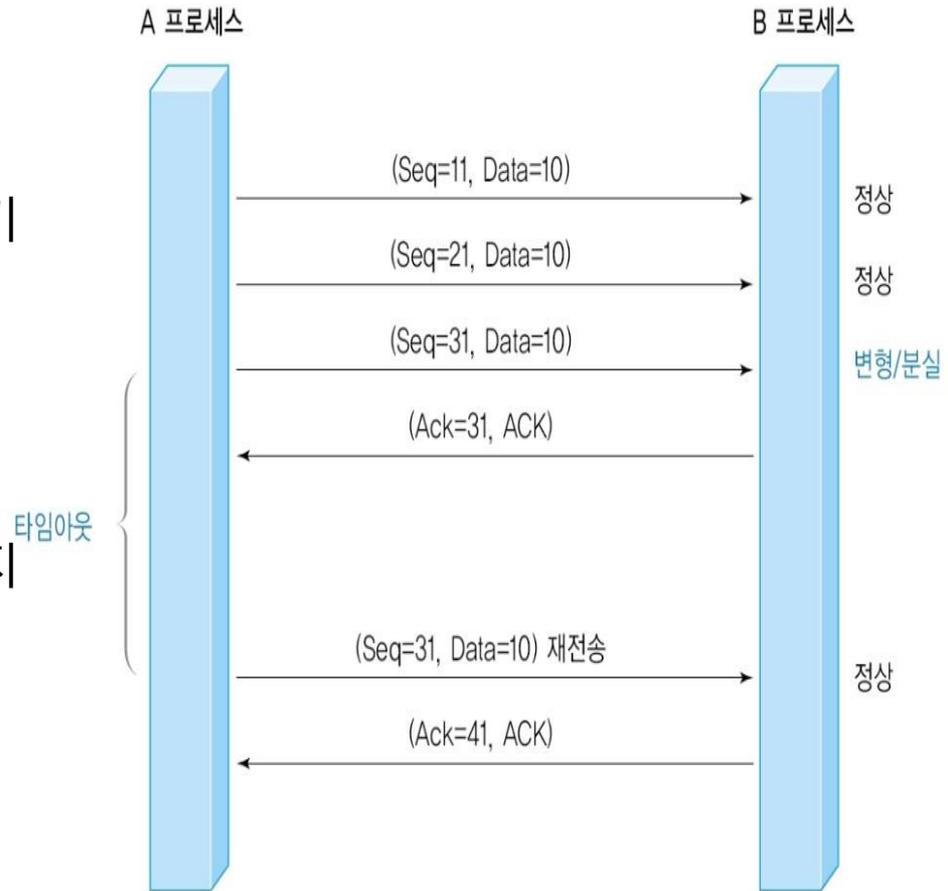
- 정상적인 데이터 전송
 - 3번째 연결설정 단계에서부터 데이터 전송 가능
 - 데이터 크기만큼 순서번호 증가
 - 흐름제어는 윈도우 필드 이용



[그림 9-11] TCP 데이터 전송

TCP의 데이터 전송 (2)

- 데이터 전송 오류
 - 동일한 순서번호
 - 중복으로 판단하고 폐기
 - 순서번호가 빠지는 경우
 - 분실로 판단
 - NAK는 지원하지 않음
 - 분실 또는 변형시 수신 프로세스는 응답을 하지 않고, 송신측에서 타임아웃기능으로 재전송



[그림 9-12] 전송 오류

TCP의 연결 해제

- 해제하고자 하는 측에서 FIN 전송
- FIN 수신측에서 보낼 데이터가 있는 경우 계속 송신할 수 있음



[그림 9-13] TCP 연결 해제

혼잡 제어

- ECN(Explicit Congestion Notification) 기능 : TCP의 혼잡 제어 기능을 지원
 - ECN 기능을 사용 : (a)처럼 SYN, ACK, ECE 플래그를 지정하여 응답
 - ECN 기능을 사용 안함 : (b)처럼 SYN, ACK 플래그만 지정하여 응답

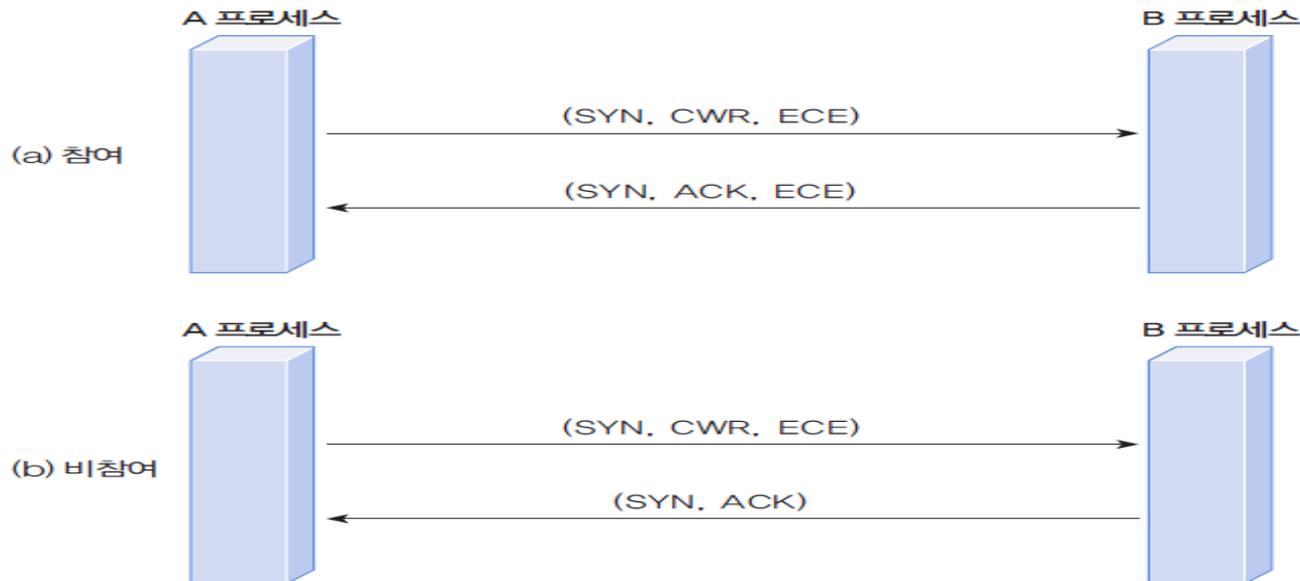


그림 9-14 TCP 연결 설정(ECN)

ECN의 동작 원리

- 수신 프로세스의 중개^{Echo}를 거쳐 간접적으로 송신 프로세스에 혼잡을 통지
 - 혼잡을 인지한 라우터 다음의 라우터들이 ECN 기능을 반복적으로 수행하지 못하도록 함



그림 9-15 ECN의 동작 원리