

컴퓨터 네트워크

제 11 장 데이터 링크 프로토콜

목 차

11.1 회선제어방식

11.2 비동기 데이터 링크 프로토콜

11.3 동기 데이터 링크 프로토콜

11.1 회선제어방식(1/11)

종류

- ✓ 회선경쟁선택 시스템(contention-based system)
 - ENQ/ACK : 대등-대-대등(peer-to-peer) 통신
- ✓ 폴링/셀렉션(Polling/Selection)
 - Poll/Select : 주국-종국(primary-secondary) 통신

결정요소

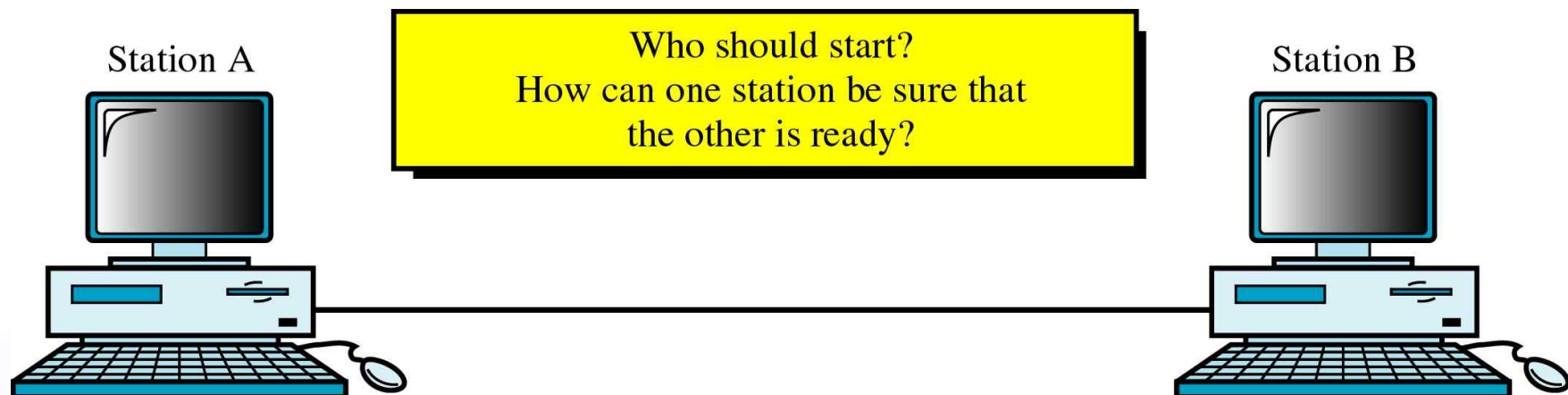
- ✓ 통신 회선의 배치 구조
- ✓ 트래픽 레벨
- ✓ 응답 시간

11.1 회선제어방식(2/11)

회선경쟁선택(Contention)

✓ 특징

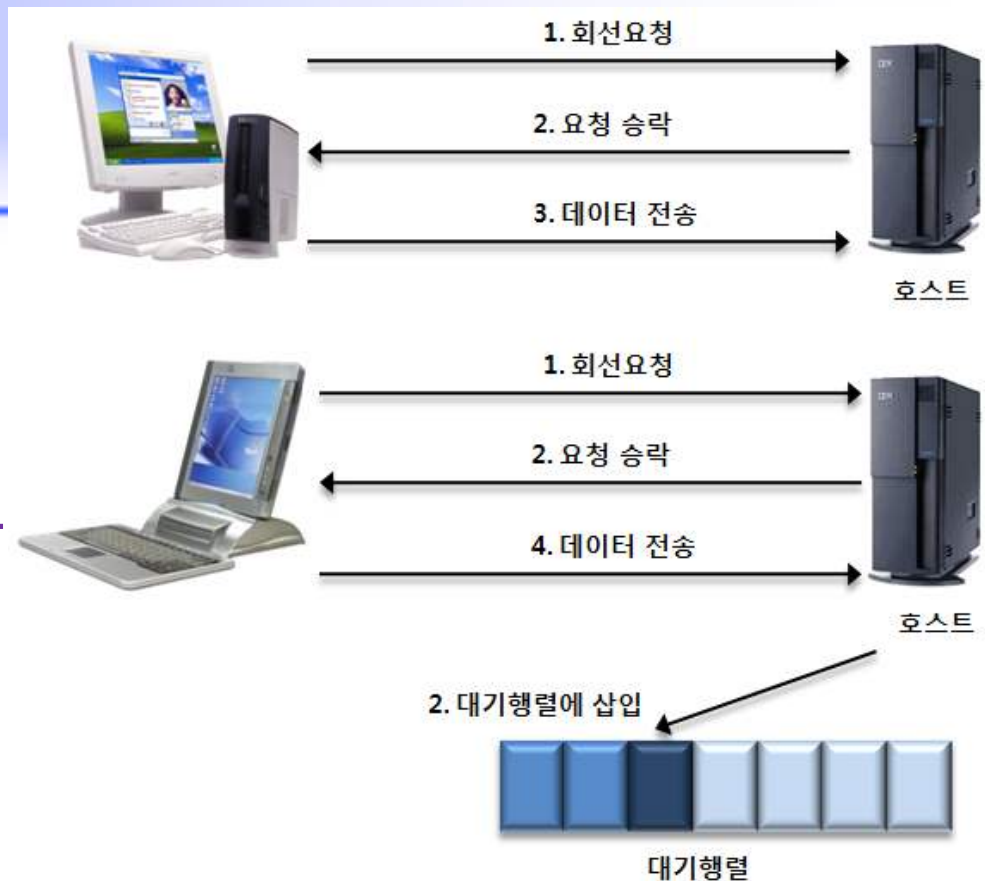
- 터미널들은 회선의 제어를 위하여 서로 경쟁
- 터미널이 회선에 대한 제어권을 획득하게 되면 회선은 해당 터미널에 의해서 점유
- 점대점(point-to-point)방식에서 주로 사용
- 일반 전화회선과 유사한 방식
- ENQ/ACK(Enquiry/acknowledgment)



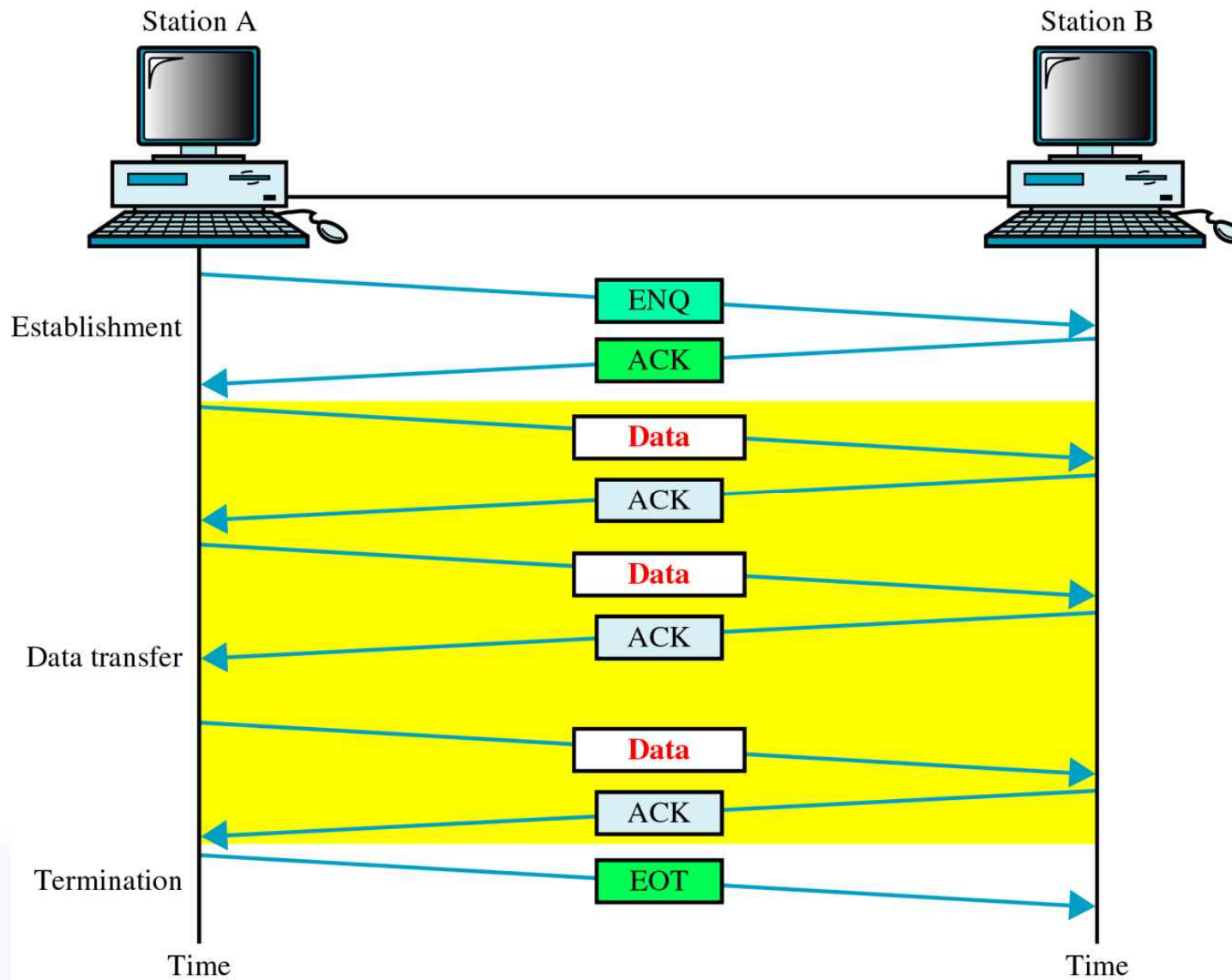
11.1 회선제어방식(3/11)

회선 제어권의 획득과정

- ✓ 터미널이 전송할 데이터가 있으면 회선 요청
- ✓ 회선이 획득되면 터미널은 데이터를 전송
- ✓ 모든 회선이 점유상태에 있으면 터미널은 대기상태로 전이
- ✓ 대기 상태로 전이되는 터미널은 호스트컴퓨터의 통신제어프로그램이 관리하며 일반적으로 대기행렬(queue)에 추가
- ✓ 대기행렬은 FCFS(First-Come First-Serve) 방식이나 필요에 따른 우선순위 방식에 의해 터미널을 관리
- ✓ 회선이 점유상태에서 풀리게 되면 대기행렬에 있던 터미널 중에 관리 방식에 따라 다음 터미널이 해당 회선을 점유
- ✓ 터미널은 회선을 점유하게 되면 데이터 전송



ENQ/ACK(Enquiry/acknowledgment)



11.1 회선제어방식(5/11)

✓ 장점

- 회선제어 형태 중 가장 간단한 방식
- 위선통신과 같은 전파지연시간(propagation delay time)이 큰 통신망에서 효율적

✓ 단점

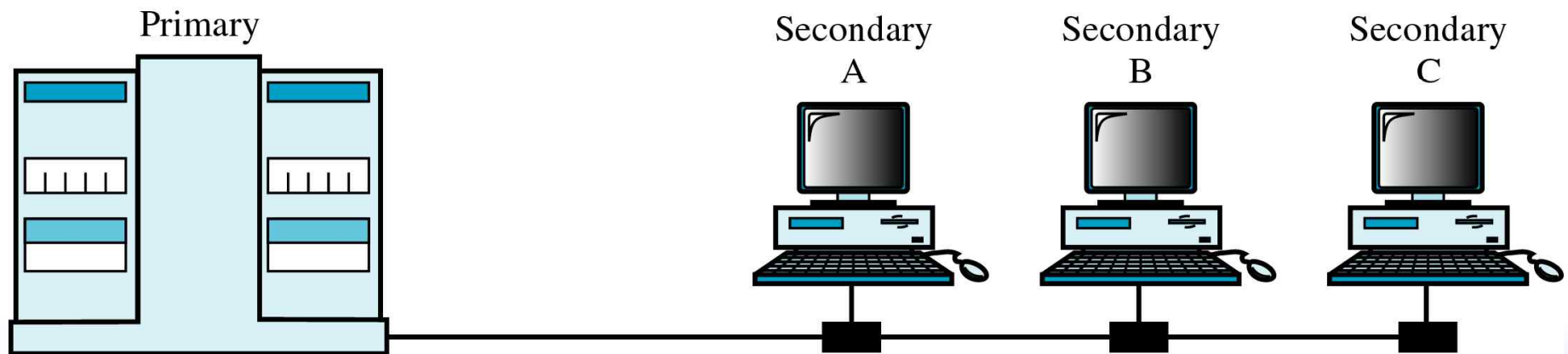
- 회선을 점유한 터미널이 실제로 데이터를 전송하고 있지 않아도 오랫동안 회선을 점유
=> **트래픽이 많은 네트워크에서는 비효율적**
- 멀티 포인트 회선 네트워크에서 두개의 터미널이 동시에 회선 점유를 요청하는 경우 문제가 발생
=> 회선경쟁선택 방식은 주로 **점대점(point-to-point) 네트워크**에서 사용.

11.1 회선제어방식(6/11)

폴링/셀렉션

- ✓ 호스트와 터미널은 주(主) · 종(從)관계
 - ✓ 폴링 : 전송할 데이터가 있는지 묻는 것
 - ✓ 셀렉션 : 전송할 데이터가 있을 때 수신준비를 하라는 것
- ※ 위 두 동작의 주체는 호스트가 되며, 종은 터미널임

Who has the right to the channel?



11.1 회선제어방식(7/11)

✓ 특징

- 네트워크는 **멀티포인트(Multi-point)** 형식의 구조
- 터미널간의 충돌은 호스트가 관리
- 호스트는 **한 시점에 하나의 터미널과 통신**하게 되어있고, **터미널은 호스트의 폴링/셀렉션에 의해서만 동작**
- 호스트는 터미널에게 가변적인 우선순위를 부여
- 우선순위는 주소의 순서와 빈도수에 의해 결정
- 관리자는 이 값을 임의로 조정

11.1 회선제어방식(8/11)

폴링(Polling)

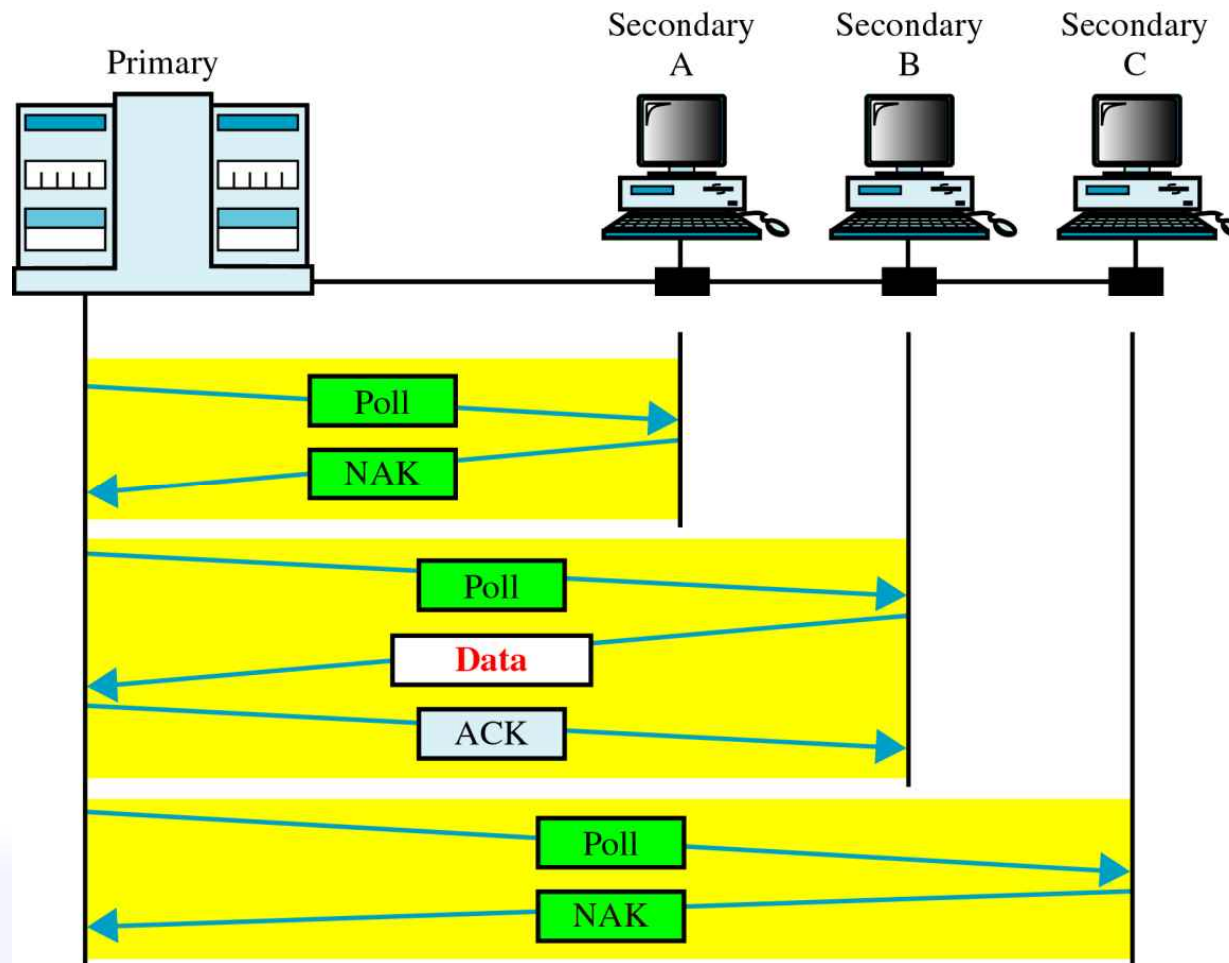
- ✓ 호스트는 터미널에게 전송할 데이터가 있는지 폴링을 수행
- ✓ 대상 터미널이 데이터가 없는 경우는 호스트에게 없음을 알리는 데이터 송신
- ✓ 이를 수신한 호스트는 다음 터미널에게 폴링을 수행
- ✓ 터미널이 전송할 데이터가 있는 경우, 호스트는 해당 터미널에게 점유권을 부여
- ✓ 점유권을 받은 터미널은 송신하고자 하는 데이터를 전송



회선 제어(계속)

폴링(Polling)

✓ 주국이 종국에게 데이터 전송을 요구할 때



선택(Selection)

- ✓ 셀렉션은 호스트가 터미널에게 보낼 데이터가 있는 경우에 수행
- ✓ 호스트는 전송할 데이터의 목적지로서 터미널을 선택
- ✓ 수신준비 하라는 데이터를 전달
- ✓ 터미널은 이에 수신준비가 되었다는 응답
- ✓ 호스트는 이 응답에 이어 실제의 데이터를 전송하기 시작



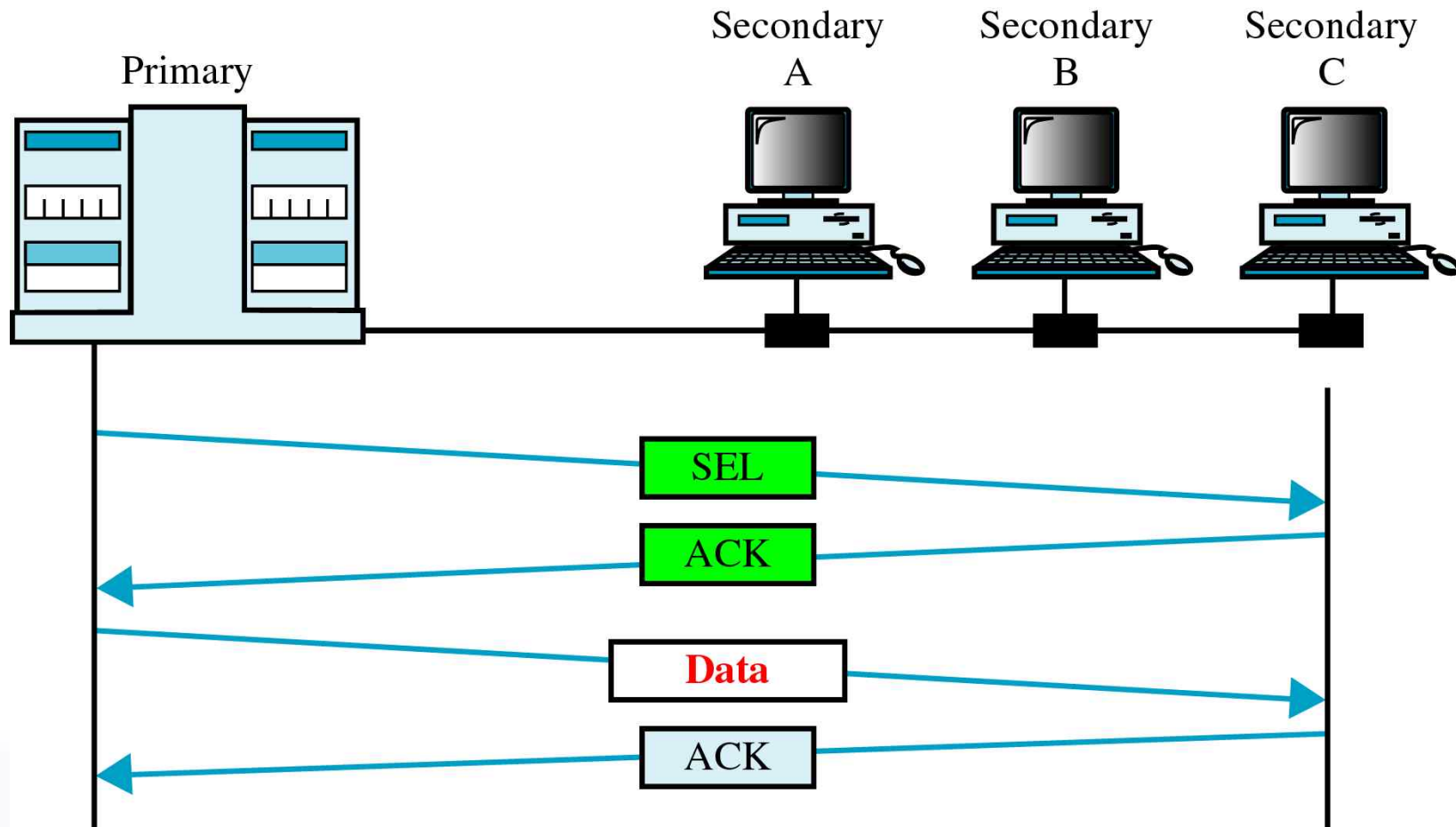
보낼테니까 준비해라



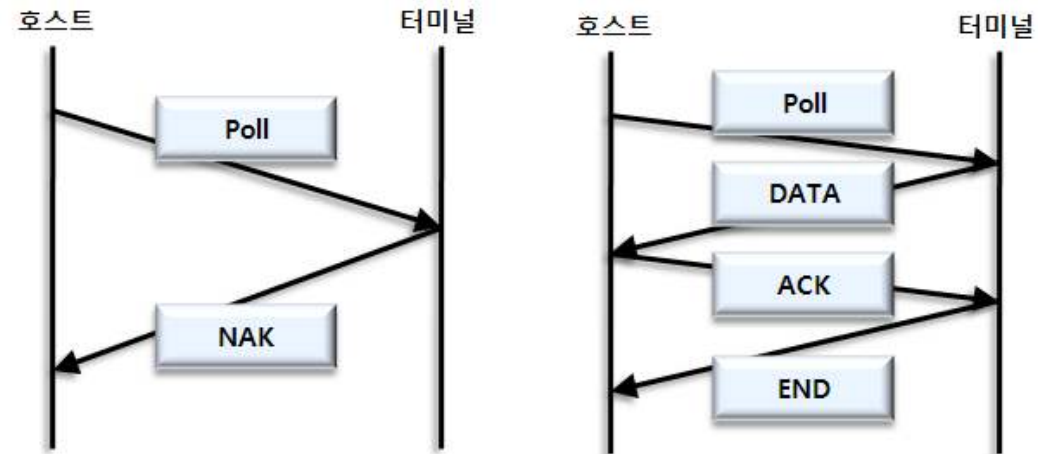
회선 제어(계속)

선택(Selection)

✓ 주국이 데이터를 전송할 때

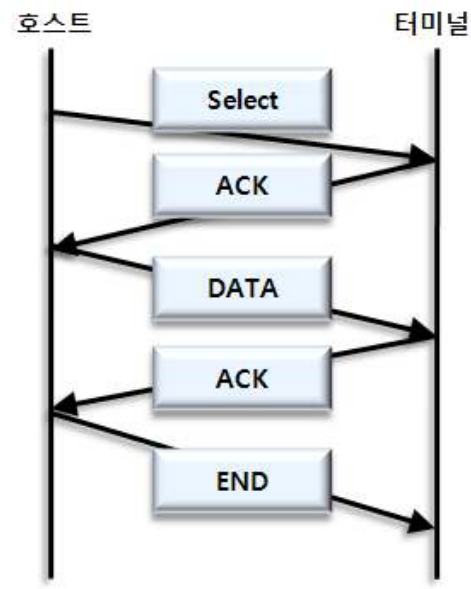


11.1 회선제어방식(9/11)



전송할 데이터가 없는 경우

데이터를 전송하는 경우



11.1 회선제어방식(10/11)

장점

- ✓ 호스트가 터미널 선택함으로써 터미널간의 충돌 없음
- ✓ 하나의 회선으로 여러 개의 터미널이 회선을 공유 함으로 회선비용 절감

단점

- ✓ 터미널에게 폴링을 수행하는 동안에 상당한 제어 오버헤드가 수반
- ✓ 터미널이 원하는 시간에 메시지를 보낼 수 없고 오직 셀렉션을 받은 다음에만 전송이 가능
- ✓ 즉각적이고 지속적인 연결을 원하는 응용프로그램에서는 사용이 곤란
- ✓ 위성 망과 같이 전파지연시간이 큰 네트워크에는 폴링/셀렉션에 지연되는 시간이 크기 때문에 비효율적

11.1 회선제어방식(11/11)

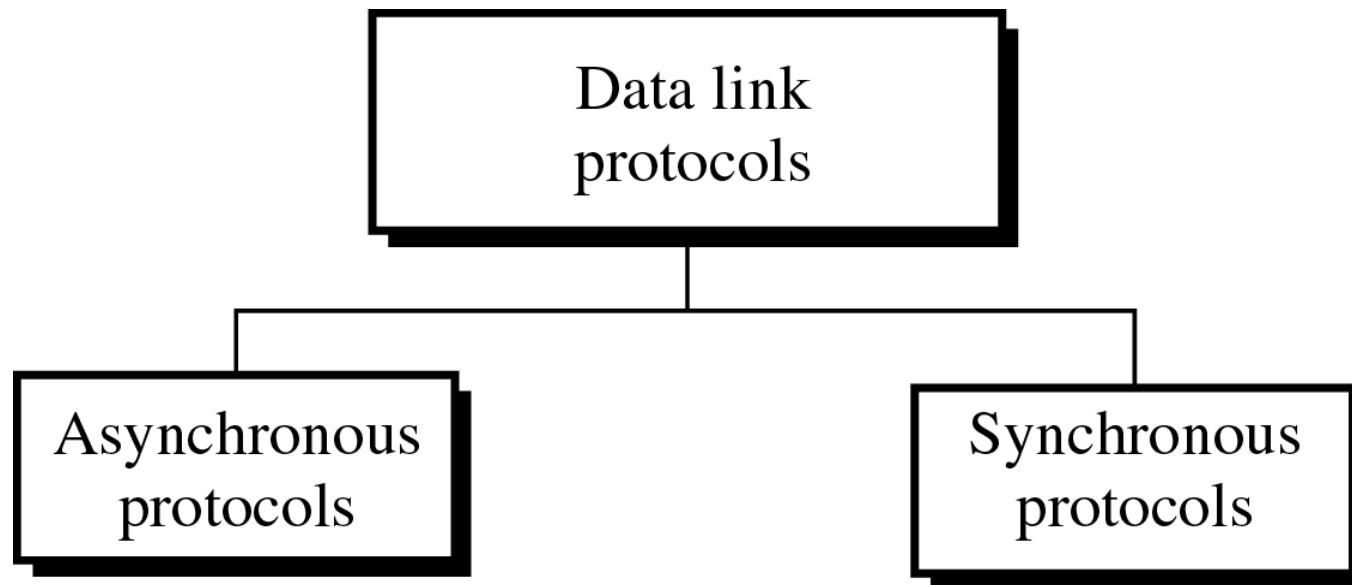
✓ 회선경쟁방식과 폴링/셀렉션의 비교

비 고	회선경쟁방식	폴링/셀렉션
회선할당 방식	터미널이 호스트에게 요청	호스트가 폴링으로 할당
장 점	<ul style="list-style-type: none"> - 설계 단순 - 전화망과 유사 - 전파지연시간이 긴 경우 유리 	<ul style="list-style-type: none"> - 터미널간 충돌 없음 - 회선 공유, 회선비용 절감
단 점	<ul style="list-style-type: none"> - 휴지(idle)한 경우에도 회선 점유 가능성 존재 - 두개의 터미널이 동시에 회선 요청시 문제 발생 	<ul style="list-style-type: none"> - 폴링/셀렉션 시간동안 제어 오버헤드가 큼 - 터미널이 원할때 회선 할당 어려움 - 즉각적이고 지속적인 연결 어려움 - 전파지연시간이 긴 경우 불리

데이터 링크 프로토콜

데이터 링크 프로토콜

- ✓ 데이터 링크 층 구현에 사용된 규약



- ✓ 비동기 프로토콜

- 비트 스트림에 있는 각 문자를 독립적으로 다룸

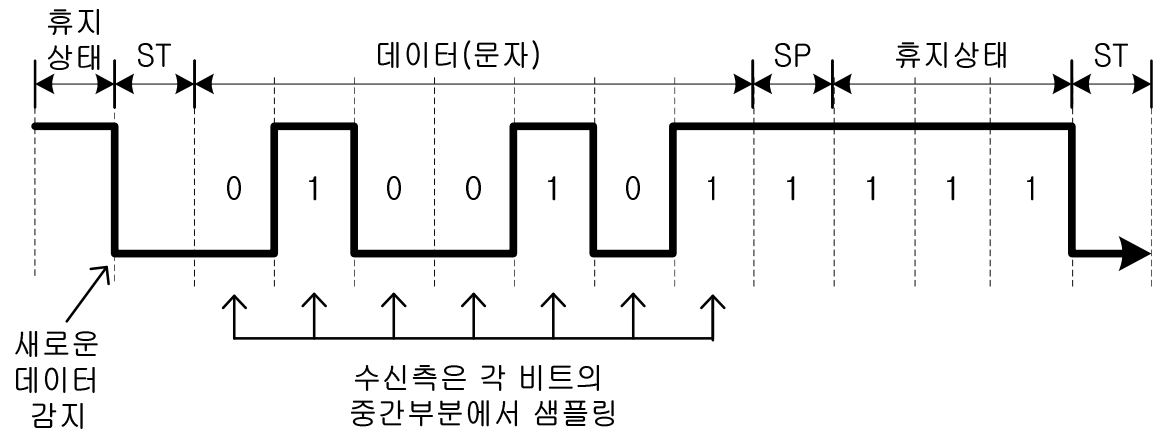
- ✓ 동기 프로토콜

- 전체 비트 스트림을 같은 크기의 문자들로 나누어 처리

11.2 비동기 데이터 링크 프로토콜

정의

- ✓ 통신이 송신측과 수신측 사이의 **기계적인 클록(clock)의 동기화 없이 이루어짐**
- ✓ 송신측에서 프레임의 **시작과 끝을 알리는** 수신측과 약속된 비트 삽입

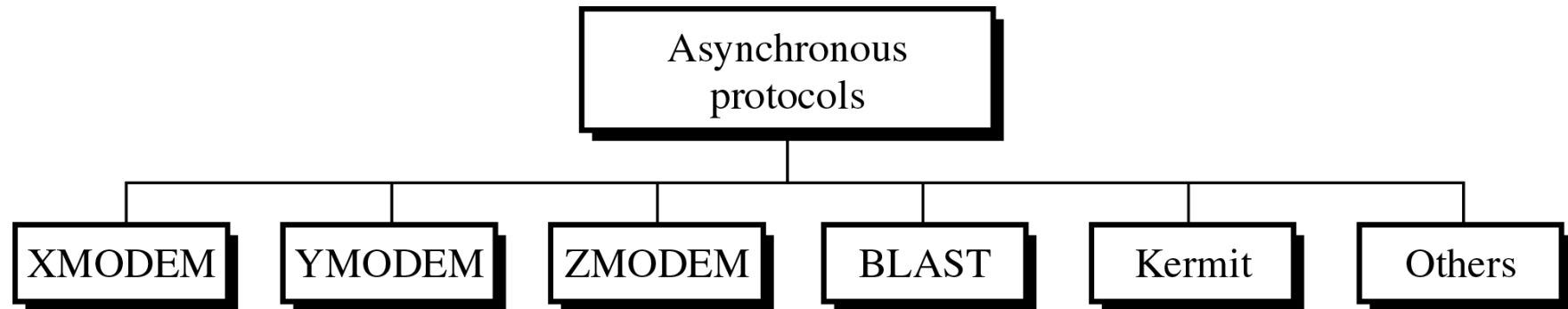


특징

- ✓ 송수신측간의 **클록의 동기화가 불필요**
- ✓ 프로토콜의 설계 단순
- ✓ 동기식에 비해 **구현상의 비용절감 효과**
- ✓ 전송 데이터의 시작과 끝에 **시작 비트 및 정지 비트**를 추가함으로써 데이터임을 구분

11.2 비동기 데이터 링크 프로토콜

주로 모뎀에서 사용하며, 시작과 정지 비트, 문자 사이에 가변 길이
갭을 가짐

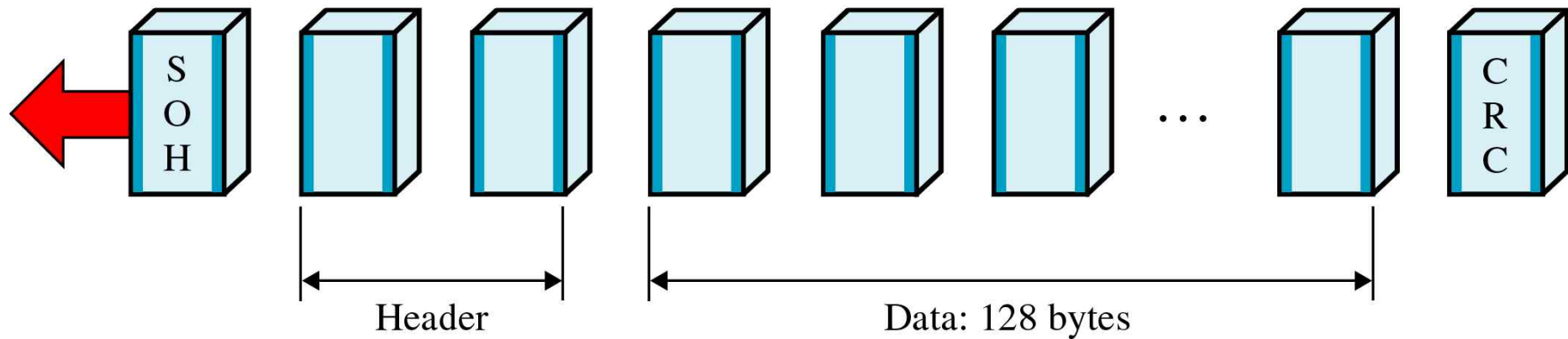


11.2 비동기 데이터 링크 프로토콜

XMODEM 프레임

- ✓ 1978년에 Ward Christensen에 의해 개발되어 사실상의 표준
- ✓ Xmodem 프로토콜을 사용하는데 동의한 모뎀은 데이터를 128 바이트 크기의 블록으로 보냄
- ✓ 블록이 성공적으로 수신되면, 포지티브 응답 (ACK), 에러가 감지되면, 네거티브 응답 (NAK)이 되돌려지며, 그 블록은 재송신
- ✓ SOH(헤더 시작) : 1 바이트
- ✓ 헤더 : 2 바이트(순서 번호, 순서번호 유효성 검사)
- ✓ 데이터(Binary, ASCII, Boolean, Text 등) : 128 바이트
- ✓ CRC : 데이터 필드 오류 검사
- ✓ 모뎀용 에러교정 프로토콜

Each character contains start and stop bits (dark portion of the box). Characters are separated from each other by gaps. The header consists of two bytes: sequence number and its one's complement.



11.2 비동기 데이터 링크 프로토콜

YMODEM

- ✓ XMODEM과 유사한 프로토콜
- ✓ 데이터 단위 : 1024 바이트
- ✓ 2개의 CAN(취소문자)은 전송을 정지하기 위해 송신된다
- ✓ ITU-T CRC-16은 오류 검사용
- ✓ 다중 파일을 동시에 전송 가능

11.2 비동기 데이터 링크 프로토콜

ZMODEM

- ✓ XMODEM과 YMODEM의 특징을 조합한 새로운 프로토콜
- ✓ CRC-32 에러 검출방식을 채택하여 에러검출이 뛰어남
- ✓ 처리속도가 빠름
- ✓ 송수신중에 끊긴 파일의 이어 받기 가능
- ✓ 선로 상태에 따라 패킷 크기를 자동 변경하여 통신의 정확성 및 안정성 최상

BLAST(Blocked Asynchronous Transmission)

- ✓ 슬라이딩 윈도우 흐름 제어를 이용한 전이중 방식

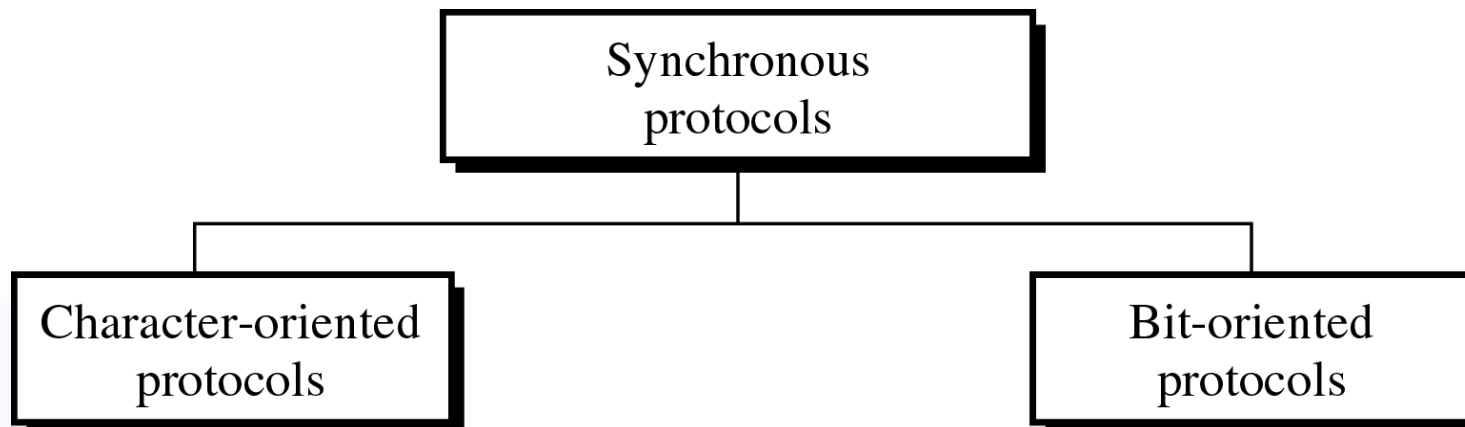
Kermit

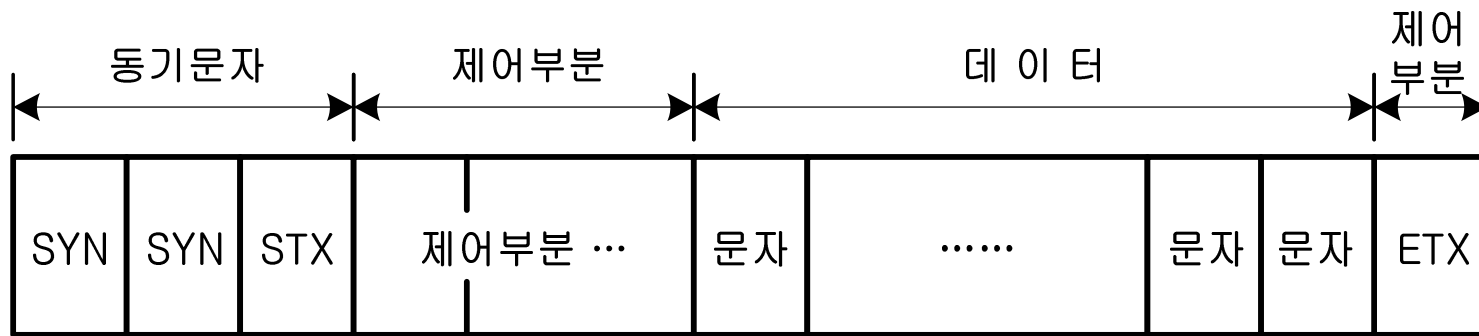
- ✓ 콜롬비아 대학에서 개발
- ✓ 가장 많이 사용되고 있는 비동기 프로토콜

11.3 동기 데이터링크 프로토콜

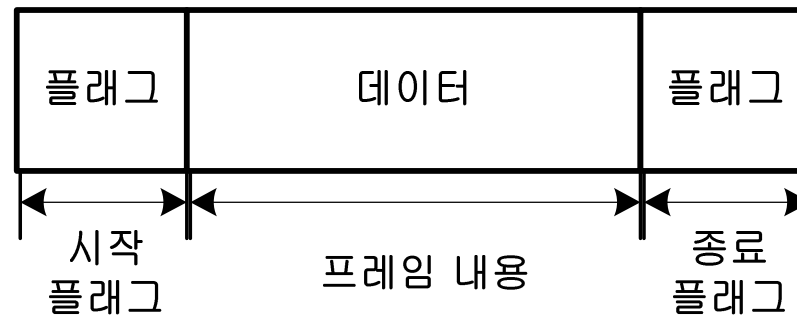
정의

- ✓ 비동기 전송에 비해 **속도 면에서 우수**
- ✓ LAN, MAN, WAN에서 사용
- ✓ 프로토콜
 - **문자중심 데이터 링크 프로토콜**
 - 프레임 또는 패킷을 **문자의 연속**으로 해석
 - **비트중심 데이터 링크 프로토콜**
 - 프레임 또는 패킷을 **비트의 연속**으로 해석





(a) 문자 전송 방식



(b) 비트 전송 방식

11.3 동기 데이터링크 프로토콜

문자중심 데이터 링크 프로토콜

✓ BSC

- 다중점 접속을 지원하는 범용 데이터 링크 제어 방식
- 반이중 프로토콜
- 폴링/셀렉션 기반
- 비트-중심 프로토콜보다 비효율적이므로 **오늘날 거의 사용되지 않음**
- 기능
 - **전송블록 형식화** : 블록들의 크기를 정하고 메시지들을 블록으로 나누기 위해 사용 (SYN, SOH, STX, ETB, ITB, ETX등의 제어문자)
 - 스테이션 간의 대화 : 데이터의 반이중 교환을 제어하기 위해 사용(ENQ)
 - **투명 모드제어** : 임의의 데이터에 제어문자들의 비트 패턴을 포함하여 전송할 때 구분하기 위해 사용(DLE)
- BSC(Binary synchronous communication)

제어 문자			공백 문자			구두점			숫자			알파벳		
10진	16진	문자	10진	16진	문자	10진	16진	문자	10진	16진	문자	10진	16진	문자
0	0x00	NUL	32	0x20	SP	64	0x40	@	96	0x60				
1	0x01	SOH	33	0x21	!	65	0x41	A	97	0x61	a			
2	0x02	STX	34	0x22	"	66	0x42	B	98	0x62	b			
3	0x03	ETX	35	0x23	#	67	0x43	C	99	0x63	c			
4	0x04	EOT	36	0x24	\$	68	0x44	D	100	0x64	d			
5	0x05	ENQ	37	0x25	%	69	0x45	E	101	0x65	e			
6	0x06	ACK	38	0x26	&	70	0x46	F	102	0x66	f			
7	0x07	BEL	39	0x27	'	71	0x47	G	103	0x67	g			
8	0x08	BS	40	0x28	(72	0x48	H	104	0x68	h			
9	0x09	HT	41	0x29)	73	0x49	I	105	0x69	i			
10	0x0A	LF	42	0x2A	*	74	0x4A	J	106	0x6A	j			
11	0x0B	VT	43	0x2B	+	75	0x4B	K	107	0x6B	k			
12	0x0C	FF	44	0x2C	,	76	0x4C	L	108	0x6C	l			
13	0x0D	CR	45	0x2D	-	77	0x4D	M	109	0x6D	m			
14	0x0E	SO	46	0x2E	.	78	0x4E	N	110	0x6E	n			
15	0x0F	SI	47	0x2F	/	79	0x4F	O	111	0x6F	o			
16	0x10	DLE	48	0x30	0	80	0x50	P	112	0x70	p			
17	0x11	DC1	49	0x31	1	81	0x51	Q	113	0x71	q			
18	0x12	DC2	50	0x32	2	82	0x52	R	114	0x72	r			
19	0x13	DC3	51	0x33	3	83	0x53	S	115	0x73	s			
20	0x14	DC4	52	0x34	4	84	0x54	T	116	0x74	t			
21	0x15	NAK	53	0x35	5	85	0x55	U	117	0x75	u			
22	0x16	SYN	54	0x36	6	86	0x56	V	118	0x76	v			
23	0x17	ETB	55	0x37	7	87	0x57	W	119	0x77	w			
24	0x18	CAN	56	0x38	8	88	0x58	X	120	0x78	x			
25	0x19	EM	57	0x39	9	89	0x59	Y	121	0x79	y			
26	0x1A	SUB	58	0x3A	:	90	0x5A	Z	122	0x7A	z			
27	0x1B	ESC	59	0x3B	;	91	0x5B	[123	0x7B	{			
28	0x1C	FS	60	0x3C	<	92	0x5C	\	124	0x7C				
29	0x1D	GS	61	0x3D	=	93	0x5D]	125	0x7D	}			
30	0x1E	RS	62	0x3E	>	94	0x5E	^	126	0x7E	~			
31	0x1F	US	63	0x3F	?	95	0x5F	_	127	0x7F	DEL			

11.3 동기 데이터링크 프로토콜

BSC(Binary Synchronous Communication)

- ✓ IBM에 의해 1964년에 설계
- ✓ 점-대-점과 다중점 구성에 사용 가능
- ✓ stop-and-wait ARQ 흐름 제어와 오류 수정을 이용한 반이중 전송을 지원
- ✓ 전이중 전송 또는 슬라이딩 윈도우 프로토콜은 지원하지 않음

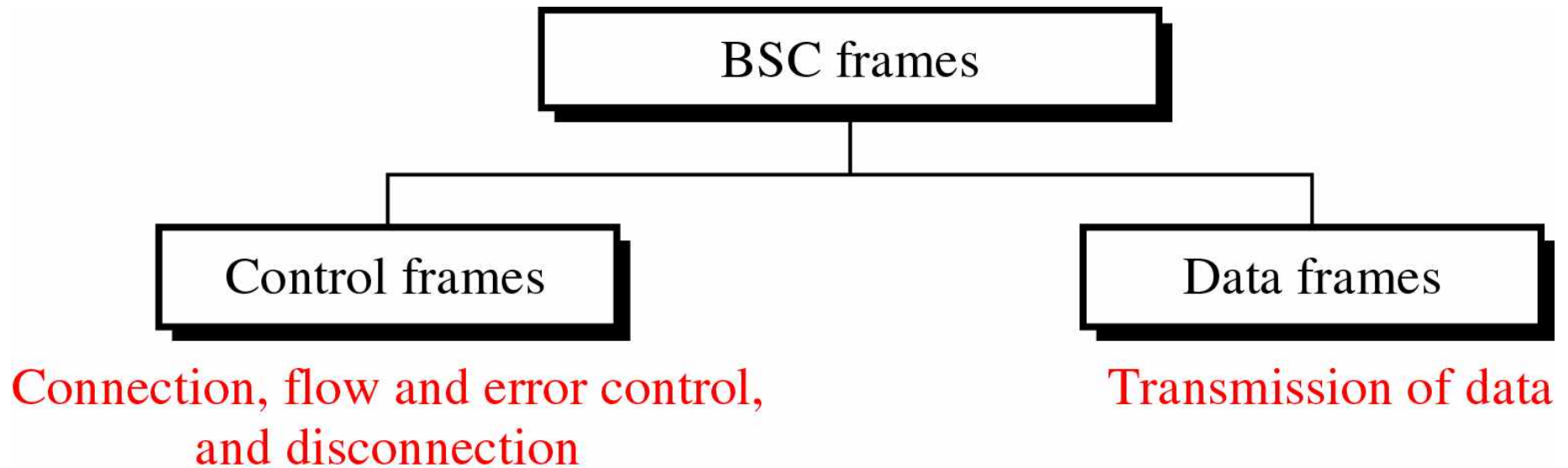
11.3 동기 데이터링크 프로토콜

BSC 프로토콜 제어문자

Character	ASCII Code	Function
ACK 0	DLE and 0	Good even frame received or ready to receive
ACK 1	DLE and 1	Good odd frame received
DLE	DLE	Data transparency maker
ENQ	ENQ	Request for a response
EOT	EOT	Sender terminating
ETB	ETB	End of transmission block; ACK required
ETX	ETX	End of text in a message
ITB	US	End of intermediate block in a multiblock transmission
NAK	NAK	Bad frame received nothing to send
NUL	NULL	Filler character
RVI	DLE and <	Urgent message from receiver
SOH	SOH	Header information beings
STX	STX	Text beings
SYN	SYN	Alerts receiver to incoming frame
TTD	STX and ENQ	Sender is pausing but not relinquishing the line
WACK	DLE and ;	Good frame received but not ready to receive more

11.3 동기 데이터링크 프로토콜

BSC 프레임

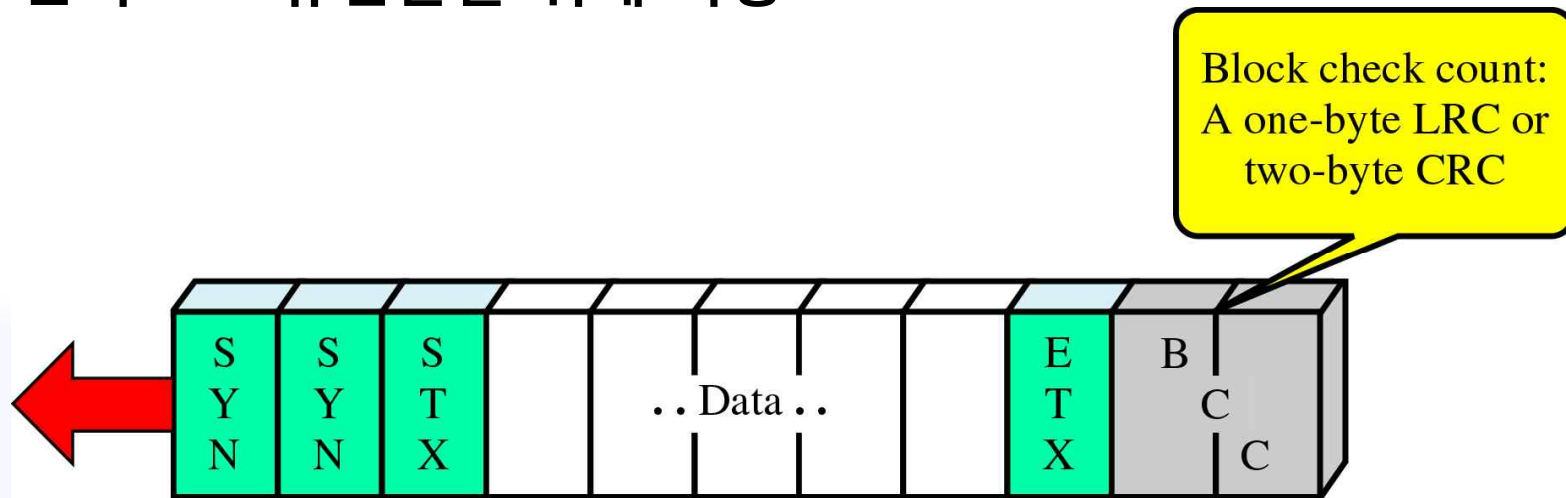


11.3 동기 데이터링크 프로토콜

✓ 데이터 프레임

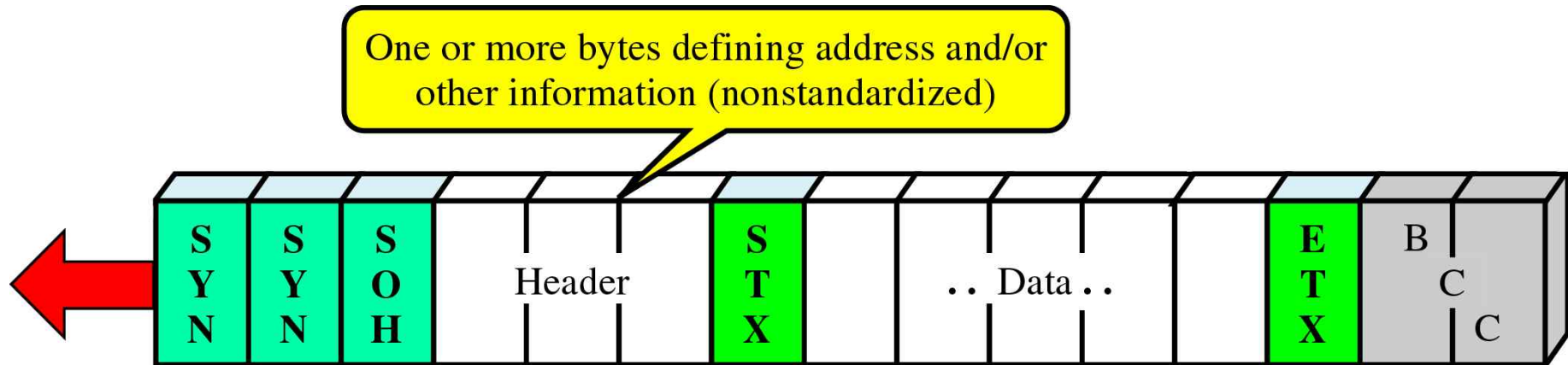
➤ 기본데이터 프레임

- 동기문자 (SYN : SYNchronous) : 수신측에게 새로운 프레임의 도착을 알리고 송신측과의 타이밍을 맞추기 위해 수신 장치에 의해 사용되는 비트 패턴을 제공
- 시작문자 (STX : Start of TeXt) : 제어정보가 끝났으며 다음 바이트는 데이터라는 것을 의미
- 종료문자 (ETX : End of TeXt) : 문서의 끝을 나타내며 제어문자로의 전이를 의미
- 블록검사계산 (BCC : Block Check Count) : 하나 이상의 문자로 오류검출을 위해 사용



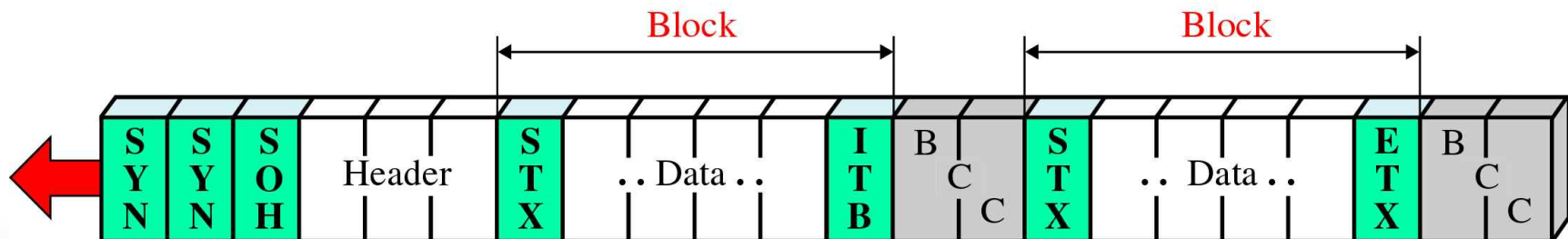
11.3 동기 데이터링크 프로토콜

헤더 프레임



다중블럭 프레임

- ✓ 메시지 텍스트를 여러 개의 블록으로 나누어 전송

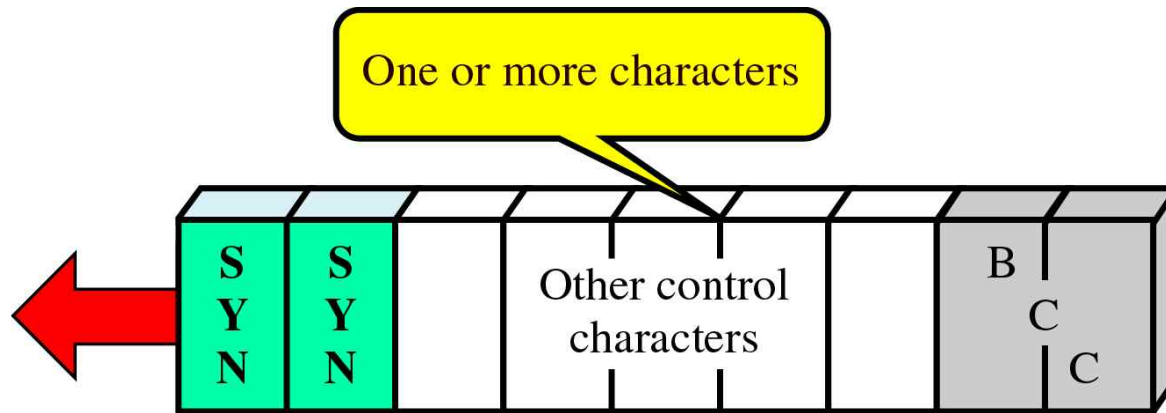


- ✓ ITB : End of intermediate block in a multiblock transmission

11.3 동기 데이터링크 프로토콜

제어 프레임

- ✓ 명령어 전송에 사용

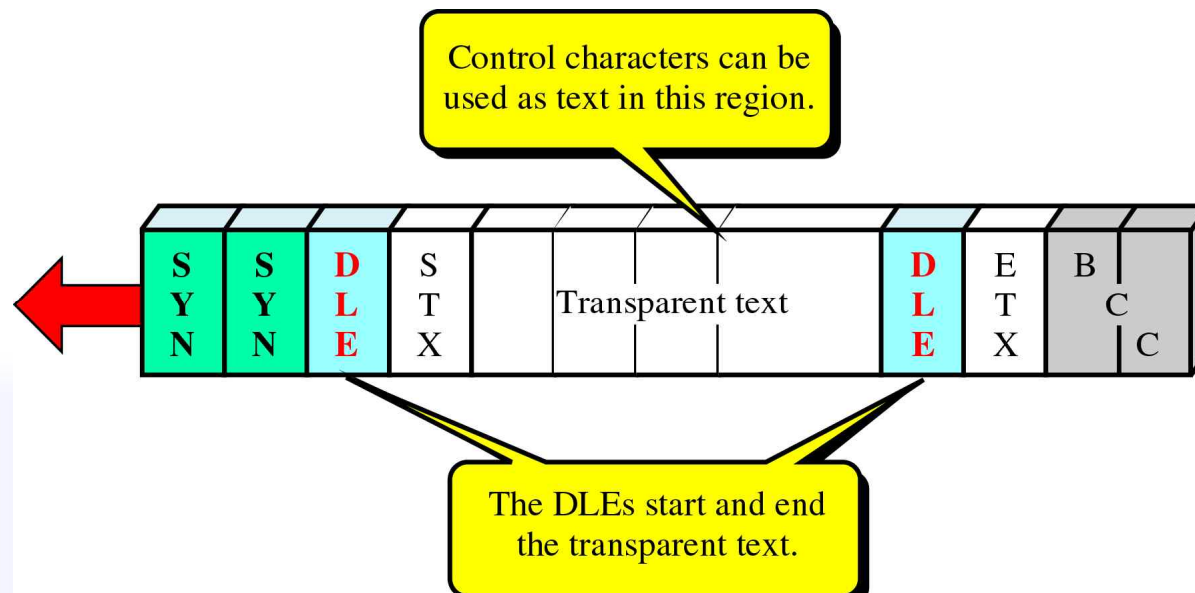


- ✓ 3가지의 목적에 사용
 - 연결 확립(establishing connections)
 - 데이터 전송시 흐름 유지 및 오류 제어
 - 연결 해제(terminating connection)

11.3 동기 데이터링크 프로토콜

데이터 투명성 제어(DLE ; Data Link Escape)

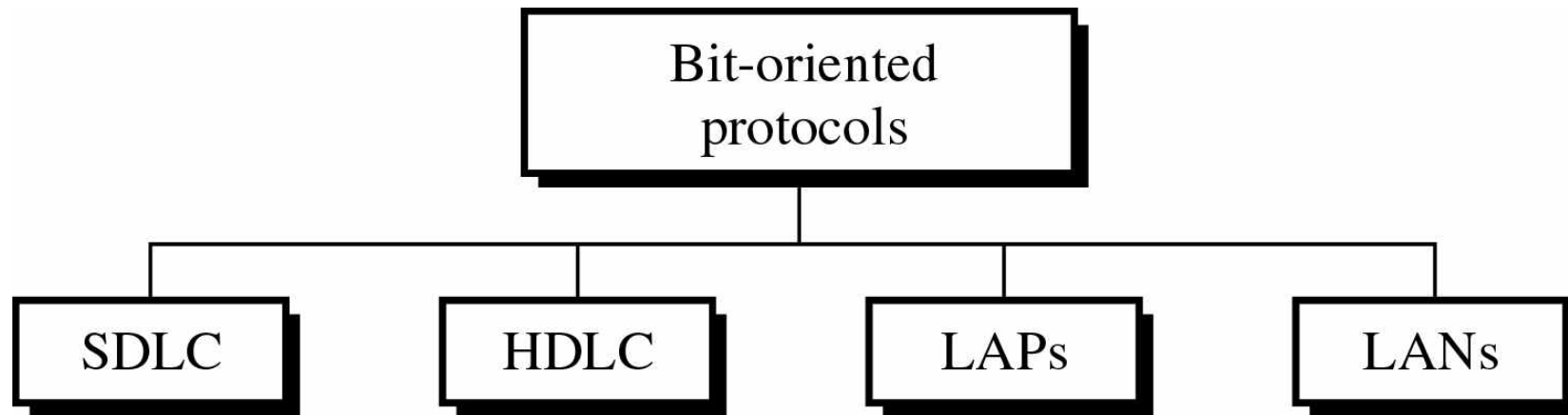
- ✓ 데이터에 들어 있는 제어문자를 실제 제어문자로 인식하지 못하도록 비트 스템핑(bit stuffing) 혹은 **바이트 스템핑(Byte stuffing)**
- ✓ **DLE문자를 적절한 위치에 채움**
- ✓ 텍스트 블록의 끝을 식별할 수 있는 방법이 필요
 - ETB, ETX 그리고 ITB에 대한 투명 모드에 대응
 - 데이터 필드 앞에 DLE ETB, DLE ETX 그리고 DLE ITB를 정의
- ✓ 송신측은 텍스트 블록에 포함된 임의의 DLE앞에 DLE를 삽입
- ✓ 수신측은 **첫번째 DLE는 무시하고 두 번째 DLE를 데이터로 취급**

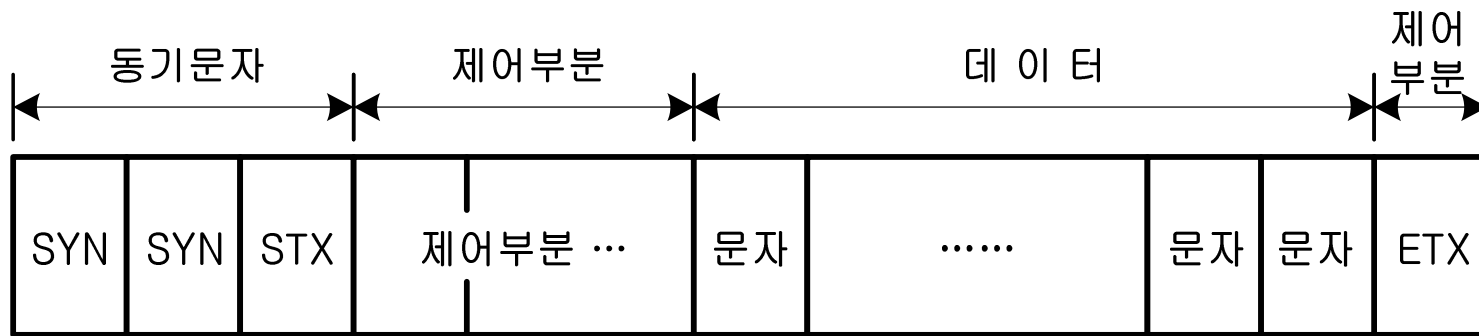


11.3 동기 데이터링크 프로토콜

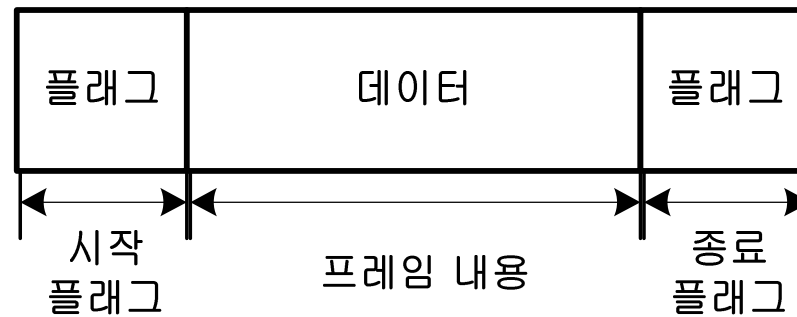
비트중심 데이터 링크 프로토콜

- ✓ 보다 짧은 프레임에 많은 정보를 전송
- ✓ 문자-중심 프로토콜에 있는 투명성 문제 해결





(a) 문자 전송 방식



(b) 비트 전송 방식

11.3 동기 데이터링크 프로토콜

SDLC(Synchronous Data Link Control)

- ✓ IBM에 의해 1975년에 개발
- ✓ WAN 구간을 위한 SNA(systems network architecture) 환경에서 데이터 링크 계층 프로토콜로서 중요한 역할을 수행

HDLC(High-Level Data Link Control)

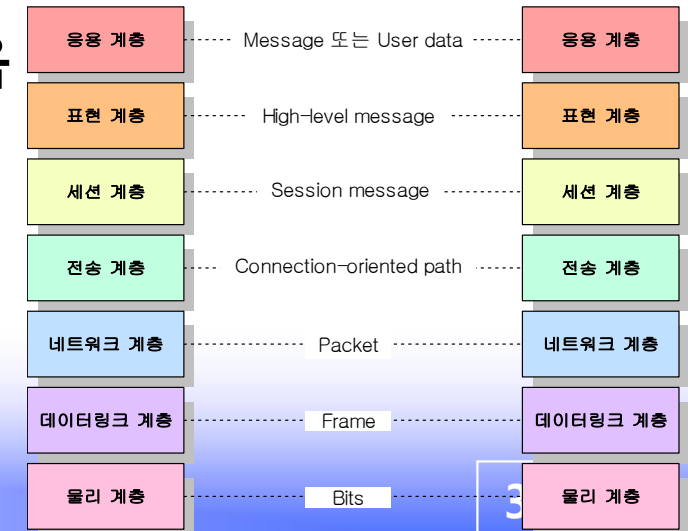
- ✓ ISO에 의해 1979년에 개발

LAPs(LAPS, LAPD, LAPM, LAPX, etc)

- ✓ ITU-T에 의해 1981년 이후로 개발되어 왔음

PPP, frame relay

- ✓ ITU-T와 ANSI에 의해 개발



11.3 동기 데이터링크 프로토콜

비트중심 데이터 링크 프로토콜

✓ HDLC (High-level Data Link Control)

- 모든 비트-중심 프로토콜은 ISO에서 규정한 상위-레벨 데이터 링크 제어와 연관됨
- HDLC은 점-대-점과 다중점 구성에서 반이중과 전이중 모드를 지원
- 모든 비트중심 데이터 링크 프로토콜의 모체
- 점대점 방식이나 다중점 방식의 통신 모두 지원
- HDLC는 지국의 형태, 구성, 응답 모드에 따라 구분

11.3 동기 데이터링크 프로토콜

스테이션의 형식(type)

✓ 주국(Primary Station)

- 데이터 회선을 제어하는 스테이션으로서 채널상의 보조국들에게 **명령 프레임**을 전송
- 그 후 보조국의 응답을 수신

✓ 보조국(Secondary Station)

- 주국으로부터 수신된 명령에 대해서 **응답**을 전송
- 주국과 관계하는 세션은 오직 한개만 유지
- 회선의 제어에 관한 일체의 책임은 없음

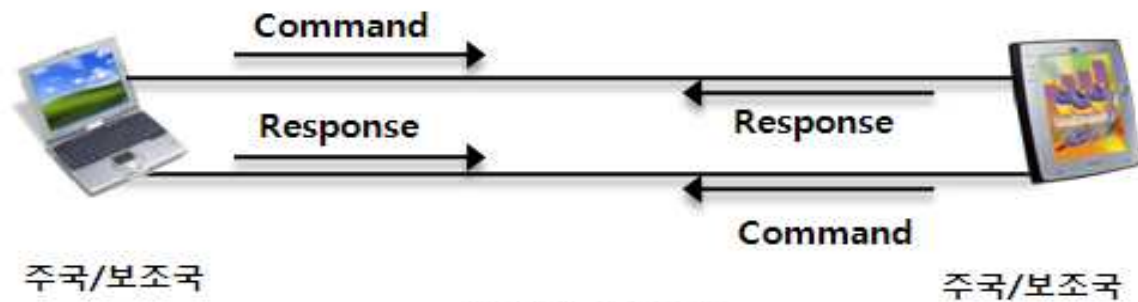
✓ 복합국(Combined Station)

- **명령과 응답**을 모두 발생할 수 있음
- 복합국은 서로 연결된 대등장치 중 하나
- 전송의 성격과 방향에 따라 주국 또는 보조국으로 수행

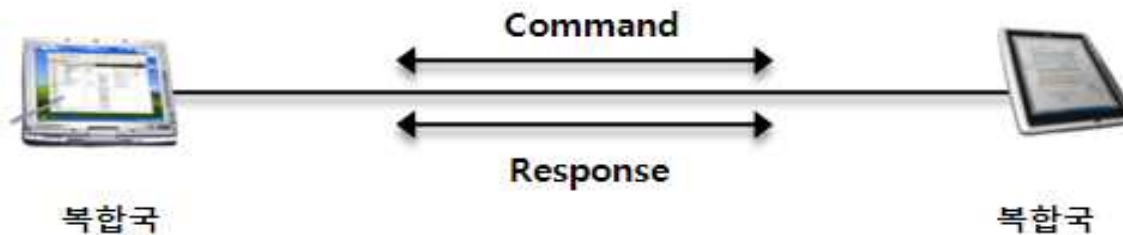
11.3 동기 데이터링크 프로토콜



(a) 비균형 구성방식



(b) 대칭 구성방식



(c) 평형 구성방식

11.3 동기 데이터링크 프로토콜

프레임

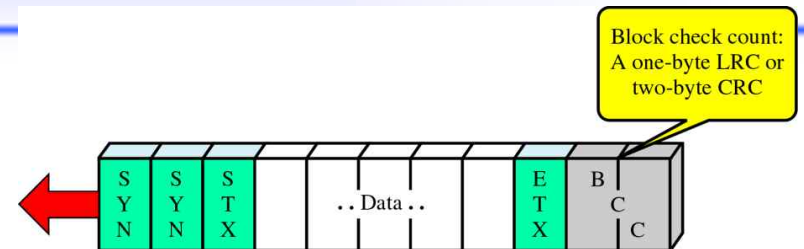
✓ 6개 필드로 구성

- 시작 플래그(beginning flag)
- 주소(address)
- 제어(control)
- 정보(information)
- FCS(Frame Check Sequence)
- 끝 플래그(ending flag)

11.3 동기 데이터링크 프로토콜(12/22)

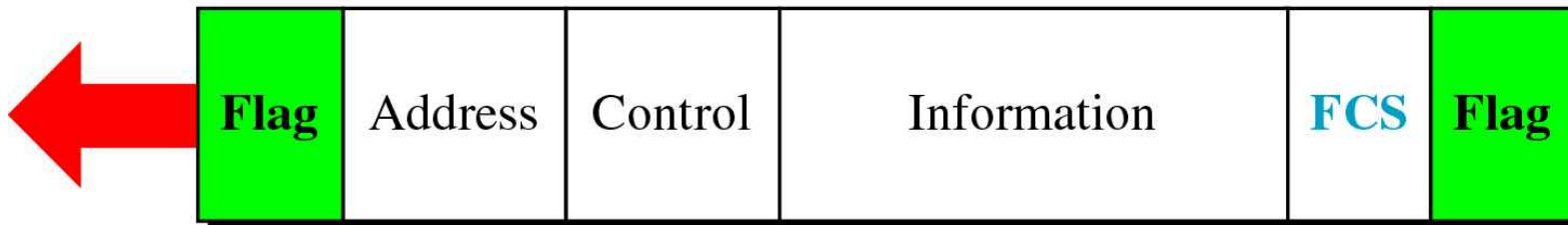
플래그 필드

- ✓ 수신자를 위한 동기 패턴으로 제공
- ✓ 01111110의 8개의 비트로 구성
- ✓ 문자중심 데이터 링크 프로토콜인 BSC의 SYN과 같은 역할
- ✓ 제어용 문자이기 때문에 투명성 보장 필요



The flag is 8 bits of a fixed pattern.
It is made of 6 ones enclosed in 2 zeros.
There is 1 flag at the beginning and 1 at the end of the frame. The ending flag of 1 frame can be used as the beginning flag of the next frame.

01111110



11.3 동기 데이터링크 프로토콜(13/22)

✓ 데이터 투명성

➤ 정의

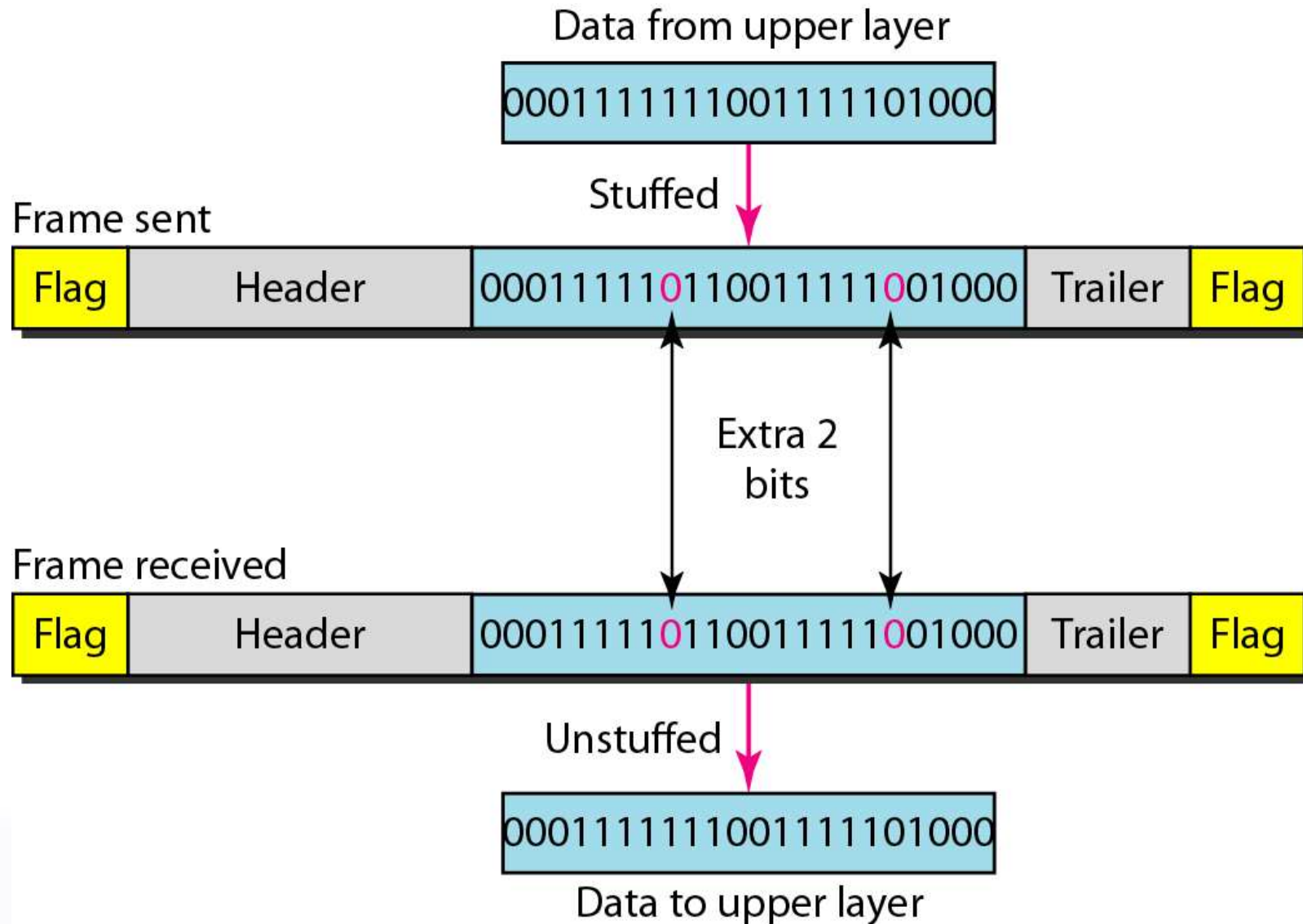
- 데이터 내에 제어비트와 동일한 패턴이 존재할 경우 수신측이 해당 비트패턴을 제어비트로 인식해서 프레임이 손실될 수 있는 문제를 해결하기 위해
- 비트 스테핑(Bit Stuffing)을 사용

➤ 송신측의 비트 스테핑 (Bit stuffing)

- 연속으로 다섯 개 이상의 1을 전송하려 하면 비트 스테핑을 수행
- 수신자가 플래그와 데이터를 혼동하지 않게 하기 위해 전송되는 데이터 중에서 1이 연속으로 5번 오면 0을 추가
- 이는 원래의 6번째 비트와는 상관없이 덧붙여 짐

11.3 동기 데이터링크 프로토콜

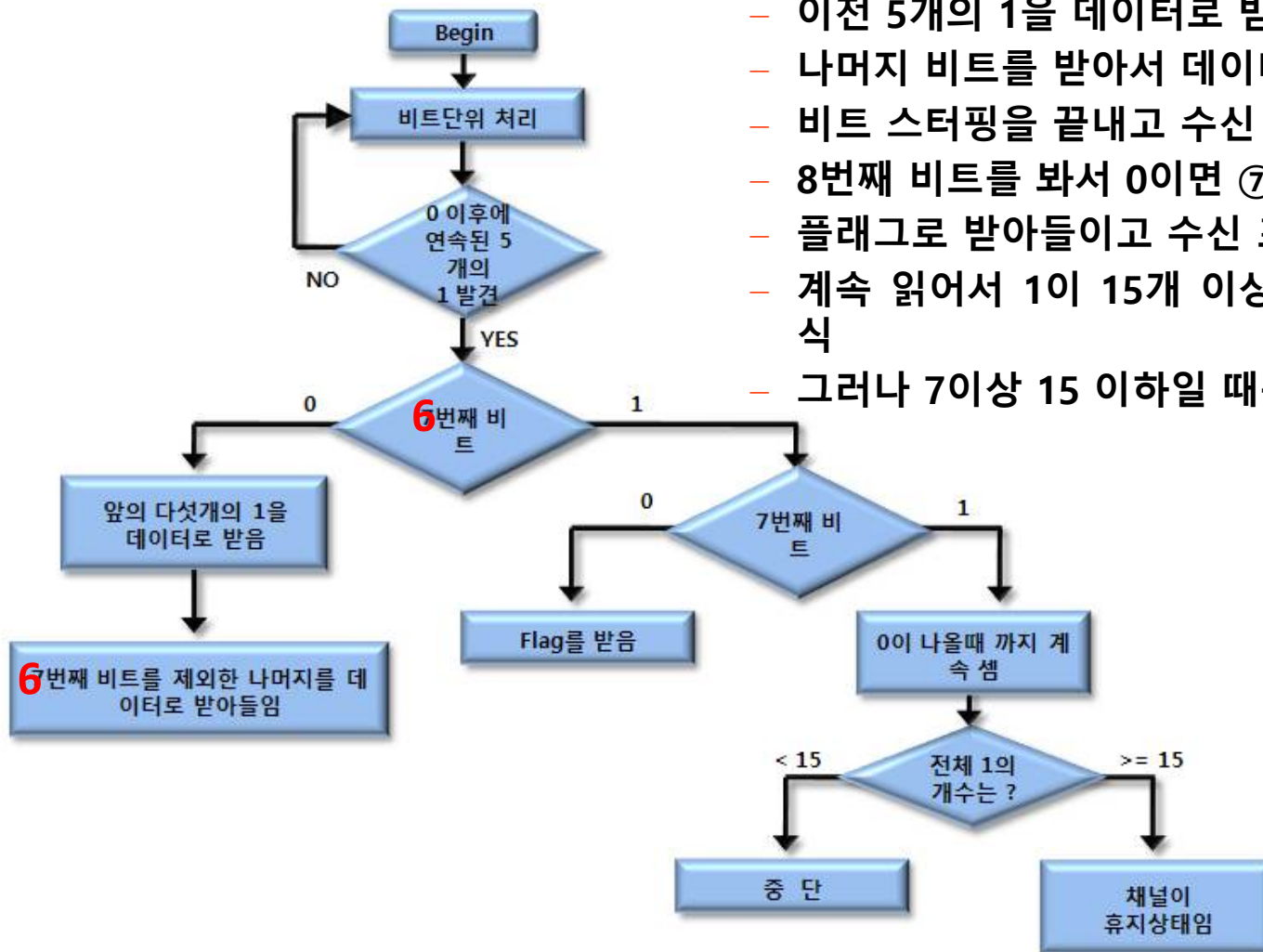
□ 비트 채우기(stuffing)와 빼기(unstuffing)



11.3 동기 데이터링크 프로토콜(14/22)

수신측의 비트스터핑

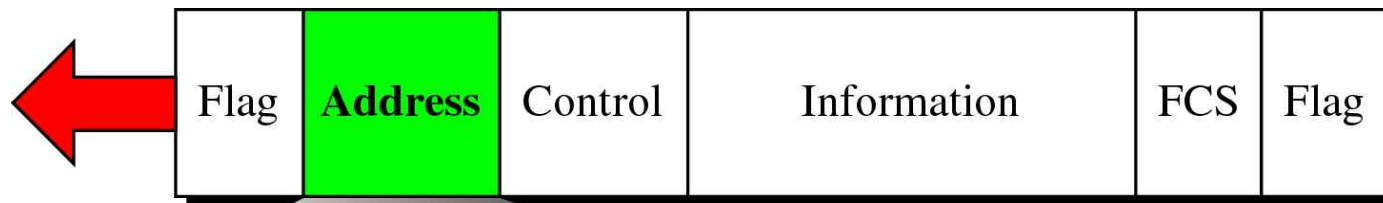
- 0 이후에 1이 연속적으로 5번 나온다고 하면 스텝핑 루틴으로,
- 7번째 비트를 조사한다. 7번째 비트가 0이라면 ③으로 1이라면 ⑥로,
- 이전 5개의 1을 데이터로 받아들이고 현재의 0을 무시하고
- 나머지 비트를 받아서 데이터로 처리
- 비트 스텝핑을 끝내고 수신 프레임을 계속 읽음
- 8번째 비트를 봐서 0이면 ⑦로 1이면 ⑧로, **<error 7=>8>**
- 플래그로 받아들이고 수신 프레임을 계속 읽음
- 계속 읽어서 1이 15개 이상 나오면 휴지채널(idle channel)로 인식
- 그러나 7이상 15 이하일 때는 중지상태(Abort)로 인식



11.3 동기 데이터링크 프로토콜

주소 필드

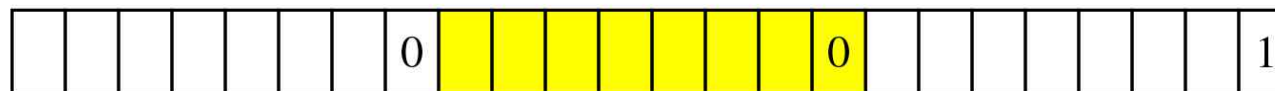
- ✓ 항상 종국의 주소
- ✓ 프레임 발신지나 목적지인 종국의 주소를 포함
 - 주국(Primary)이 주소를 만들 경우 : 목적지 종국 주소
 - 종국(Secondary)이 주소를 만들 경우 : 발신지 종국 주소



The address is one byte (8 bits)
or a multiple of bytes.



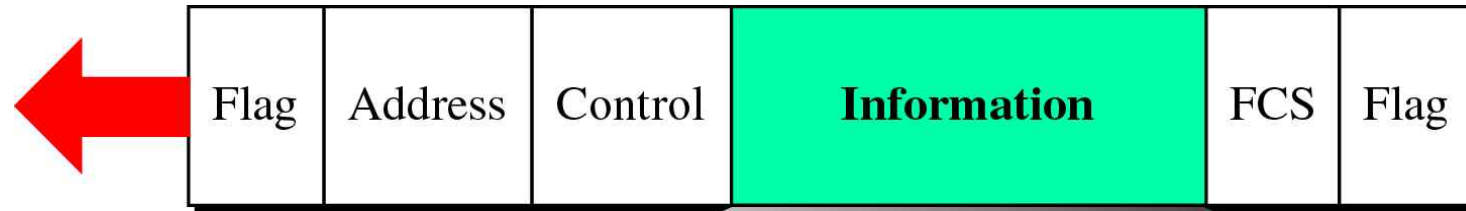
One-byte address



Multibyte address

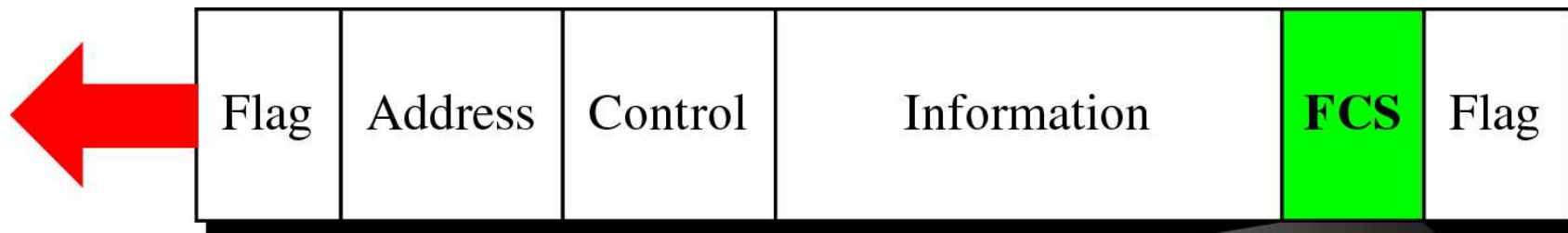
11.3 동기 데이터링크 프로토콜(17/22)

정보필드



It contains user data in an I-Frame.
It does not exist in an S-Frame.
It contains management information in a U-frame.

FCS 필드



Frame check sequence is the error detection field.
It can be a two-byte or a four-byte CRC.