4장. 데이터 전송의 기초

컴퓨터 네트워크

데이터 전송 방식

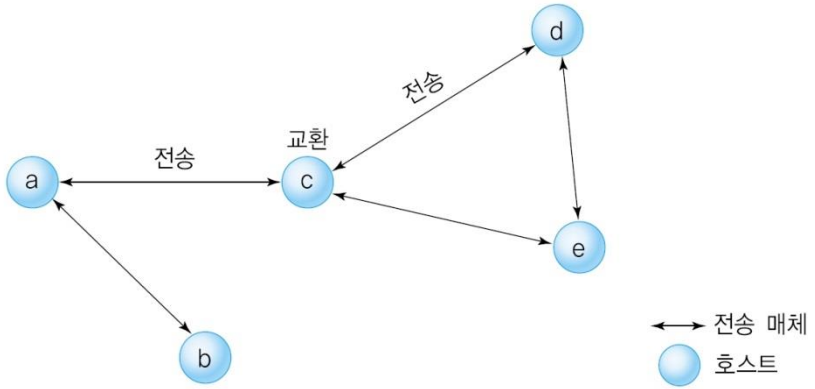
네트워크 효과

- 자원공유
  - 컴퓨터 하드웨어 외에 물리적, 논리적 정보 공유
- 병렬 처리에 의한 성능 향상
  - 네트워크 속도 제한에 의한 한계
  - 네트워크 성능 개선으로 병렬 처리 가능
- 중복 저장으로 인한 신뢰성 향상
  - 일관성 문제
  - 데이터 갱신 비용

전송과 교환

- 전달(Transfer)
  - 교환(Switching) + 전송(Transmission)
- 교환
  - 둘 이상의 경로 중에 어느 방향으로 전달할지 선택
- 전송
  - 물리적으로 1:1 연결된 시스템 사이의 데이터 전송

- a에서 d로 갈 때
  - a – c 전송
  - c에서 교환
  - c – d 전송



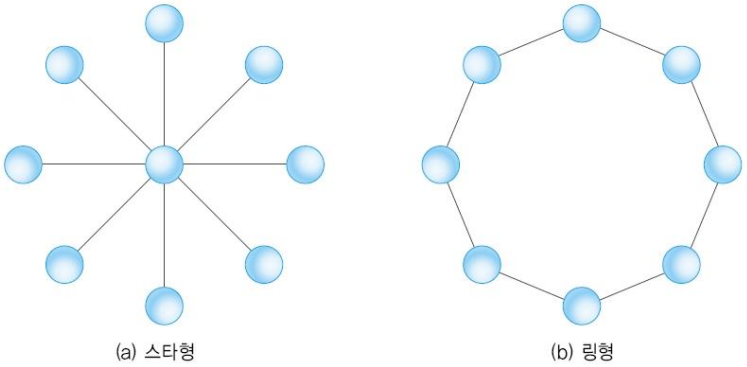
[그림 4-1] 전송과 교환

전송 방식의 종류

- 지리적 분포
  - LAN
  - MAN
  - WAN

- 전송 교환기술에 의한 분류
  - 점대점(Point-to-point)
    - 호스트들이 물리적으로 1:1 형식으로 연결
  - 브로드캐스팅(Broadcasting)
    - 호스트들이 공유 전송 매체에 연결

점대점 방식 (1)

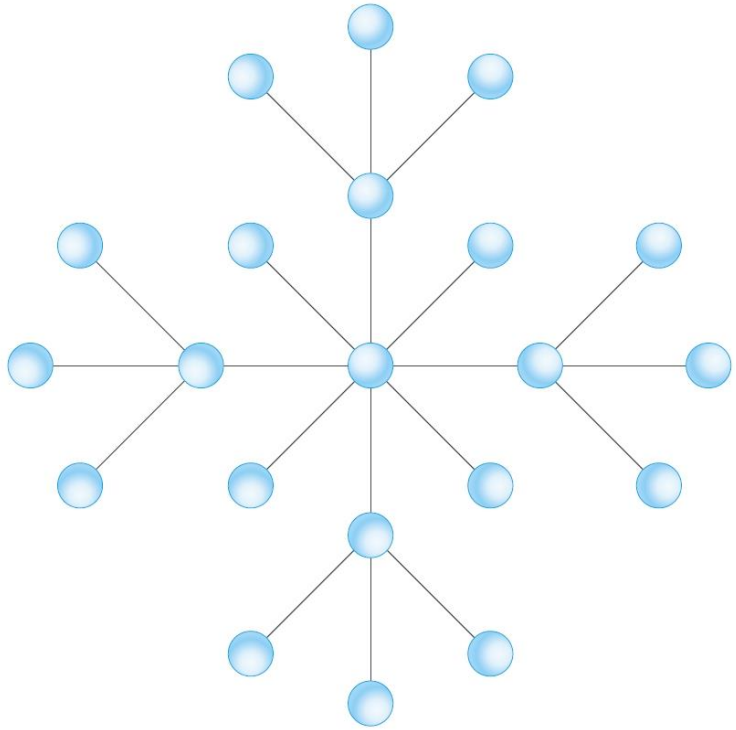


[그림 4-2] 점대점 방식

- 교환호스트가 송수신 호스트 중간에 위치
- 직접 연결하거나 중계기능을 통해 전달
- 연결개수가 많아지면 성능면에서 우수하나 매체 길이가 증가하여 비용 증가
- 연결 개수가 적어지면 네트워크 혼잡도 증가

점대점 방식 (2)

- 스타(star)형
  - 중앙에 있는 하나의 중개 호스트(허브 : hub) 주위로 여러 호스트를 1:1로 연결
  - 중앙 호스트의 성능과 신뢰성이 중요
  - 트리(tree)형



[그림 4-3] 트리 구조

점대점 방식 (3)

- 링형
  - 호스트의 연결이 순환 구조를 이룸
  - 모든 호스트가 전송과 교환 기능을 수행
  - 시계방향 또는 반시계방향 선택 가능
  - 현실적으로 한 방향으로만 전송
  - 토큰
    - 데이터를 전송할 수 있는 권리
    - 데이터 전송을 원하는 호스트는 미리 토큰을 확보해야 함
    - 데이터 전송이 완료되면 호스트는 토큰을 반납해야 함
    - 데이터를 전송하는 호스트가 없으면 오직 하나의 토큰이 링을 순환함

점대점 방식 (4)

- 링형 (계속)
  - 데이터 전송 원리
    - 먼저, 토큰을 링에서 회수하여 확보한 후,
    - 데이터를 링에 전송함
    - 데이터는 링을 한 바퀴 순환한 후, 다시 송신 호스트에게 돌아옴
    - 이 과정에서 링에 연결된 모든 호스트가 데이터를 수신함
    - 단, 자신을 목적지로 하는 호스트만 데이터를 보관하고, 다른 호스트는 버림
    - 마지막으로, 송신 호스트는 데이터를 회수한 후에 토큰을 링에 돌려줌

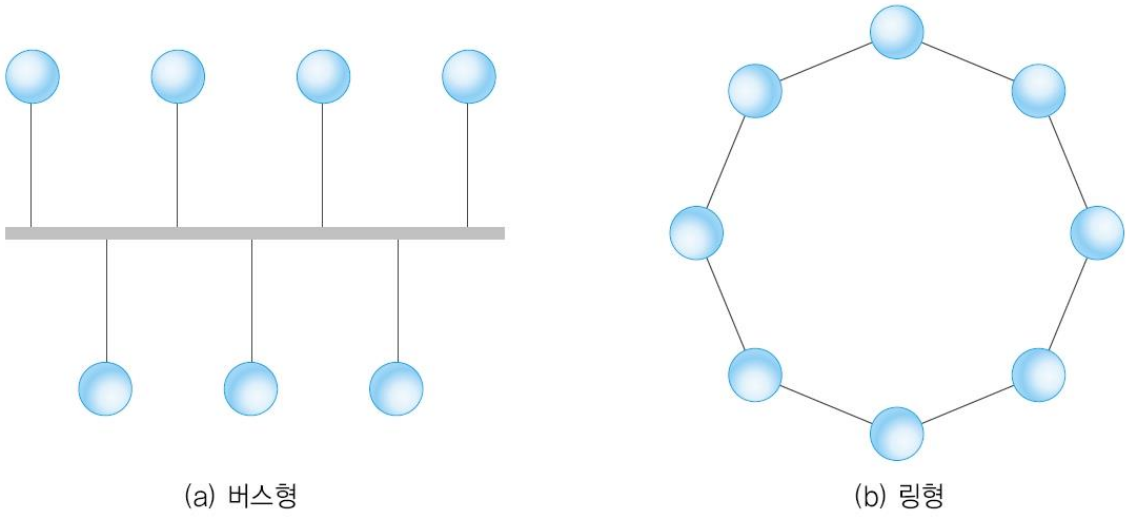
점대점 방식 (5)

- 완전형
  - 네트워크에 존재하는 모든 호스트를 1:1로 연결
  - 교환 기능이 불필요
  - 극단적으로 비효율적인 방식

- 불규칙형
  - 트레픽이 많은 지역은 연결의 수가 많지만,
  - 트레픽이 적은 지역은 연결의 수가 적음

브로드캐스팅 방식 (1)

- 네트워크에 연결된 모든 호스트에게 데이터를 전달하는 방식
- 주로 LAN 환경에서 사용
- 버스형과 링형이 존재



(a) 버스형 (b) 링형

[그림 4-4] 브로드캐스팅 방식

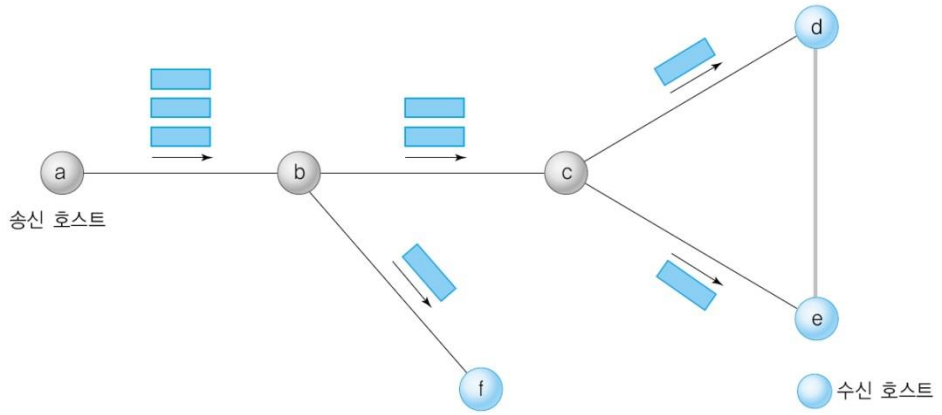
브로드캐스팅 방식 (2)

- 버스형
  - 공유 버스에 모든 호스트를 연결
  - 둘 이상의 호스트가 데이터를 전송하면 충돌 발생
  - 충돌 문제의 해결 방법
    - 사전 예방: 전송 시간대를 다르게 하는 방법과 토큰 제어 방식이 가능
    - 사후 해결: 충돌을 감지하는 기능이 필요 (예: 이더넷)
- 링형
  - 호스트를 순환 구조로 연결
  - 송신 호스트가 전송한 데이터는 링을 한 바퀴 순환한 후 송신 호스트에 되돌아옴
  - 중간의 호스트 중에서 수신 호스트로 지정된 호스트만 데이터를 내부에 저장
  - 데이터를 전송하기 위해서는 토큰 확보가 필수

멀티포인트 통신 (1)

- 하나의 송신 호스트를 기준으로
  - 유니포인트: 하나의 수신 호스트와 연결
  - 멀티포인트: 다수의 수신 호스트와 연결
- 송신 호스트가 한번의 전송으로
  - 유니캐스팅: 하나의 수신 호스트에 데이터를 전송
  - 멀티캐스팅: 다수의 수신 호스트에 데이터를 전송

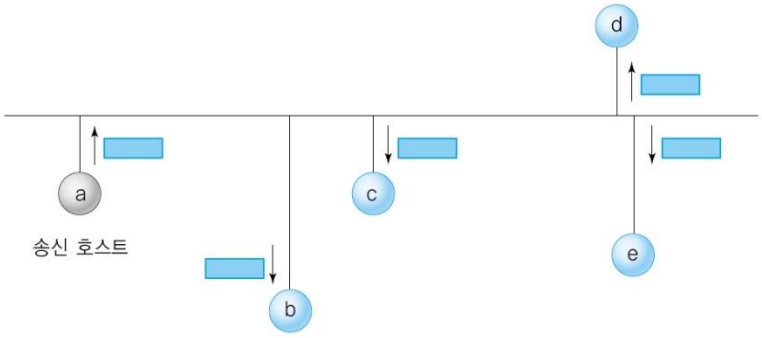
- 멀티포인트 유니캐스팅
  - 유니캐스팅 방식을 이용하여 일대다 통신을 지원
  - 호스트 a가 호스트 d, e, f에게 데이터를 전송하려면 3번의 송신 절차가 필요
  - 수신 호스트의 수가 증가하면 성능에 문제점 발생



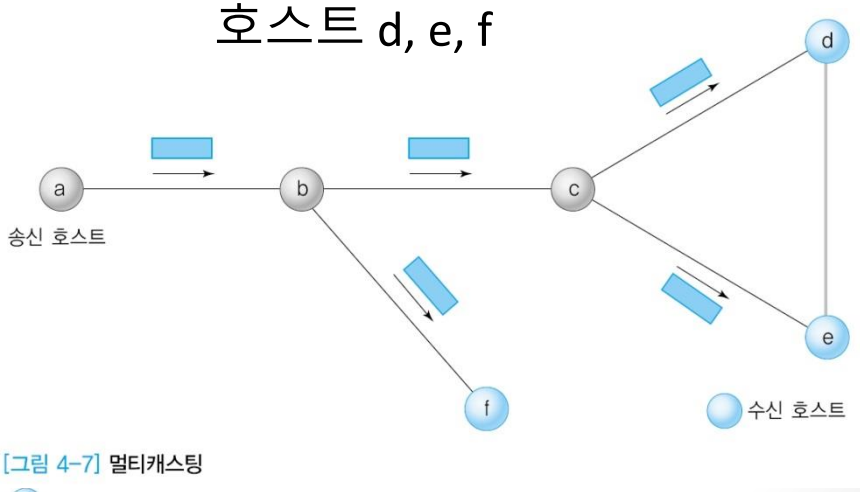
[그림 4-5] 멀티포인트 유니캐스팅

멀티포인트 통신 (2)

- 브로드캐스팅
  - 네트워크에 연결된 모든 호스트에게 데이터 전송
  - 자신을 목적지로 하는 호스트만 데이터를 내부에 저장하고, 다른 호스트는 데이터를 무시함
- 멀티캐스팅
  - 1:다 전송 기능을 지원
  - 송신 호스트는 한번의 데이터 전송으로 여러 호스트에게 데이터를 전송할 수 있음
  - 예: 송신 호스트 a, 수신 호스트 d, e, f



[그림 4-6] 브로드캐스팅



[그림 4-7] 멀티캐스팅

수신 호스트

오류 제어

오류 제어 (1)

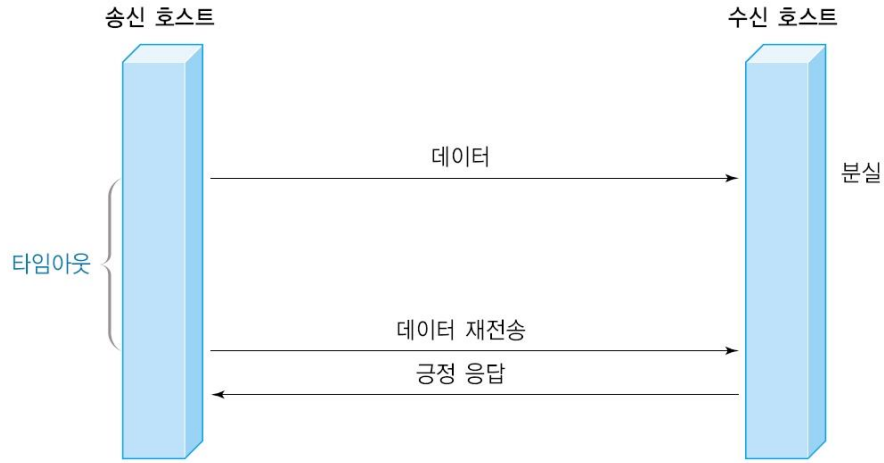
- 수신 호스트의 응답 프레임
  - 긍정 응답
  - 부정 응답

- 송신 호스트의 타이머 기능
  - 프레임 분실 오류시 수신 호스트의 인지 불가능
  - 일정 시간 내 긍정 응답이 없으면 타임아웃(timeout) 기능을 동작시켜 재전송

- 순서번호(sequence number)
  - 긍정 응답이 분실되는 경우 재전송으로 인한 중복 수신 가능
  - 이를 구별하기 위해 순서번호 기록

오류 제어 (2)

[그림 4-8] 정상적인 데이터 전송

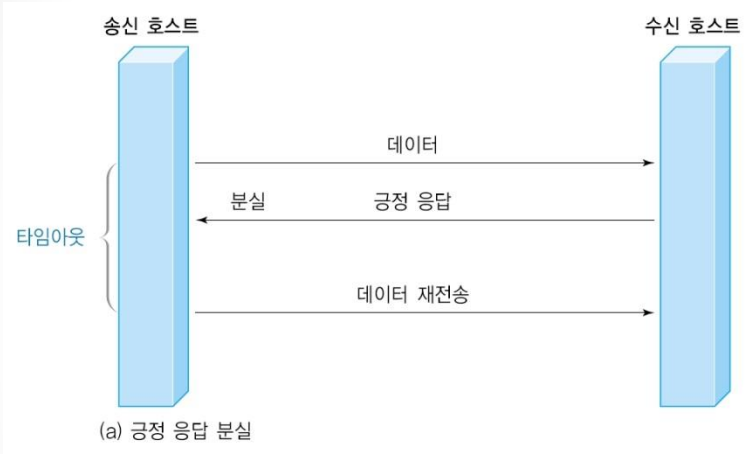


[그림 4-10] 프레임 분실 오류

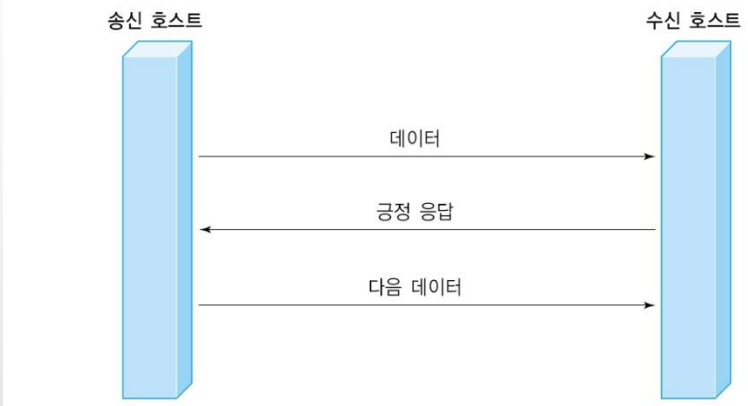
[그림 4-9] 프레임 변형 오류

오류 제어 (3)

- 순서번호



(a) 긍정 응답 분실



(b) 긍정 응답 도착

[그림 4-11] 순서 번호가 없는 경우

(a) 긍정 응답 분실

(b) 긍정 응답 도착

[그림 4-12] 순서 번호가 있는 경우

흐름제어

- 전송 데이터의 속도 조절
- 송신 호스트는 수신 호스트가 감당할 수 있을 정도의 전송속도를 유지하면서 전송
- 흐름제어가 없는 경우 데이터의 손실, 재전송으로 이어짐
- 기본 원리
  - 다음에 수신할 프레임의 전송 시점을 송신 호스트에게 통지하는 방식
- 대표적인 방식
  - 슬라이딩 윈도우(sliding window)

프레임

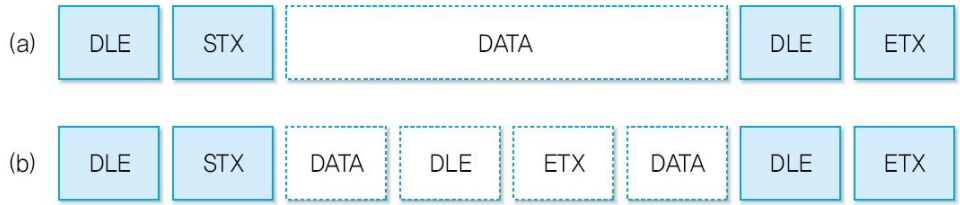
데이터 링크 계층 기능

- 프레임(frame) 단위로 나누어 처리
  - 전송 데이터
  - 오류 확인을 위한 체크썸(checksum),
  - 송수신호스트 주소,
  - 기타 프로토콜에서 사용되는 제어코드 같은 정보 포함

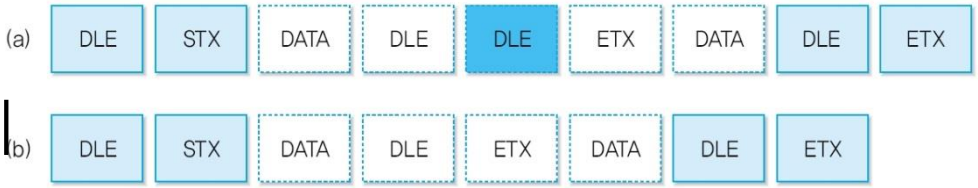
- 프레임 구분
  - 내부 정보를 표현하는 방식에 따라
    - 문자 프레임
    - 비트 프레임

문자 프레임

- 프레임의 내용이 문자로만 구성됨
- 프레임의 시작과 끝에 특수 문자 사용
  - 시작: DLE / STX
  - 끝: DLE / ETX
- 전송 데이터에 특수 문자가 포함되면 혼선이 발생
  - 문자 스터핑(stuffing)
    - 데이터에 DLE가 있으면 강제로 DLE 하나 더 추가
    - 수신측에서는 두 개의 DLE가 나오면 뒤에 있는 DLE 제거



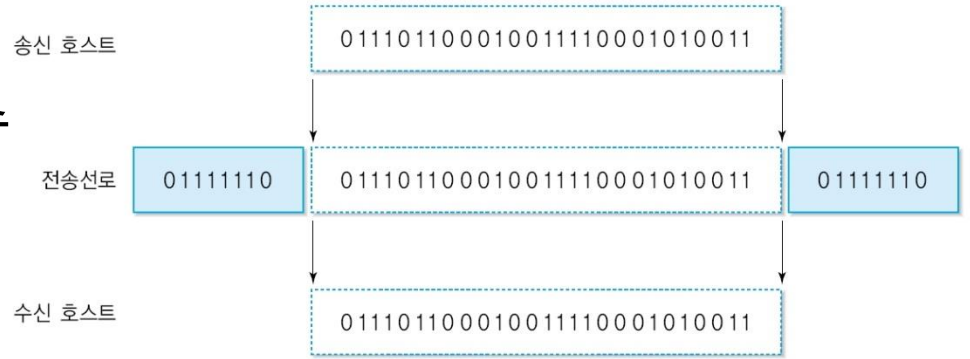
[그림 4-13] 문자 프레임의 구조



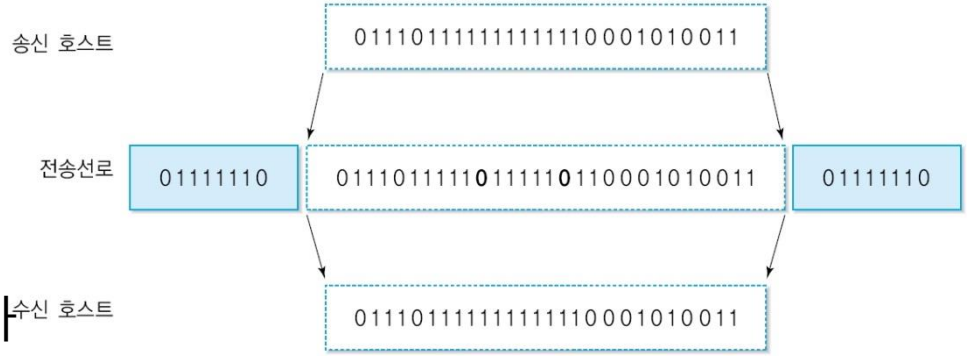
[그림 4-14] 문자 스터핑

비트 프레임

- 임의의 비트 패턴 전송 가능
- 프레임과 시작과 끝을 나타내는 플래그(flag) 사용
- 전송 데이터에 플래그와 같은 패턴이 등장할 수 있음
  - 비트 스터핑
    - 1이 연속해서 5개 발생하면 강제로 0 추가



[그림 4-15] 비트 프레임의 구조



[그림 4-16] 비트 스터핑

다항 코드

오류 극복 방법

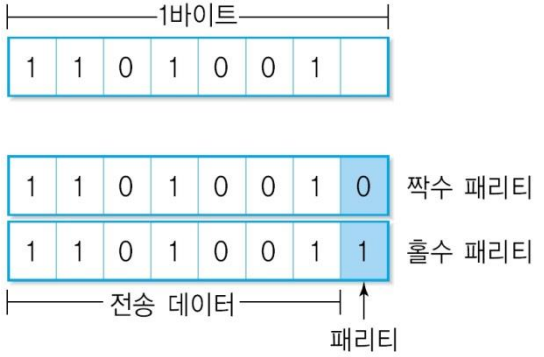
- 오류 검출코드를 넣어 수신호스트가 오류 검출 후 재전송으로 복구
  - CRC(Cyclic Redundancy Check)

- 오류복구 코드를 넣어 수신호스트가 오류 검출과 복구를 동시에 수행
  - 해밍코드(hamming code)
  - 순방향 오류복구(FEC: Forward Error Control)

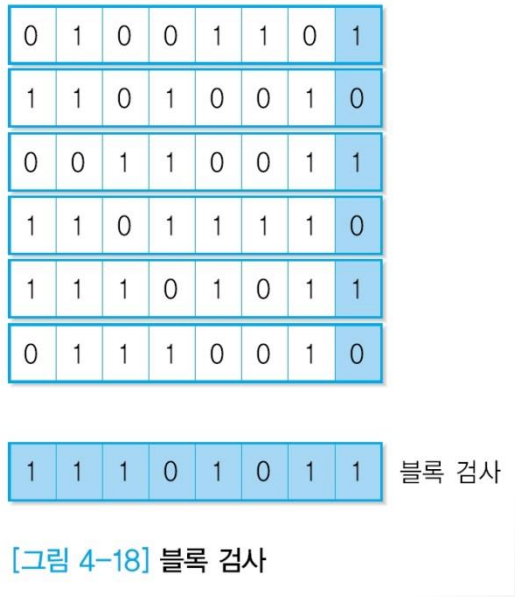
오류 검출 (1)

- BEC (Backward Error Control)
- ARQ(Automatic Repeat reQuest)

- 패리티(parity)
  - 짝수 패리티
  - 홀수 패리티
- 블록 검사 (Block sum check)



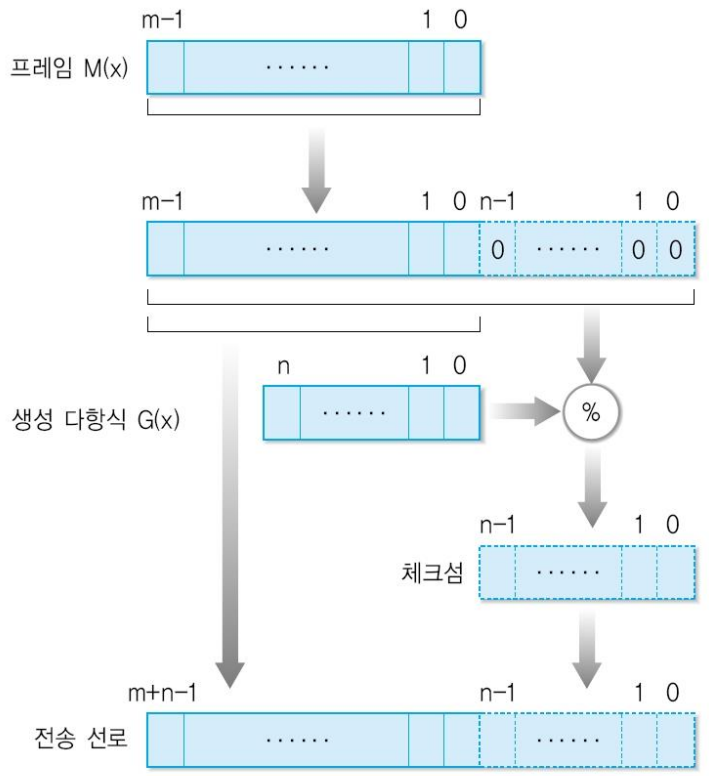
[그림 4-17] 패리티 비트



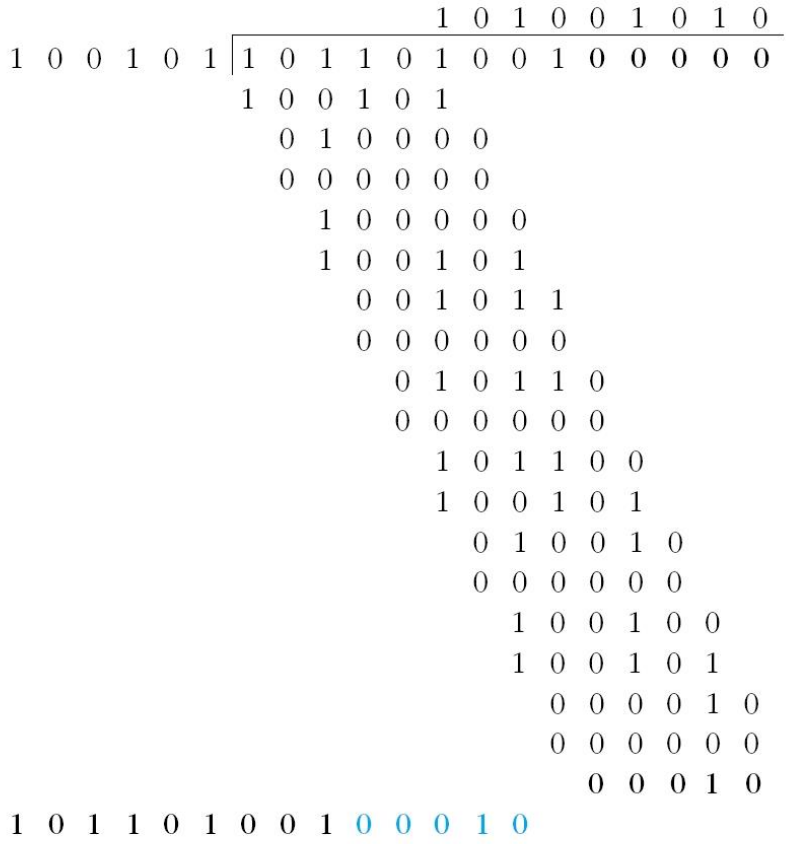
[그림 4-18] 블록 검사

오류 검출 (2)

- 다항코드(Polynomial Code)



[그림 4-19] 생성 다항식



[그림 4-20] 체크섬 계산의 예