

# IT CookBook, 쉽게 배우는 데이터 통신과 컴퓨터 네트워크(개정) 9장 연습문제 해답

본 자료의 저작권은 박기현과 한빛아카데미(주)에 있습니다.

이 자료는 강의 보조자료로 제공되는 것으로, 학생들에게 배포되어서는 안 됩니다.

1. ① 데이터 링크
2. ① 흐름 제어, ② 오류 제어
3. ① 분할, ② 병합
4. 3 단계 설정
5. TCP
6. ① 포트, ② 소켓, ③ Well-known 포트
7. ① SYN, ② ACK
8. ① 슬라이딩 윈도우, ② Window
9. ① ECE, ② CWR
10. ① 세그먼트, ② 바이트

11. ④

(설명④) 데이터 링크 계층은 물리적 선로를 통해 데이터를 직접 전달하기 때문에 네트워크 계층의 중개 기능이 필요 없다.

12. ①, ③

(설명①) 양 끝단의 호스트에서 실행되는 프로세스 사이의 전송 속도 조절은 흐름 제어 기능이 담당한다.

(설명③) 전송 계층에서 발생하는 오류는 선로 오류보다 논리적으로 구축된 각 계층의 소프트웨어가 동작하는 과정에서 데이터를 분실하는 경우가 대부분이다.

13. ①, ②, ③, ⑤

(설명④) 3 단계 연결 설정에서 마지막 단계의 응답은 신뢰성 확보를 위한 추가적인 수단이며 상대방의 응답을 잘 받았다는 의미이다. 따라서 자신이 전송할 데이터가 있으면 이를 전송함으로써 대체할 수 있다.

14. ①, ②, ④, ⑤

(설명③) TCP 프로토콜은 세그먼트를 하나의 단위로 간주하여 순서 번호를 관리하지 않고, 전송되는 데이터의 바이트 수를 순서 번호에 반영한다. 따라서 1개의 세그먼트로 1000 바이트를 전송하면 순서번호도 1000이 증가해야 한다.

15. ①, ③, ⑤

(설명②) Source Port와 Destination Port 필드는 TCP로 연결되는 가상 회선 양단의 송수신 프로세스에 할당된 네트워크 포트 주소이며, TCP와 UDP가 별도의 독립적인 주소 공간을 갖지 때문에 같은 번호를 사용할 수 있다.

(설명④) Acknowledgement Number 필드는 수신 프로세스가 제대로 수신한 바이트의 수를 응답하기 위해 사용한다. ACK 플래그 비트가 지정된 경우에만 유효하고, 다음에 수신을 기대하는 데이터의 순서 번호를 표기한다.

16. ④, ⑤

(설명④) SYN 비트는 연결 설정 요구를 의미하는 플래그 비트이므로 가상 회선 연결을 설정하는 과정에서 사용한다.

(설명⑤) FIN 비트는 한쪽 프로세스가 더는 전송할 데이터가 없어 연결을 종료하고 싶다는 의사 표시를 상대방에게 알리는데 사용한다. 연결 해제는 양쪽 프로세스 모두가 FIN 플래그를 전송해야 완료된다.

17. ①, ②, ③, ④, ⑤

18. ①, ②, ③, ⑤

(설명④) TCP는 부정 응답 기능을 사용하지 않는다. 따라서 세그먼트의 내용이 변형되어도 수신 프로세스가 응답을 하지 않아 데이터 분실과 동일하게 처리된다. 즉, 송신 프로세스의 타임아웃 기능에 의하여 오류 복구가 진행된다.

19. 없음

20.

전송 계층(Transport Layer) 프로토콜은 오류 제어, 흐름 제어, 데이터 순서화 등의 기능면

에서 데이터 링크 계층과 특징이 유사하다. 그러나 데이터 링크 계층이 물리적인 전송 선로로 직접 연결된 두 물리적 호스트(컴퓨터나 라우터) 사이의 데이터 전송을 담당하는 반면, 전송 계층은 네트워크 끝단에 위치하는 통신 주체가 중간의 논리적 선로(라우터로 연결된 컴퓨터 네트워크)를 통해 데이터를 주고받는다.

21.

전송 계층 프로토콜을 설계하는 과정에서는 다음과 같이 여러 기능을 고려해야 한다.

: 흐름 제어, 오류 제어, 분할과 병합, 서비스 프리미티브

#### ■ 흐름 제어

데이터 링크 계층에서도 라우터 사이의 프레임 전송 과정에서 흐름 제어기능을 수행한다. 그러나 라우터를 연결하는 물리적 선로의 수는 전송 계층이 관여하는 호스트 사이의 논리적 연결 수보다 상대적으로 적다. 따라서 두 계층이 지원하는 흐름 제어 기능은 기본 목적을 비롯해 유사한 특징이 많음에도 불구하고, 서로 다른 버퍼 관리 방법이 필요하다. 수신자가 송신자의 전송 속도보다 느리게 데이터를 수신하면 버퍼 용량이 초과하여 데이터를 분실할 수 있다. 이 경우에 송신자는 타임아웃 기능을 통해 재전송 과정을 수행하므로 전체 네트워크의 전송 효율이 점점 느려진다. 이론적으로 흐름 제어 기능은 수신자가 슬라이딩윈도우 프로토콜의 윈도우 하단 값을 조정한다. 즉, 송신자가 보낼 수 있는 패킷의 한계를 지정하는 방법으로 문제를 해결한다.

#### ■ 오류 제어

전송 오류의 발생에 따라 수신 데이터의 내용이 깨지거나 분실되면 데이터 재전송에 의한 오류 제어(Error Control) 기능에 의해 복구 절차가 진행된다. 데이터 재전송 절차는 수신자의 요구에 의해 이루어질 수도 있고, 송신자 스스로 판단해 이루어질 수도 있다. 전송 계층에서 발생하는 오류는 선로 오류보다 논리적으로 구축된 각 계층의 소프트웨어가 동작하는 과정에서 데이터를 분실하는 경우가 대부분이다.

#### ■ 분할과 병합

상위 계층에서 전송을 요구한 데이터 크기가 전송 계층에서 처리할 수 있는 데이터 크기보다 크면 데이터를 쪼개 전송해야 한다. 데이터를 전송하기 전에 적합한 크기로 나누는 과정을 분할(Segmentation)이라 한다. 반대로 수신 측에서 수신한 데이터를 원래 크기로 다시 모으는 과정을 병합(Reassembly)이라고 한다. 수신 측에서 상위 계층에 데이터를 보낼 때는 반드시 데이터의 병합 과정을 거쳐야 계층 구조의 틀이 유지된다.

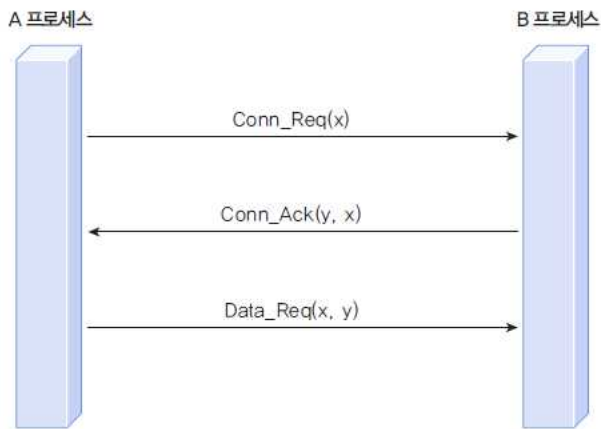
#### ■ 서비스 프리미티브

전송 서비스 프리미티브(Primitive)는 전송 계층 사용자가 전송 계층 서비스를 사용하기

위한 인터페이스다. 네트워크 계층에서 제공하는 서비스는 일반적으로 비신뢰성을 바탕으로 한 비연결형 서비스 프리미티브가 정의된다. 전송 계층에서는 비연결형 서비스뿐만 아니라, 신뢰성이 향상된 연결형 서비스도 제공한다. 그러므로 이러한 특징이 반영되어 전송 계층이 네트워크 계층의 서비스 프리미티브보다 간단하다고 볼 수 있다.

22.

[그림 9-4]는 오류가 발생하지 않은 정상 상태에서 3단계 설정(Three-Way Handshake)의 동작 과정을 보여준다. A 프로세스는 임의의 송신 순서 번호  $x$ 를 지정하여 연결 설정을 요청한다. 이를 수신하는 B 프로세스는 연결 설정 요구에 대한 순서 번호  $x$ 에 대해 응답을 보낸다. 이때 순서 번호  $y$ 는 자신의 초기 송신 번호를 왼쪽 프로세스에 알려주기 위해 사용한다.



연결 요구에 대한 응답 Conn\_Ack를 받은 A 프로세스는 Conn\_Ack를 잘 받았다고 B 프로세스에 응답해야 한다. 따라서 Conn\_Ack와 다른 종류로 응답 정보를 보내야 하지만, 그림에서는 이 과정을 생략하고 바로 데이터를 전송하고 있다. 이는 Data\_Req가 데이터 전송과 함께 응답 기능을 함께 수행하기 때문이다. 만일 A 프로세스가 전송할 데이터가 없으면 Conn\_Ack에 대한 응답을 따로 해야 한다. A 프로세스는 Data\_Req(x)처럼 자신이 설정한 순서 번호  $x$ 를 사용해 데이터를 전송하고 있다. 이때 데이터 전송과 함께 B 프로세스가 설정한 순서 번호  $y$ 에 대해서도 응답해야 한다.

23.

#### ■ 일방적 연결 해제 절차

일방적 연결 해제 절차 방식에서는 통신하는 한쪽 프로세스가 일방적으로 Disc\_Req를 전

송해 연결 종료를 선언할 수 있다. Disc\_Req에 대한 상대 프로세스의 동의가 없어도 연결이 끊어지는 방식이다. 이 방식에서는 A 프로세스가 B 프로세스에 전송할 데이터가 남거나, 전송 중이지만 아직 완료되지 않은 시간  $t$ 에 상대 프로세스의 연결 종료가 이루어진다. 그러면 데이터 전송을 지원하는 원통 기능이 정지되어 데이터 전송을 완료할 수 없다는 문제가 발생한다(그림 9-5).

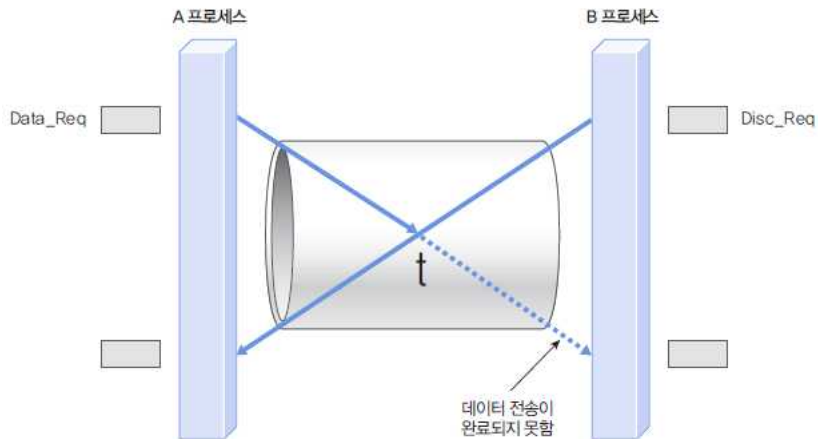


그림 9-5 일방적 연결 해제 절차

#### ■ 점진적 연결 해제 절차

점진적 연결 해제 절차 방식은 하나의 연결에 두 개의 단방향 연결을 지원하는 원통이 논리적으로 존재하는 것과 같다. 따라서 A 프로세스의 데이터 전송 과정 중에 B 프로세스의 연결 해제 요구가 발생해도 아래쪽 원통 기능만 정지하고, 위쪽 원통은 여전히 정상 기능을 수행할 수 있다. 즉, A 프로세스가 전송하는 데이터를 계속 전송할 수 있다. 두 프로세스 사이의 연결을 완전히 종료하려면 양쪽에서 자신에게 할당된 단방향의 원통 기능을 명시적으로 정지해야 된다. 점진적 연결 해제 절차 방식에서 연결을 해제하려면 두 프로세스 모두 Disc\_Req를 전송해야 한다.

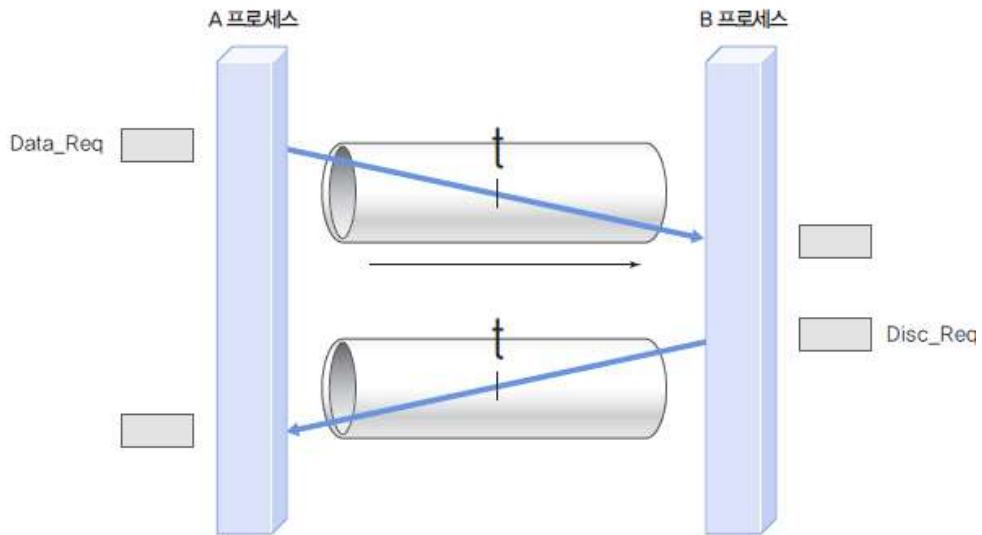


그림 9-6 점진적 연결 해제 절차

24.

TCP의 헤더 구조는 다음 그림과 같다.

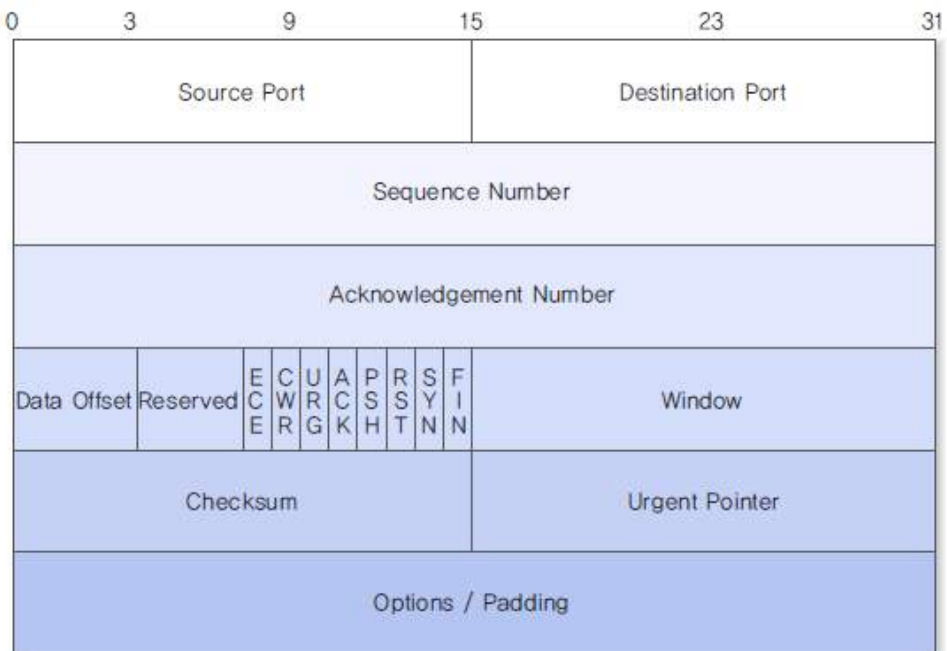


그림 9-8 TCP 헤더의 구조

TCP 헤더에서 정의된 필드의 의미와 용도는 다음과 같다.

- Source Port/Destination Port(송신 포트/수신 포트) : TCP로 연결되는 가상 회선 양단의 송수신 프로세스에 할당된 네트워크 포트 주소다. 인터넷 환경에서는 통신을 원하는 프로세스가 실행되는 호스트 주소와 호스트 내부에서 다른 프로세스와 구분할 수 있는 프로세스 전용의 고유 주소가 필요하다. 호스트 주소는 IP 프로토콜에서 정의된 호스트의 IP 주소를 사용해 구분할 수 있다. 프로세스에 할당되는 네트워크 자원을 포트(Port)라고 하는데, 포트를 구분하기 위한 고유 주소가 포트 번호다. 포트 번호는 TCP와 UDP가 별도의 주소 공간을 갖기 때문에 같은 번호를 사용할 수 있다.
- Sequence Number(순서 번호) : 송신자가 지정하는 순서 번호다. 세그먼트 전송 과정에서 전송되는 바이트 수를 기준으로 증가한다. 즉, TCP에서는 전송 데이터의 각 바이트마다 순서 번호가 존재한다. 크기가 32비트인 필드로 표시할 수 있고 최대 범위가 232개 가능해 크기가 충분히 크므로 순서 번호가 쉽게 중복되지 않는다. 송신자가 최초 데이터를 전송할 때는 임의의 순서 번호를 선택해 전송한다. 이는 전송 연결이 예기치 않은 이유로 끊어졌을 때 순서 번호가 혼선되는 것을 방지하기 위함이다.
- Acknowledgement Number(응답 번호) : 수신 프로세스가 제대로 수신한 바이트의 수를 응답하기 위해 사용한다. 필드 값은 ACK 플래그 비트가 지정된 경우에만 유효하며, 다음에 수신을 기대하는 데이터의 순서 번호를 표시해야 한다. ACK 응답을 받은 송신 프로세스는 Acknowledgement Number-1까지의 모든 데이터가 올바르게 전송되었음을 확인할 수 있다. 연결 설정이나 연결 해제처럼 데이터 세그먼트가 없는 경우에도 순서 번호가 1씩 증가한다.
- Data Offset(데이터 오프셋) : TCP 세그먼트가 시작되는 위치를 기준으로 데이터의 시작 위치를 나타내므로 TCP 헤더의 크기가 된다. 32비트 워드 단위로 표시된다.
- Reserved(예약) : 예약 필드다.
- Window(윈도우) : 슬라이딩 윈도우 프로토콜에서 수신 윈도우의 버퍼 크기를 지정하려고 사용하며, 수신 프로세스가 수신할 수 있는 바이트의 수를 표시한다. 수신자의 버퍼 용량초과 등으로 인해 데이터를 더는 수신할 수 없으면 Window 필드 값을 0으로 지정한다. 이 경우에 송신 프로세스는 데이터를 더 전송하면 안 된다.
- Checksum(체크섬) : TCP 세그먼트에 포함되는 프로토콜 헤더와 데이터 모두에 대한 변형오류를 검출하려고 사용한다. IP 프로토콜에서 사용하는 오류 검출 알고리즘을 사용한다.
- Urgent Pointer(긴급 포인터) : 긴급 데이터를 처리하기 위한 것으로, URG 플래그 비트가 지정된 경우에만 유효하다. 이 필드를 사용해 송신 프로세스가 긴급히 처리하려는 데이터를 전송할 수 있다. 예를 들어, Sequence Number =2,000, Urgent Pointer =100으로 지정한 패킷을 전송한다면 순서 번호 2,000번~2,100번의 데이터는 긴급 데이터로 전송되고 2,101번 이후는 다시 정상 데이터로 전송된다.

25.

인터넷 환경에서 많이 사용하는 네트워크 응용 서비스의 서버 프로세스에 할당된 포트 번호를 Well-known 포트라고 하는데, 전 세계의 모든 컴퓨터가 동일한 포트 번호를 사용하도록 권고되어 있다. 예를 들어, 다음 표처럼 FTP, 전자메일(SMTP), 웹 서비스(HTTP), DNS처럼 인터넷에서 자주 사용하는 응용 서비스는 모두 고정된 포트 번호를 사용한다.

표: Well-know 포트

서비스	포트번호
FTP(데이터채널)	20
FTP(제어채널)	21
Telnet(텔넷)	23
SMTP	25
DNS	53
HTTP	80
rlogin	513
rsh	514
portmap	111

네트워크 서비스를 제공하는 포트 번호는 컴퓨터의 파일 시스템에 보관되므로 일반 사용자가 포트 번호를 직접 지정하는 경우는 없다. 사용자가 연결을 원하는 서버의 호스트 IP 주소만 클라이언트 프로그램에게 지정하고, 포트 번호 선택은 프로그램에서 자동으로 해준다. TCP와 UDP는 별도의 포트 주소 공간을 관리하므로 동일한 포트 번호를 사용할 수 있다. 즉, 두 프로토콜에서 동일한 포트 번호를 할당해도 서로 다른 포트로 간주된다.

26.

TCP를 사용하는 프로세스가 가장 먼저 실행하는 연결 설정은 [그림 9-10]과 같은 3단계 설정(Three-Way Handshake) 방식이다. 그림에서는 A 프로세스가 연결 설정을 요구하고, B프로세스가 이를 수락하는 형식이다.



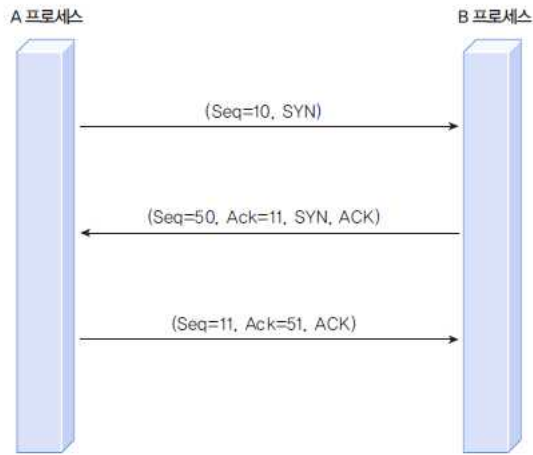


그림 9-10 TCP 연결 설정

먼저 A 프로세스는 TCP 헤더의 SYN 플래그를 지정한 세그먼트를 전송함으로써 연결 설정을 요구한다. 순서 번호 10번은 임의로 설정한 것이다. 연결 설정 요구를 받은 B 프로세스가 연결을 수락하려면 이에 대한 긍정 응답을 해야 한다. 이를 위해 SYN과 ACK 플래그를 지정해 연결에 대한 긍정 응답을 표시하였다. SYN 플래그가 지정한 세그먼트에는 전송 데이터가 포함되지 않지만 순서 번호는 1이 증가한다. 따라서 SYN 세그먼트의 순서 번호 10에 1을 더한 11번을 Acknowledgment Number 필드에 지정해 회신해야 한다. Acknowledgment Number 값을 유효하게 하기 위해 ACK 플래그를 지정했으며, B 프로세스의 순서 번호 50번은 임의로 지정하였다.

마지막에 있는 세 번째 세그먼트는 B 프로세스가 전송한 연결 수락 세그먼트가 제대로 도착했음을 알린다. [그림 9-10]은 A 프로세스가 전송할 데이터가 없을 때 처리하는 방식이고, 전송 데이터가 있으면 세 번째 세그먼트에서 바로 데이터를 전송해도 된다.

27.

#### ■ 정상적인 데이터 전송

TCP에서의 데이터 전송은 [그림 9-11]처럼 양쪽 프로세스가 동시에 데이터를 전송할 수 있는 전이중 방식을 지원한다. 그림은 A 프로세스가 먼저 순서 번호 11번부터 5바이트의 데이터를 전송하고, B 프로세스는 순서번호 51번부터 10바이트의 데이터를 전송한다. 순서 번호와 응답 번호의 조합은 전송되는 데이터양에 따라 적절히 조정되고 있다.

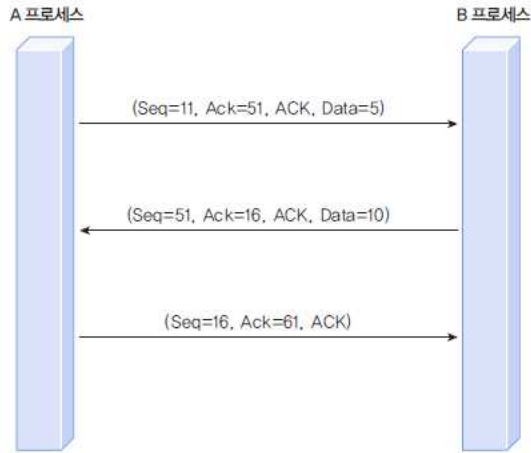


그림 9-11 TCP 데이터 전송

TCP에서의 흐름 제어는 프로토콜 헤더의 Window 필드를 사용한다. Window 필드에 지정한 값 이상으로 데이터를 전송할 수 없으므로 수신자는 자신의 처리 능력에 맞는 값으로 지정해 송신자의 전송 속도를 제어할 수 있다. 예를 들어, 송신자가 데이터를 1,000번부터 전송했을 때 수신자로부터 Window = 300으로 지정된 세그먼트가 도착하면 1,299번까지의 데이터를 연속으로 전송할 수 있다. 그러나 다음 데이터는 수신자로부터 응답이 도착했을 때만 전송할 수 있다.

#### ■ 데이터 전송 오류가 발생한 경우

TCP 데이터를 세그먼트(Segment)라 부르며, 순서 번호와 함께 전송된다. 순서 번호는 상위 계층에서 보낸 데이터의 바이트 수에 기초하여 부여되고, 수신자는 이 번호를 근거로 데이터의 순서를 올바르게 정렬할 수 있다. 순서 번호가 동일한 데이터가 도착하면 중복으로 판단하여 해당 데이터를 버리고, 중간 순서 번호가 빠지면 데이터를 잃어버렸다고 판단할 수 있다.

TCP는 부정 응답 기능인 NAK를 사용하지 않는다. 따라서 수신 프로세스에 도착한 데이터 세그먼트의 내용이 변형되어도 수신 프로세스가 응답을 하지 않아 데이터 분실과 동일하게 처리된다. 데이터 변형과 데이터 분실 오류가 발생하면 수신 프로세스로부터 회신을 받을 수 없으므로 송신 프로세스의 타임아웃(Timeout) 기능에 의해 오류가 복구된다. [그림 9-12]는 A 프로세스가 TCP 세그먼트 세 개를 연속으로 전송하고, 이중 맨 마지막 세그먼트에 오류가 발생했다고 가정한 경우다.



그림 9-12 전송 오류

A 프로세스가 순서 번호 11~20, 21~30, 31~40인 데이터 세그먼트를 전송하고, 이들이 모두 정상적으로 수신되면 41번 응답 번호를 긍정 응답으로 회신 받을 수 있다. 그러나 그림에 표시된 것처럼 맨 마지막의 31~40번 데이터 세그먼트에 오류가 발생하면 이에 대한 오류 제어를 수행해야 한다. B 프로세스에서는 11~20, 21~30번의 데이터를 올바르게 수신했으므로, 다음에 수신할 데이터는 31번이 된다. 따라서 ACK 플래그가 지정된 응답 세그먼트의 순서 번호는 31번으로 지정된다. 한편 A 프로세스는 31~40번 데이터에 대한 긍정 응답을 받지 못했으므로 타임아웃 기능을 통해 해당 세그먼트를 재전송해야 한다. 그림에는 표시하지 않았지만 긍정 응답 세그먼트에 오류가 발생하면 수신 프로세스가 정상적으로 수신한 세그먼트를 재전송할 수 있다. 따라서 세그먼트가 중복으로 수신될 수 있기 때문에 수신 프로세스는 순서 번호를 기준으로 중복 여부를 처리할 수 있어야 한다.

28.

CP의 연결 해제 단계는 [그림 9-13]처럼 연결을 해제하고자 하는 쪽에서 FIN 플래그를 지정해 요구할 수 있다. 연결 해제는 양쪽 프로세스의 동의하에 진행되기 때문에 연결 해제 세그먼트를 받은 프로세스가 FIN 플래그로 응답할 때까지 연결은 계속 유지된다.

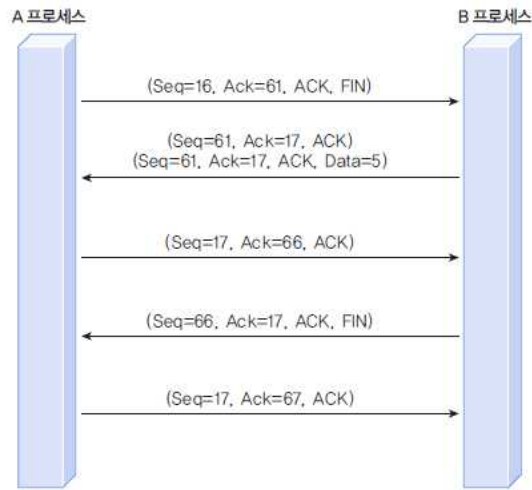


그림 9-13 TCP 연결 해제

29. TCP/IP 환경에서 혼잡 제어 기능의 동작 원리에 대하여 설명하시오.

ECN 기능이 동작하는 TCP 연결을 사용하여 데이터를 전송하는 과정의 혼잡 제어 처리는 [그림 9-15]와 같다. 먼저 TCP 세그먼트를 전송하는 A 프로세스의 IP 프로토콜은 IP 헤더 내의 ECN 필드 값을 ECT(01 혹은 10)로 설정하여 전송한다. 그림에서 (a)로 표시된 부분이며, 전송되는 모든 데이터는 항상 이 값이 설정된 상태로 전송된다. 참고로 IP 헤더에 ECT가 설정되었다는 의미는 TCP가 ECN 기능을 지원한다는 의미이다.

이후, 임의의 시점에서 네트워크에 혼잡이 발생하고, 라우터가 이를 감지한 경우가 (b)이다. 혼잡을 감지한 라우터는 IP 헤더에 ECT가 설정된 패킷들에 대해 CE를 설정하여 혼잡이 발생했음을 알린다. 이때, 라우터가 송신 프로세스인 A 프로세스에 직접 혼잡을 통지하지 않고, 수신 프로세스인 B 프로세스에 통지하는 것에 주의한다. 즉 수신 프로세스의 중개Echo를 거쳐 간접적으로 송신 프로세스에 혼잡을 통지한다.

이와 같이 수신 프로세스의 중개를 거쳐야 하는 이유는 TCP와 IP 프로토콜이 혼재하여 동작하는 상황에서 네트워크 계층을 수행하는 라우터가 지니는 구조적인 한계 때문이다. 더불어, 혼잡을 인지한 라우터 다음에 위치한 라우터들이 ECN 기능을 반복적으로 수행하지 못하도록 하는 효과도 얻는다.

라우터로부터 IP 헤더의 CE 값을 받은 B 프로세스는 모든 TCP 세그먼트에 대해 TCP 헤더의 ECE 플래그를 지정하여 라우터가 감지한 혼잡을 A 프로세스에게 알려주며, 이는 (d)로 표현되었다. 혼잡을 인지한 A 프로세스는 송신 윈도우를 조절하여 전송되는 데이터의 양을 줄이고, TCP 헤더의 CWR 플래그를 지정함으로써 혼잡에 적절한 조치를 취했음을 통지한다.

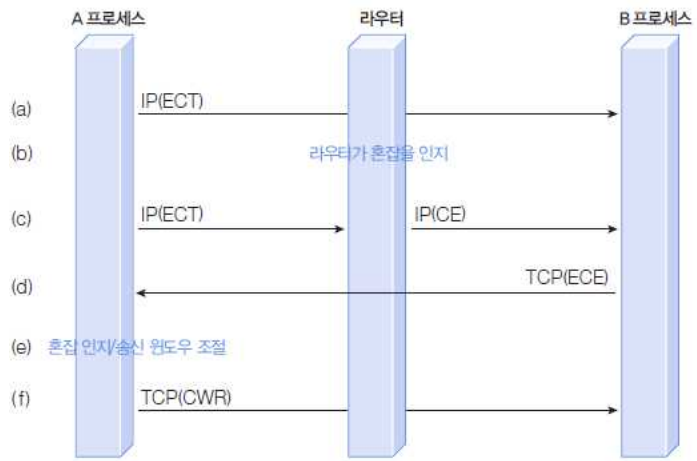


그림 9-15 ECN의 동작 원리