

데이터통신과 네트워킹

Data Communication
& Networking Ch. 13



CHAPTER

13

전송 계층

Section

- 01 전송 계층
- 02 연결 설정 및 해제

1. 전송 계층의 이해

- 전송계층의 대표 프로토콜은 TCP이며, 네트워크 계층과 응용 프로그램의 연결을 담당.
- 프로그램들은 TCP를 통해서 인터넷을 이용할 수 있음.
- TCP의 중요한 역할은 여러 프로그램들이 데이터를 보낼 수 있는 인터페이스를 만들어 주는 것.
- 인터페이스가 TSAPTansport Service Access Point

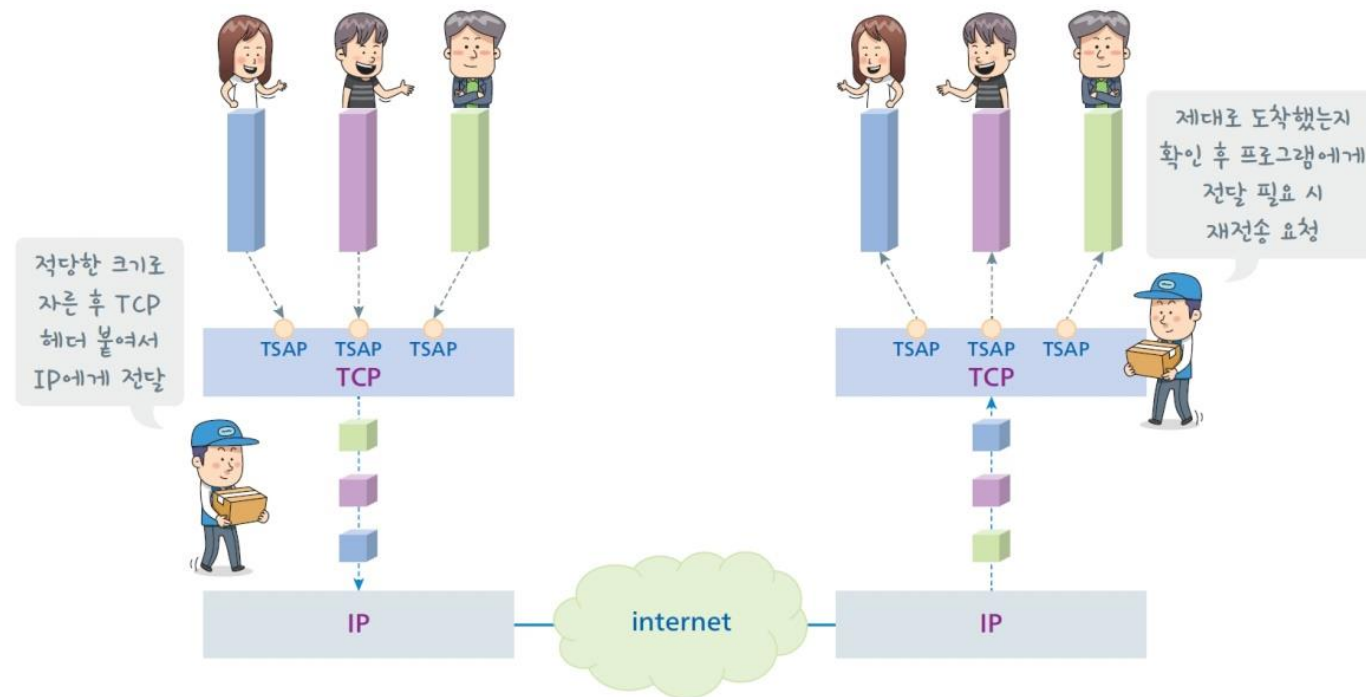


그림 13-1 전송 계층의 작업

전송 계층

- 전송계층 데이터의 단위 TPDU(Transport Protocol Data Unit).
- 전송 계층은 목적지에 도착한 데이터가 순서가 뒤바뀌거나 사라진 것은 없는지 확인 -> 일련번호와 ACK 사용.
- 슬라이딩 윈도우 프로토콜을 사용하여 혼잡 제어(congestion control)



그림 13-2 TPDU의 구조

전송 계층

- IP는 목적지까지 데이터를 전송만하기 때문에 연결^{connection}이라는 개념이 없음.
- 응용 프로그램의 연결을 설정하고 해제하는 작업을 지원하는 것도 전송계층의 역할.

표 13-1 전송 계층 주요작업

전송 계층 주요작업	설명
분할과 병합	데이터가 큰 경우 자르고, 받는 쪽에서 원래 크기로 합쳐줌
멀티 인터페이스 제공	포트 _{port} 와 소켓 _{socket}
데이터의 무결성	일련번호(seq)와 ACK 사용(데이터 링크 계층 참고)
혼잡 제어	슬라이딩 윈도우 크기 조절(데이터 링크 계층 참고)
연결 설정 및 연결 해제	3방향 핸드셰이크 _{3way handshake}

2. 포트와 소켓

- 클라이언트는 서비스를 요청하는 컴퓨터 -> 웹 브라우저는 클라이언트 소프트웨어를 대표하는 단어, 웹 브라우저 제품명으로는 익스플로러, 크롬, 파이어 폭스가 있음.
- 웹 브라우저는 서버 컴퓨터에 접속하여 웹 페이지를 요청 -> 이 때 사용하는 프로토콜은 하이퍼 텍스트(웹 페이지)를 전송하라는 프로토콜, Hyper Text Transfer Protocol이며, 약자로 HTTP.
- 서버가 요청을 받으면, HTML Hyper Text Markup Language을 사용하여 만들어진 웹 페이지를 클라이언트에게 전송 -> 웹 페이지를 웹 브라우저가 화면에 보여줌.

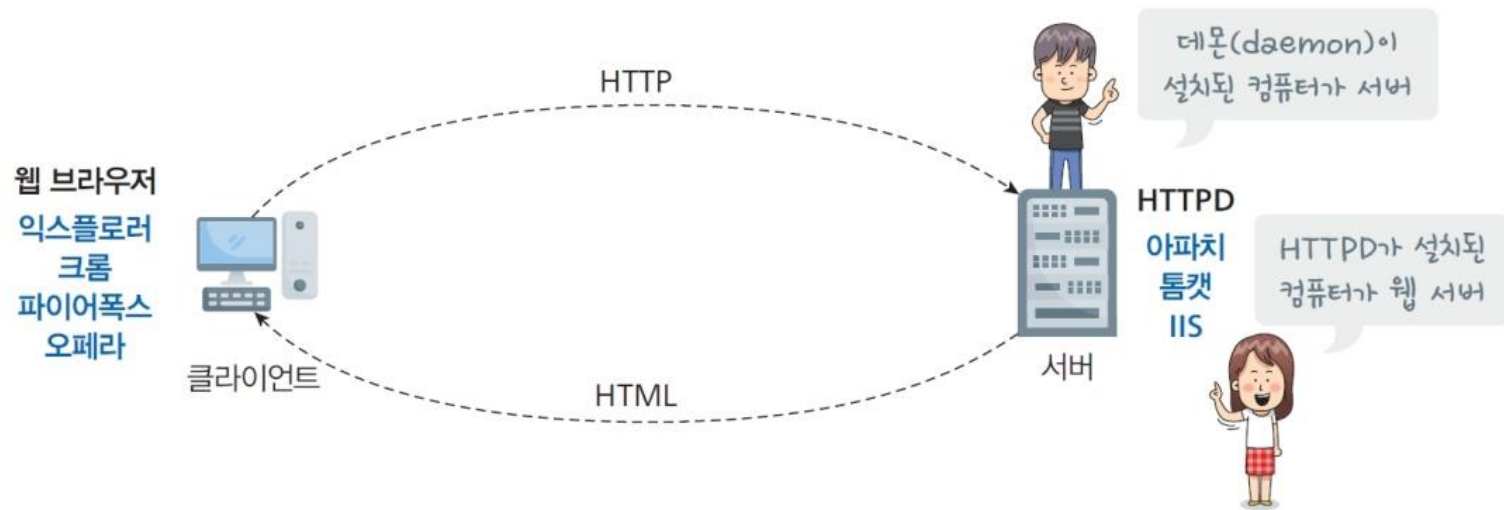


그림 13-3 웹 시스템의 클라이언트-서버 구조

전송 계층

- 클라이언트 소프트웨어는 사용자가 필요할 때 사용했다가 필요 없으면 종료.
- 서버 소프트웨어는 클라이언트가 언제 서비스를 요청할지 모르기 때문에 항상 대기 상태로 기다려야 함 -> 죽지 않고 살아서 서비스를 계속하는 프로그램을 **데몬**, 영어로 **daemon**이라 부름.
- 웹 시스템의 서버 쪽에는 HTTP(웹 서비스)를 받아 줄 데몬이 설치되어 있어야 함 -> 웹 데몬을 HTTPD(Hyper Text Transfer Protocol Daemon)이라 부름 -> 보통의 경우 프로토콜 이름 뒤에 D를 붙임.
- HTTPD의 제품명으로는 아파치(Apache), 톰캣(Tomcat), IIS(Internet Information Service) 등이 있음.
- 서버는 데몬이 설치된 컴퓨터 -> HTTPD가 설치된 컴퓨터를 웹 서버라고 부름.
- 웹 서버를 구축한다는 뜻은 아파치, 톰캣, IIS와 같은 HTTPD(웹 데몬)를 컴퓨터에 설치한다는 의미.

전송 계층

- 인터넷을 통해 파일을 주고받는 것도 웹 시스템과 마찬가지로.
- 파일 전송에 사용되는 프로토콜이 FTP이며, File Transfer Protocol의 약자.
- 인터넷을 통해 누군가에게 파일을 보내거나 누군가로부터 파일을 받고 싶은 경우 파일 서버를 구축하면 됨.
- 파일 서버를 구축한다는 것은 컴퓨터에 FTPD를 설치하는 것.

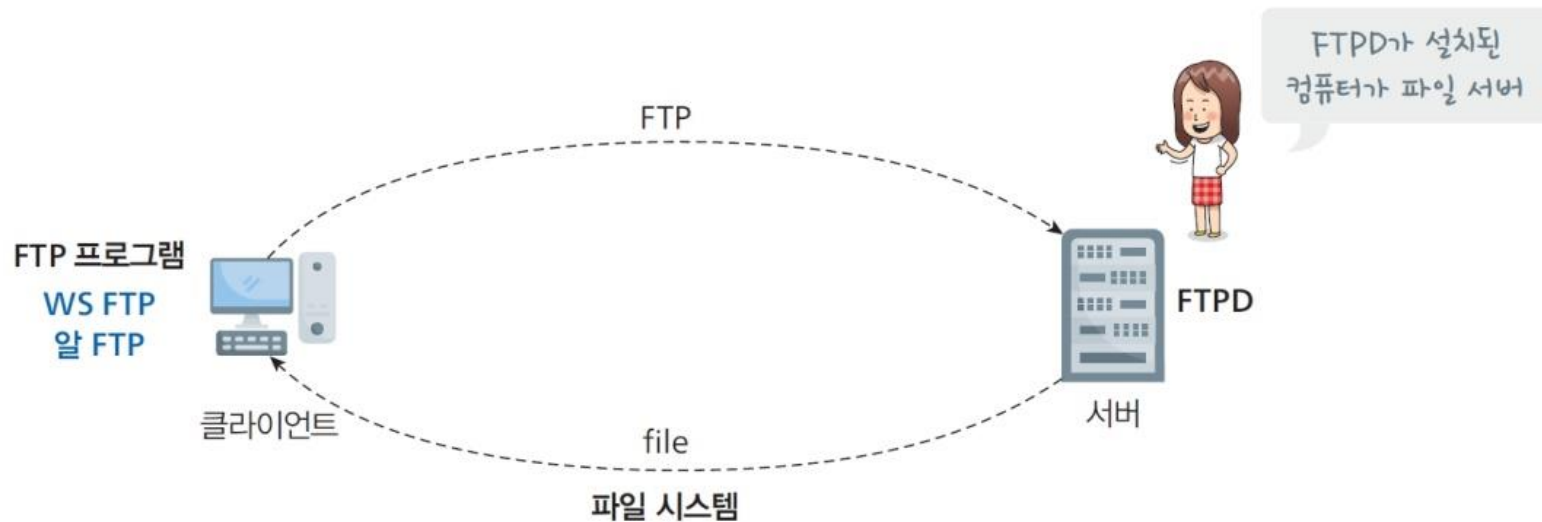


그림 13-4 파일서버

- 이메일의 경우도 웹 시스템과 마찬가지로.
- 이메일에 사용되는 프로토콜은 SMTP이며 영어로 Simple Mail Transfer Protocol의 약자.
- 어떤 컴퓨터를 이메일 서버로 만들고 싶다면 SMTPD(Simple Mail Transfer Protocol Daemon)를 설치하면 됨.
- 서버는 특정 하드웨어를 가리키는 단어가 아니라 데몬이 설치된 컴퓨터를 지칭하는 단어.
- 서버의 역할을 하기 위해서는 성능이 높은 CPU와 24시간 안정적으로 작동하는 내구성이 필요함.
그래서 고사양의 컴퓨터를 '서버급 컴퓨터'라 부름.

전송 계층

- IPark 아파트 203동의 어느 집에서 중국집 '진짜루'에 짜장면을 시켰다고 가정.
 - 두 집에서는 짬뽕을 시켰으나 방번호(호수)를 말하지 않으면 배달할 수 없음.
- IP 주소는 인터넷에 있는 특정 컴퓨터까지 오는 데 사용-> 아파트의 동 번호와 같음.
- 여러 응용 프로그램들을 구분하기 위한 방번호(주소)가 필요 -> **전송계층이 사용하는 주소를 포트, 영어로 port라 부름.**



그림 13-5 중국집 진짜루의 음식 배달

- 원래 포트는 항구에서 배의 화물을 싣거나 내리는 장소.
- 포트 번호의 크기는 16비트이며, 각 컴퓨터에는 0에서 65535사이의 포트가 있음.
- 운영체제는 0에서 65535 사이의 빈 포트 번호 중 하나를 네트워크를 사용하려는 프로그램에게 줌. 해당 프로그램은 할당받은 포트를 사용하여 원격지 호스트와 데이터를 주고 받음.
- 포트는 전송 계층이 여러 프로그램에게 제공하는 주소인 동시에 멀티 인터페이스 -> TSAP = 포트.

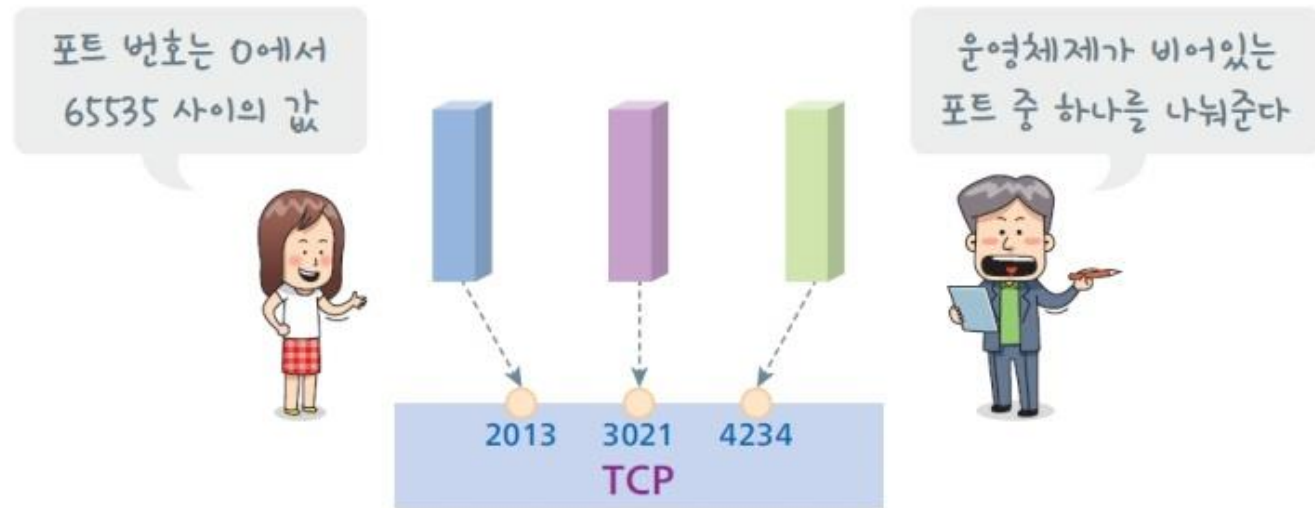


그림 13-6 포트 번호

전송 계층

- 첫 번째 크롬에서 세 번째 크롬까지 포트 번호를 각각 2013, 3021, 4234번을 할당. 각 크롬들은 네이버, 구글, 다음 웹 서버에 접속할 때 자신의 포트 번호를 알려주며 통신 함.
- 서버에는 웹 데몬(HTTPD) 뿐 아니라 FTPD, SMTPD도 설치되어 있을 수 있으며, 데몬들도 포트 번호를 사용 -> 클라이언트가 HTTPD(웹 데몬)와 처음 통신할 때, HTTPD의 포트 번호를 알아야 통신 할 수 있음 -> 만약 HTTPD의 포트번호도 운영체제에 의해 임의로 할당된다면, 매번 서버에게 물어 봐야 함.

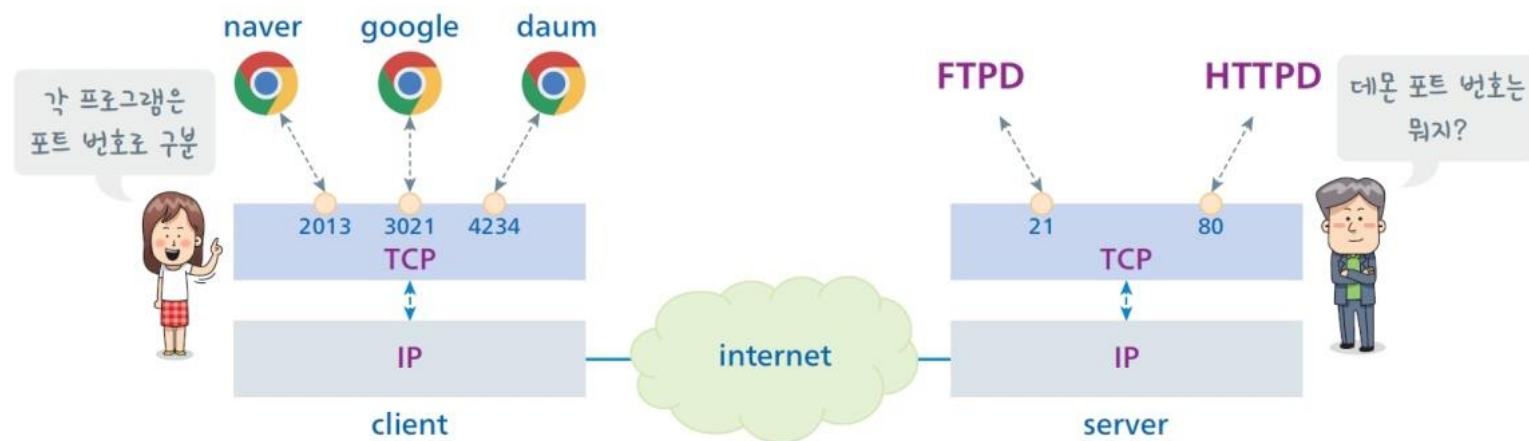


그림 13-7 well-known 포트 번호

전송 계층

- 자주 사용하는 데몬이나 중요한 프로그램의 포트 번호를 고정 시켰음.
- 이를 잘 알려진 포트번호 혹은 **well-known 포트 번호**라 부름.
- 다음 표는 주요 well-known 포트번호를 나타냄.

표 13-2 주요 well-known 포트 번호

서비스	포트 번호	비고
FTP	21(제어), 20(데이터)	FTP의 경우 21번 포트로 접속하여 작업을 하는데 실제 데이터는 20번 포트로 전송된다.
telnet	23	원격지 로그인에 사용
SMTP	25	이메일 전송 프로토콜(서버-서버)
DNS	53	도메인 이름 - 주소 변환
DHCP(BOOTP)	67, 68	동적 IP 주소 할당
HTTP	80	웹 시스템
HTTPS	443	HTTP의 암호 프로토콜

- URL 형식은 '프로토콜://도메인 이름:포트 번호'
- http://naver.com:80이 정확한 주소 -> http:// 및 포트번호 :80은 생략가능.

전송 계층

- 네이버에는 하루에도 수만 명의 사람들이 몰려드는데, 모두 `http://naver.com:80`으로 방문 -> 네이버 서버 입장에서 보면 80번 포트에 수만 명의 사람들이 몰려오는 것.
- 벽에 220V 콘센트가 하나뿐인데 노트북, 스마트폰, 블루투스 이어폰을 동시에 충전해야 하는 경우 멀티 탭 사용 -> 80번 포트에 멀티 탭을 꽂은 후, 몰려드는 사용자에게 멀티 탭의 콘센트를 하나씩을 나눠줌.
- **소켓은 같은 포트에 연결되어 여러 명을 동시에 처리할 수 있는 소프트웨어적인 접속장치** -> 네트워크를 이용한 프로그래밍을 소켓 프로그래밍이라 부름.

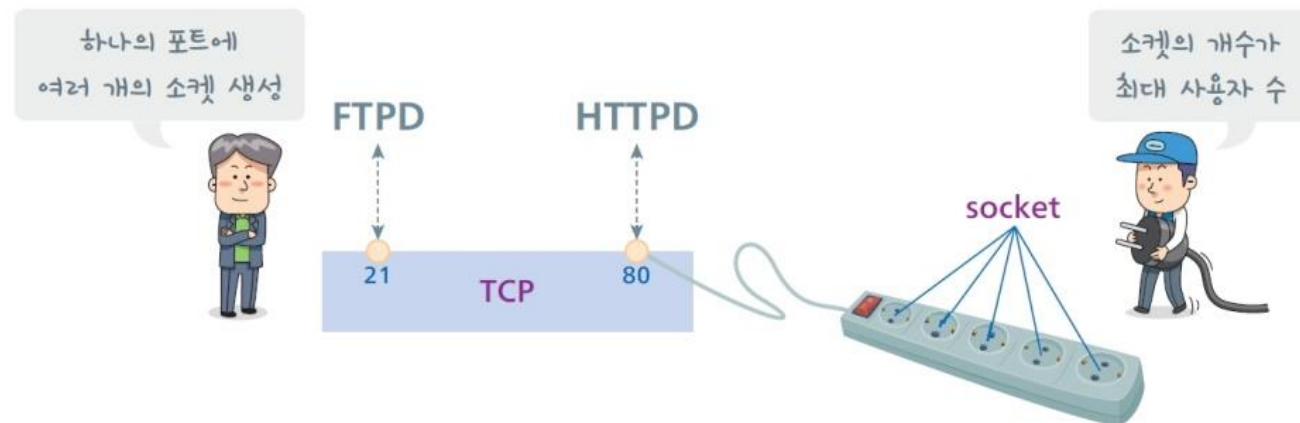


그림 13-8 포트와 소켓

전송 계층

- 행사나 이벤트로 인하여 어떤 홈페이지에 갑자기 사용자가 몰려 서버가 다운되었다는 것은 소켓과 연관 있음.
- 동시 최대 접속자수란 동시에 접속 시킬 수 있는 최대 인원을 설정 -> 소켓을 몇 개 준비할 것인가와 같은 의미.
 - 소켓 개수가 작으면 클라이언트는 빈 소켓을 얻지 못하여 서비스가 지연되고, 서버가 다운된 것처럼 느끼게 됨.
 - 사람이 많이 올 것이라 예상하여 소켓을 무작정 많이 열면 컴퓨터가 느려지고 최악의 경우 서버가 다운 됨.

연결 설정 및 해제

1. 연결 설정의 어려움

- 연결설정은 CR(Connection Request)을 보냄으로서 이루어짐.
- CR을 보내고 데이터를 보내는 경우.
 - CR을 보냈다고 해서 호스트 B가 연결을 허락한 것은 아님.
 - 이런 상태에서 데이터가 도착하게 되면, 호스트 B는 해당 데이터를 무시.

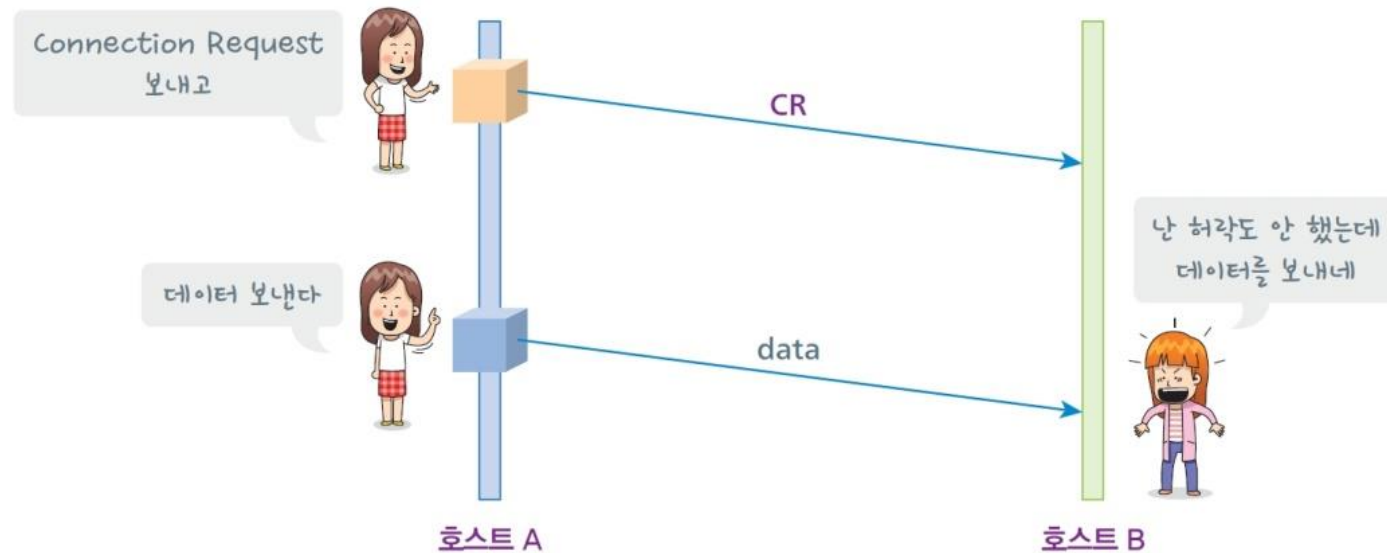


그림 13-9 CR과 데이터 전송의 문제

연결 설정 및 해제

- 호스트 A는 CR을 보내고, 호스트 B는 연결을 승낙한다는 뜻으로 CR에 대한 응답 메시지 CR_ACK를 보냄 -> 호스트 A는 CR_ACK를 받은 이후부터 데이터를 호스트 B에게 보낼 수 있음.
- 호스트 B가 보낸 CR_ACK가 정상적으로 전송되었는지를 호스트 B는 알 수 없음 -> 호스트 B가 CR_ACK를 보낸 직후에 도착하는 데이터가 호스트 A가 보낸 데이터라는 보장이 없음.

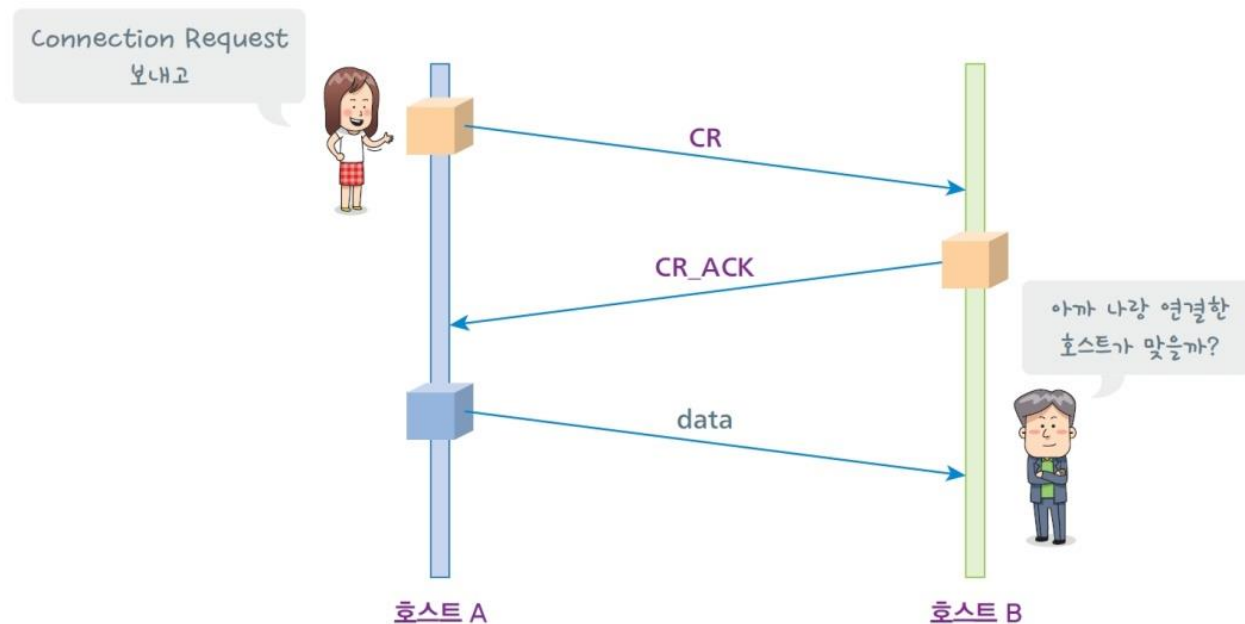


그림 13-10 CR_ACK 사용

연결 설정 및 해제

- 호스트 A는 CR을 보내고, 호스트 B는 CR에 대한 허락의 의미로 CR_ACK를 보냄 -> 호스트 A는 CR_ACK를 받았다는 증거로, 보내는 데이터에 CR_ACK_ACK를 같이 넣어 보냄(피기백킹piggy backing), 보내는 데이터가 없는 경우에는 CR_ACK_ACK만 보냄.
- CR -> CR_ACK -> CR_ACK_ACK, 3번의 합의를 거쳐 연결이 이루어지는데, 이를 **3방향 핸드셰이크** 3way handshake라 부름.

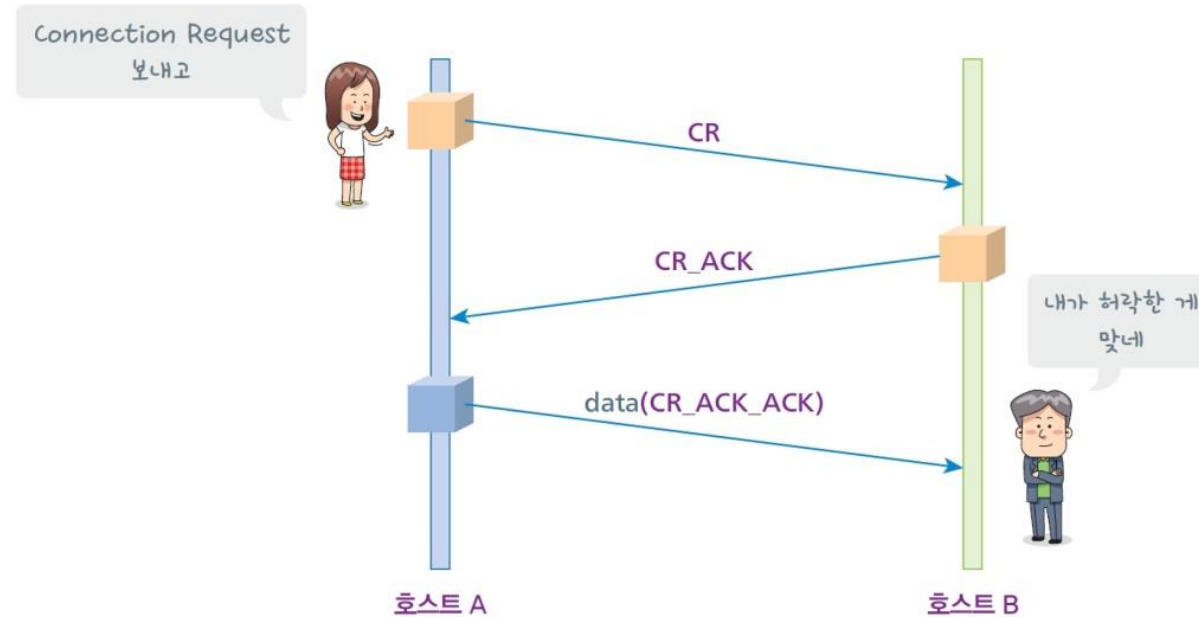


그림 13-11 3방향 핸드셰이크

2. 연결 설정

- TCP에서 실제로 연결 설정이 이루어지는지 과정.
 - TCP에서 CR^{Connection Request}에 해당하는 필드의 이름이 SYN -> 호스트 A는 연결 설정을 위하여 SYN을 호스트 B로 보냄(호스트 A의 일련번호(seq)는 x) -> 호스트 B도 연결 설정의 허락의 의미로 SYN을 보냄(B의 일련번호(seq)는 y, ACK 번호는 x) -> 호스트 B로부터 SYN을 받으면 호스트 A는 데이터를 전송(seq = x + 1, ack = y).

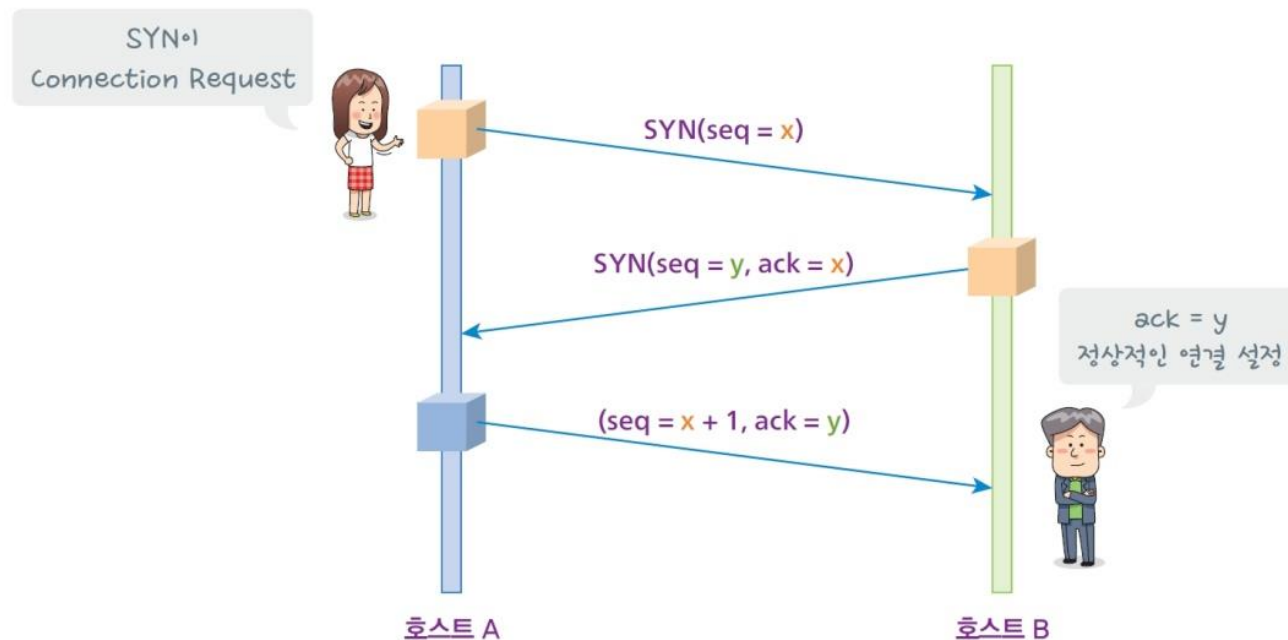


그림 13-12 3방향 핸드셰이크를 사용하여 연결 설정

3. 연결 해제

- 연결을 해제(DR^{Disconnection Request}) 할 때에도 서로간의 합의가 필요.
 - 호스트 A는 계속적으로 호스트 B에게 데이터를 보내는 중간에 호스트 B가 일방적으로 DR을 보냄.
 - DR을 보냈다는 것은 연결이 끊겼다는 의미, 이후 도착한 데이터는 모두 무시 -> 호스트 A는 데이터를 보냈지만, 호스트 B는 그 데이터를 처리하지 못하는 문제 발생 -> 반대의 경우도 마찬가지.

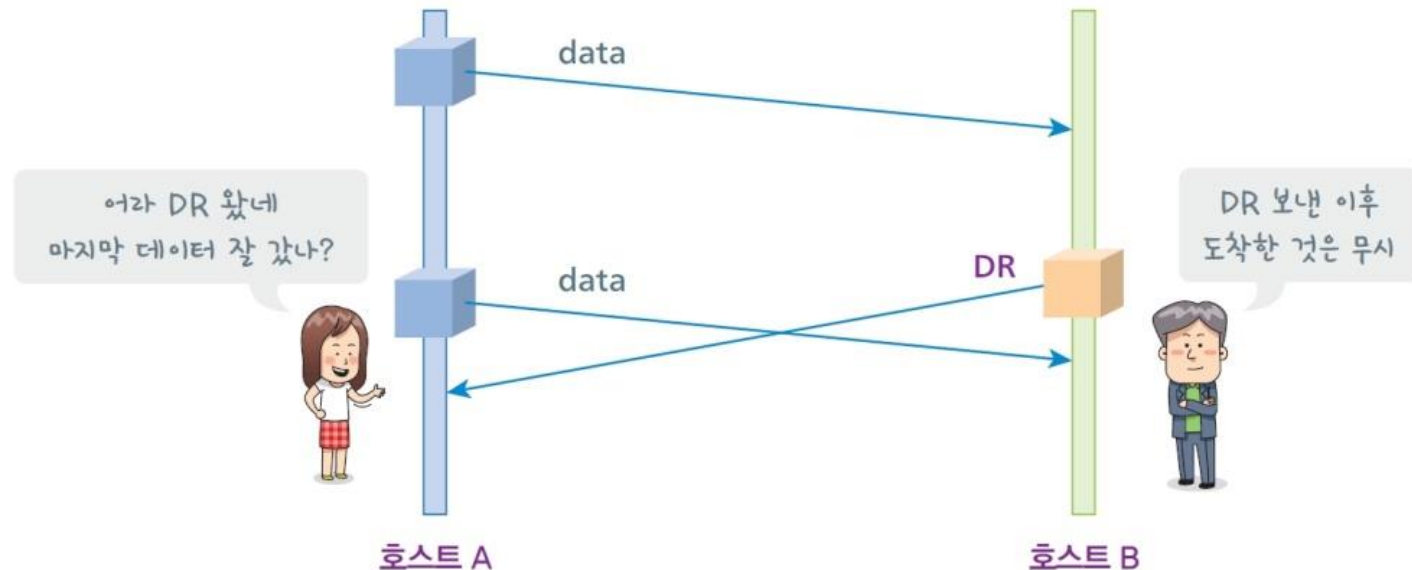


그림 13-13 일방적 DR의 문제점

연결 설정 및 해제

- 연결 해제의 가장 큰 문제.
 - 호스트 A가 연결해제를 위해 보낸 DR이 사라진 경우, 이때 악의적인 사용자로부터 위조된 데이터가 호스트 B에 도착 -> 호스트 B는 호스트 A가 보낸 DR이 사라졌다는 것을 알지 못하기 때문에 데이터를 처리 할 수 밖에 없음.

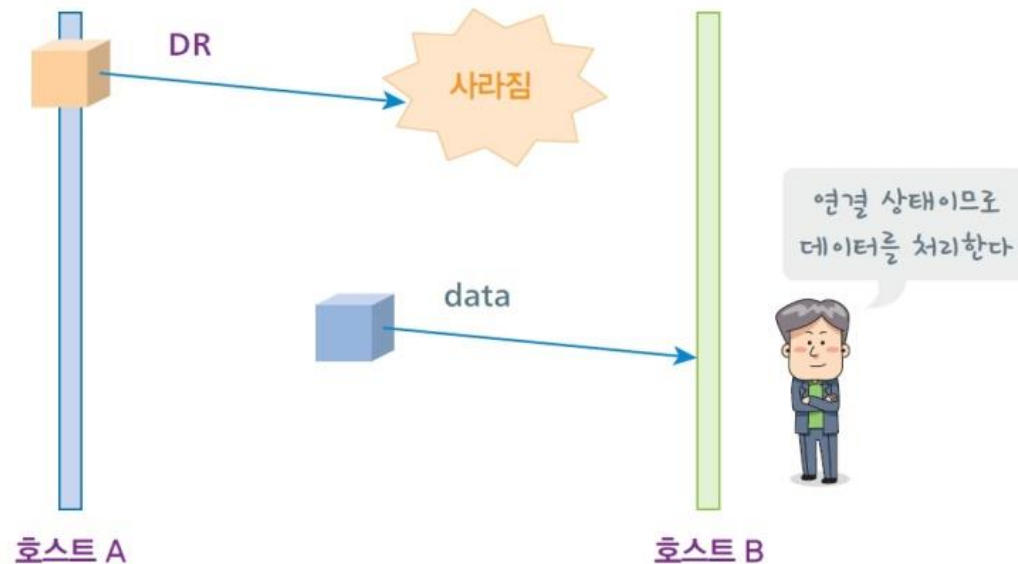


그림 13-14 DR이 사라지는 경우

연결 설정 및 해제

- TCP에서 DR^{Disconnection Request}에 해당하는 필드의 이름이 FIN.
- 연결 해제에도 연결설정과 똑같이 3방향 핸드 셰이크를 사용.
 - 호스트 A는 연결 해제를 위하여 $\text{FIN}(\text{seq} = x)$ 을 호스트 B로 보냄 -> 호스트 B는 연결 해제를 동의 한다는 의미로 $\text{FIN}(\text{seq} = y, \text{ack} = x)$ 을 보냄 -> 호스트 B로부터 FIN을 받으면 호스트 A는 $\text{ACK}(\text{seq} = x + 1, \text{ack} = y)$ 을 보냄.

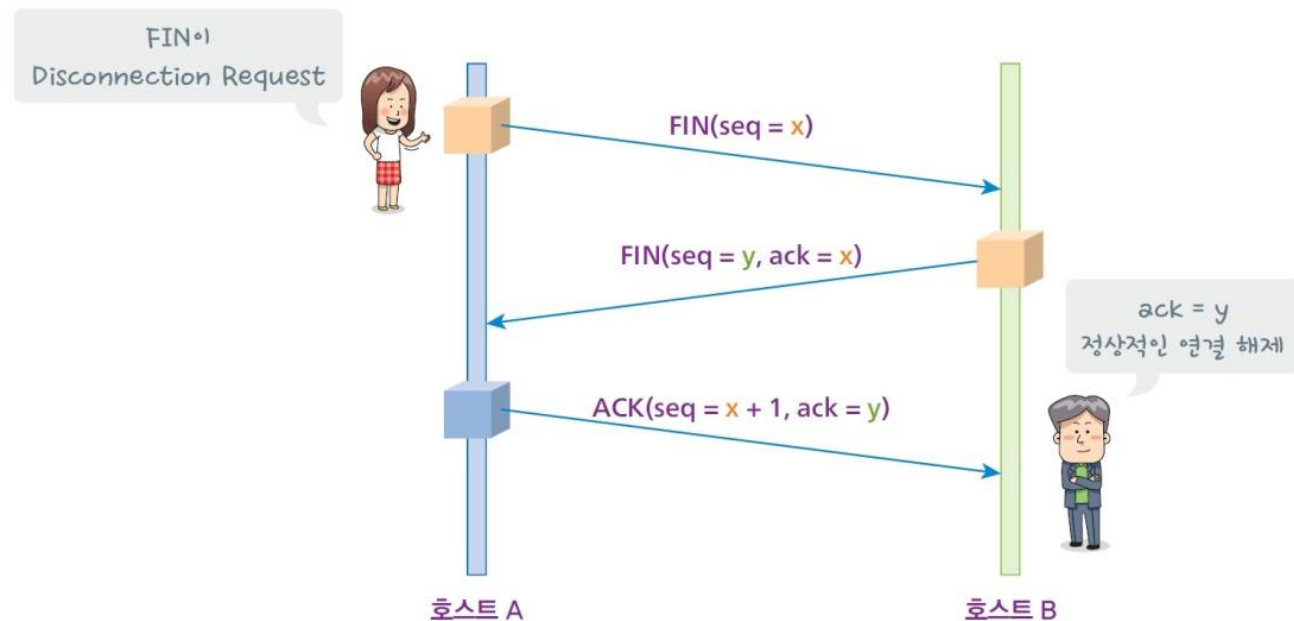


그림 13-15 3방향 핸드셰이크로 연결 해제