

5 MAC 계층

가천대학교 - 2019학년도 1학기 -

❖ 전송매체

- 유선매체 : 트위스티드 페어^{Twisted Pair}, 동축케이블, 광케이블 등
- 무선매체 : 적외선, 레이저, 마이크로파, 라디오 등

❖ 전화망

- 아날로그 방식을 이용하는 음성 데이터 전송망
- 통신 시스템 중 가장 먼저 개발됨. 현재도 폭넓게 사용중
- 디지털 방식의 컴퓨터가 전화망을 통해 통신하려면 전화망과 컴퓨터 단말기 사이에 모뎀이 필요

❖ 디지털 통신방식

- 무선 휴대 통신 기기, 인터넷 보급으로 디지털 통신 환경 보편화
- 아날로그 방식보다 오류률이 낮고 전송률은 높음
- 음성, 영상, 컴퓨터 데이터 등 멀티미디어 데이터 처리에 효과적

- ❖ LAN 환경에서의 네트워크 표준 모델 2계층(Data link layer)
 - LLC가 3계층에 가깝고 MAC은 1계층에 가까움 LLC와 MAC 계층으로 역할 구분
 - WAN과 LAN 환경에서 계층 2의 역할 차이 이해 필요
 - LAN 네트워크 연결 구조가 바뀔에 따라 MAC 계층도 다양한 구조 필요

Contents

❖ 학습목표

- LAN 환경에서 MAC 계층과 LLC 계층의 차이와 역할을 이해한다.
- 이더넷의 동작 원리와 프레임 구조를 이해한다.
- 토큰 버스에서 토큰의 역할과 프레임 구조를 이해한다.
- 토큰 링의 프레임 구조와 제어 필드를 이해한다.

❖ 내용

- MAC 계층과 IEEE 802 시리즈
- 이더넷
- 토큰 버스
- 토큰 링

01_MAC 계층과 IEEE 802 시리즈

❖ WAN

- 원거리에 있는 호스트를 연결

❖ LAN

- 근거리에 분포된 호스트를 연결
- 효율적인 전송 관리를 통해 네트워크 이용 효율 극대화 필요
- 네트워크 전송 효율은 주로 전송 매체의 성능에 의해 좌우됨.
연결 방식이나 전송 방법에도 영향을 많이 받음

01_MAC 계층과 IEEE 802 시리즈

❖ MAC 계층의 이해

- LAN 환경에서 네트워크 자원의 효율적 활용을 위해 계층 2 기능을 MAC 과 LLC 계층으로 분리
 - LLC Logical Link Control 계층 : 데이터 링크 계층의 기본 기능
 - MAC Medium Access Control 계층 : 물리 계층 기능(물리적인 전송 선로의 특징, 매체 간의 연결 방식에 따른 제어 부분)

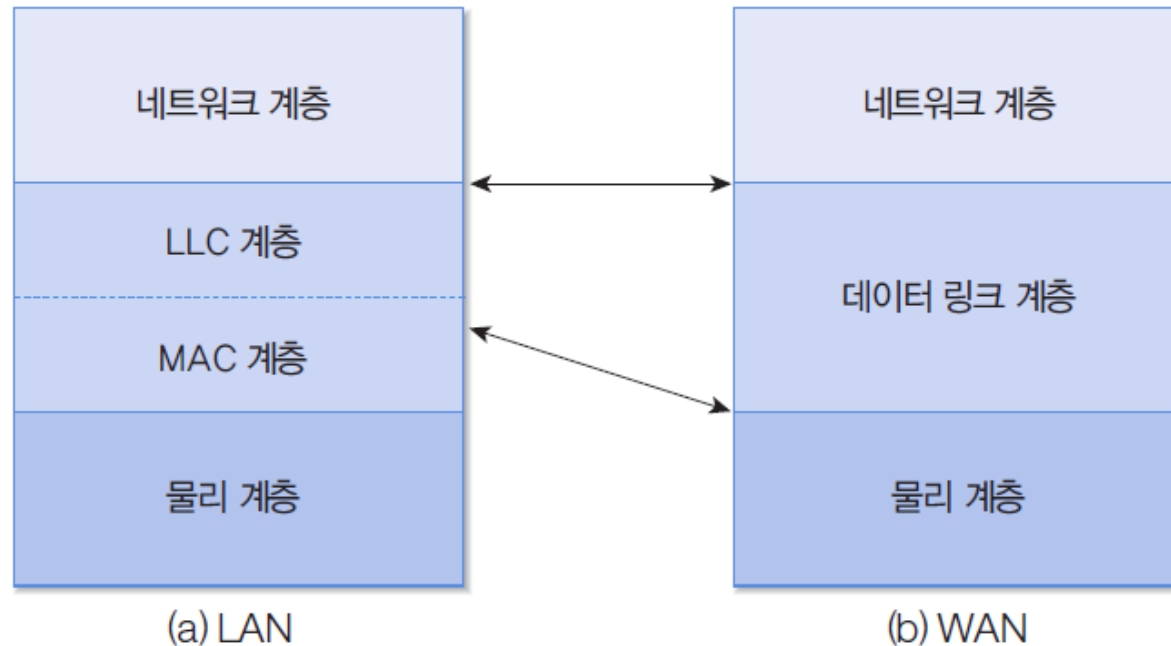


그림 5-1 MAC과 LLC 계층의 관계

01_MAC 계층과 IEEE 802 시리즈

■ MAC 계층

- 전송 매체의 물리적인 특성 반영
- LAN의 종류에 따라 특성이 구분됨
- 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

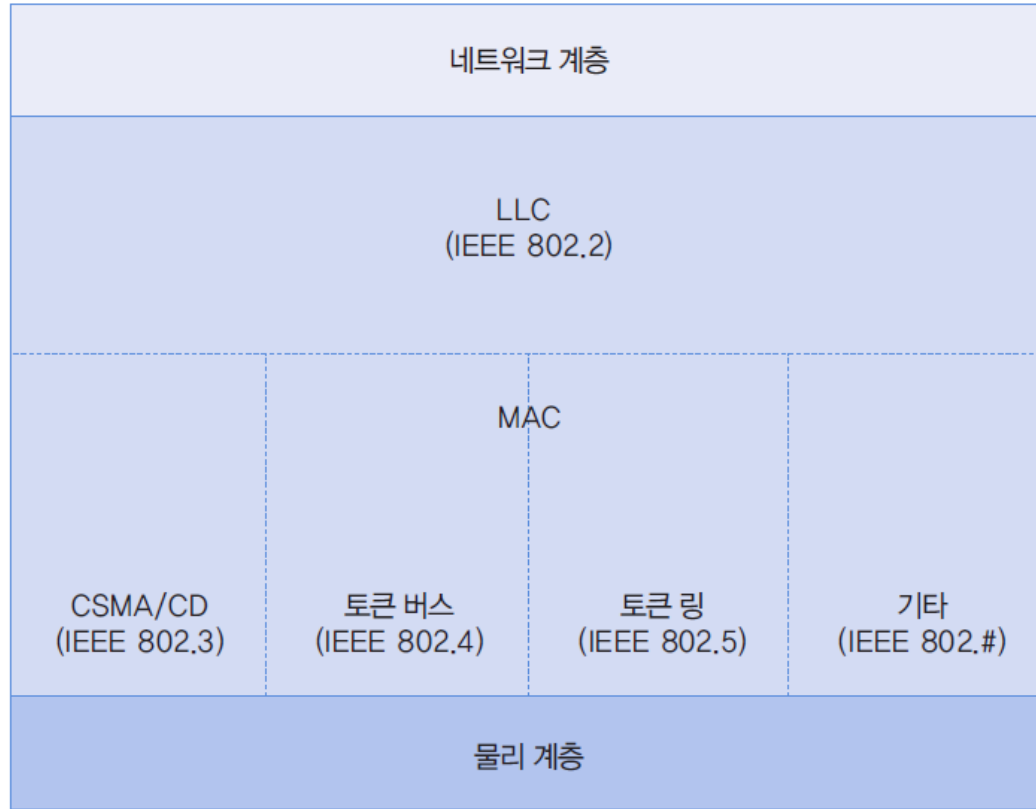
■ LLC 계층

- WAN 환경의 데이터 링크 계층 기능을 수행
- LAN의 특성에 부분적으로 영향을 받을 수 있음
 - 이더넷의 LLC : 프레임 전송 과정에서 슬라이딩 윈도우 프로토콜 사용
 - 토큰링, 토큰버스 : 송신 호스트가 수신 호스트의 도움 없이 보낸 프레임의 전송 오류를 검출하고 재전송하는 기능 수행 가능. 슬라이딩 윈도우 프로토콜 불필요

01_MAC 계층과 IEEE 802 시리즈

❖ IEEE에서 데이터 링크 계층과 관련된 다양한 LAN 표준안 발표

- IEEE 802.1 : 표준안 전체를 소개
- IEEE 802.2 : LLC 계층을 소개
- IEEE 802.3 부터 : 다양한 환경의 MAC 계층을 소개



LAN 종류에 큰 영향을 받지 않으므로 공통 프로토콜인 IEEE 802.2 사용

물리 계층의 종류에 따라 다양하게 사용될 수 있으므로 각각의 표준안 정의

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
 - 충돌이 자주 발생하면 재전송이 많이 이루어지므로 네트워크 전체 성능 저하
 - 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

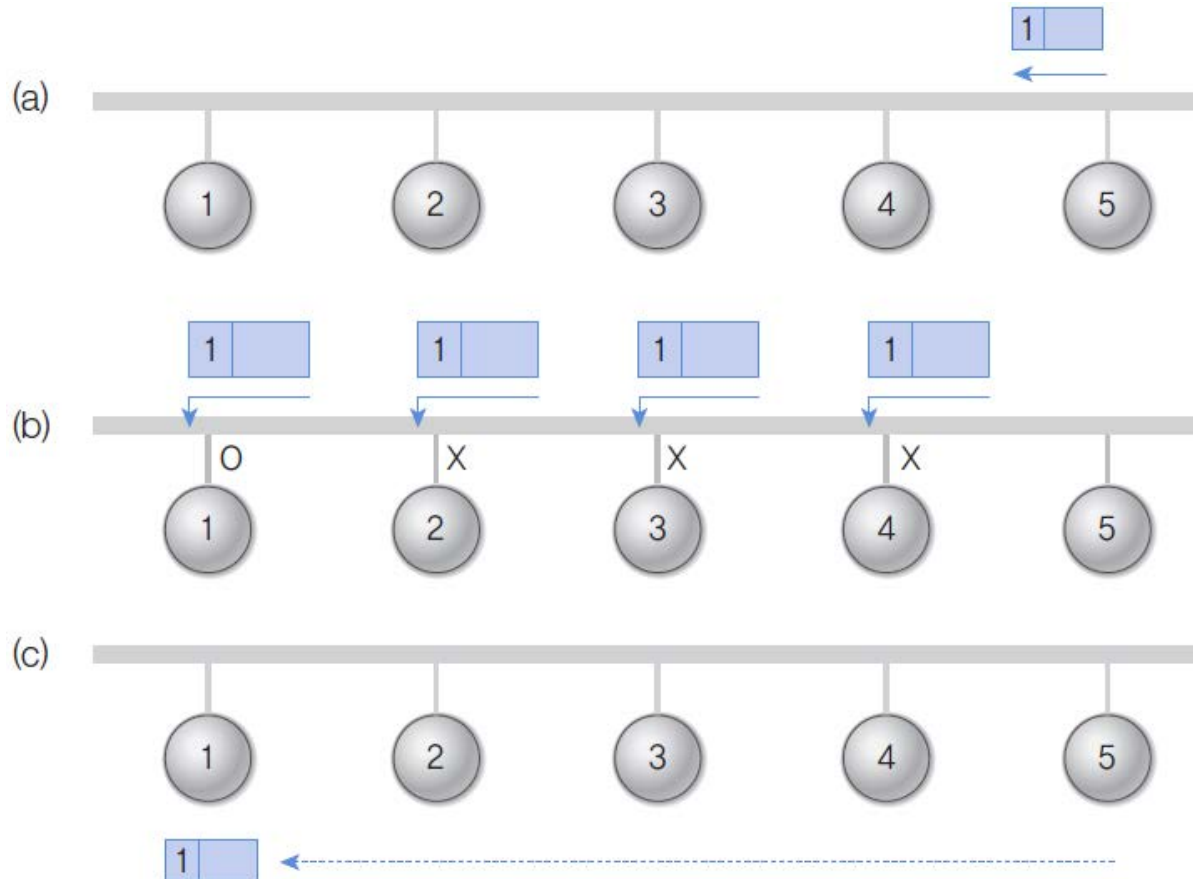


그림 5-3 공유 버스에서의 데이터 전송

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
- 호스트가 이웃하는 순서는 각 호스트의 고유 번호와 관련
예) 고유 번호가 높은 번호부터 네트워크에 연결, 토큰 전달도 먼저 이루어짐

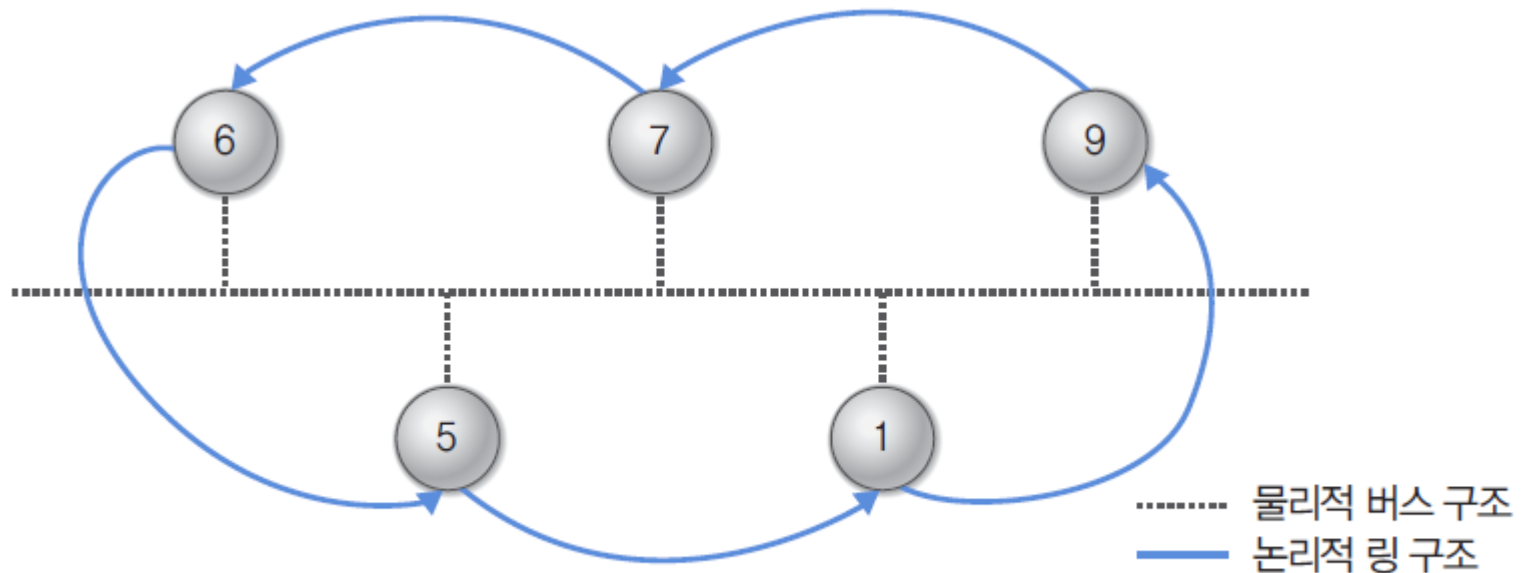


그림 5-4 토큰 버스

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

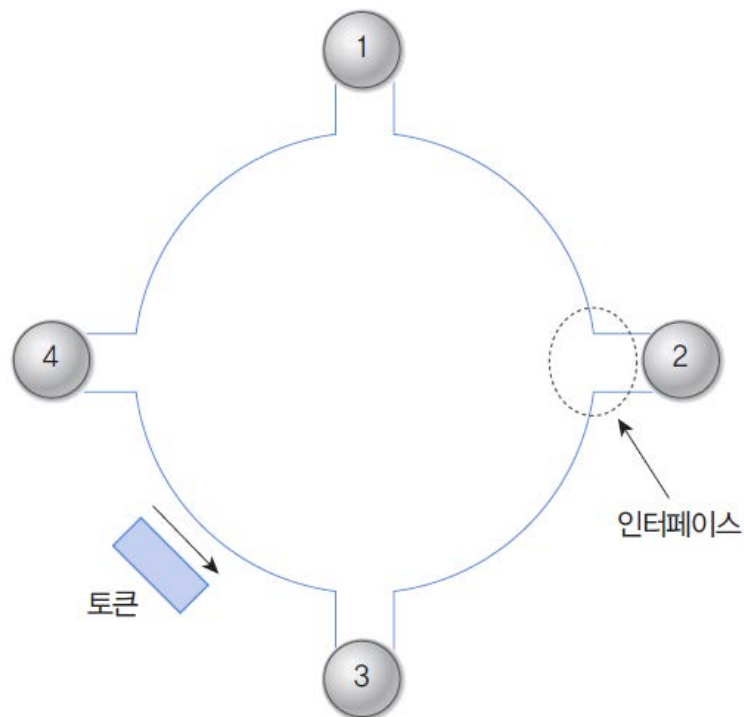


그림 5-5 토큰 링

- 송신 호스트가 전송한 프레임이 링을 한 바퀴 돈 후 송신 호스트에 되돌아옴
 - 프레임 목적지 주소가 자신의 주소와 동일한 호스트는 해당 프레임을 수신하고, 프레임 내부의 특정 위치에 수신 성공 표시
 - 송신 호스트는 자신에게 돌아온 프레임의 특정 위치 값을 확인하여 전송 성공 여부 확인
- 프레임이 올바르게 전송되면 데이터 프레임을 회수하고 토큰 프레임을 링에 반환

02_이더넷

이더넷은 충돌을 감지/처리할 수 있어야 함

❖ 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
 - 둘 이상의 호스트에서 동시에 전송 채널을 유휴 상태로 판단하면 프레임 전송 과정에서 충돌 발생 가능
 - 충돌이 발생하면 임의의 시간 동안 대기 후 다시 채널 감지 시작

❖ 이더넷과 신호 감지 기능

■ 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

- CSMA/CD
 - 둘 이상의 호스트에서 채널이 유허 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, 전송 선로 신호 감지 및 충돌 감지 기능 제공
- 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

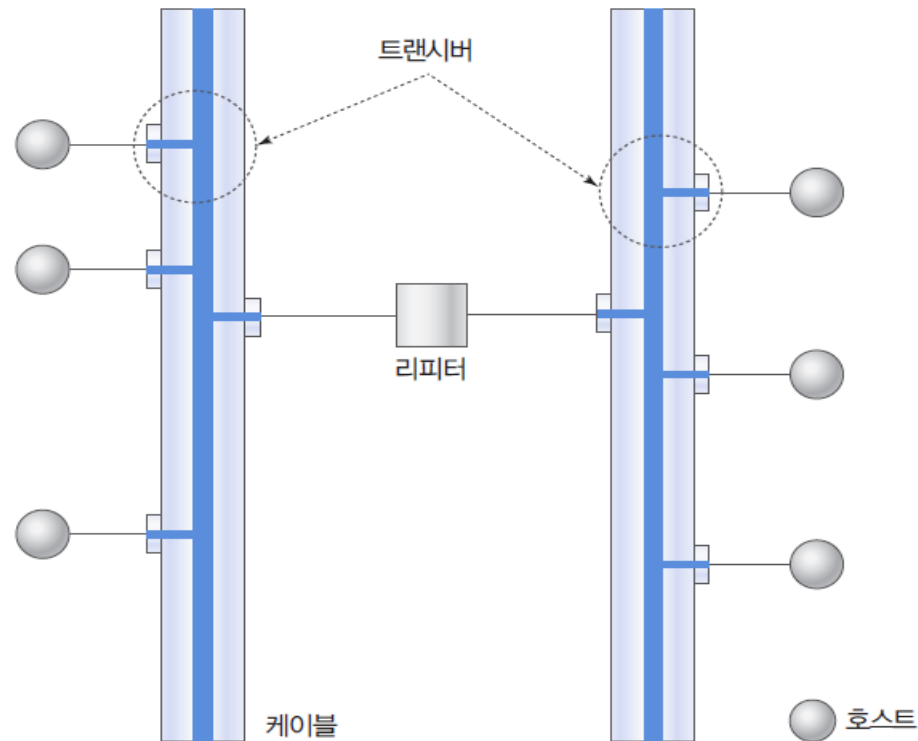


그림 5-6 이더넷의 연결

❖ 프레임 구조

- MAC 프레임^{MAC Frame} : 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^{Ethernet Frame} 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

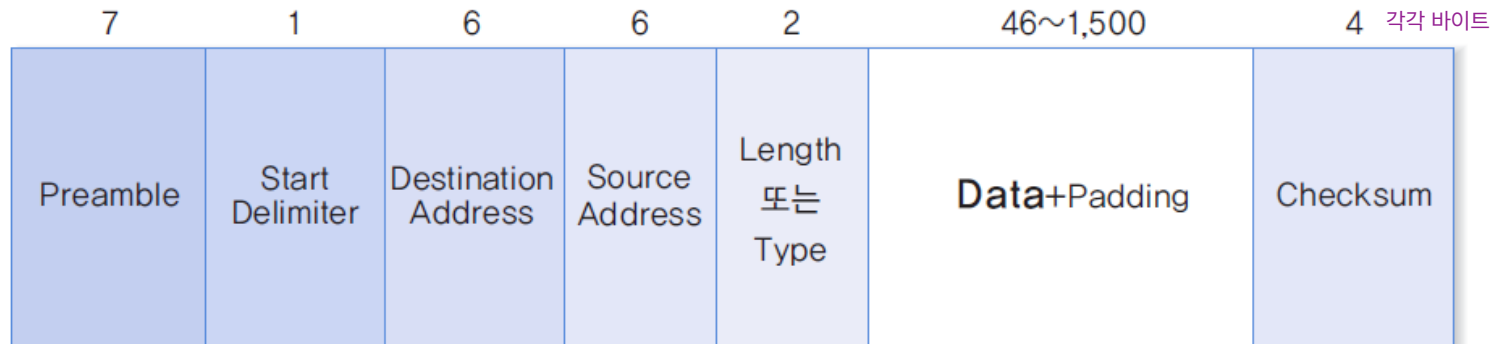
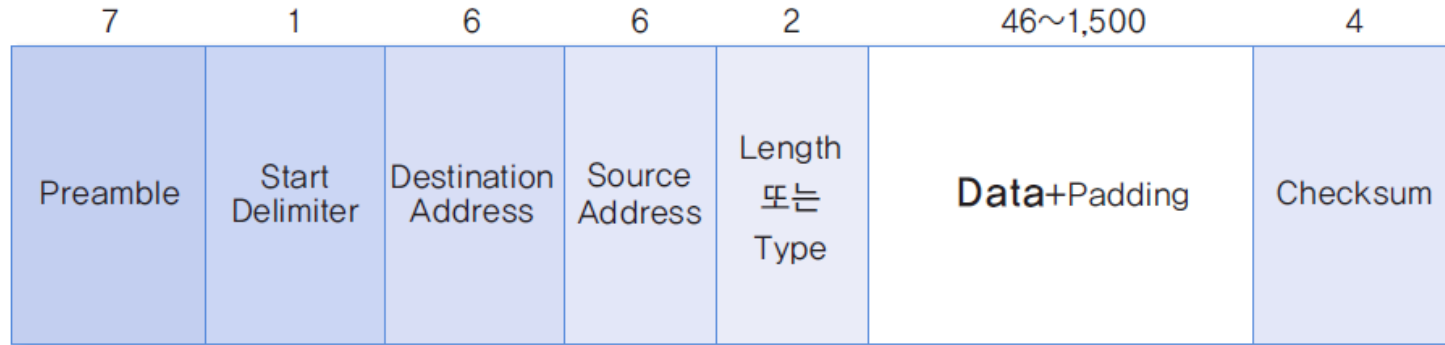


그림 5-7 이더넷 프레임의 구조

02_이더넷



- Preamble(프리앰블) : 7바이트 크기로, 수신 호스트가 송신 호스트의 클럭과 동기를 맞출 수 있도록 시간 여유를 제공하는 것이 목적. 각 바이트는 10101010 패턴
- 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. 최대값은 1,500이며, IP패킷이 이 값을 초과하면 먼저 분할 과정 필요. Length 값이 46보다 작으면 padding 필드에 해당하는 크기만큼 0으로 채움
- 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

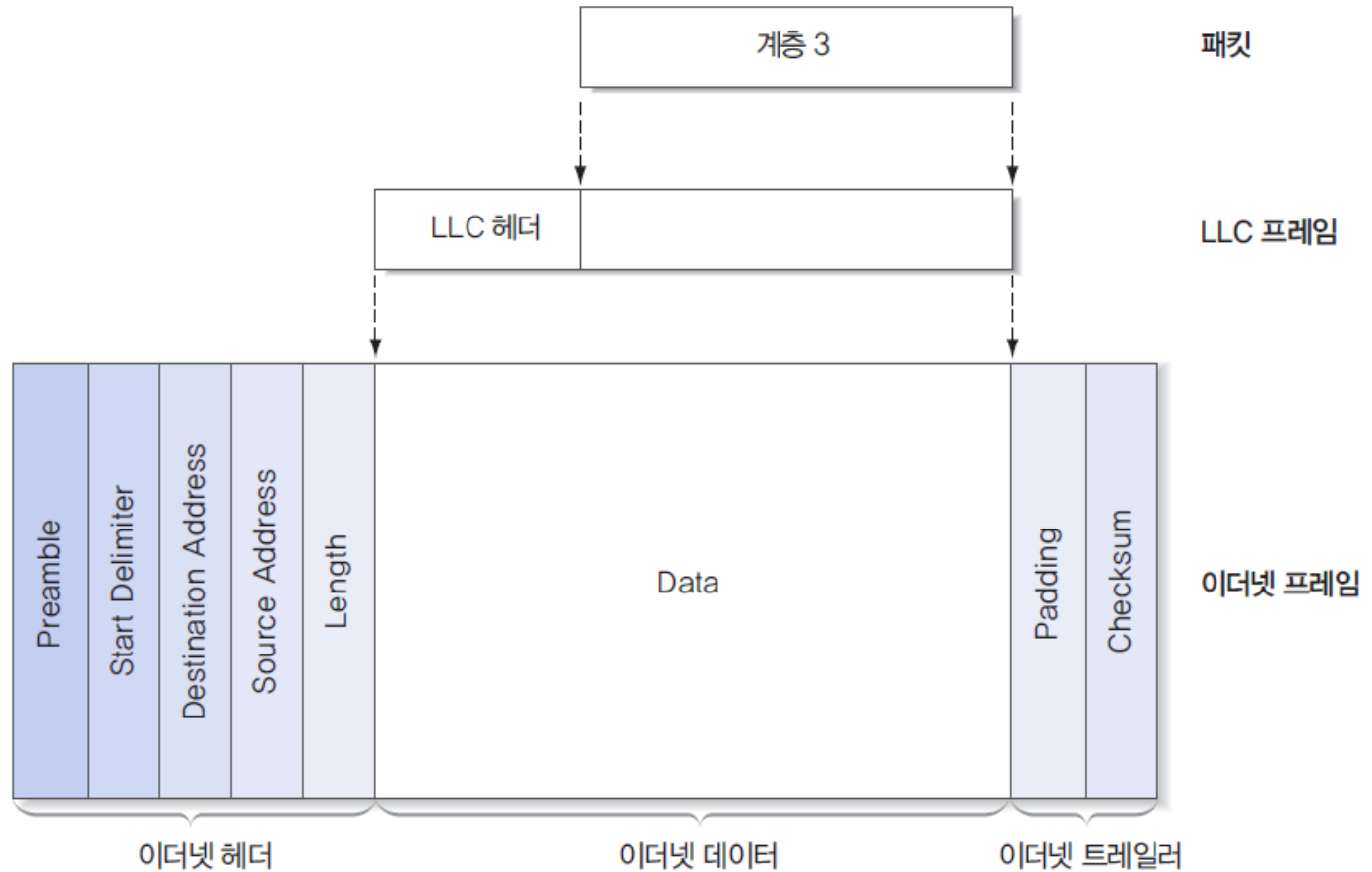


그림 5-8 이더넷 프레임의 Data 필드

02_이더넷

❖ 허브와 스위치

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

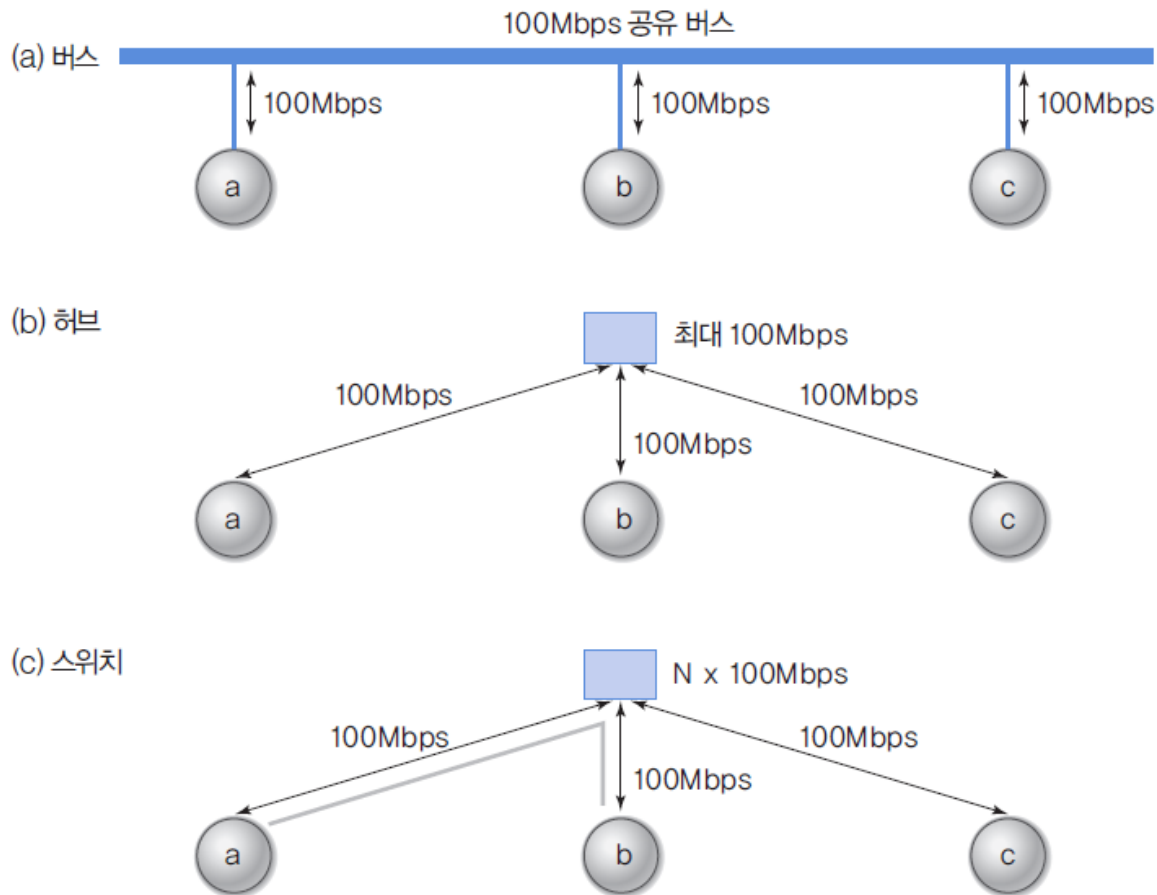


그림 5-9 허브와 스위치

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

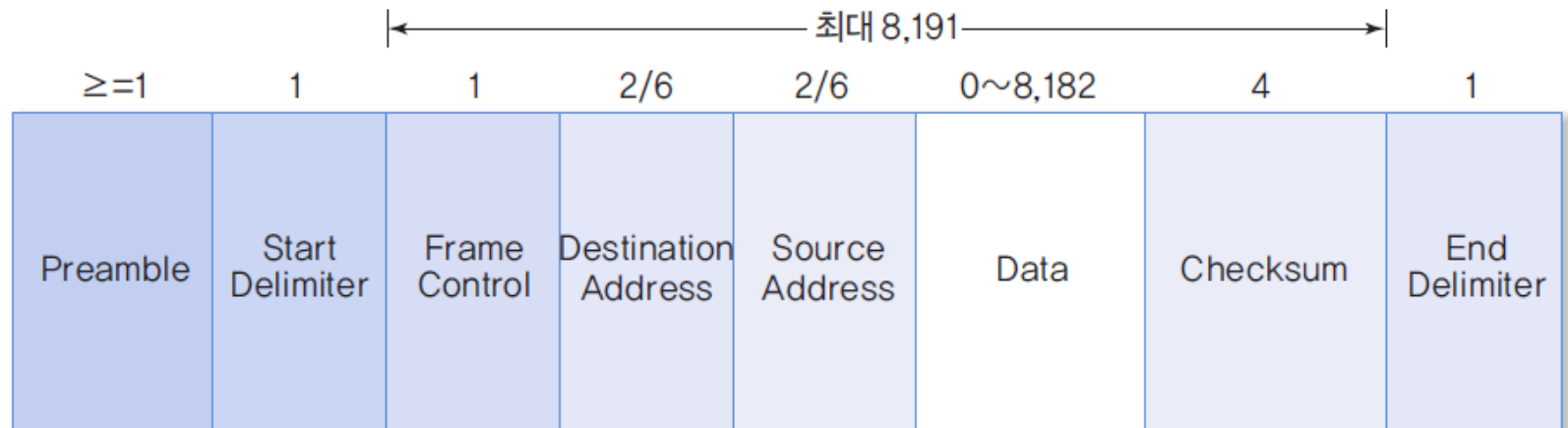
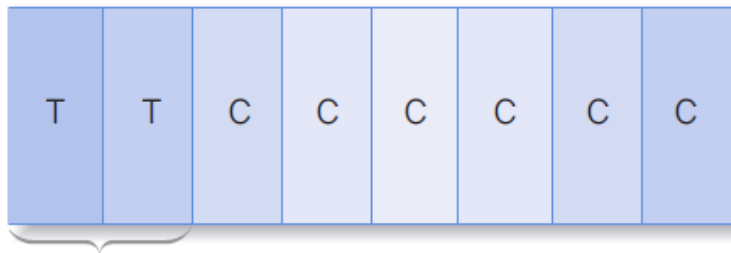


그림 5-10 토큰 버스 프레임의 구조

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	00001000 토큰 프레임
0	1 : LLC 프레임	상위 계층인 LLC계층에서 내려온 전송 데이터
1	0 : 네트워크 관리용 데이터 프레임	
1	1 : 예약	

그림 5-11 토큰 버스 프레임의 Frame Control 필드

❖ 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

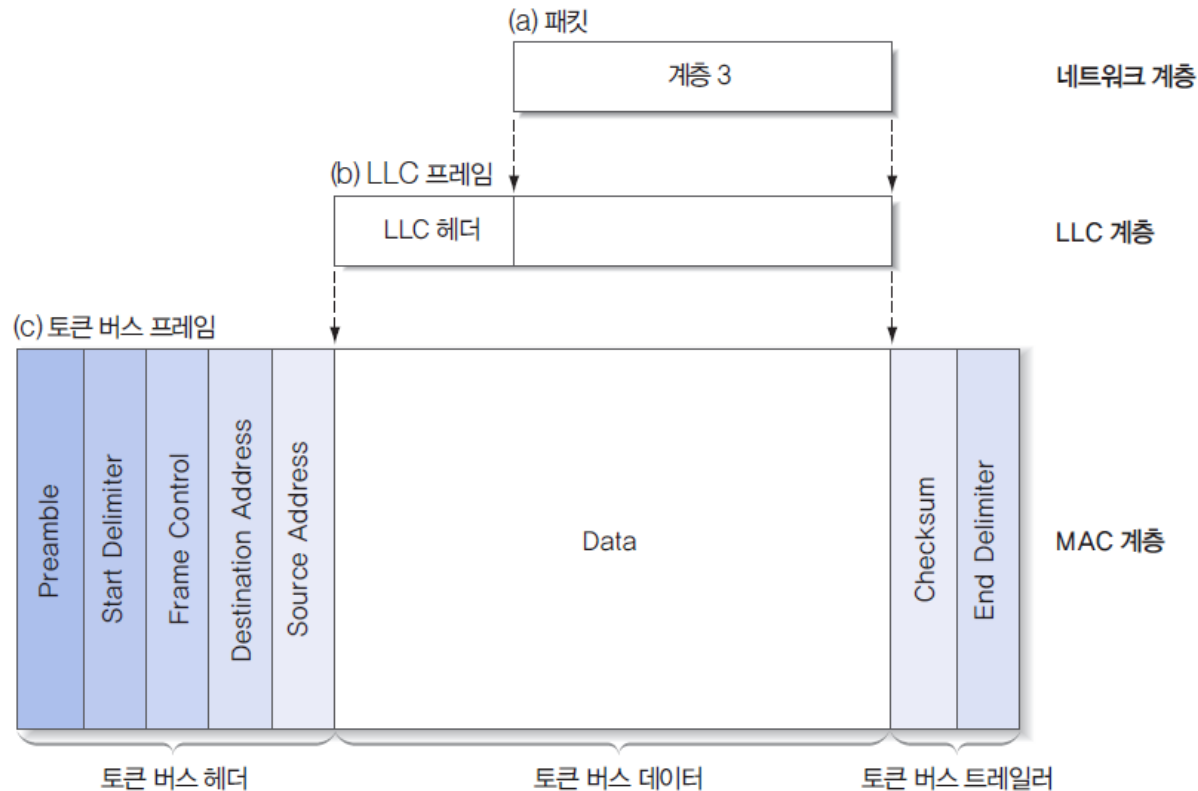
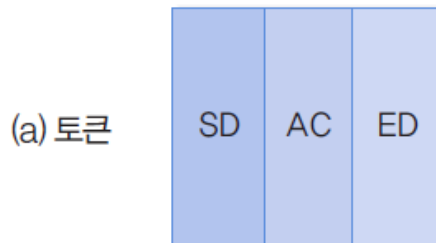


그림 5-12 토큰 버스 프레임의 Data 필드

❖ 프레임 구조

■ 토큰 링 프레임 Token Ring Frame

- 토큰 프레임 : SD, AC, ED 필드
- 데이터 프레임 [그림5-13]의 (b)
- 모니터(Monitor) : 특별한 기능을 수행하는 관리 호스트. 네트워크 정상동작을 방해하는 예기치 않은 오류 복구 수행
 - 토큰 분실 시 새로 생성, 무한 순환중인 프레임 제거 등



SD(Start Delimiter) : 시작 구분자
AC(Access Control) : 접근 제어
FC(Frame Control) : 프레임 제어
ED(End Delimiter) : 끝 구분자
FS(Frame Status) : 프레임 상태

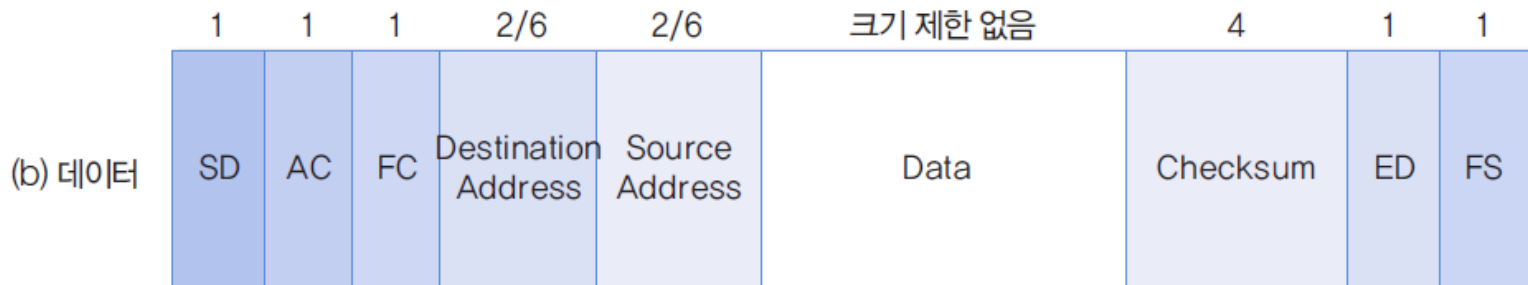


그림 5-13 토큰 링 프레임의 구조

❖ LLC 프레임 캡슐화

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

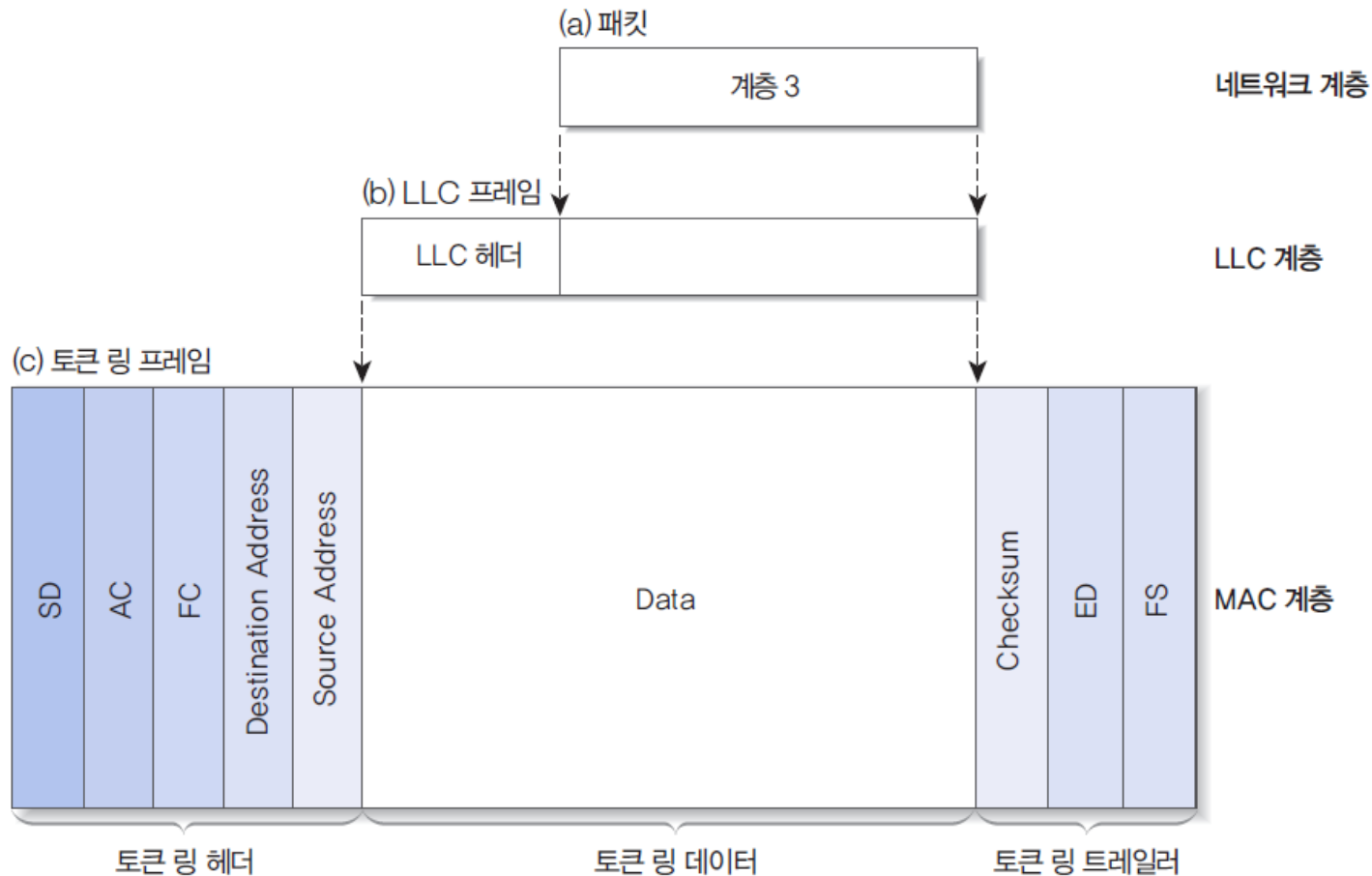


그림 5-14 토큰 링 프레임의 Data 필드

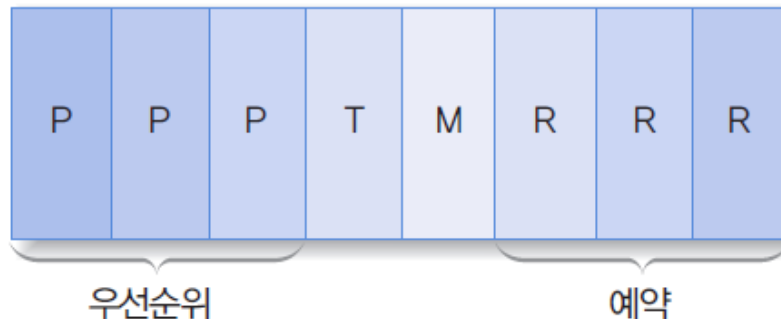
❖ 프레임 필드의 의미

■ Start Delimiter/End Delimiter

- 프레임의 시작과 끝을 구분
- End Delimiter 필드
 - 1 비트는 데이터 프레임을 여러 개로 나누어 전송하는 경우에 사용. 데이터 처음과 중간 프레임은 1 비트 값을 1로, 마지막 프레임은 0으로 지정
 - F 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 데이터 프레임이 자신을 지날 때 M을 1로 지정, M비트가 1인 프레임이 다시 지나가면 링에서 제거함

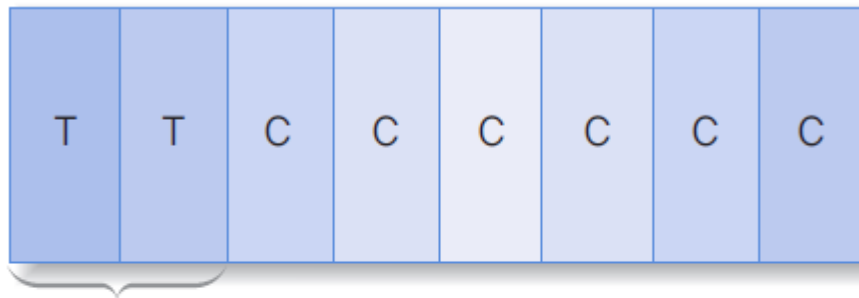


P(Priority) : 우선순위
T(Token) : 토큰
M(Monitor) : 모니터
R(Reserved) : 예약

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

■ 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^{Access} 비트 : 수신 호스트가 접근했다는 표시로 1 지정
- C^{Copy} 비트: 수신 호스트가 자신의 내부 버퍼에 보관 했다는 표시로 1 지정

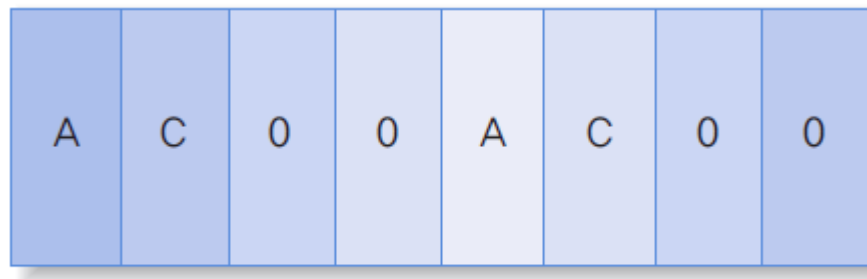


그림 5-17 Frame Status 필드



Thank You
