IT COOKBOOK

257

4차 산업혁명과 함께하는 네트워크

네트워크 개론 3판

진혜진 지음

네트워크 통신

01. 통신방식

- 다른 컴퓨터에 데이터 전송 서비스를 제공하는 컴퓨터를 '서버'라 하고, 서버에서 보내주는 데이터 서비스를 수신하는 컴퓨터를 '클라이언트'라고 한다.

- 서버는 클라이언트(사용자)한테 요청 받아 서비스를 제공하는데, 이렇게 구성된 시스템을 '클라이언트/서버 시스템'이라고 한다.

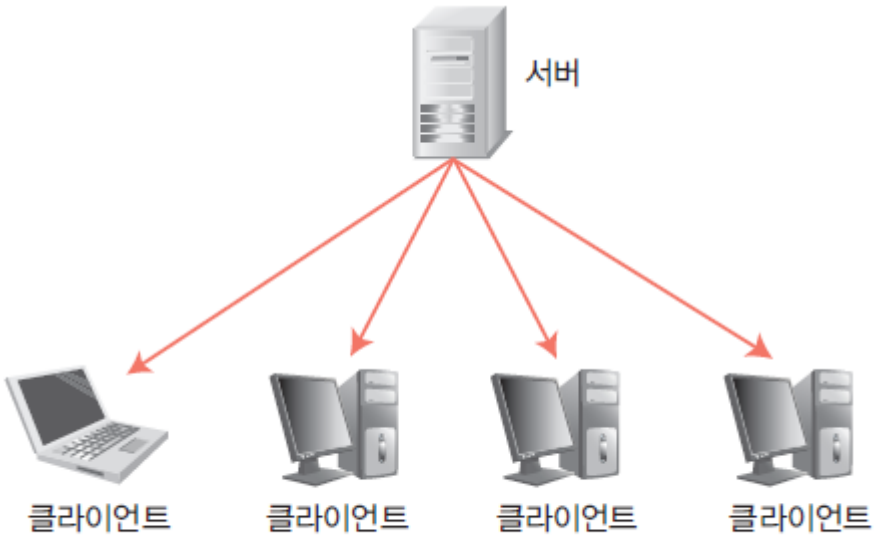

그림 3-1 서버와 클라이언트의 데이터 전송 관계

01. 통신방식

1. LAN에서 통신하는 방식

- 유니캐스트는 가장 많이 사용하는 통신 방식으로 수신지 주소(MAC 주소)를 적어 특정 컴퓨터에만 전송한다.

- 브로드캐스트는 영역 안에 있는 모든 컴퓨터에 한 번에 다 전송한다.

- 멀티캐스트는 유니캐스트와 브로드캐스트의 장점을 결합하여 특정 그룹 컴퓨터에만 한 번에 데이터를 전송하여 그룹 이외의 컴퓨터에는 영향을 주지 않는다.

01. 통신방식

- **유니캐스트(Unicast)**
  - 네트워크에서 가장 많이 사용하는 유니캐스트(Unicast)는 서버와 클라이언트 간의 일대일(1:1) 통신 방식을 말한다.
  - 데이터를 송신하려는 컴퓨터의 MAC 주소를 90-2B-35-91-E0-3F, 수신하려는 컴퓨터의 MAC 주소를 90-2B-34-92-C0-5F라고 가정해 보자.
    - 통신하려면 전송되는 프레임 안에 항상 송신지(90-2B-35-91-E0-3F)와 수신지(90-2B-34-92-C0-5F) 주소, 즉 MAC 주소가 있어야 한다.

그림 3-2 유니캐스트 발생

01. 통신방식

- 자신의 MAC 주소와 수신지 MAC 주소가 동일하다면 전송된 데이터를 수신하고, 자신의 LAN 카드 MAC 주소가 수신지 주소가 아니라고 판단되면 해당 프레임은 버린다.

그림 3-3 유니캐스트 방식

01. 통신방식

- **브로드캐스트**
  - 브로드캐스트(Broadcast)는 로컬 LAN(라우터로 구분된 공간)에 있는 모든 네트워크 단말기에 데이터를 보내는 방식으로, 서버와 클라이언트 간에 일대모두(1:모두)로 통신하는 데이터 전송 서비스다.
  - 브로드캐스트의 주소는 FF-FF-FF-FF-FF-FF로 미리 정해져 있다.

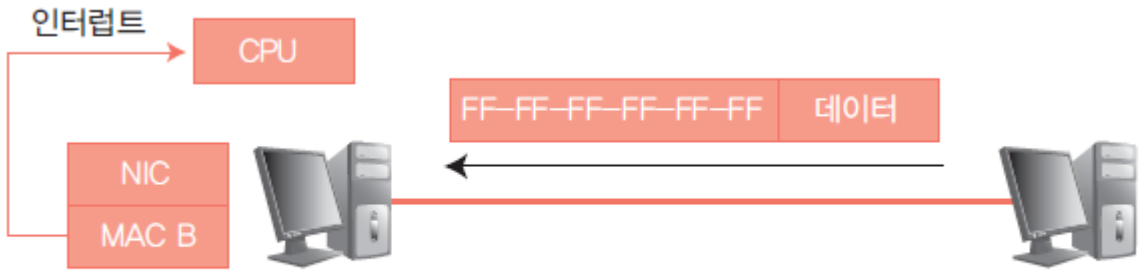

그림 3-4 브로드캐스트 발생

01. 통신방식

- 브로드캐스트는 다른 라우터를 찾거나, 라우터끼리 데이터를 교환하거나, 서버가 서비스를 제공하려고 모든 클라이언트에게 알릴 때 등 여러 상황에서 사용할 수 있다.
- 하지만 불특정다수에게 전송되는 서비스라 수신을 원치 않는 클라이언트도 수신하게 되므로 네트워크 성능 저하를 가져올 수 있다.

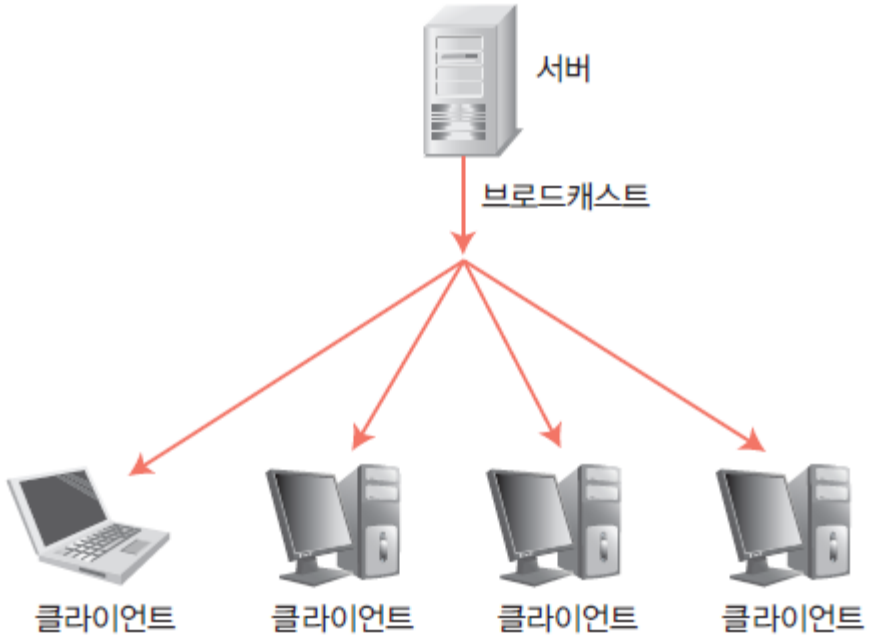

그림 3-5 브로드캐스트 방식

01. 통신방식

- **멀티캐스트**
  - 브로드캐스트는 데이터를 무조건 CPU로 전송하기 때문에 컴퓨터 자체의 성능을 떨어뜨린다. 이 문제를 가장 쉽게 해결할 수 있는 방법이 바로멀티캐스트(Multicast)이다.
  - 멀티캐스트는 전송하려는 특정 그룹에게만 한 번에 전송할 수 있기 때문에 유니캐스트처럼 반복해서 보낼 필요가 없고, 브로드캐스트처럼 전송받을 필요가 없는 컴퓨터에 보내지 않아도 된다.
  - 현재 많이 사용하는 애플리케이션에서는 이런 기능이 필요하므로 멀티캐스트가 인기를 끌고 있다.

NOTE 애니캐스트(anycast)

IPv6에 등장한 개념으로 송신 노드에서 수신자 그룹의 가장 가까운 노드로 데이터그램을 전송하는 라우팅 기법이다.

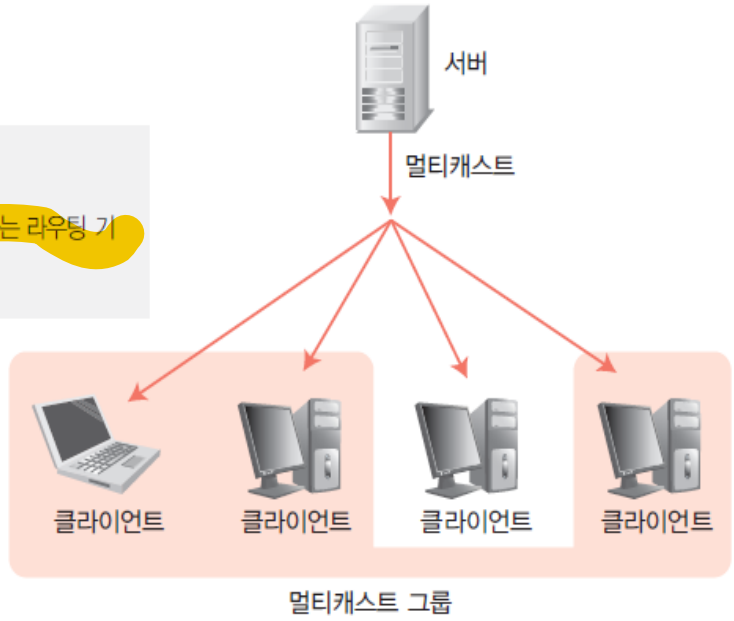

그림 3-6 멀티캐스트 방식

01. 통신방식

2. 전송 방향에 따른 통신 방식

- 통신은 떨어져 있는 두 지점 간에 정보를 전송하는 것을 말한다. 두 장치 간에 데이터를 전송 할 때는 데이터 전송 방향에 따라 단방향 simplex 통신과 양방향 duplex 통신으로 나눌 수 있다. 양방향 통신은 정보를 주고받는 시점에 따라 다시 반이중 half-duplex 통신과 전이중 full-duplex 통 신으로 나뉜다.

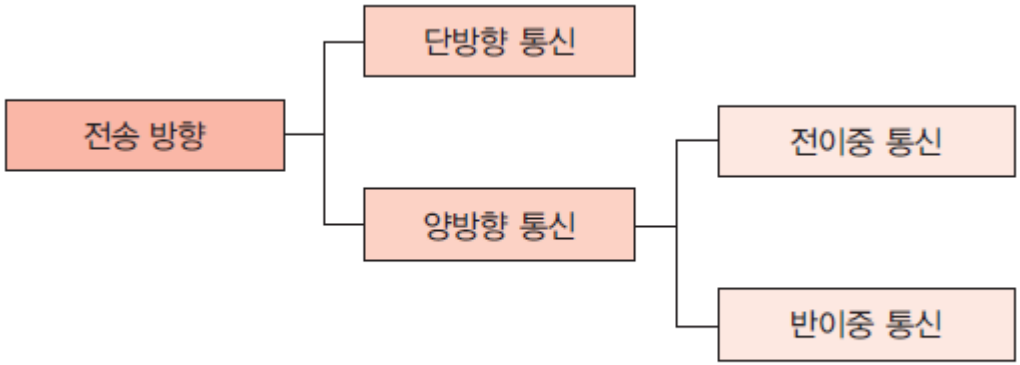

그림 3-7 전송 방향에 따른 통신 방식

01. 통신방식

- **단방향(Simplex) 통신**
  - 송신 측과 수신 측이 미리 고정되어 있고, 통신 채널을 통해 접속된 단말기 두 대 사이에서 데이터가 한쪽 방향으로만 전송되는 통신 방식을 말한다.
  - 단방향 통신에서 전기적으로 신호를 보내려면 송신 측과 수신 측을 연결하는 회로를 구성해야 하므로, 비록 단방향 전송일지라도 전송로는 두 개가 필요하다.
  - 대표적인 단방향 통신에는 예전에 많이 사용하던 무선호출기나 라디오, 아날로그 TV 방송, 모니터, 키보드 등이 있다.

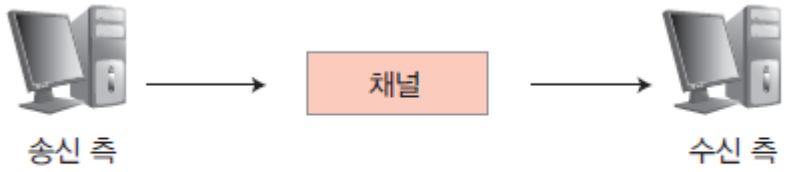

그림 3-8 단방향 통신

01. 통신방식

- **양방향(Duplex) 통신**
  - 통신 채널을 통해 접속된 두 대의 단말기 사이에서 데이터의 송수신이 모두 가능한 방식으로, 데이터의 송수신을 한 번씩 번갈아 가면서 할 수 있는 반이중 통신과 송수신을 동시에 할 수 있는 전이중 통신으로 구분된다.
  - 반이중(Half-Duplex) 통신
    - 통신 채널에 접속된 두 대의 단말기 중 어느 한쪽이 데이터를 송신하면 상대방은 수신만 할 수 있는 통신 방식이다. 송신 측과 수신 측이 정해져 있지 않으며, 양쪽 단말기의 상호 협력에 따라 송수신 방향이 바뀐다.
    - 대표적인 예로, 휴대용 무전기와 모뎀을 이용한 데이터 통신을 들 수 있다.

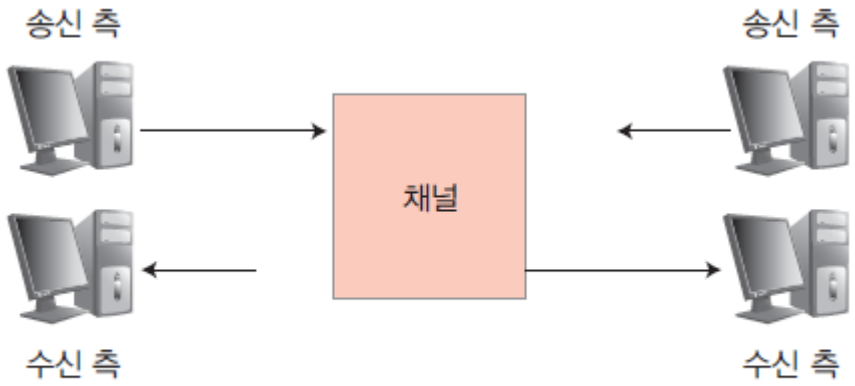

그림 3-9 반이중 통신

01. 통신방식

- **전이중(Full-Duplex) 통신**
  - 통신 채널에 접속된 단말기 두 대가 동시에 데이터를 송수신할 수 있는 통신 방식을 말한다 .
  - 전이중 통신은 통신 채널 두 개를 이용하여 한 번에 데이터를 송수신할 수 있다.

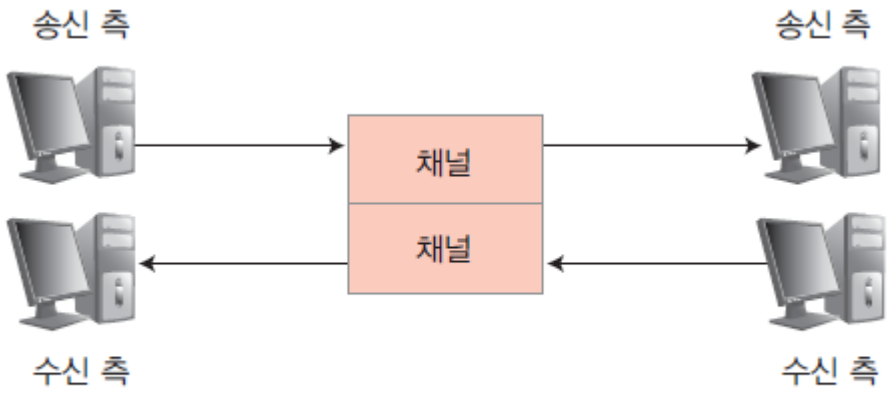

그림 3-10 전이중 통신

01. 통신방식

3. 직렬 전송과 병렬 전송

- 장치 사이에 데이터를 전송할 때 몇 가지 고려해야 할 사항이 있다. 예를 들어 여러 데이터를 한꺼번에 전송할 것인가, 아니면 한 번에 하나씩만 전송할 것인가, 그리고 데이터를 하나씩 전송한다면 어떤 방식을 사용할 것인가 등을 고려해야 한다.
- 데이터 전송은 2진 데이터를 전압이나 전류의 변화로 표현한 신호에 실어 보내는 것을 말하며, 데이터 비트를 전송하는 방법에 따라 직렬 전송과 병렬 전송으로 나눌 수 있다.

01. 통신방식

- **직렬 전송**
  - 하나의 정보를 나타내는 각 데이터 비트를 직렬로 나열한 후 하나의 통신회선을 사용하여 순차적으로 1비트씩 송신하는 방식이다.
  - 하나의 통신회선을 사용하기 때문에 송신 측에서는 데이터를 1비트씩 송신하고, 수신 측에서는 수신되는 비트를 일정한 단위로 모아서 사용한다.
  - 병렬 전송에 비해 데이터 전송속도가 느린 반면, 원거리 데이터 전송에서는 통신회선이 한 개만 필요하므로 경제적이다.

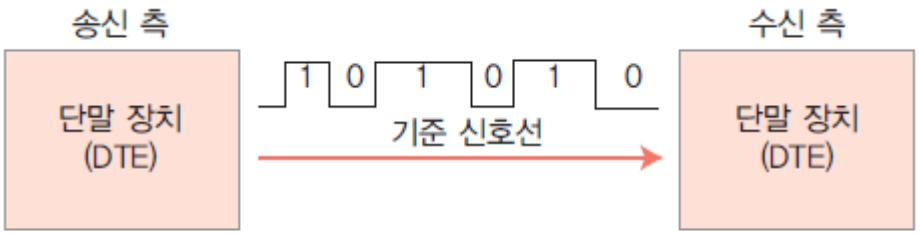

그림 3-11 **직렬 전송**

01. 통신방식

- **병렬 전송**
  - 부호를 구성하는 비트 수와 같은 양의 통신회선을 사용하여 여러 데이터 비트를 동시에 병렬로 전송하는 방식으로, 비트 n개를 전송하려고 회선 n개를 사용한다.
  - 송신 측과 수신 측 단말기 간에 여러 개의 통신회선을 사용하기 때문에 여러 비트의 데이터를 한 번에 송신한다.
  - 병렬 전송은 거리에 비례해서 선로비용이 많이 들기 때문에 전송속도가 빨라야 하는 짧은 거리의 데이터 전송에 주로 사용한다.

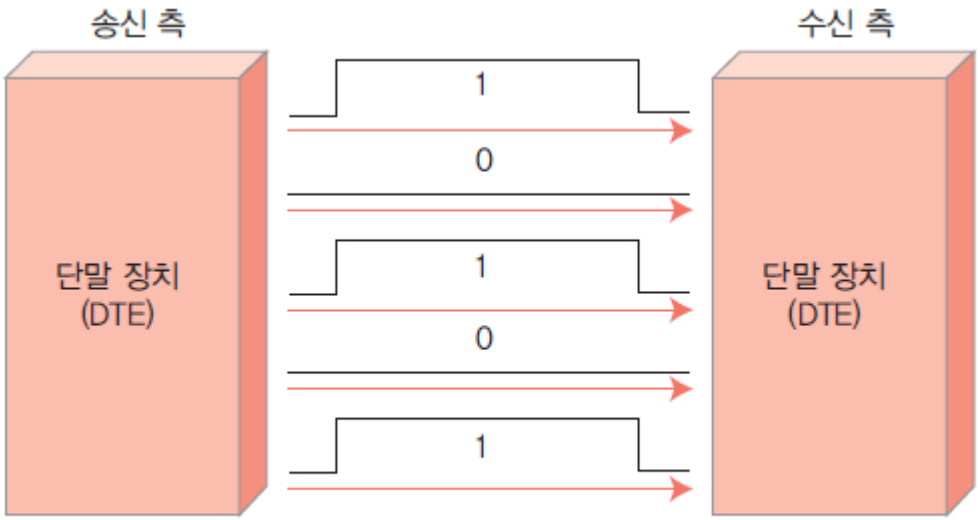

그림 3-16 병렬 전송

02. 통신오류검출

- 수신 측으로 전송한 데이터는 송신 측의 데이터와 동일해야 하지만, 다양한 원인 때문에 데이터 오류가 발생할 수 있다.

- 따라서 신뢰할 수 있는 네트워크 통신을 하려면 오류를 검출·수정해야 한다.

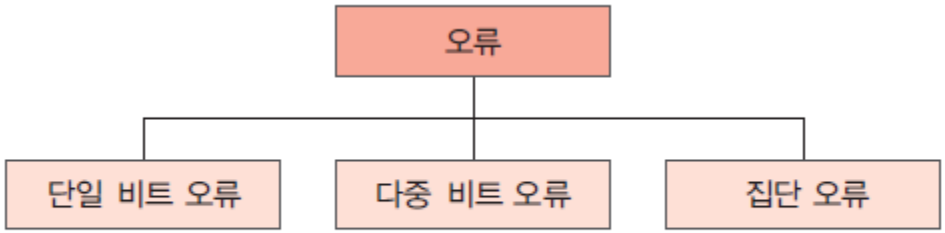

그림 3-17 오류의 종류

02. 통신오류검출

- **단일-비트 오류(Single-bit Error)**
  - 데이터 단위 중 하나의 비트만 변경하는 오류를 말한다.

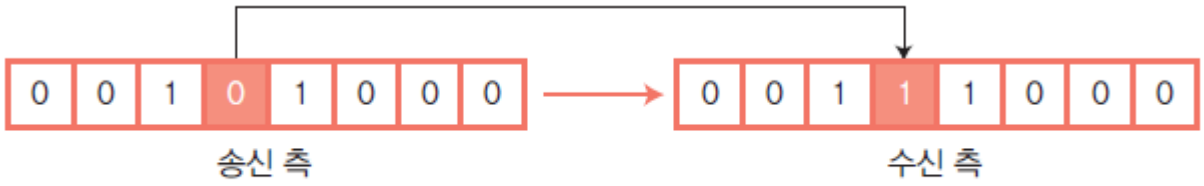

그림 3-18 단일 비트 오류

- **다중-비트 오류(Multiple-bit Error)**
  - 데이터 단위 중 두 개 이상의 비연속적인 비트를 변경하는 오류를 말한다

그림 3-19 다중 비트 오류

02. 통신오류검출

- **집단 오류(Burst Error)**
  - 데이터 단위 중 두 개 또는 그 이상의 연속적인 비트를 변경하는 오류를 말한다.
  - 송신 측이 보내려는 데이터 외에 별도로 잉여(중복)분의 데이터를 추가해서 전송하면 수신 측은 이 잉여 데이터를 검사하여 오류를 검출할 수 있다.
  - 오류를 검출하는 방식에는 패리티 비트 검사 parity bit check, 블록 합 검사 block sum check, 순환 중복 검사 Cyclic Redundancy Check, CRC 등이 있다.

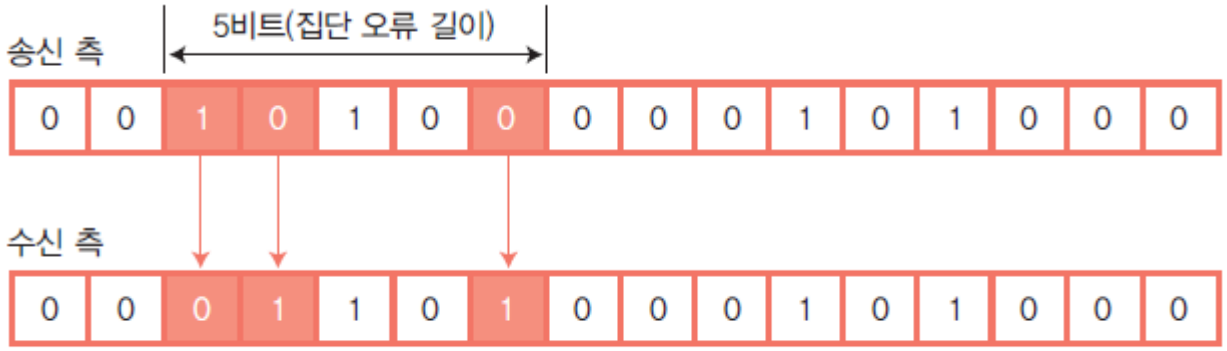

그림 3-20 집단 오류

02. 통신오류검출

1. 패리티 비트 검사

- 패리티 비트 검사(Parity Bit Check)는 전송하는 데이터마다 패리티 비트를 하나씩 추가하여 홀수 또는 짝수 검사 방법으로 오류를 검출한다.
- 예를 들어, 7비트 데이터를 전송할때 1비트 검사 비트를 추가로 전송하여 수신 측에서 데이터 전송 중 발생한 오류를 검출할수 있도록 하는 방식이다.
- 추가로 전송되는 1비트를 '패리티 비트'라고 한다.
- 패리티 비트의 값은 데이터 코드 내에 있는 1의 수를 계산함으로써 결정된다.

그림 3-21 패리티 비트 검사

02. 통신오류검출

- **홀수 패리티 방식(Odd Parity)**
  - 전체 비트에서 1의 개수가 홀수가 되도록 패리티 비트를 정하는 것을 말한다.
  - 데이터 비트에서 1의 개수가 짝수면 패리티 비트를 1로 정하여 전송되는 전체 데이터에 있는 1의 개수는 홀수가 된다.

- **짝수 패리티 방식(Even Parity)**
  - 전체 비트에서 1의 개수가 짝수가 되도록 패리티 비트를 정하는 것을 말한다.
  - 데이터 비트에서 1의 개수가 홀수면 패리티 비트를 1로 정하여 전송되는 전체 데이터에 있는 1의 개수는 짝수가 된다.

표 3-2 패리티 비트 검사 적용 예

7비트 데이터	패리티 포함 8비트	
	짝수 패리티	홀수 패리티
0100101(3)	10100101	00100101
0111010(4)	00111010	10111010
1101011(5)	11101011	01101011

02. 통신오류검출

- 홀수 패리티 검사
- 전송하려는 데이터가 1101001이라고 가정해 보자. 1의 개수를 홀수로 만들려고 패리티 비트를 1로 지정한다(11101001).
- 데이터를 전송 받은 수신 측은 패리티 비트를 포함한 데이터 내 1의 개수를 세어 홀수인지 판단한 후 홀수가 아니면 재전송을 요청한다.

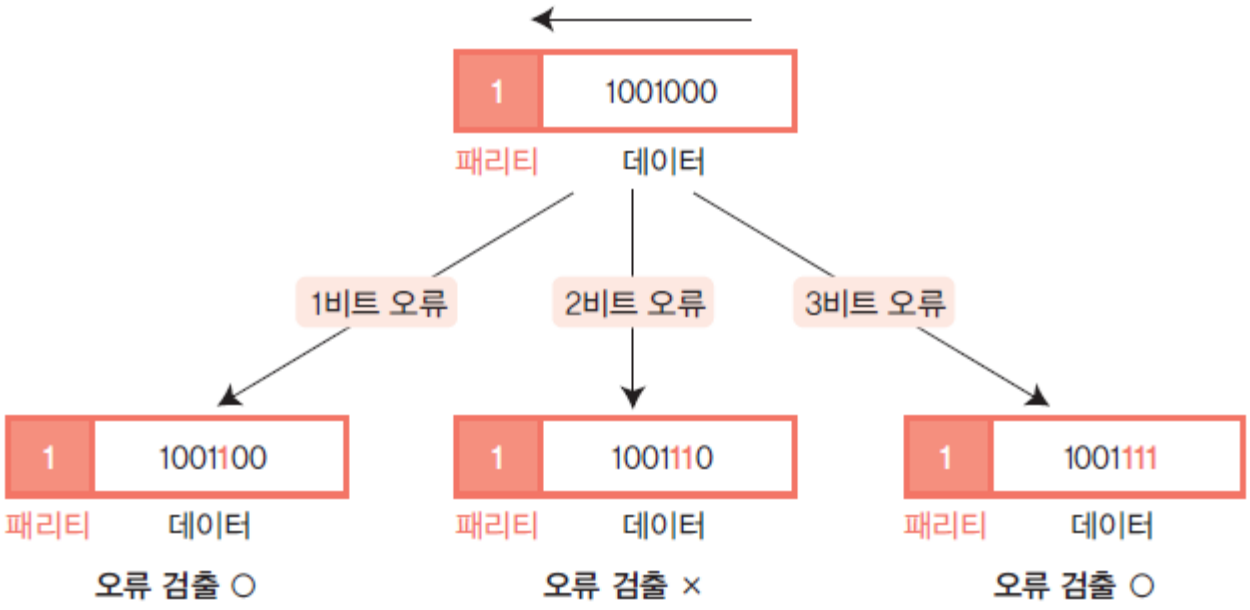

그림 3-22 홀수 패리티 검사 오류 검출 예

02. 통신오류검출

- 패리티 검사는 모든 단위 비트열 오류를 검출할 수 있다. 데이터를 전송할 때 홀수 개의 비트에 오류가 발생하면 오류가 검출되지만, 짝수 개의 비트에 오류가 발생하면 오류를 검출할 수 없다(11011001: 짝수 개의 오류가 발생하여 1의 개수가 홀수인 경우).
- 전송 중에 두비트가 변경되어 비트열이 손상되었음에도 패리티 검사를 통과하기 때문이다. 예를 들어 HyeJin이라는 단어를 전송할 때 패리티 검사를 적용해보자. 다음은 짝수 패리티 방식을 적용한 경우이다. 문자에 해당하는 2진 값은 [표 3-4]를 참고한다.

'HyeJin'이라는 단어 송신
1001000 1111001 1100101 1001010 1101001 1101110

패리티 비트 적용(even) 후 전송
01001000 11111001 01100101 11001010 01101001 11101110

수신된 정보(accept)
01001000 11111001 01100101 11001010 01101001 11101110

수신된 정보(reject, 재전송 요구: 1의 개수가 홀수인 경우)
01101000 11111001 01101101 11001010 01101001 11101110

오류 검출이 불가능한 경우(짝수 개의 비트 오류 발생) → 블록 합 검사로 오류 검출 가능
01111000 11100001 01100101 11001010 01101001 11101110

02. 통신오류검출

표 3-4 문자에 해당하는 2진 값

2진수	8진수	10진수	16진수	문자	2진수	8진수	10진수	16진수	문자
100 0001	101	65	41	A	110 0001	141	97	61	a
100 0010	102	66	42	B	110 0010	142	98	62	b
100 0011	103	67	43	C	110 0011	143	99	63	c
100 0100	104	68	44	D	110 0100	144	100	64	d
100 0101	105	69	45	E	110 0101	145	101	65	e
100 0110	106	70	46	F	110 0110	146	102	66	f
100 0111	107	71	47	G	110 0111	147	103	67	g
100 1000	110	72	48	H	110 1000	150	104	68	h
100 1001	111	73	49	I	110 1001	151	105	69	i
100 1010	112	74	4A	J	110 1010	152	106	6A	j
100 1011	113	75	4B	K	110 1011	153	107	6B	k

100 1011 – 110 0111 – 110 1010 -> 홀수
>1100 1011 – 0110 0111 – 1110 1010

02. 통신오류검출

100 1100	114	76	4C	L	110 1100	154	108	6C	l
100 1101	115	77	4D	M	110 1101	155	109	6D	m
100 1110	116	78	4E	N	110 1110	156	110	6E	n
100 1111	117	79	4F	O	110 1111	157	111	6F	o
101 0000	120	80	50	P	111 0000	160	112	70	p
101 0001	121	81	51	Q	111 0001	161	113	71	q
101 0010	122	82	52	R	111 0010	162	114	72	r
101 0011	123	83	53	S	111 0011	163	115	73	s
101 0100	124	84	54	T	111 0100	164	116	74	t
101 0101	125	85	55	U	111 0101	165	117	75	u
101 0110	126	86	56	V	111 0110	166	118	76	v
101 0111	127	87	57	W	111 0111	167	119	77	w
101 1000	130	88	58	X	111 1000	170	120	78	x
101 1001	131	89	59	Y	111 1001	171	121	79	y
101 1010	132	90	5A	Z	111 1010	172	122	7A	z

02. 통신오류검출

2. 블록 합 검사

- 문자를 블록으로 전송하면 오류 확률이 높아지는데, 오류 검출 능력을 향상시키려고 문자 블록에 수평 패리티와 수직 패리티를 2차원적으로 검사하는 방법이 바로 블록 합 검사(Block Sum Check)이다.
- 행 단위 패리티에 열 단위의 오류 검사를 수행할 수 있는 열 패리티 문자를 추가하여 이중으로 오류 검출 작업을 수행한다.
- 추가된 열 패리티 문자를 '블록 검사 문자(BCC, Block Check Character)'라고 한다.
- 블록 합 검사를 사용하면 한 데이터에서 짝수 개의 오류가 발생하더라도 오류를 검출할 수 있다.

02. 통신오류검출

그림 3-23 블록 합 검사

02. 통신오류검출

- 'HyeJin'이라는 단어를 전송하는 데 블록 합 검사를 적용한 경우를 살펴보자.

- **송신 측** : 수평 패리티 검사는 홀수 패리티를 적용하고, 수직 패리티 검사는 짝수 패리티를 적용하여 프레임(단어) 끝에 1 0 0 1 1 0 0 1 BCC 를 추가해서 전송한다.

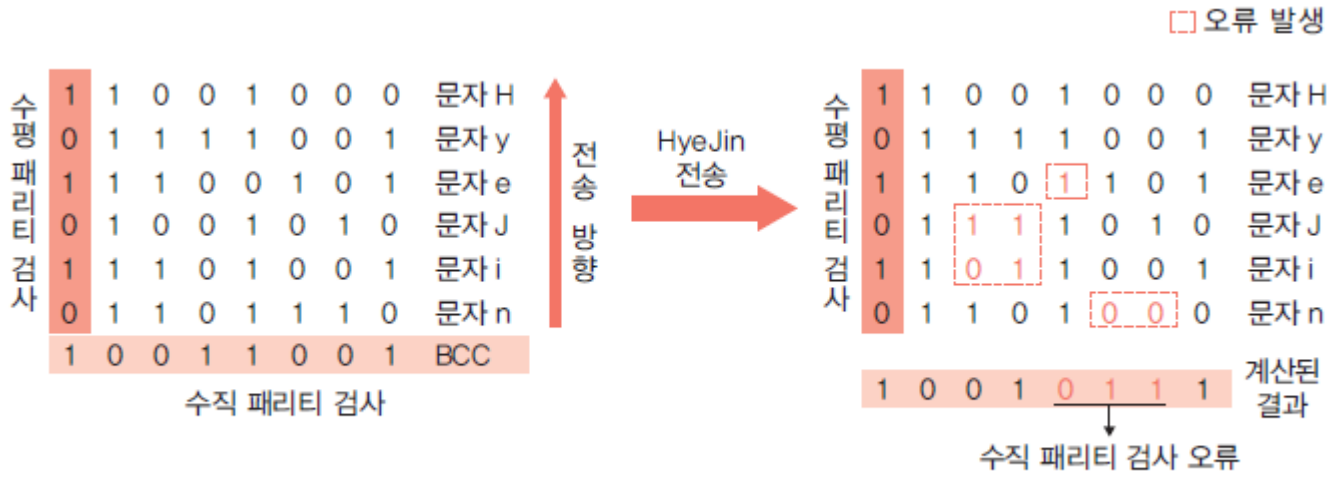

그림 3-24 'HyeJin'을 전송할 때의 블록 합 검사

02. 통신오류검출

- **수신 측** : 수신된 프레임의 문자에 대해 BCC를 계산(1 0 0 1 0 1 1 1 계산된 결과)하고 송신 측이 전송한 BCC와 비교하여 오류를 검출한다. 이때 두 값이 같으면 오류가 없는 것이다. 만약 두 값이 다르면 1비트 오류는 위치까지 알 수 있고, 2비트 오류의 경우 위치는 알수 없으나 오류 여부는 판단이 가능하다. 블록 합 검사라도 연속된 2개의 문자에서 같은 위치의 2비트가 오류일 때는 오류를 검출할 수 없다. J(11) 문자와 i(01) 문자를 전송하다가 각 문자당 2비트에 오류가 발생했는데도 불구하고 수직 패리티 검사를 해보면 송신 측의 BCC(01)와 일치하기 때문에 오류 검출이 불가능하다

송신 측 BCC와 수신 측의 계산된 결과가 같으면 오류가 없는 것이다.

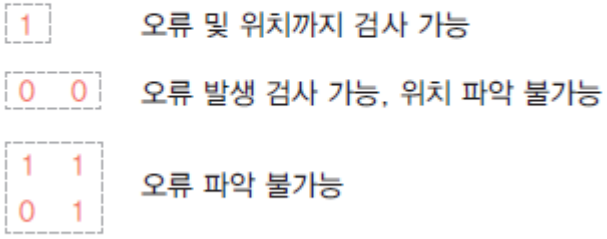

그림 3-25 블록 합 오류 검출 여부

02. 통신오류검출

3. 순환 중복 검사

- 순환 중복 검사(CRC, Cyclic Redundancy Check)는 정확하게 오류를 검출하려고 다항식 코드를 사용하는 방법이다.
- 오류가 없을 때는 계속 발생하지 않다가 오류가 발생하면 그 주위에 집중적으로 오류를 발생시키는 집단 오류를 검출하는 능력이 탁월하고, 구현이 단순하다.

▪ 다항식(Polynomial)

- CRC 발생기는 0과 1의 스트링 보다는 대수 다항식으로 표현하며, 하나의 다항식은 하나의 제수(Divisor)를 표현한다.

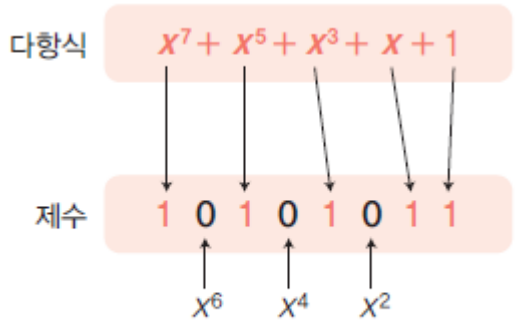

그림 3-26 다항식 표현의 예

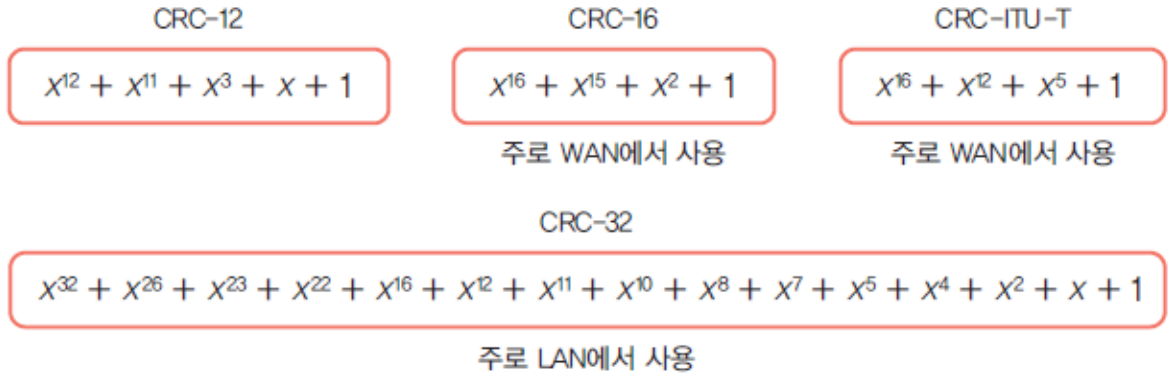

그림 3-27 생성 다항식

02. 통신오류검출

- 다항식을 이용한 순환 중복 검사의 오류 검출 과정을 살펴보자.
  1. 송신 측이 데이터를 전송하기 전에 송수신 측은 동일한 생성 다항식을 결정한다
  2. 송신 측에서는 K비트의 전송 데이터를 생성 다항식으로 나눈 n비트의 나머지 값을 구한다. K비트의 전송 데이터에 n비트의 나머지 값을 추가하여 K+n비트의 데이터를 수신 측으로 전송한다.
  3. 수신 측에서는 수신된 K+n비트의 데이터를 생성 다항식으로 나눈다. 나눈 나머지가 0이면 오류가 없는 것이고, 0이 아니면 오류가 발생한 것이다.

02. 통신오류검출

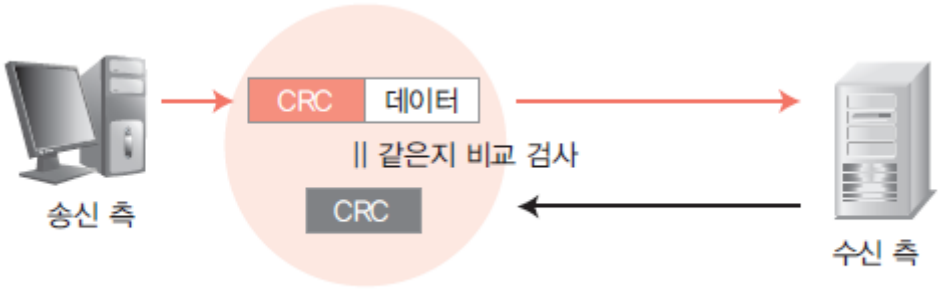

그림 3-28 순환 중복 검사 과정

① 송신 측

a. 데이터 전송

b. 오류 검출코드 계산

c. CRC 추가

② 수신 측

a. 데이터 수신

b. 오류 검출코드 계산

c. 수신된 CRC와 계산된 CRC 비교 검사

d. 동일하지 않으면 오류 검출 신호 발생

02. 통신오류검출

- CRC는 패리티 검사처럼 데이터마다 추가 비트를 붙이지 않아도 되지만, 프레임의 실제 내용으로 계산되는 프레임 검사 순서Frame Check Sequence, FCS를 프레임 끝에 추가해서 전송해야 한다.

- 순환 중복 검사 과정을 좀 더 자세히 살펴보자.
  - 전송 데이터 다항식 P(x) = 10101101($x^7+x^5+x^3+x^2+1$)
  - 생성 다항식 G(x) = 11101($x^4+x^3+x^2+1$)
  - FCS 비트 수는 생성 다항식(제수)보다 1비트 작은 4비트이다.

✓ **1단계**

- P′(x): 전송 데이터 다항식 P(x)를 FCS의 비트 수만큼 왼쪽 시프트 shift한다(왼쪽 시프트한 후 빈자리에는 0이 들어간다).
- P(x) = 10101101을 FCS 비트 수만큼 왼쪽 시프트
- P′(x) = 101011010000

02. 통신오류검출

✓ 2단계

P′(x)를 G(x)로 나누어 나머지를 계산한다.
나눗셈 연산은 캐리(overflow, borrow)를 무시하고 XOR(mod-2)을 수행한다.

XOR 연산: 서로 다른 값일 때는 1,
같은 값일 때는 0을 반환한다.

송신 측 과정

```
11101 | 10101101 0000
        11101
         10001
         11101
          11000
          11101
           10110
           11101
            10110
            11101
             10110
             11101
              10110
              11101
               1011
```

11101: 생성 다항식
0000: FCS 비트 수
1011: 나머지

✓ 3단계

P″(x) = 101011011011

✓ 4단계

P″(x)를 수신 측으로 전송한다.

✓ 5단계

수신 측에서는 전송받은 P″(x)를 생성 다항식으로 나눈다.
나머지가 0이면 오류가 없는 것이고, 0이 아니면 오류가 발생한 것이다.

02. 통신오류검출

수신 측 과정

```
11101 | 10101101 1011
        11101
         10001
         11101
          11000
          11101
           10111
           11101
            10100
            11101
             10011
             11101
              11101
              11101
                  0
```

11101: 생성 다항식
1011: 송신 측 나머지 값
0: 나머지