

**데이타링크 계층**

**편집: 김혜영**

# 01\_MAC 계층과 IEEE 802 시리즈

## ❖ MAC 계층의 이해

- LAN 환경에서 계층 2 기능을 MAC과 LLC 계층으로 분리
  - LLC 계층 : 데이터 링크 계층의 기본 기능
  - MAC 계층 : 물리 계층 기능

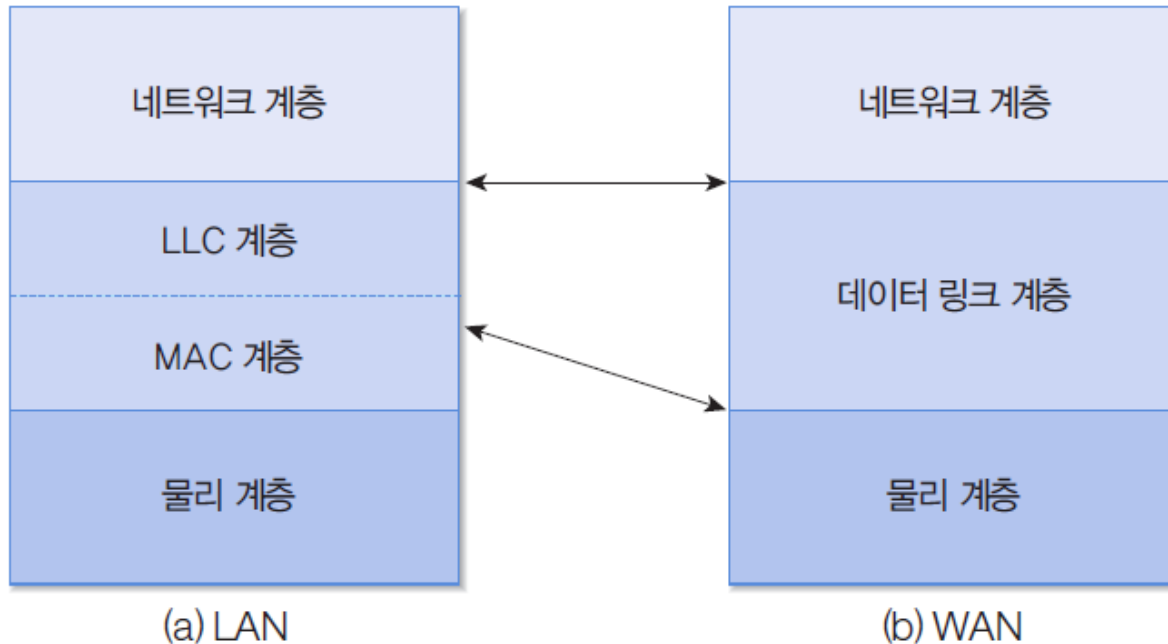


그림 5-1 MAC과 LLC 계층의 관계

# 01\_MAC 계층과 IEEE 802 시리즈

## ■ MAC 계층

- MAC<sup>Medium Access Control</sup> 계층 : 물리적인 특성 반영
- 이더넷
  - 공유 버스 방식 지원
  - 둘 이상의 호스트에서 동시에 데이터 프레임 전송을 시도하면 충돌 발생
- 토큰 링 방식
  - 링 구조를 지원
  - 토큰이라는 특정 패턴의 제어 데이터가 링을 순환
  - 토큰을 확보하여 전송 후 토큰은 링에 내려 놔야 함
  - 각 호스트에 전송 우선순위를 부여할 수 있음

## ■ LLC 계층

- WAN 환경의 데이터 링크 계층과 기능을 수행

# 01\_MAC 계층과 IEEE 802 시리즈

- IEEE: Institute of Electrical and Electronics Engineers 전지전자기술자협회
- IEEE 802.1 : 표준안 전체를 소개
- IEEE 802.2 : LLC 계층을 소개
- IEEE 802.3 부터 : 다양한 환경의 MAC 계층을 소개

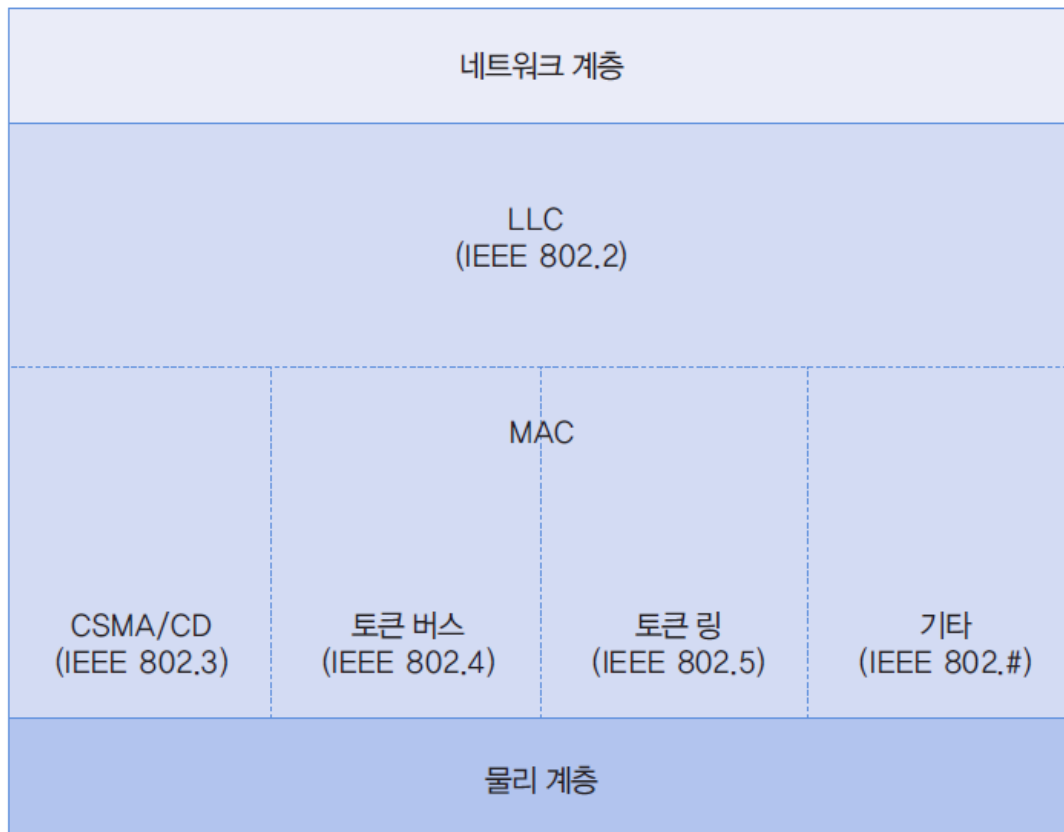


그림 5-2 IEEE 802 시리즈의 계층 구조

# 01\_MAC 계층과 IEEE 802 시리즈

## ❖ CSMA/CD Carrier Sense Multiple Access/Collision Detection

- 공유 매체에서의 충돌 문제 해결
  - 충돌 허용 방식: CSMA/CD
  - 충돌 회피 방식: 전송 시간대를 달리하는 타임 슬롯 방식
- CSMA/CD
  - 충돌 감지 기능이 필수적으로 요구됨
  - 충돌 감지 후, 재전송 기능으로 오류 복구
  - 공유 매체의 길이가 길수록 전송 지연이 증가하여 충돌 가능성이 높아짐
  -

# 01\_MAC 계층과 IEEE 802 시리즈

- 데이터 전송 원리
  - 호스트 5가 호스트 1에게 데이터 전송
  - 전송 프레임에 목적지 주소 1을 기록하여 수신 호스트를 지정함

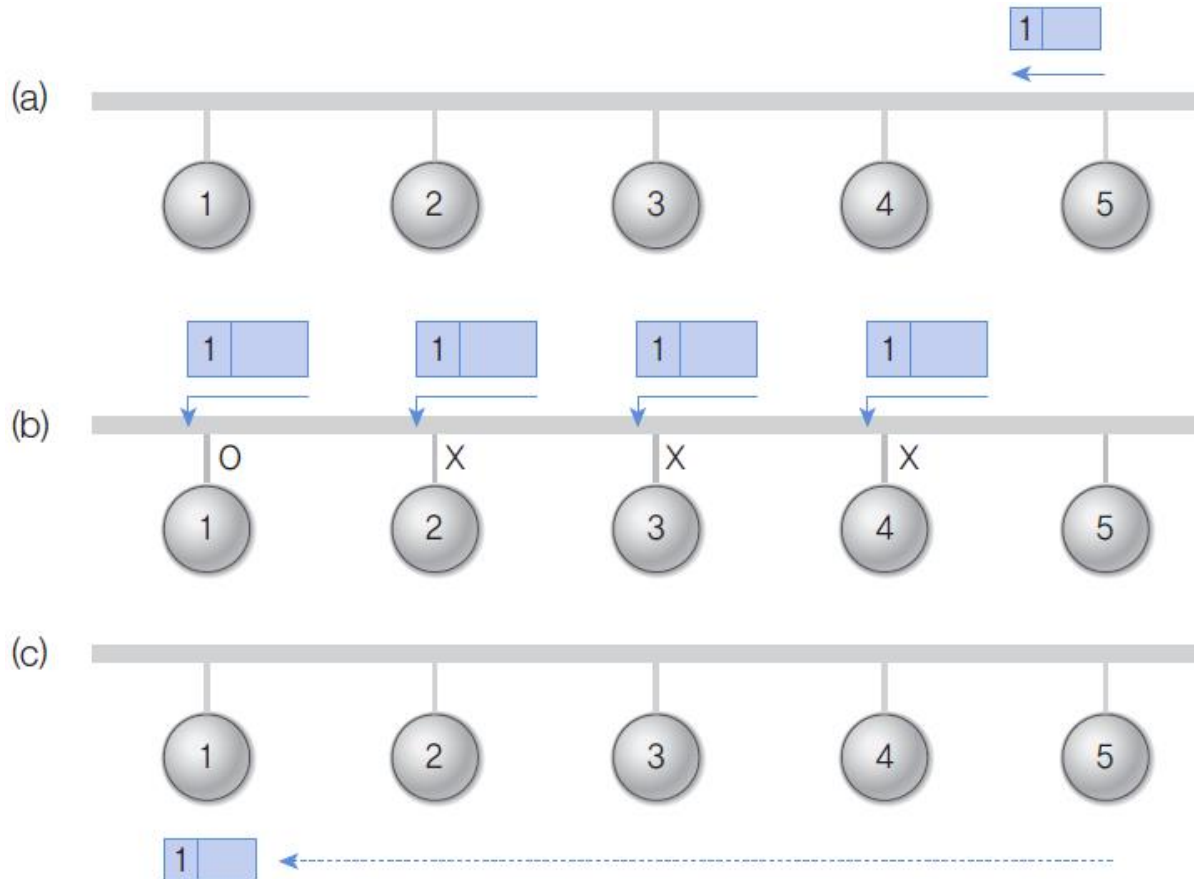


그림 5-3 공유 버스에서의 데이터 전송

# 01\_MAC 계층과 IEEE 802 시리즈

## ■ 토큰 버스 Token Bus

- 물리적인 버스 구조(점선)로 연결되지만 논리적인 프레임 전달은 링 구조(실선)
- 토큰이라는 제어 프레임을 사용, 반드시 토큰을 확보해야 함

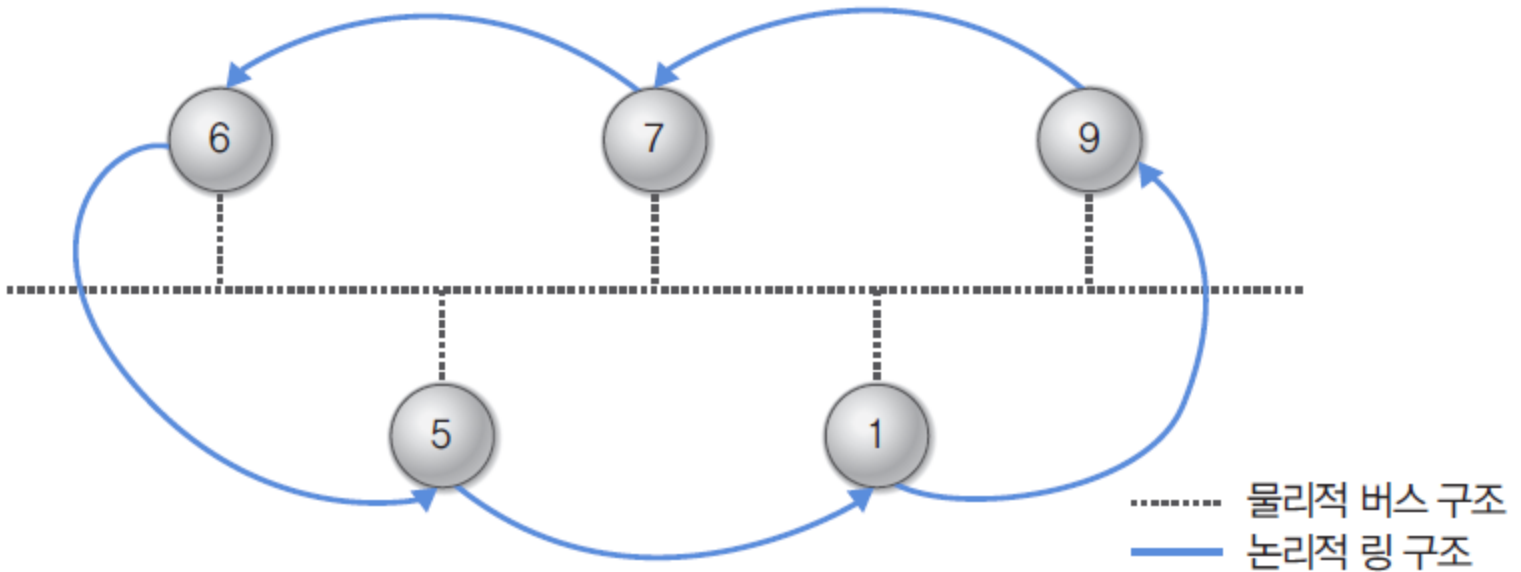


그림 5-4 토큰 버스

# 01\_MAC 계층과 IEEE 802 시리즈

## ■ 토큰 링

- 물리적으로 링 구조를 지원
- 호스트의 동작 방식 : 대기 모드, 전송 모드
- 대기 모드
  - 입력단으로 들어온 데이터를 바로 출력단으로 보냄
  - 호스트가 다운되거나 기타 장애가 발생해도 네트워크에 영향을 주지 않기 위함
- 전송 모드
  - 토큰을 확보해 데이터를 전송할 수 있는 권한을 보유한 상태
  - 호스트가 중개 기능을 수행하며, 출력단을 통해 데이터를 전송할 수 있음



# 01\_MAC 계층과 IEEE 802 시리즈

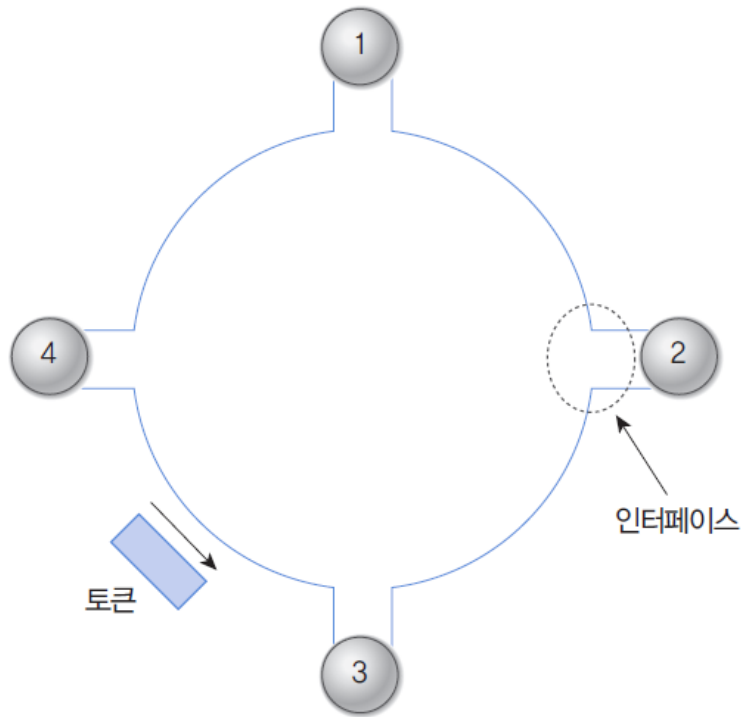


그림 5-5 토큰 링

- 송신 호스트가 전송한 프레임이 링을 한 바퀴 돈 후 송신 호스트에 되돌아옴
- 프레임이 올바르게 전송되면 데이터 프레임을 회수하고 토큰 프레임을 링에 반환

## 02\_이더넷

- CSMA/CD
  - 둘 이상의 호스트에서 채널이 유헤 상태라고 판단할 수 있음
  - 이런 경우 프레임 전송 과정에서 충돌이 발생
  - 따라서 충돌 감지 기능이 필수적으로 요구됨
  - 충돌이 감지되면 진행중인 프레임의 전송을 중지

## 02\_이더넷

- 이더넷의 연결 [그림 5-6]

- 굵고 긴 전송케이블로 된 전송 매체에 트랜시버 장비로 보조선을 연결해 각 호스트를 연결
- 트랜시버는 호스트를 전송 케이블에 연결하기 위한 송수신 장치
- 트랜시버는 충돌이 발생했음을 알려줌으로써 무의미한 프레임 전송을 억제
- 리피터는 단순히 신호를 증폭하여 이웃하는 케이블로 넘겨주는 기능
- 왼쪽 케이블에서 전송되는 신호는 리피터를 통해 오른쪽으로 전달, 반대 방향으로도 신호를 증폭해 전달

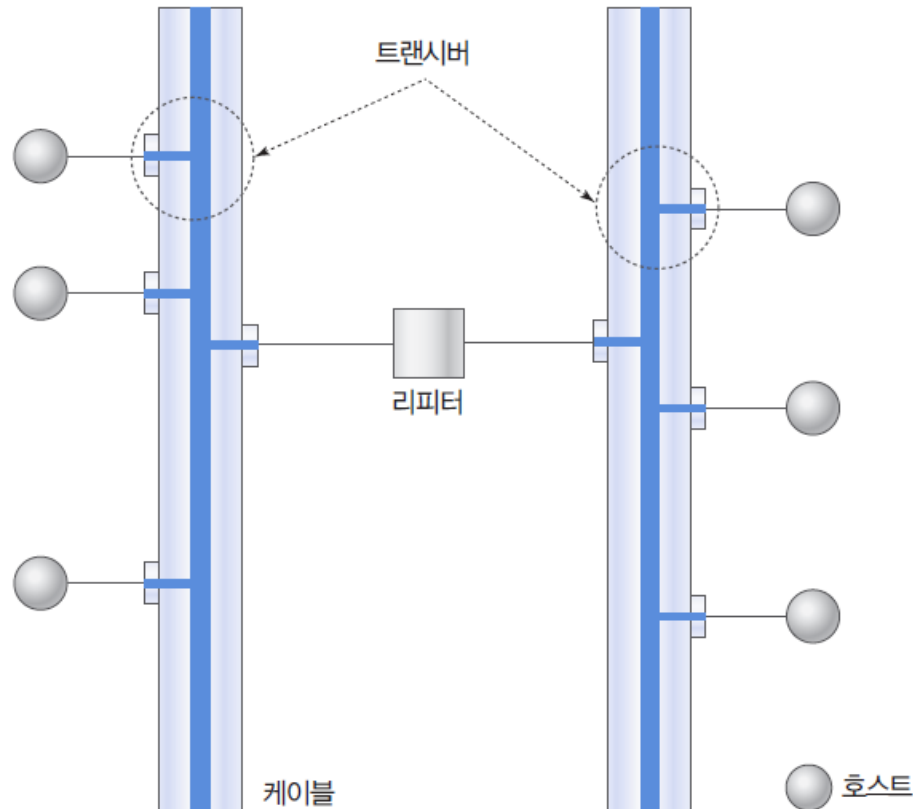


그림 5-6 이더넷의 연결

## 02\_이더넷

### ■ 프레임 구조

- MAC 프레임<sup>MAC Frame</sup> : MAC 계층 프로토콜에 정의된 MAC 헤더와 트레일러 정보를 추가한 것
- MAC 프레임은 LLC 계층에서 보낸 정보를 전송 데이터로 취급, 데이터 앞에는 헤더가, 뒤에는 트레일러가 위치
- 이더넷 프레임<sup>Ethernet Frame</sup> 구조
  - 필드의 단위는 바이트
  - Source Address와 Destination Address 필드 : 6바이트의 MAC 주소를 사용
  - Data와 Padding 필드 : 가변 길이를 지원
  - Data 필드 왼쪽에 위치한 필드들은 헤더, 오른쪽은 트레일러에 속함

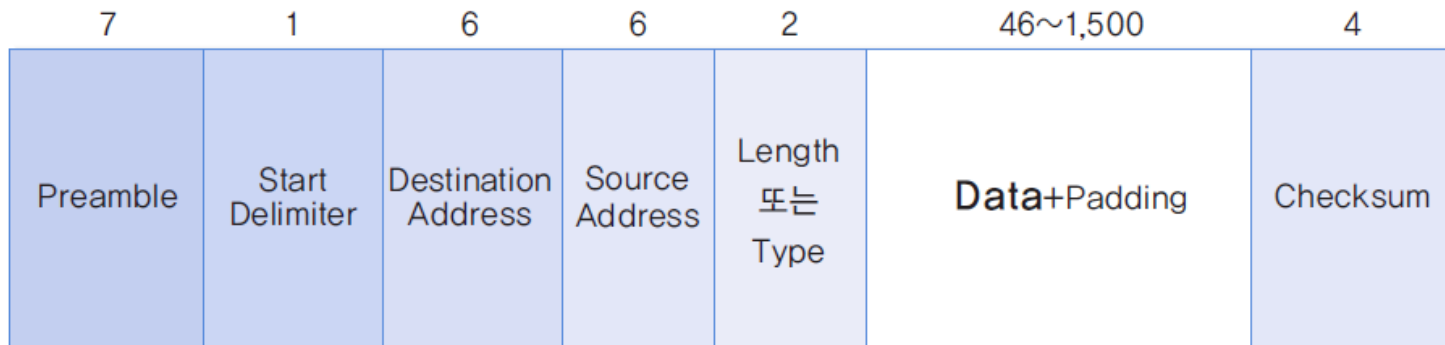


그림 5-7 이더넷 프레임의 구조

## 02\_이더넷

- 헤더와 트레일러에서 정의한 필드의 의미
  - Preamble(프리앰블) : 7바이트 크기로, 수신 호스트가 송신 호스트의 클럭과 동기를 맞출 수 있도록 시간 여유를 제공하는 것이 목적
  - Start Delimiter(시작 구분자) : 프레임이 시작된다는 의미로 사용, Preamble 필드와 구분하기 위해 10101011의 값을 가짐
  - Source Address/Destination Address(송신 호스트 주소/수신 호스트 주소) : MAC 계층에서는 호스트를 구분하는 고유의 MAC 주소를 사용
  - Length/Type : 필드 값이 1,500 이하이면 Data 필드의 데이터 크기를 의미하는 Length로 해석하고, 그렇지 않으면 Type으로 해석
  - Length(길이) : Data 필드에 포함된 가변 길이의 전송 데이터 크기
  - Type(종류) : 이더넷 프레임에 캡슐화된 상위 프로토콜의 패킷 종류를 구분
  - Checksum(체크섬) : 데이터 전송 과정에서 데이터 변형 오류의 발생 여부를 수신 호스트가 확인할 수 있도록 송신 호스트가 값을 기록

## 02\_이더넷

### ❖ LLC 프레임 캡슐화

- 이더넷 프레임에서 Data 필드를 제외한 필드들이 MAC 계층에서 추가하는 정보

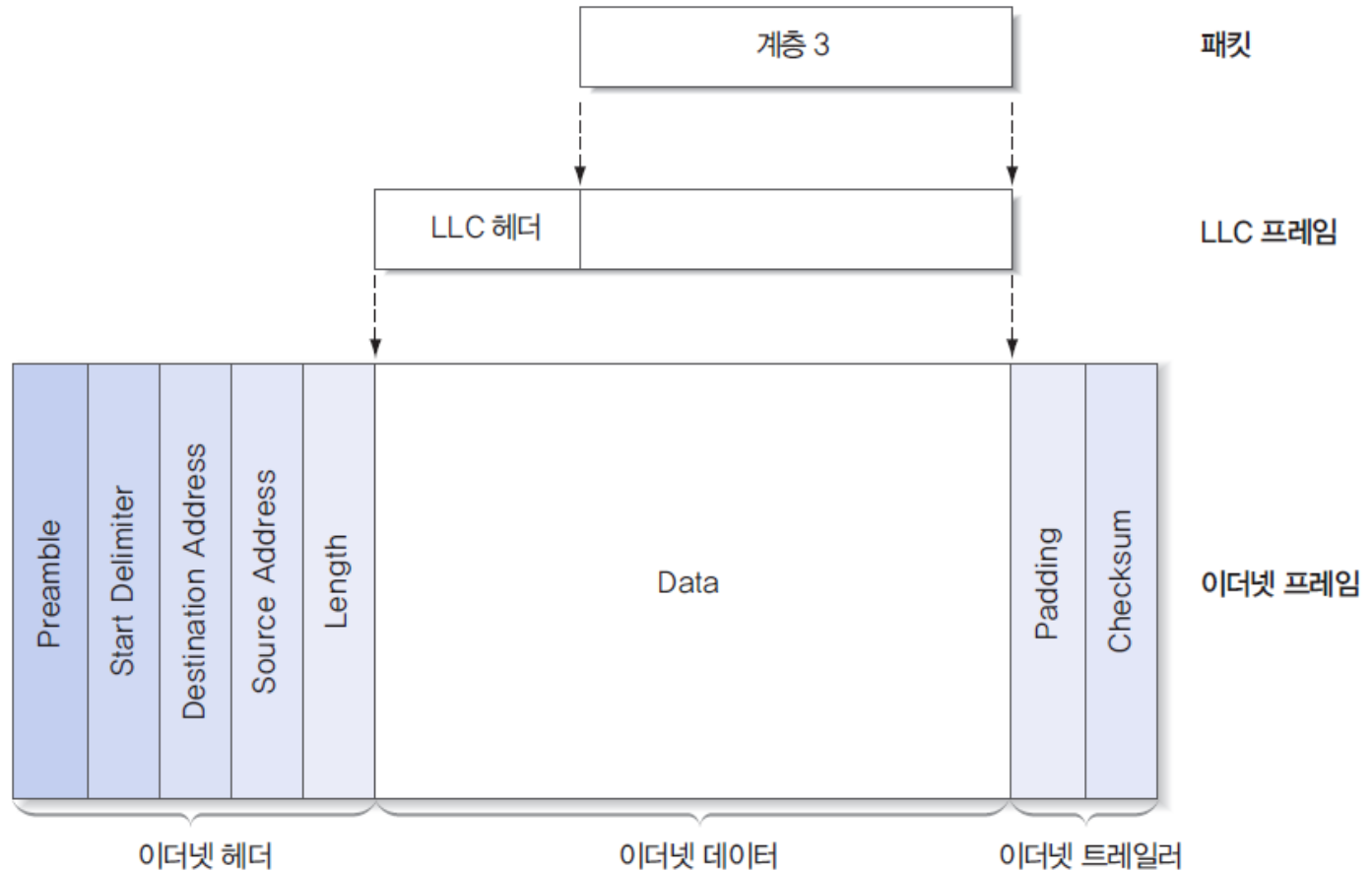


그림 5-8 이더넷 프레임의 Data 필드

## 02\_이더넷

### ❖ 허브와 스위치

- 허브 : 박스 형태의 장비에 잭을 연결해서 이더넷 네트워크를 구성
- 스위치 허브 : 허브의 성능을 향상시킨 장비

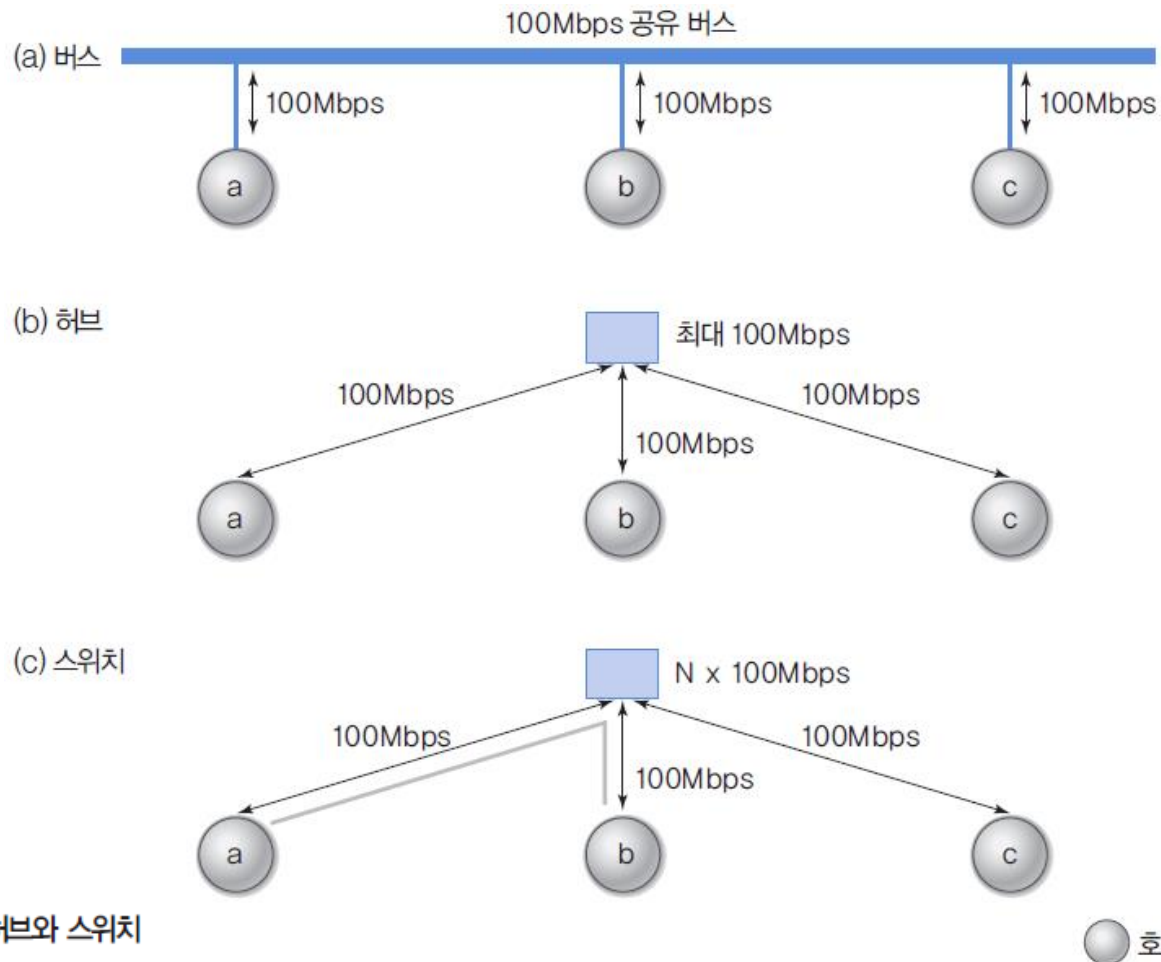


그림 5-9 허브와 스위치

## 02\_이더넷

### ■ 허브 Hub

- 각 호스트는 외형상 허브에 스타형 구조로 연결
- 허브의 내부 동작은 공유 버스 방식

### ■ 스위치

- 모든 호스트에게 프레임을 전송하지 않음
- 목적지로 지정된 호스트에게만 프레임 전송
- 따라서 동시에 여러 호스트가 데이터를 전송할 수 있음
- 장점
  - 스위치 허브의 용량이 허용되면 각각의 호스트는 할당된 LAN 용량을 모두 사용함
  - 일반 허브를 스위치 허브로 교체하는 과정이 간단함



## 03\_토큰 버스

### ❖ 프레임 구조

- LLC 계층에서 내려온 LLC 프레임을 수신 호스트에 전달하려면 토큰 버스 프레임을 만들어야 함
- 토큰 버스 프레임의 구조
  - 데이터 프레임과 토큰 프레임을 구분하기 위한 Frame Control 필드가 추가됨

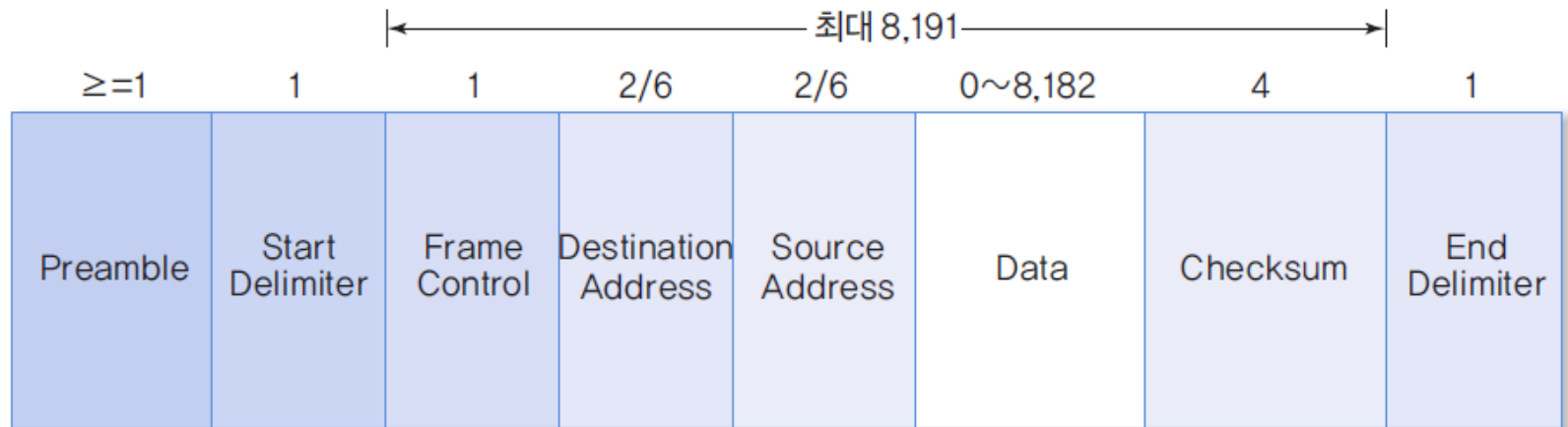
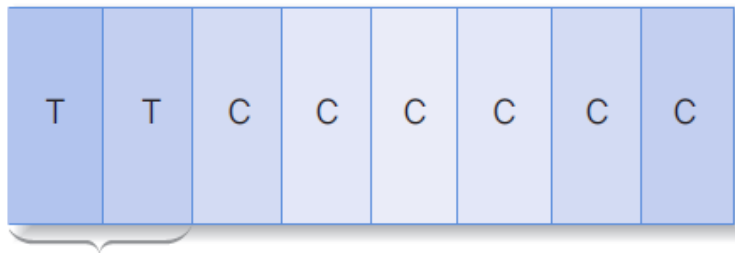


그림 5-10 토큰 버스 프레임의 구조

## 03\_토큰 버스

- 헤더와 트레일러에 정의된 필드의 기능
  - Start Delimiter/End Delimiter(시작 구분자/끝 구분자)
    - 프레임의 시작과 끝을 의미하는 경계를 표시
  - Preamble/Source Address/Destination Address/Checksum(프리앰블/송신 호스트 주소/수신 호스트 주소/체크섬)
    - 이더넷 프레임과 기능이 동일
  - Frame Control(프레임 제어)
    - 데이터 프레임과 제어 프레임을 구분
- Frame Control 필드의 용도



0	0 : 제어용 MAC 프레임
0	1 : LLC 프레임
1	0 : 네트워크 관리용 데이터 프레임
1	1 : 예약

그림 5-11 토큰 버스 프레임의 Frame Control 필드

# 04\_토큰 링

## ❖ 프레임 구조

- 토큰 링 프레임 Token Ring Frame
  - 토큰 프레임 : SD, AC, ED 필드
  - 데이터 프레임 [그림5-13]의 (b)
  - 모니터<sup>Monitor</sup> : 특별한 기능을 수행하는 관리 호스트

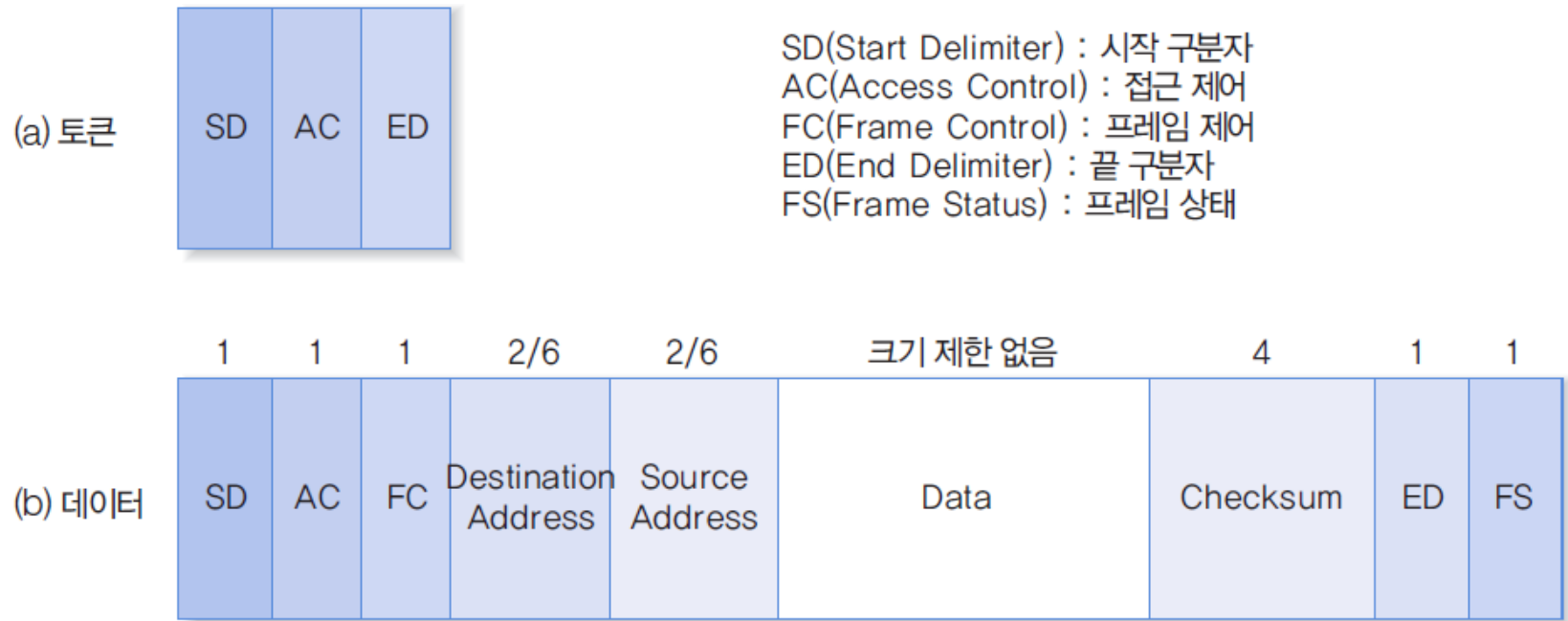


그림 5-13 토큰 링 프레임의 구조

# 04\_토큰 링

## ❖ LLC 프레임 캡슐화

- LLC 계층에서 MAC 계층으로 전송 요청이 내려온 LLC 프레임을 토큰 링 프레임의 구조로 캡슐화하는 과정

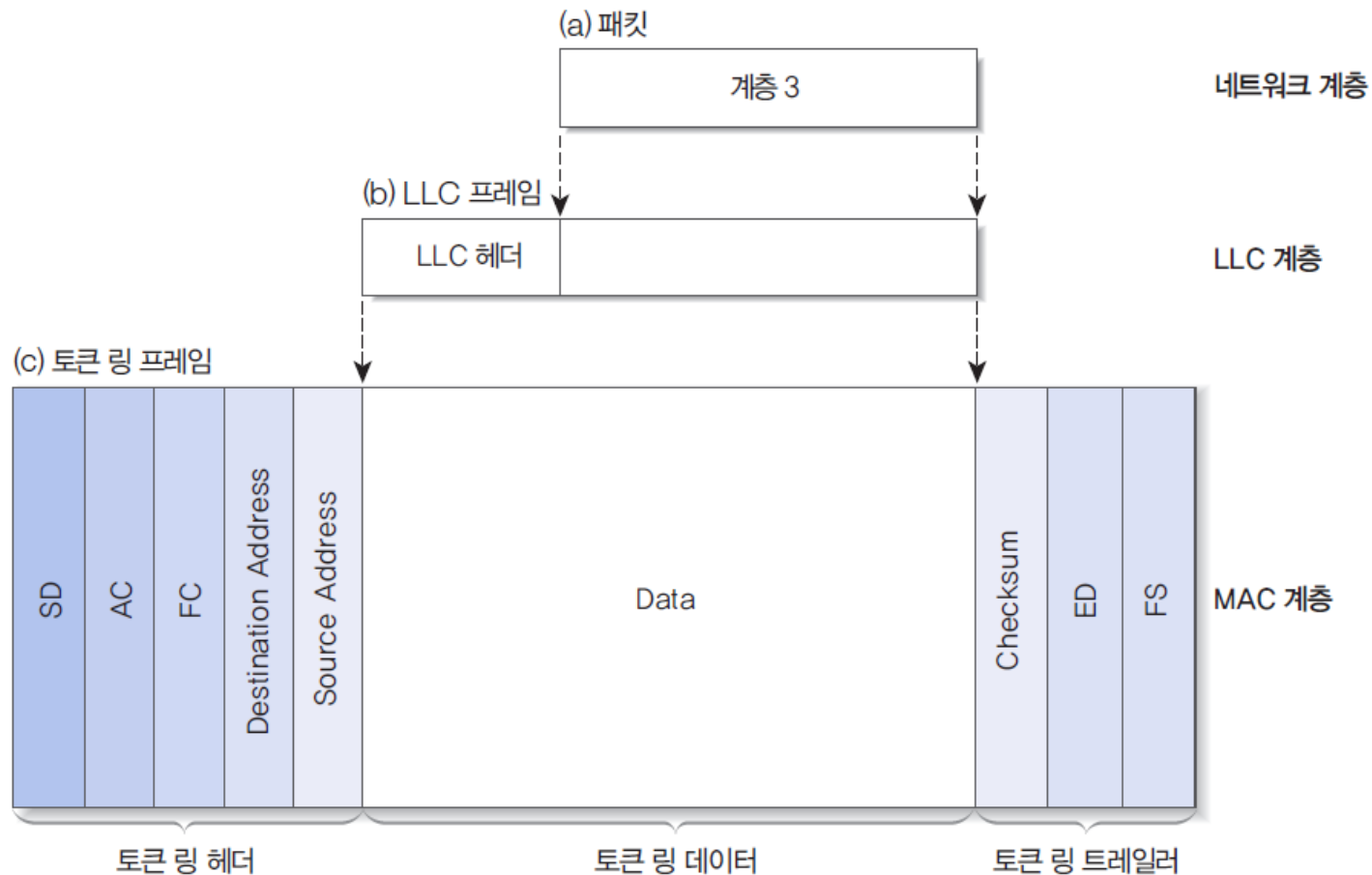


그림 5-14 토큰 링 프레임의 Data 필드

# 01\_데이터 링크 계층 프로토콜의 기초

## ❖ 프레임의 종류

- 정보 프레임 ( I 프레임 )
  - 상위 계층이 전송을 요구한 데이터를 수신 호스트에 전송하는 용도로 사용
  - 순서번호, 송수신 호스트 정보 등이 포함됨
  
- 긍정 응답 프레임 ( ACK 프레임 )
  - 전송 데이터가 올바르게 도착했음을 회신하는 용도
  - 데이터를 수신한 호스트가 데이터를 송신한 호스트에게 전송
  
- 부정 응답 프레임 (NAK 프레임 )
  - 전송 과정에서 프레임 변형 오류가 발생했음을 회신하는 용도
  - 원래의 정보 프레임을 재전송하도록 요청
  - 송신 호스트는 오류가 발생한 프레임을 동일한 순서 번호로 다시 전송

# 01\_데이터 링크 계층 프로토콜의 기초

## ❖ 오류·흐름 제어가 없는 프로토콜

### ■ 가정 : 가장 이상적인 통신환경

- 단방향 통신 : 데이터는 송신 호스트에서 수신 호스트로만(한쪽 방향으로만) 전달
- 전송 오류 없는 물리 매체 : 통신 채널에서는 전송 오류가 발생하지 않음
- 무한 개의 수신 버퍼 : 수신 호스트의 버퍼 수는 무한함

### ■ 단순 프로토콜

- 송신 호스트는 원하는 만큼 자유롭게 프레임을 전송할 수 있음
- 오류 제어 : 프레임 분실/변형 오류가 발생하지 않음
- 흐름 제어 : 수신 버퍼가 무한이므로 분실 오류 없음

# 01\_데이터 링크 계층 프로토콜의 기초

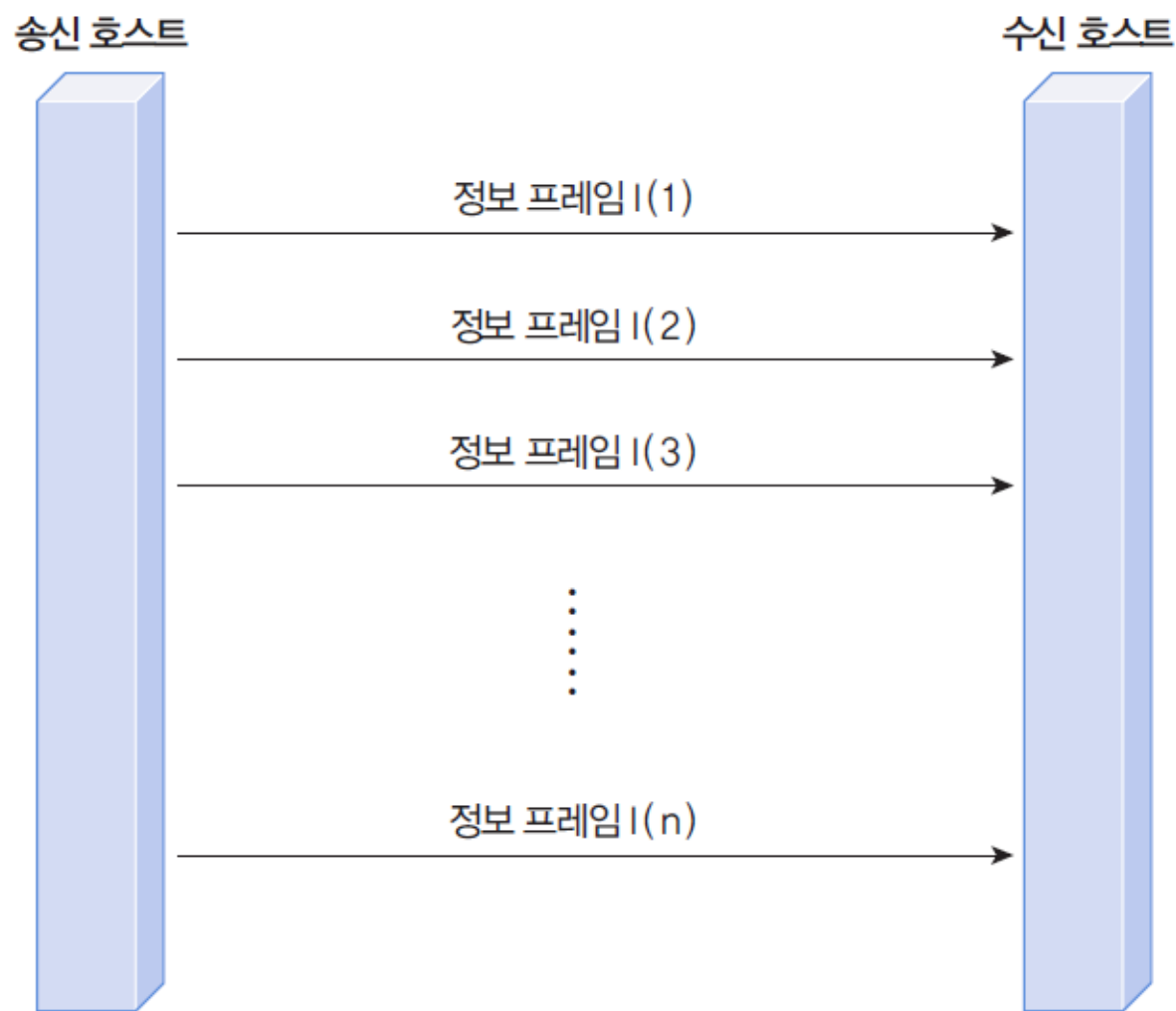


그림 6-2 단순 프로토콜

# 01\_데이터 링크 계층 프로토콜의 기초

## ❖ 오류 제어가 없는 프로토콜

- 가정 : 수신 호스트의 버퍼 개수가 유한(버퍼 개수 제한)
  - 단방향 통신 : 데이터는 송신 호스트에서 수신 호스트로만(한쪽 방향으로만) 전달
  - 전송 오류 없는 물리 매체 : 통신 채널에서는 전송 오류가 발생하지 않음
- 정지-대기 프로토콜 1
  - 수신 호스트 버퍼 개수가 제한일 경우 흐름 제어 기능으로 송신 호스트의 전송 속도를 조절함
  - ACK 프레임 : 송신 호스트에 긍정 응답의 기능을 수행, 다음 프레임을 전송하도록 지시하는 흐름 제어 기능도 수행
  - 정지-대기<sup>Stop-and-Wait</sup> 방식 : 수신 호스트가 회신하는 ACK 프레임이 도착해야 다음 프레임을 전송할 수 있는 프로토콜 방식



# 01\_데이터 링크 계층 프로토콜의 기초



그림 6-3 정지-대기 프로토콜 1

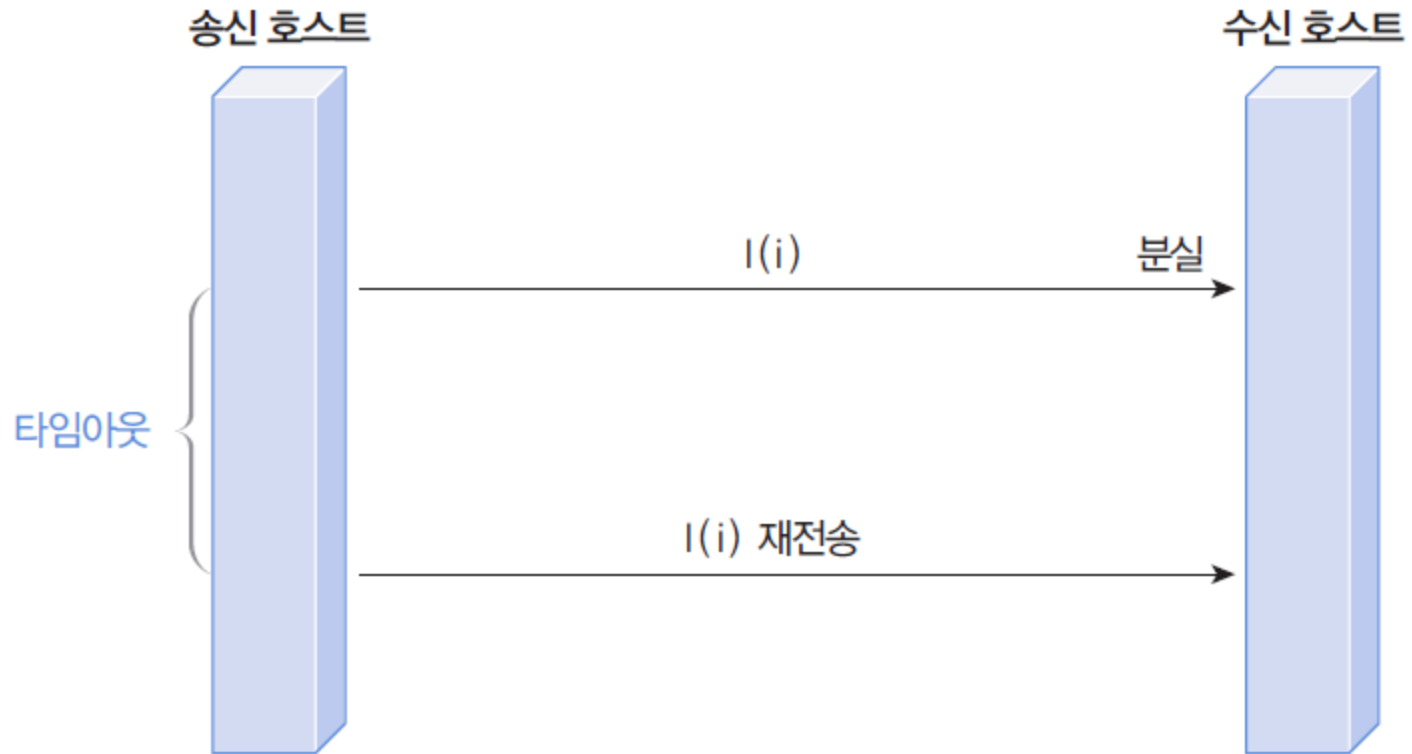
# 01\_데이터 링크 계층 프로토콜의 기초

## ❖ 단방향 프로토콜

- 가정 : 오류 제어와 흐름 제어 기능 지원
  - 단방향 통신 : 데이터는 송신 호스트에서 수신 호스트로만(한쪽 방향으로만) 전달
- 오류 제어와 흐름 제어가 모두 필요
- 프레임 변형 오류를 해결하기 위한 수신 호스트의 NAK 기능 필요
- 프레임 분실 오류를 해결하기 위한 송신 호스트의 타임아웃 기능 필요

# 01\_데이터 링크 계층 프로토콜의 기초

- NAK가 없는 경우
  - 정보 프레임 분실: 송신 호스트의 타임아웃 기능으로 오류 복구

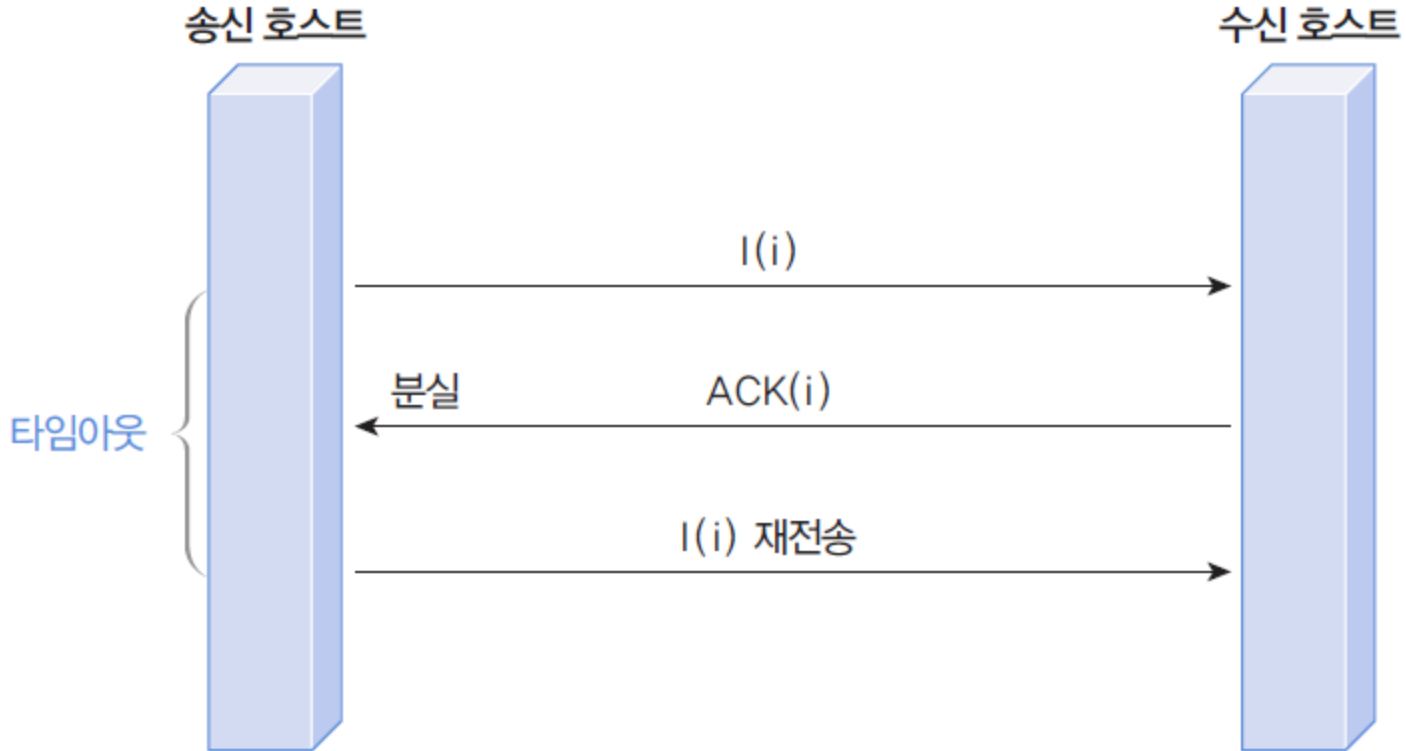


(a) 정보 프레임 분실 오류

그림 6-4 정지-대기 프로토콜 2 : NAK가 없는 경우

# 01\_데이터 링크 계층 프로토콜의 기초

- ACK 프레임 분실: 송신 호스트의 타임아웃 기능으로 오류 복구



(b) ACK 프레임 분실 오류

그림 6-4 정지-대기 프로토콜 2 : NAK가 없는 경우

# 01\_데이터 링크 계층 프로토콜의 기초

- 프레임 변형 오류 : 송신 호스트의 타임아웃 기능으로 오류 복구

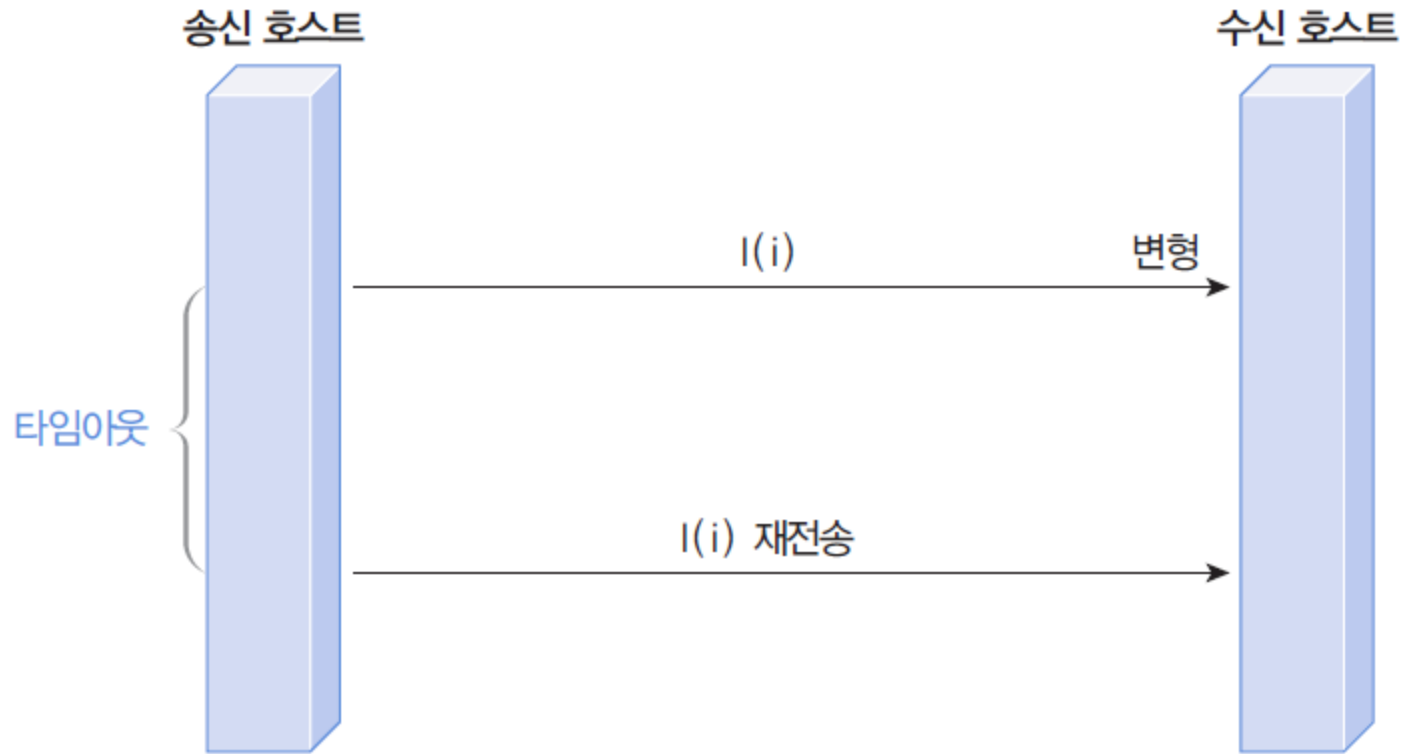


그림 6-5 프레임 변형 오류

# 01\_데이터 링크 계층 프로토콜의 기초

- NAK가 있는 경우
  - 프레임 변형 오류 : 수신 호스트의 NAK 프레임 응답으로 오류 복구

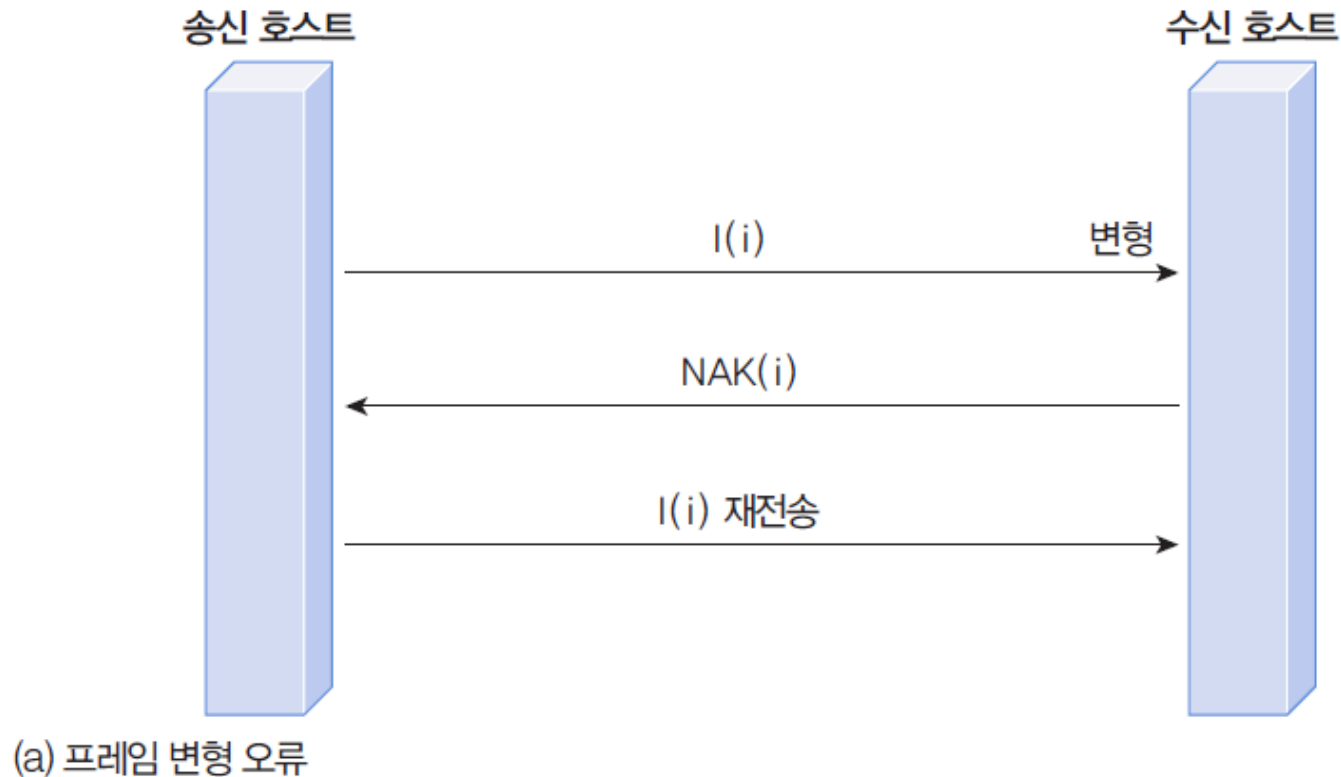
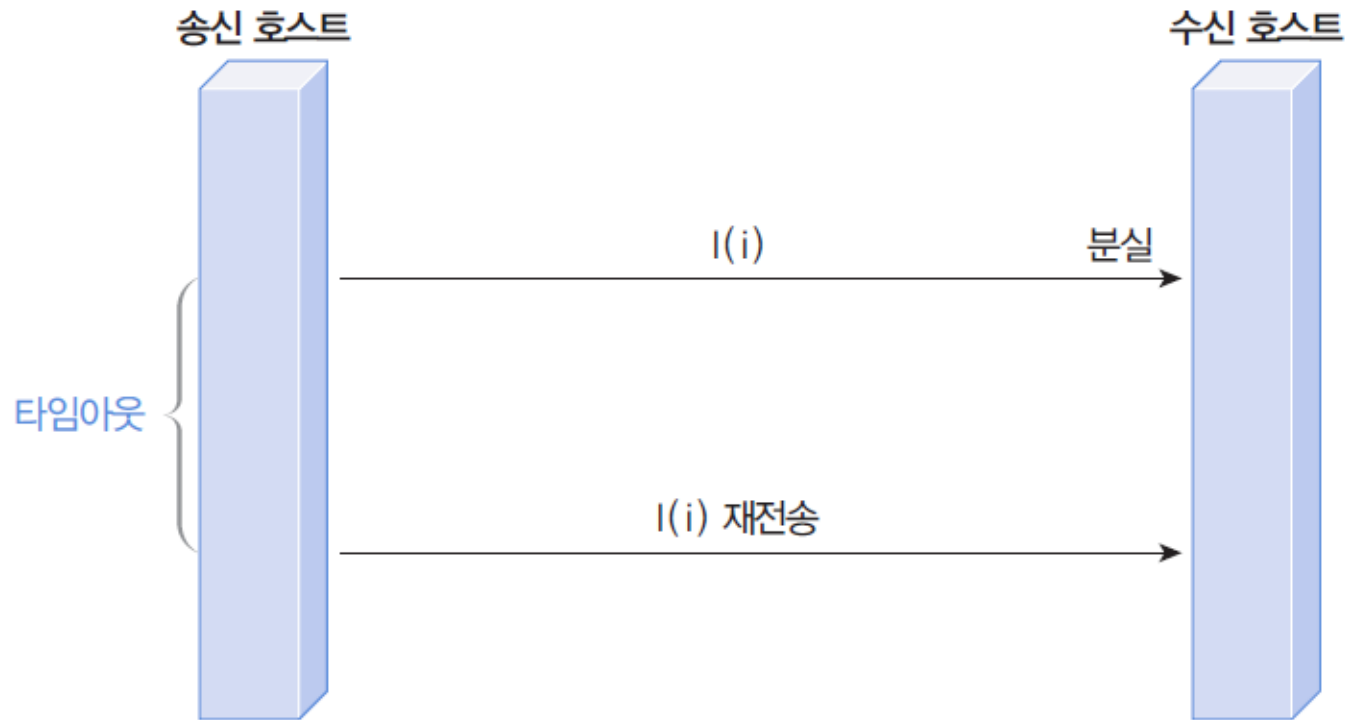


그림 6-6 정지-대기 프로토콜 3 : NAK가 있는 경우

# 01\_데이터 링크 계층 프로토콜의 기초

- 프레임 분실 오류 : 송신 호스트의 타임아웃 기능으로 오류 복구



(b) 프레임 분실 오류

그림 6-6 정지-대기 프로토콜 3 : NAK가 있는 경우

## 02\_슬라이딩 윈도우 프로토콜

- 양방향 통신을 지원
- 오류 제어와 흐름 제어 기능을 모두 지원

### ■ 기본 절차

- 송신 호스트는 정보 프레임(전송 데이터, 순서 번호, 오류 검출 코드)을 순서 번호에 따라 순차적으로 전송함
- 정보 프레임을 수신한 수신 호스트가 응답하는 순서 번호는 정상적으로 수신한 번호가 아닌 다음에 수신하기를 기대하는 번호를 회신하는 것이 일반적임
- 송신 호스트가 관리하는 송신 윈도우는 전송은 되었지만 긍정 응답이 회신되지 않은 프레임을 보관함
- 수신 호스트가 관리하는 수신 윈도우는 프로토콜의 방식에 따라 크기가 다름
  - 선택적 재전송 Selective Retransmission 방식에서는 송신 윈도우 크기와 같음
  - 고백 N Go-Back-N 방식에서는 크기가 1 임



## 02\_슬라이딩 윈도우 프로토콜

### ❖ 흐름 제어

#### ■ 순서 번호

- 프레임 별로 부여되는 일련 번호
- 0 부터 임의의 최댓값까지 순환 방식으로 사용
- 일반적으로 순서 번호의 최댓값이 송신 윈도우 크기보다 커야 함
- 프레임에서 순서 번호의 공간 크기 =  $n$  비트 : 순서 번호의 범위는  $0 \sim 2^n - 1$

#### ■ 윈도우 크기

- 수신 호스트로부터 긍정 응답 프레임을 받지 않고 전송할 수 있는 정보 프레임의 최대 개수

## 02\_슬라이딩 윈도우 윈도우 프로토콜

- 슬라이딩 윈도우 프로토콜의 동작 과정 1(송신 윈도우 크기=3)



그림 6-7 슬라이딩 윈도우 프로토콜의 동작 과정 1(송신 윈도우 크기=3)

## 02\_슬라이딩 윈도우 프로토콜

- 슬라이딩 윈도우 프로토콜의 동작 과정 2(송신 윈도우 크기=3)

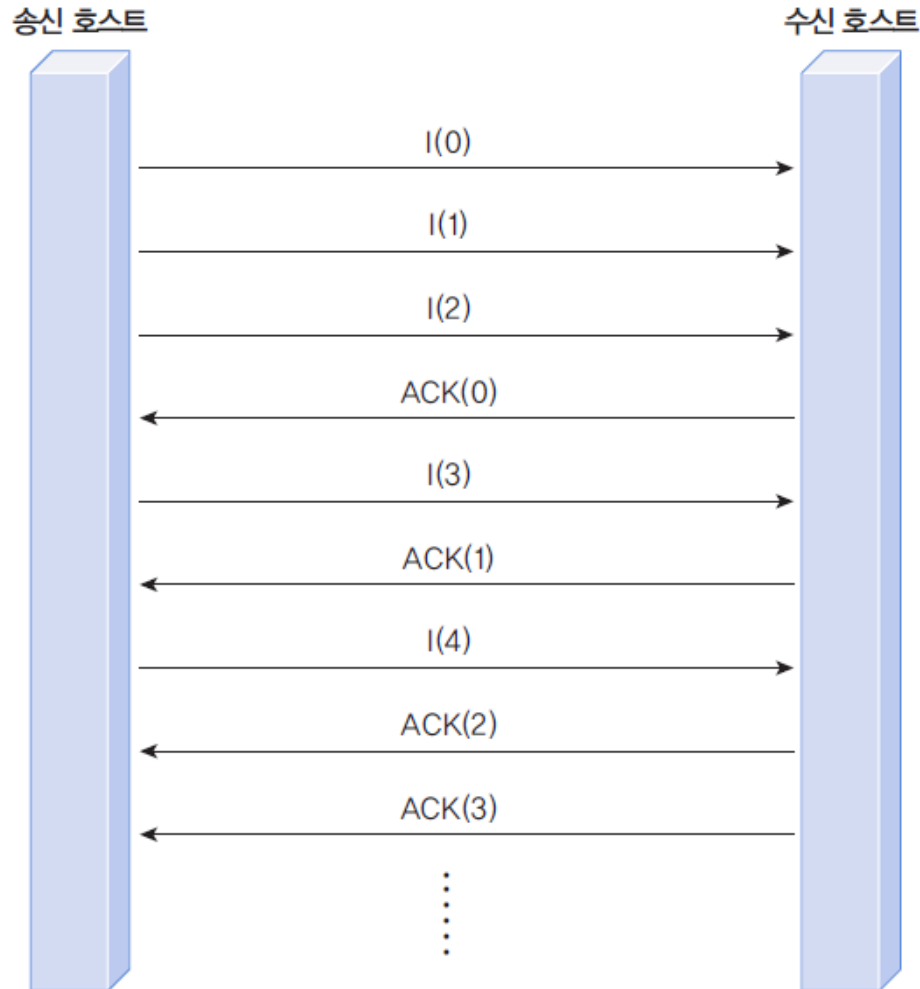


그림 6-8 슬라이딩 윈도우 프로토콜의 동작 과정 2(송신 윈도우 크기=3)

## 02\_슬라이딩 윈도우 프로토콜

### ❖ 연속형 전송

- 정지-대기 프로토콜은 송신 윈도우 크기가 1인 경우
- 연속형<sup>Pipelining</sup> 정지: ACK 프레임을 받지 않고 여러 프레임을 연속 전송
- 장점 : 오류 가능성이 적은 환경에서 효율적
- 오류 해결 방법
  - 선택적 재전송 : 오류가 발생한 프레임만 재전송
  - 고백 N : 오류가 발생한 프레임 이후의 모든 프레임을 재전송

## 02\_슬라이딩 윈도우 프로토콜

### ■ 고백 N<sup>Go-Back-N</sup> 방식

- 오류가 발생한 12번 프레임을 포함해 이후에 전송된 모든 정보 프레임을 재전송

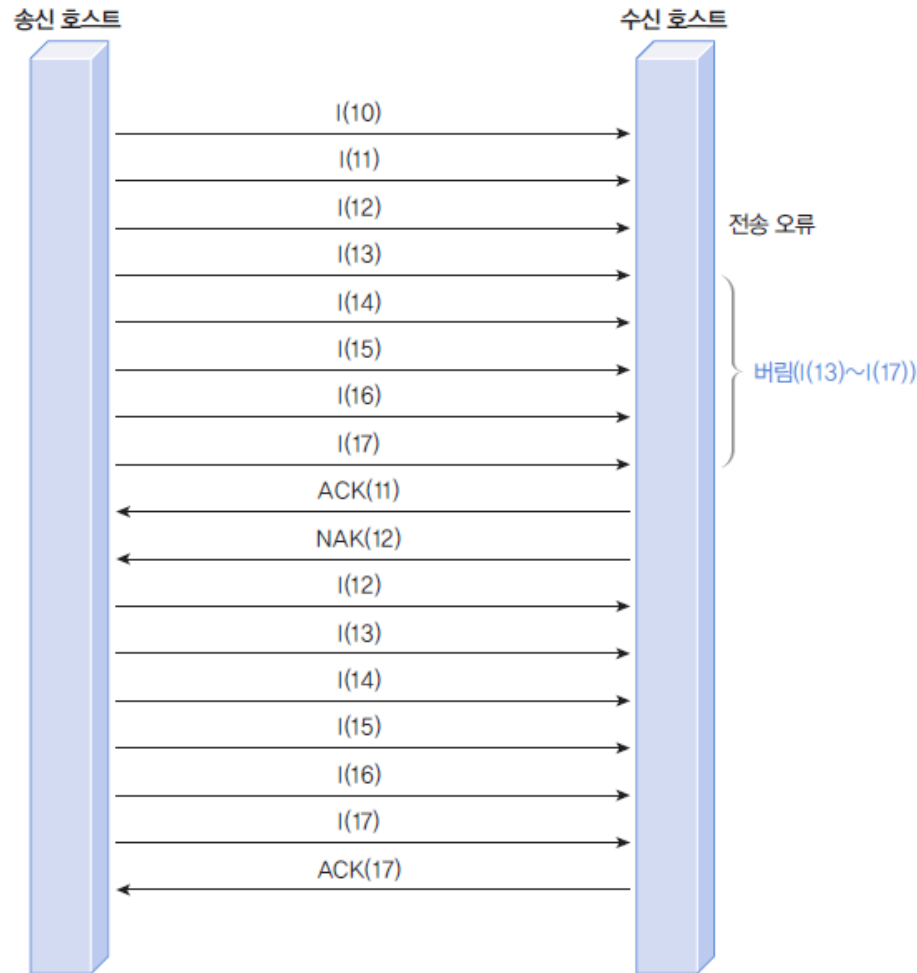


그림 6-9 고백 N<sup>Go-Back-N</sup>

## 02\_슬라이딩 윈도우 프로토콜

- 선택적 재전송 Selective Retransmission 방식 : 오류가 발생한 프레임만 선택적으로 복구하는 방식
- 부정 응답 프레임을 사용해 오류가 발생한 정보 프레임을 처리하는 경우

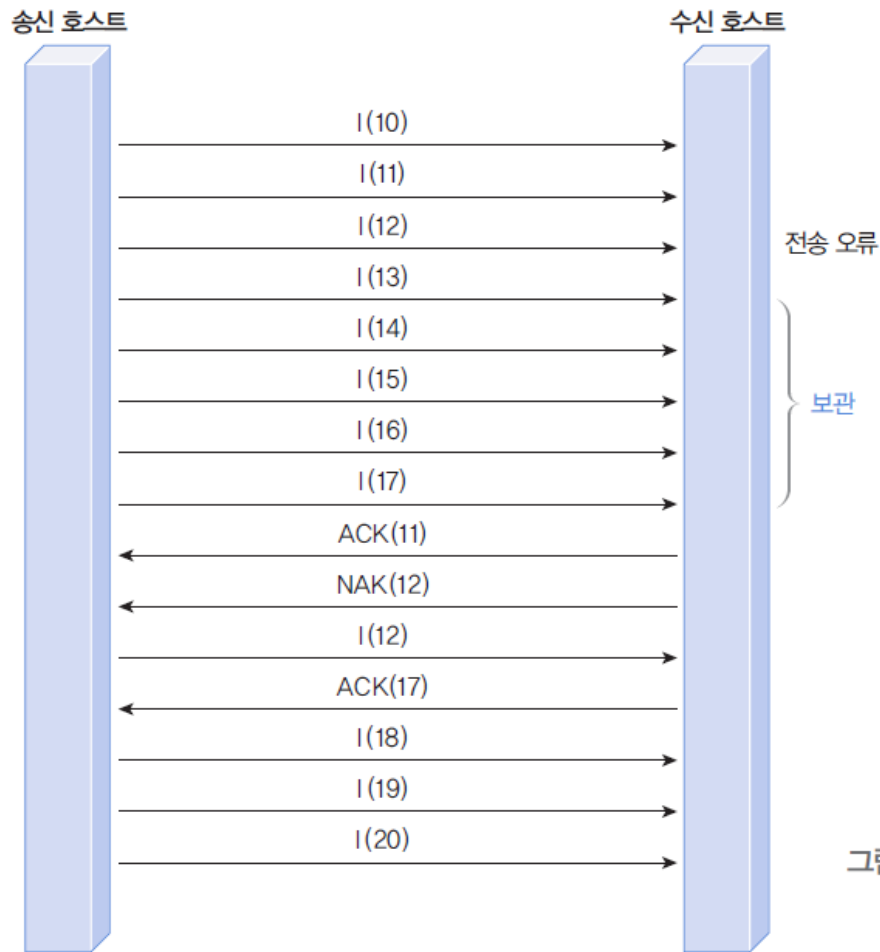


그림 6-10 선택적 재전송

## 03\_HDLC 프로토콜 High-level Data-Link Control

### ❖ 프레임 구조

#### ■ HDLC 프레임 구조

- 상단의 숫자는 비트 수.
- 프레임의 좌우에 위치한 01111110 플래그는 프레임의 시작과 끝을 구분
- Address(주소) : 일대다 환경에서 특정 호스트를 구분, 지칭하는 목적으로 사용
- Control(제어) : 프레임의 종류를 구분
- Data(데이터) : 가변 크기의 전송 데이터가 포함
- Checksum(체크섬) : CRC-CCITT를 생성 다항식으로 하는 오류 검출 용도로 사용

# 03\_HDLC 프로토콜

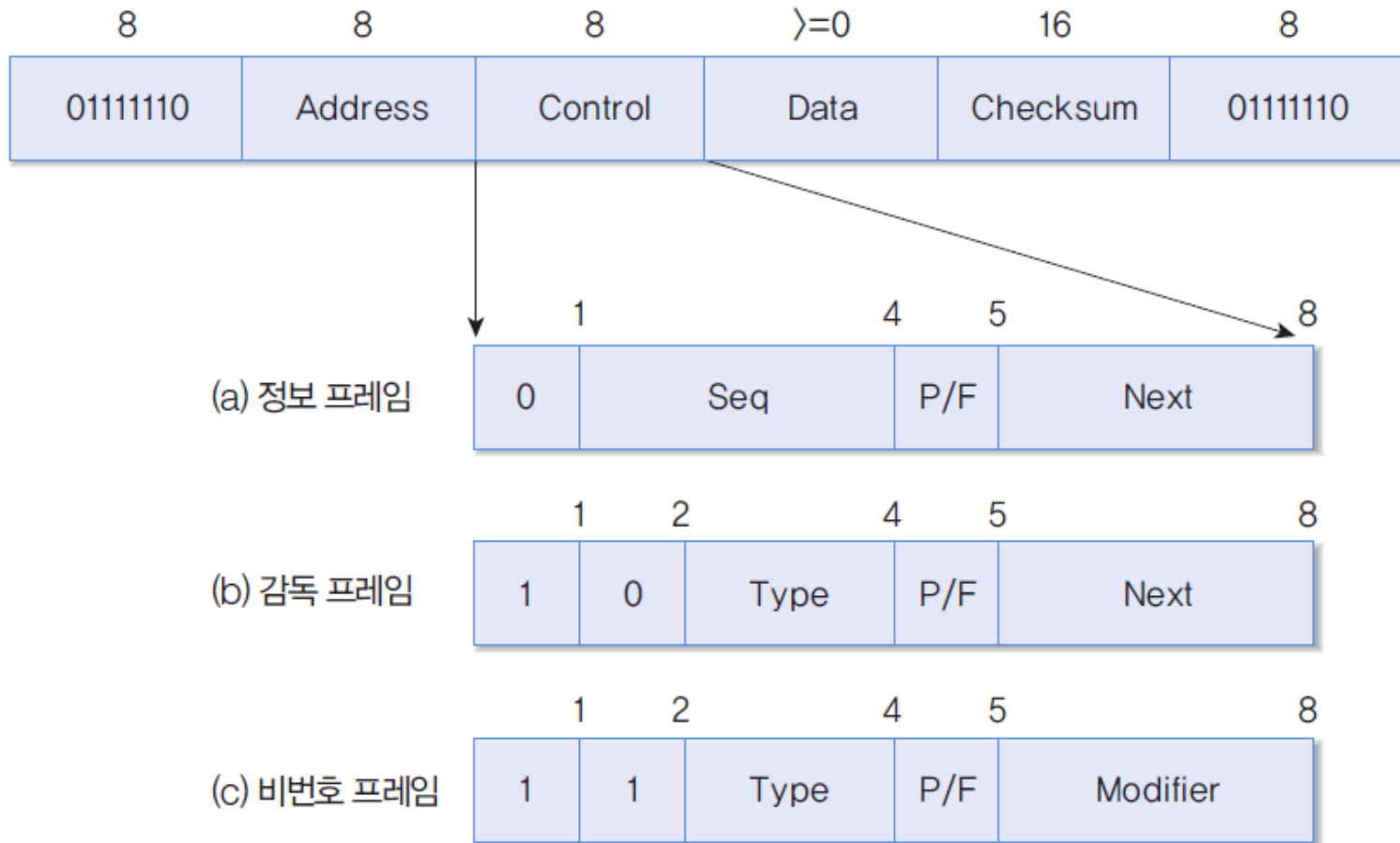


그림 6-12 HDLC 프레임



## 참조자료

# 02\_이더넷

## ❖ 이더넷과 신호 감지 기능

- 신호 감지 Carrier Sense 프로토콜
  - 전송 매체의 신호를 감지해 프레임의 전송 여부를 결정
- 1-persistent CSMA
  - 프레임을 전송하기 전에 채널 사용 여부를 확인
  - 채널이 사용 중이면 유휴 상태가 될 때까지 대기
  - 채널이 유휴 상태가 되면 확률 1의 조건으로 프레임을 전송
- Non-persistent CSMA
  - 프레임을 전송하기 전에 채널 사용 여부를 확인
  - 채널이 사용 중이면 더 이상 유휴 상태를 확인하지 않음
  - 대신 임의의 시간 동안 대기 후 다시 채널 감지를 시작
  - 1-persistent 방식보다 충돌 확률을 줄일 수 있음
- p-persistent CSMA
  - 슬롯 채널 방식에서 많이 사용
  - 채널이 유휴 상태이면  $p$ 의 확률로 프레임을 전송, 채널이 사용 중이면 다음 슬롯을 기다린 후 앞의 과정을 반복

# 03\_토큰 버스

## ❖ LLC 프레임 캡슐화

- LLC 프레임의 좌우에 토큰 버스 프레임의 헤더와 트레일러 정보가 채워지면 물리 계층이 수신 호스트로 전송.
- 수신 호스트의 MAC 계층은 토큰 버스 프레임의 헤더와 트레일러 정보를 떼어내고, 상위 계층인 LLC 프로토콜에는 LLC 프레임 정보만 올림

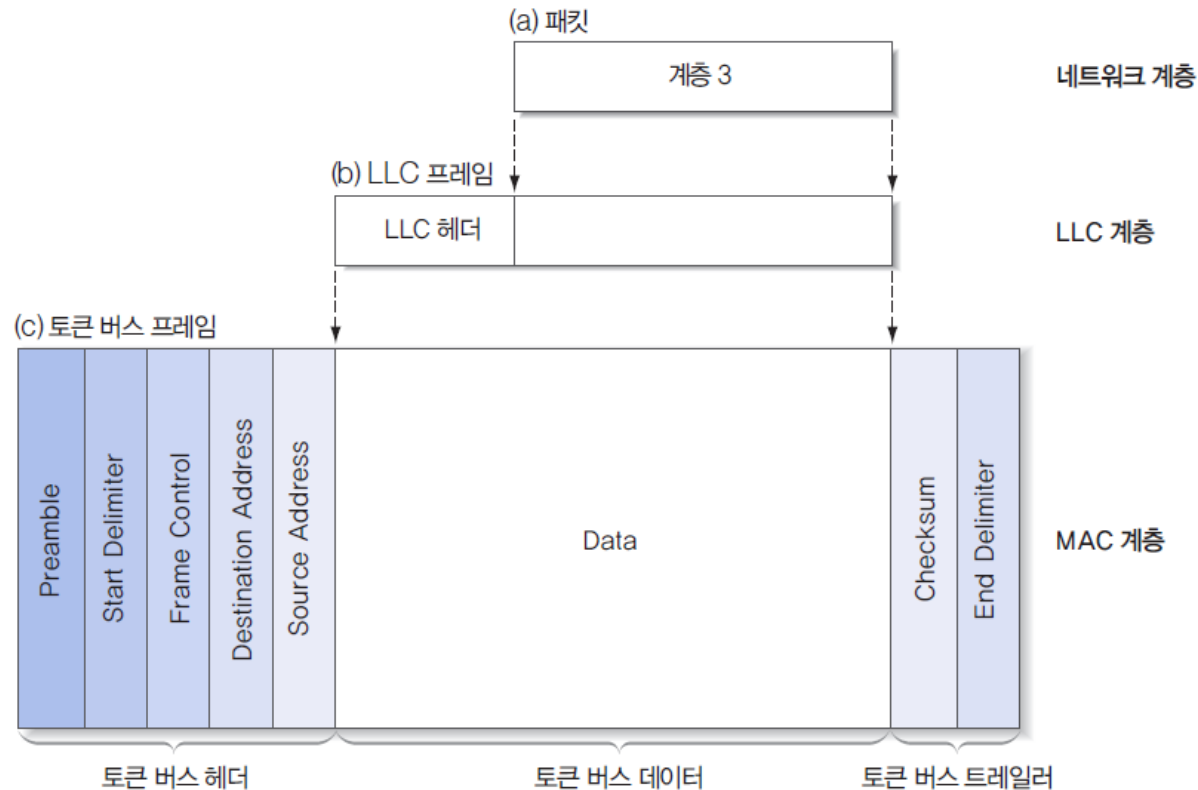


그림 5-12 토큰 버스 프레임의 Data 필드

# 04\_토큰 링

## ❖ 프레임 필드의 의미

- Start Delimiter/End Delimiter
  - 프레임의 시작과 끝을 구분
  - End Delimiter 필드
    - 1 비트는 데이터 프레임을 여러 개로 나누어 전송하는 경우에 사용
    - E 비트는 오류 검출용
- Access Control
  - 우선순위 비트 : 토큰의 우선순위보다 높은 프레임을 전송할 수 있음
  - 토큰 비트 : 토큰 프레임과 일반 프레임을 구분, 토큰 프레임은 값이 0
  - 모니터 비트 : 오류발생시 링 주위를 무한정 순환을 막기 위해 특정 호스트를 모니터로 지정, 모니터 호스트는 이 프레임을 링에서 제거함

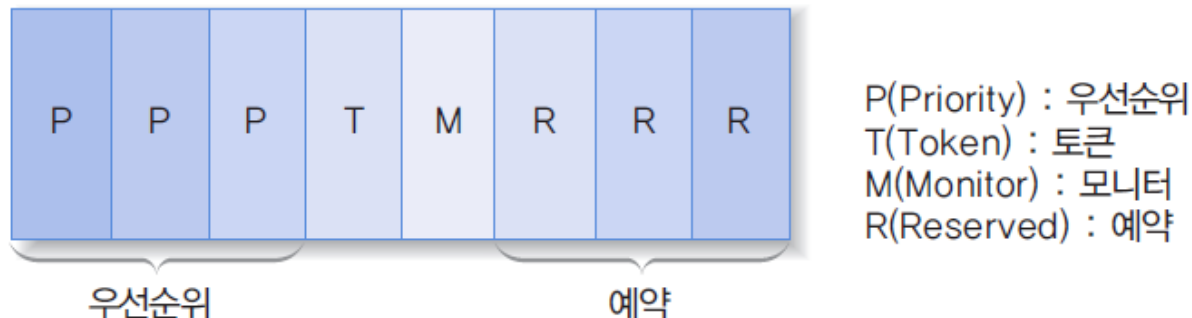


그림 5-15 Access Control 필드

## 04\_토큰 링

### ■ Frame Status

- 토큰 링 프레임의 맨 마지막에 위치, 프레임의 수신 호스트가 송신 호스트에 응답할 수 있도록 함
- 두 개의 플래그 비트 A, C 필드로 정의
  - 한 쌍의 값이 동일한 경우에만 유효한 응답으로 정의, 다르면 0으로 처리되어 무시됨
- A 비트 : 수신 호스트가 접근했다는 표시로 1 지정
- C 비트: 수신 호스트가 자신의 내부 버퍼에 보관 했다는 표시로 1 지정

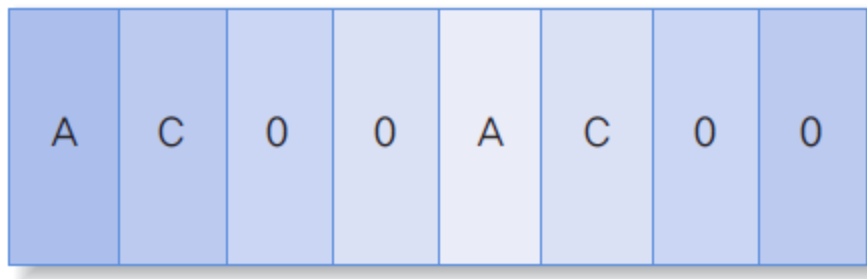


그림 5-17 Frame Status 필드

# 01\_데이터 링크 계층 프로토콜의 기초

- 데이터 링크 계층에서 두 호스트가 통신하려면 일대일(1:1)형식의 점대점 방식으로 연결해야 함
- 점대점 연결 : 주소 개념 불필요
- 멀티드롭 연결 : 주소 개념 필요
- 물리 계층에서 오류 제어 방식으로 재전송 기법을 사용

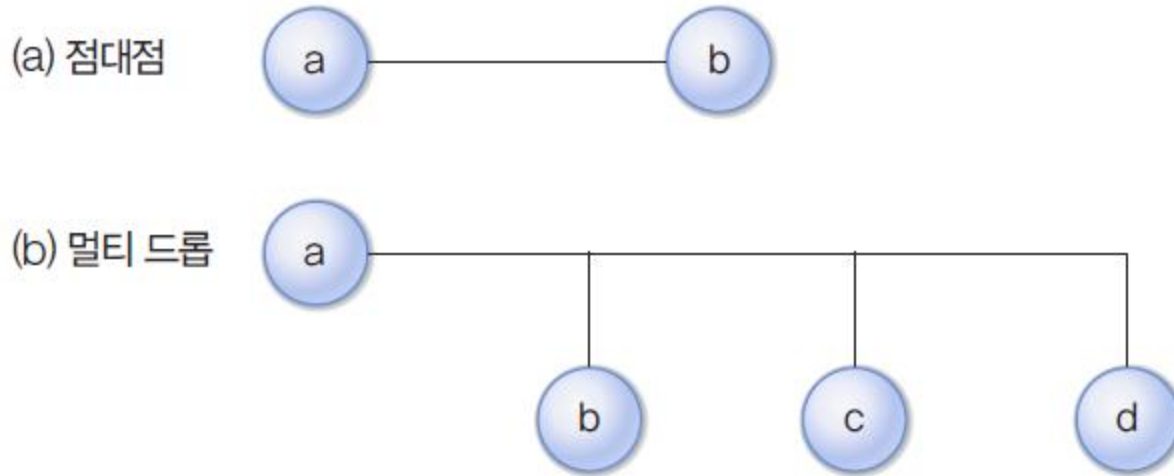
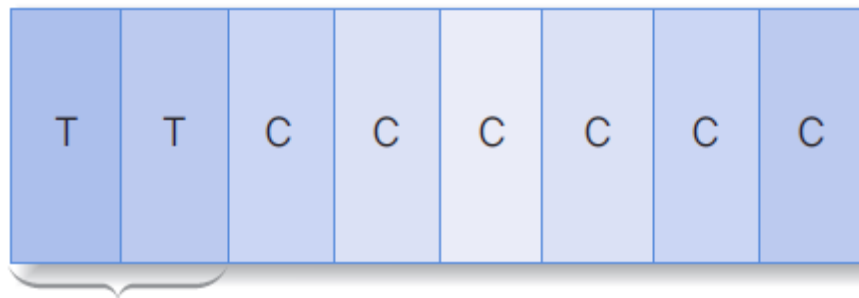


그림 6-1 연결 구성도

## 04\_토큰 링

### ■ Frame Control

- LLC 계층에서 목적지 호스트로 전송해줄 것을 요청한 LLC 프레임과 토큰 링 프로토콜에서 사용하는 제어용 프레임을 구분하는 데 사용



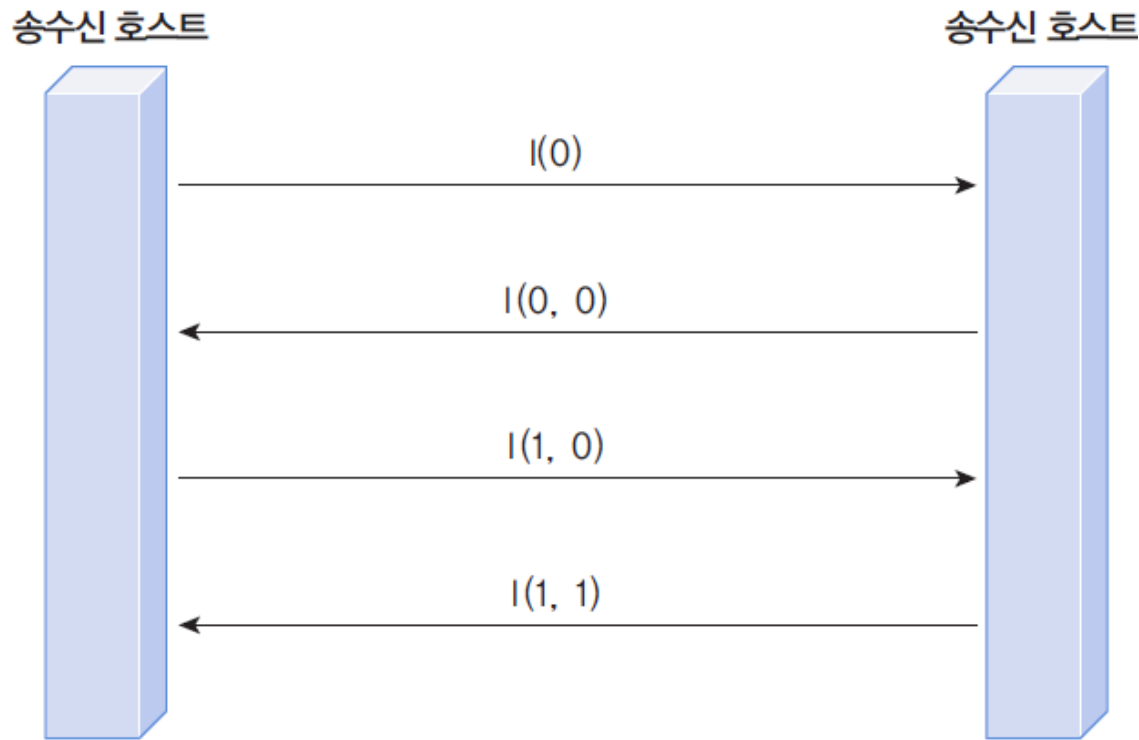
0 0: 제어용 MAC 프레임  
0 1: LLC 프레임  
1 x: 예약

그림 5-16 Frame Control 필드

- TT 비트=00: 제어 기능을 수행하는 프레임을 위해 정의, CCCCCC 비트의 코드값으로 제어 명령의 종류를 구분
- TT 비트=01: 상위 계층인 LLC 계층에서 전송을 요구한 LLC 프레임을 의미

## 02\_슬라이딩 윈도우 프로토콜

- 피기배킹을 사용하는 경우
  - 정보 프레임의 표기방식은  $I(i, j)$ 로 재정의
  - $i$ 는 자신이 전송하는 데이터의 순서 번호,  $j$ 는 제대로 수신한 프레임의 순서 번호를 의미



(b) 피기배킹을 사용하는 경우

그림 6-11 피기배킹



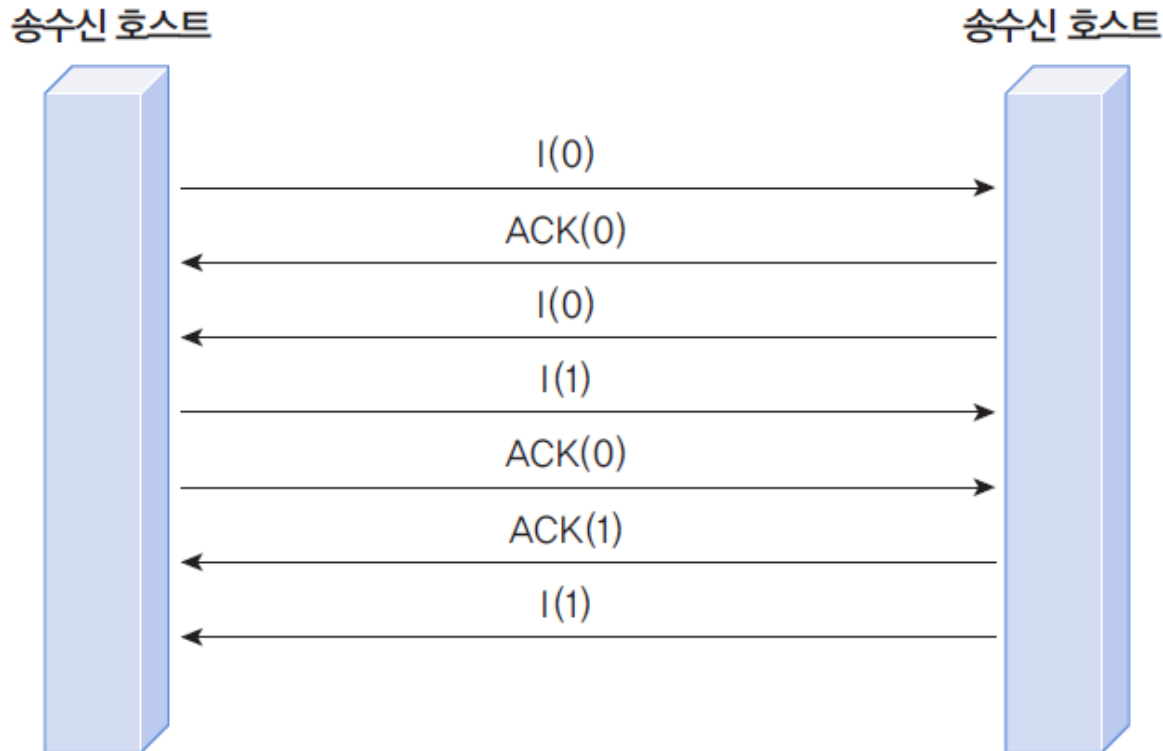
## 03\_HDLC 프로토콜

- HDLC High-level Data Link Control 프로토콜
  - 일대일 혹은 일대다로 연결된 환경에서 데이터의 송수신 기능 제공
- 호스트 종류
  - 주국 Primary Station 에서 전송되는 메시지를 명령 Command이라 정의
  - 종국 Secondary Station 의 회신을 응답 Response이라 함
  - 혼합국 Combined Station : 주국과 종국 기능을 모두 지닌 호스트

## 02\_슬라이딩 윈도우 프로토콜

### ❖ 피기배킹

- 정보 프레임을 전송하면서 응답 기능까지 함께 수행, 전송 효율 높임
  - 피기배킹을 사용하지 않는 경우
    - 개별 정보 프레임에 대해 긍정 응답 혹은 부정 응답 프레임이 순서 번호와 함께 별도 처리



(a) 피기배킹을 사용하지 않는 경우

# 03\_HDLC 프로토콜

## ❖ 프레임 종류

### ■ 정보 프레임 Information Frame

- 네트워크 계층의 데이터 전송을 위해 정의
- 3비트의 순서 번호를 이용한 슬라이딩 윈도우 프로토콜을 사용
- 순서 번호는 0~7의 순서 번호 8개를 순환하여 사용
- Seq : 정보 프레임의 송신용 순서 번호로 사용
- Next : 피기배킹을 이용한 응답 기능으로 사용
- P/F : 값이 1로 지정되었을 경우에 한하여 Poll 혹은 Final의 의미를 가짐

# 03\_HDLC 프로토콜

## ■ 감독 프레임 Supervisor Frame

- 정보 프레임에 대한 응답 기능을 수행하는 프레임
- 긍정 응답 프레임과 부정 응답 프레임으로 구분

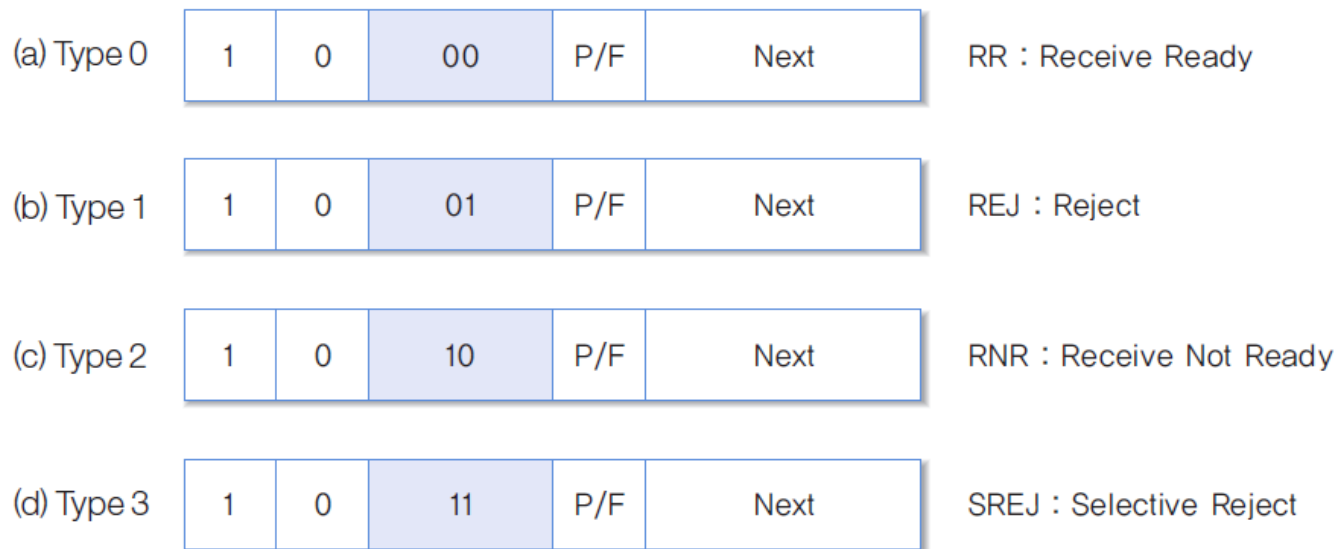


그림 6-13 감독 프레임

- Type 0 : RR로 정의된 긍정 응답 프레임
- Type 1 : REJ로 정의된 부정 응답 프레임
- Type 2 : RNR로 정의된 응답 프레임으로 흐름 제어 기능까지 제공
- Type 3 : SREJ로 정의된 프레임, 선택적 재전송 방식에서 부정 응답 기능을 지원

# 03\_HDLC 프로토콜

- 비번호 프레임 Unnumbered Frame : 순서 번호가 없는 프레임을 정의
  - 프레임 종류
    - SABM<sup>Set ABM</sup> : 비동기 균형 Asynchronous Balanced 모드의 연결 설정을 요구
    - SNRM<sup>Set NRM</sup> : 정규 응답 Normal Response 모드의 연결 설정을 요구
    - SARM<sup>Set ARM</sup> : 비동기 응답 Asynchronous Response 모드의 연결 설정을 요구
    - DISC<sup>DISConnect</sup> : 연결 설정 해제를 요구
    - RSET<sup>ReSET</sup> : 비정상적인 프로토콜의 동작에 따른 리셋 기능을 수행
    - FRMR<sup>FRaMe Reject</sup> : 비정상적인 프레임의 수신을 거부
    - UA<sup>Unnumbered ACK</sup> : 비번호 프레임에 대한 응답 기능을 수행

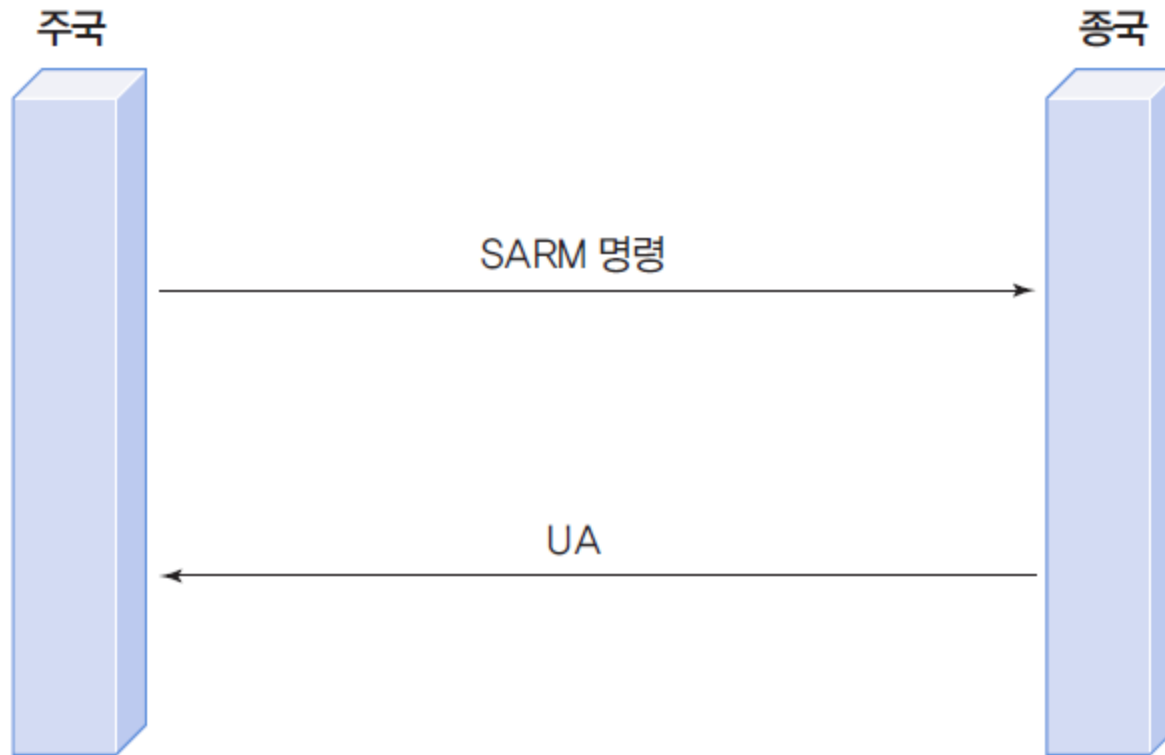
표 6-1 세 가지 연결 모드

연결 모드	설명
정규 응답 Normal Response	불균형 모드를 의미하기 때문에 호스트 하나는 주국으로 동작하고, 다른 하나는 종국으로 동작한다. SNRM을 이용하여 연결 설정을 요구하며, 종국에서 데이터를 전송하려면 반드시 주국의 허락을 받아야 한다.
비동기 균형 Asynchronous Balanced	두 개의 호스트가 동일한 능력을 갖는 혼합국으로 동작하며, 양쪽에서 명령과 응답을 모두 전송할 수 있다. 따라서 둘 중 임의의 호스트에서 SABM을 이용한 연결 설정 요구를 수행할 수 있다.
비동기 응답 Asynchronous Response	불균형 모드이나 종국이 주국의 허락 없이도 데이터를 전송할 수 있는 권한을 갖는다. 즉, 주국의 허가가 없어도(주국에 의한 P/F 비트의 Poll 지정이 없어도) 종국에서 먼저 SARM 응답을 보낼 수 있다.

# 03\_HDLC 프로토콜

## ❖ LAP 프로토콜

- LAPLink Access Protocol : 비동기 응답 모드인 ARM으로 동작하는 프로토콜
  - 주국에서 전송된 SARM 명령에 대하여 종국에서 UA 응답을 전송함으로써 완료



(a) 연결 설정 1

## 03\_HDLC 프로토콜

- 중국에서 주국으로 SARM 응답을 전송하여 연결 설정을 요구
- 중국에서 시작된 연결 설정은 3단계로 이루어지는 효과가 발생

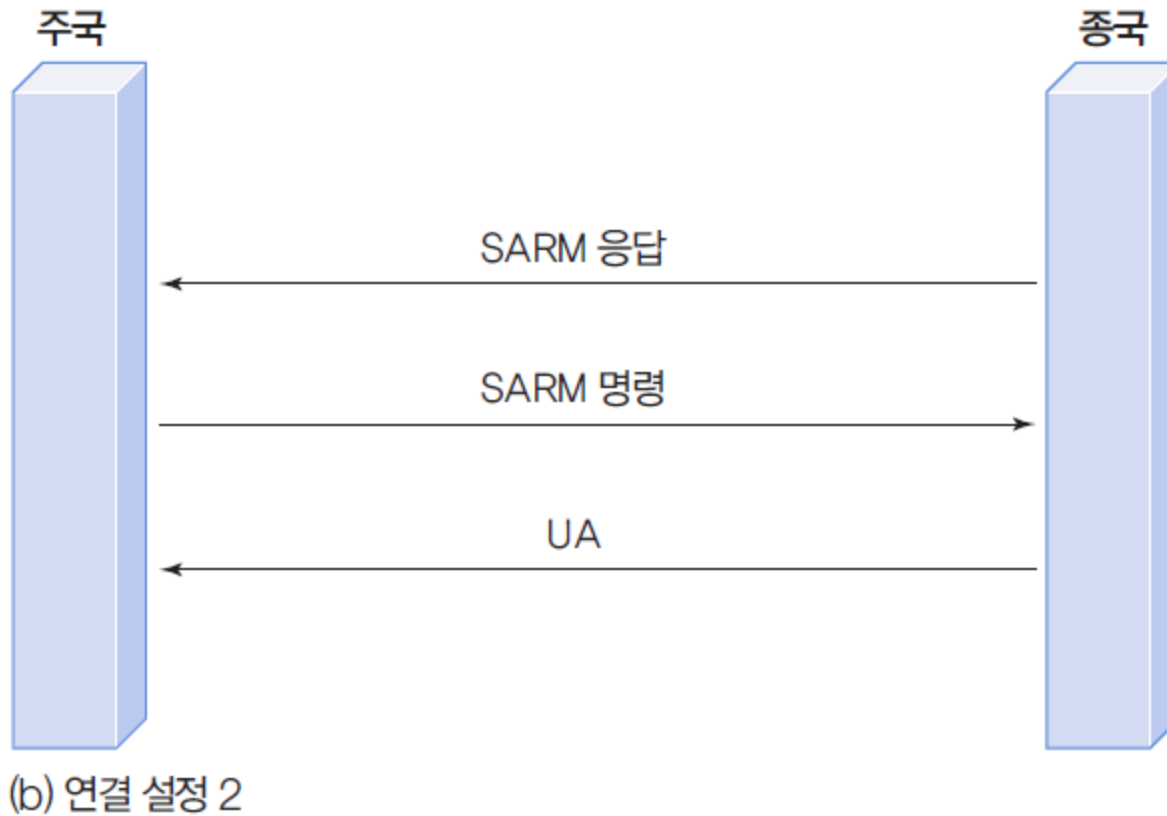


그림 6-14 LAP에서의 연결 설정

## 03\_HDLC 프로토콜

### ❖ LAPB 프로토콜

- 양쪽 호스트가 혼합국으로 동작, 누구나 먼저 명령을 전송 가능

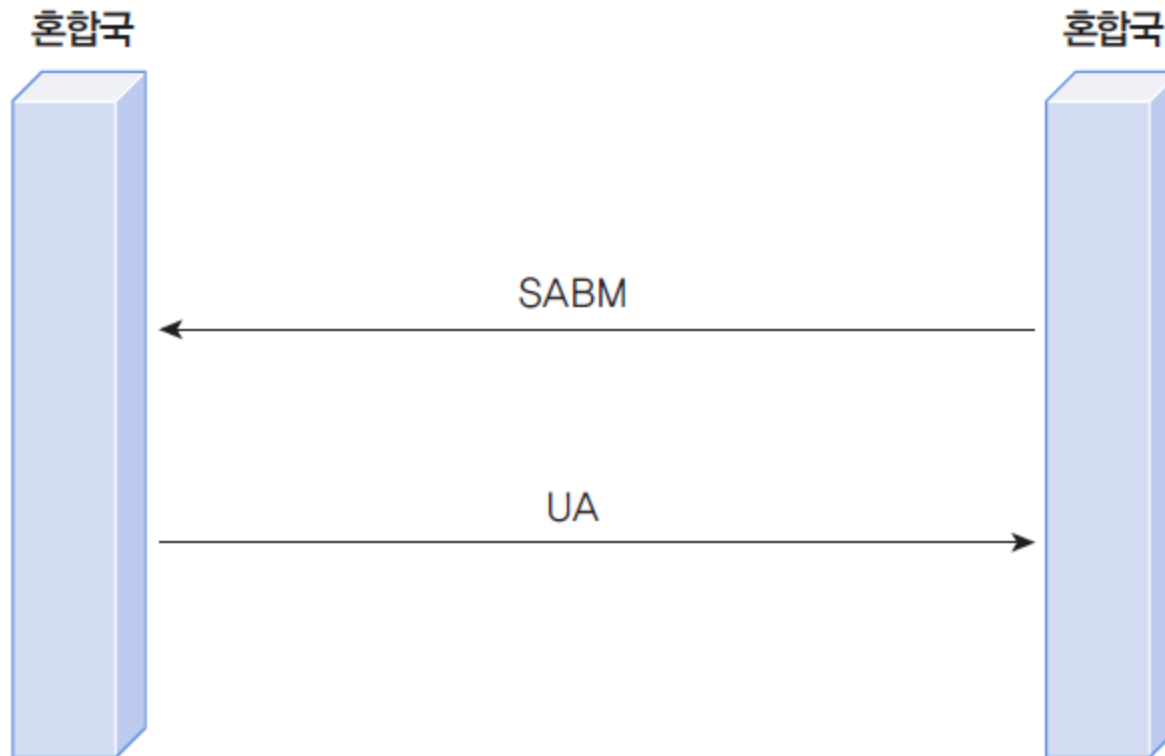


그림 6-15 LAPB에서의 연결 설정