Royal University of Phnom Penh

Data Communication II

Chapter 5_0: Data Link Control and Protocols

Lecturer: ***CHHORN SYLUN***

Email: *chhorn.sylun@rupp.edu.kh*

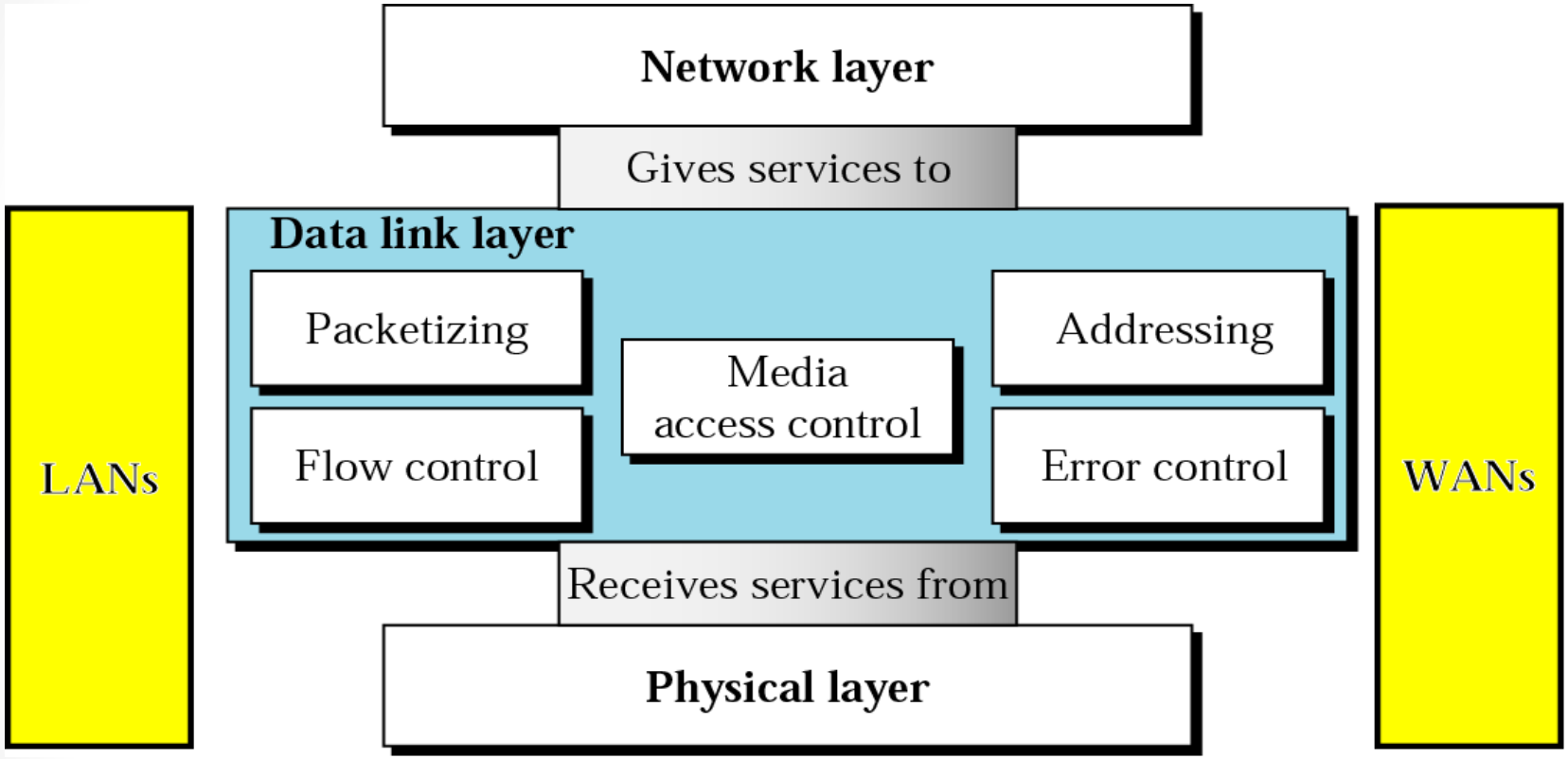
Room: A311B, RUPP

2019

Objective

- Position of Data Link Layer
- Data Link Control
- Flow and Error Control
- Data Link Protocols
- HDLC

Position of Data Link Layer



Data Link Control

- Data Link Control (DLC) deals with procedures for communication between two adjacent nodes.
- DLC functions include:
  1. Framing
  2. Flow control
  3. Error control

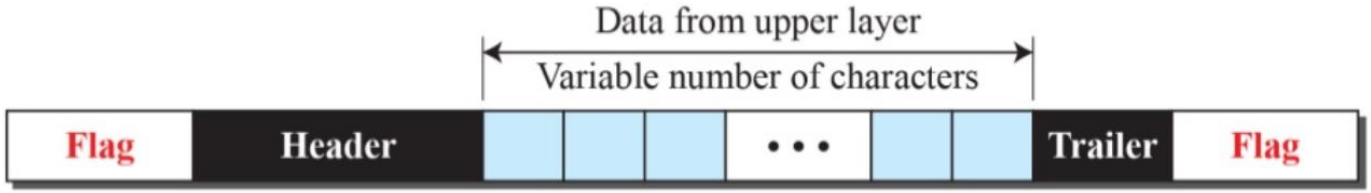
Data Link Control

- Framing: Data link layer packs bits into frames so that each frame is distinguishable from another frame.
- Framing in data link layer separates a message by encapsulating a sender address and a destination address in to a header.
- Framing techniques include:
  1. Character Oriented Framing
  2. Bit Oriented Framing

Data Link Control

Frames may have a particular bit pattern between frames to help identify the start and end of a frame.

This kind of special bit pattern is referred to as a "Flag".



Flag: indication for start and end of a frame

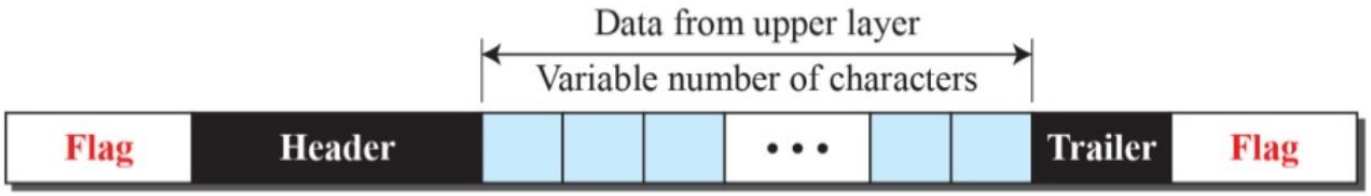
Header: source/destination address

Data from the upper layer

Trailer: error detection/correction code

Data Link Control

1. A frame in a character-oriented protocol



- Also known as byte oriented protocol
- A communications protocol in which full bytes are used as control codes.
- Byte oriented protocol often uses the ASCII codes to encode the transmitted data.

Data Link Control

1. A frame in a character-oriented protocol

Bit vs Byte

- A Bit stores just 0 and 1
- In the computer it's all 0's and 1's
- One Byte = collection of 8 bits
- Example: 0 1 0 1 1 0 1 0
- One Byte can store one character, Ex. 'A', 'x', or '$'

Data Link Control

1. A frame in a character-oriented protocol

ASCII Table

Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char	Dec	Hex	Char
0	00	Null	32	20	Space	64	40	@	96	60	`
1	01	Start of heading	33	21	!	65	41	A	97	61	a
2	02	Start of text	34	22	"	66	42	B	98	62	b
3	03	End of text	35	23	#	67	43	C	99	63	c
4	04	End of transmit	36	24	$	68	44	D	100	64	d
5	05	Enquiry	37	25	%	69	45	E	101	65	e
6	06	Acknowledge	38	26	&	70	46	F	102	66	f
7	07	Audible bell	39	27	'	71	47	G	103	67	g
8	08	Backspace	40	28	(	72	48	H	104	68	h
9	09	Horizontal tab	41	29	)	73	49	I	105	69	i
10	0A	Line feed	42	2A	*	74	4A	J	106	6A	j
11	0B	Vertical tab	43	2B	+	75	4B	K	107	6B	k
12	0C	Form feed	44	2C	,	76	4C	L	108	6C	l
13	0D	Carriage return	45	2D	-	77	4D	M	109	6D	m
14	0E	Shift out	46	2E	.	78	4E	N	110	6E	n
15	0F	Shift in	47	2F	/	79	4F	O	111	6F	o
16	10	Data link escape	48	30	0	80	50	P	112	70	p
17	11	Device control 1	49	31	1	81	51	Q	113	71	q
18	12	Device control 2	50	32	2	82	52	R	114	72	r
19	13	Device control 3	51	33	3	83	53	S	115	73	s
20	14	Device control 4	52	34	4	84	54	T	116	74	t
21	15	Neg. acknowledge	53	35	5	85	55	U	117	75	u
22	16	Synchronous idle	54	36	6	86	56	V	118	76	v
23	17	End trans. block	55	37	7	87	57	W	119	77	w
24	18	Cancel	56	38	8	88	58	X	120	78	x
25	19	End of medium	57	39	9	89	59	Y	121	79	y
26	1A	Substitution	58	3A	:	90	5A	Z	122	7A	z
27	1B	Escape	59	3B	;	91	5B	[	123	7B	{
28	1C	File separator	60	3C	<	92	5C	\	124	7C	|
29	1D	Group separator	61	3D	=	93	5D	]	125	7D	}
30	1E	Record separator	62	3E	>	94	5E	^	126	7E	~
31	1F	Unit separator	63	3F	?	95	5F	_	127	7F	□

Data Link Control

1. A frame in a character-oriented protocol

Example Code:

- **SYN = 00010110 (16 Hex)**
- **SOH = 00000001 (01 Hex)** ("start of header")
- **SOT = 00000010 (02 Hex)** ("start of text")
- **EOT = 00000011 (03 Hex)** ("end of text")
- **ESCAPE=00011011(1B Hex)**

Data Link Control

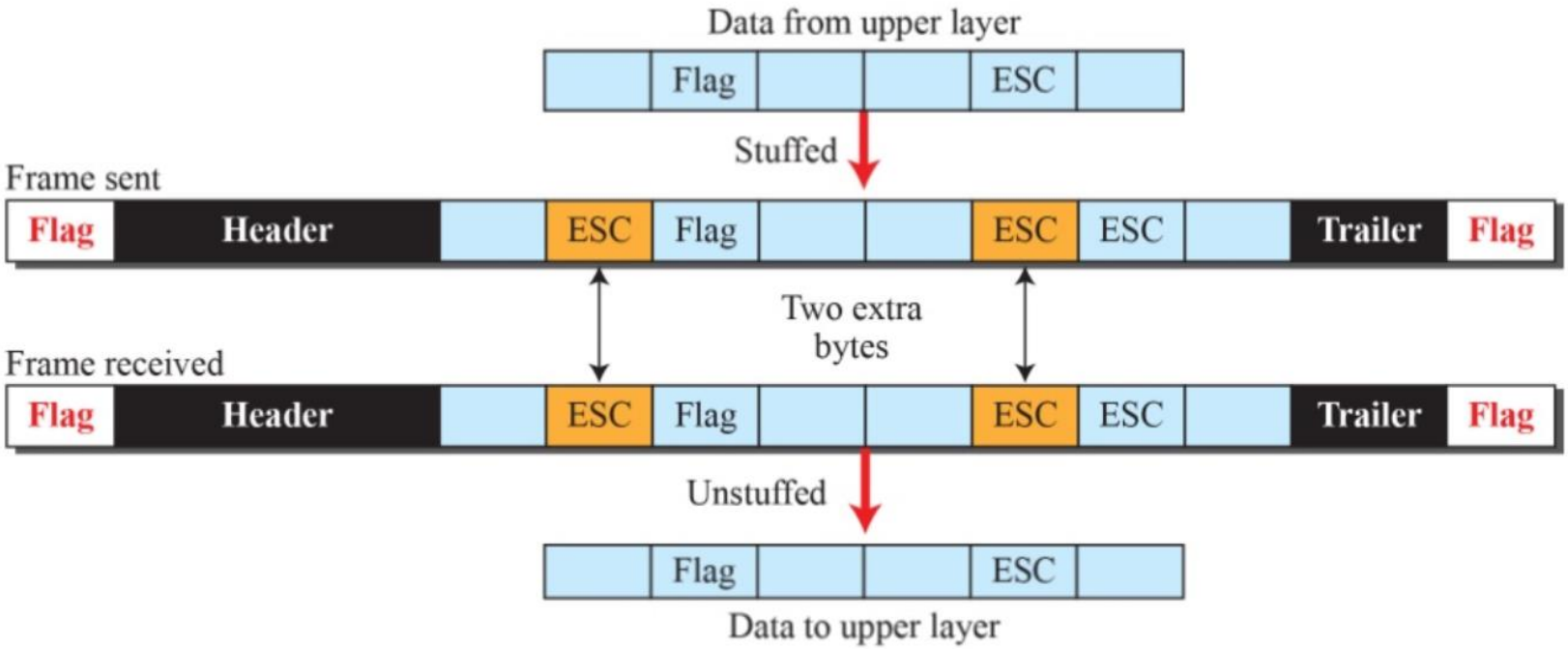
1. A frame in a character-oriented protocol

Problem: the flag bit pattern could appear in the data portion of the frame.

The problem is solved by stuffing a special byte pattern know as an Escape Character **(ESC)** into the data.

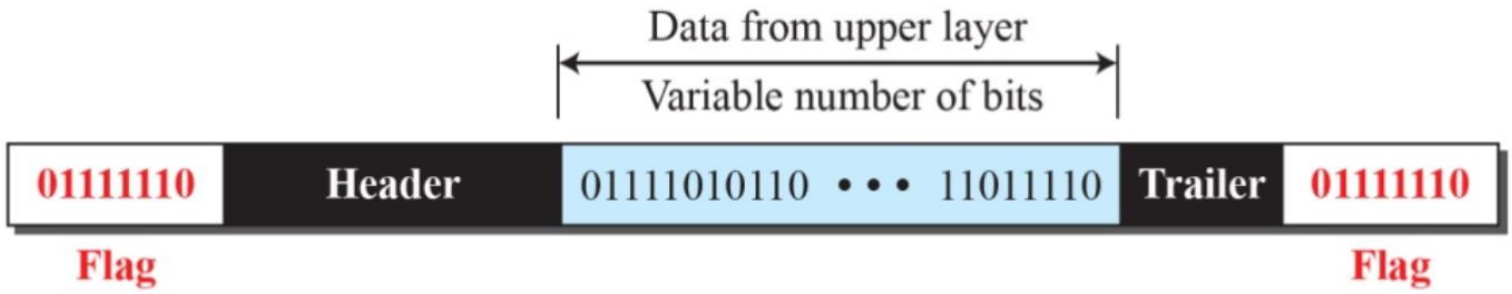
Data Link Control

Byte stuffing and unstuffing



Data Link Control

2. A frame in a bit-oriented protocol



- Control codes are defined in terms of bit sequences instead of characters.
- Each frame begins and ends with a special bit pattern **01111110**, called a flag byte.

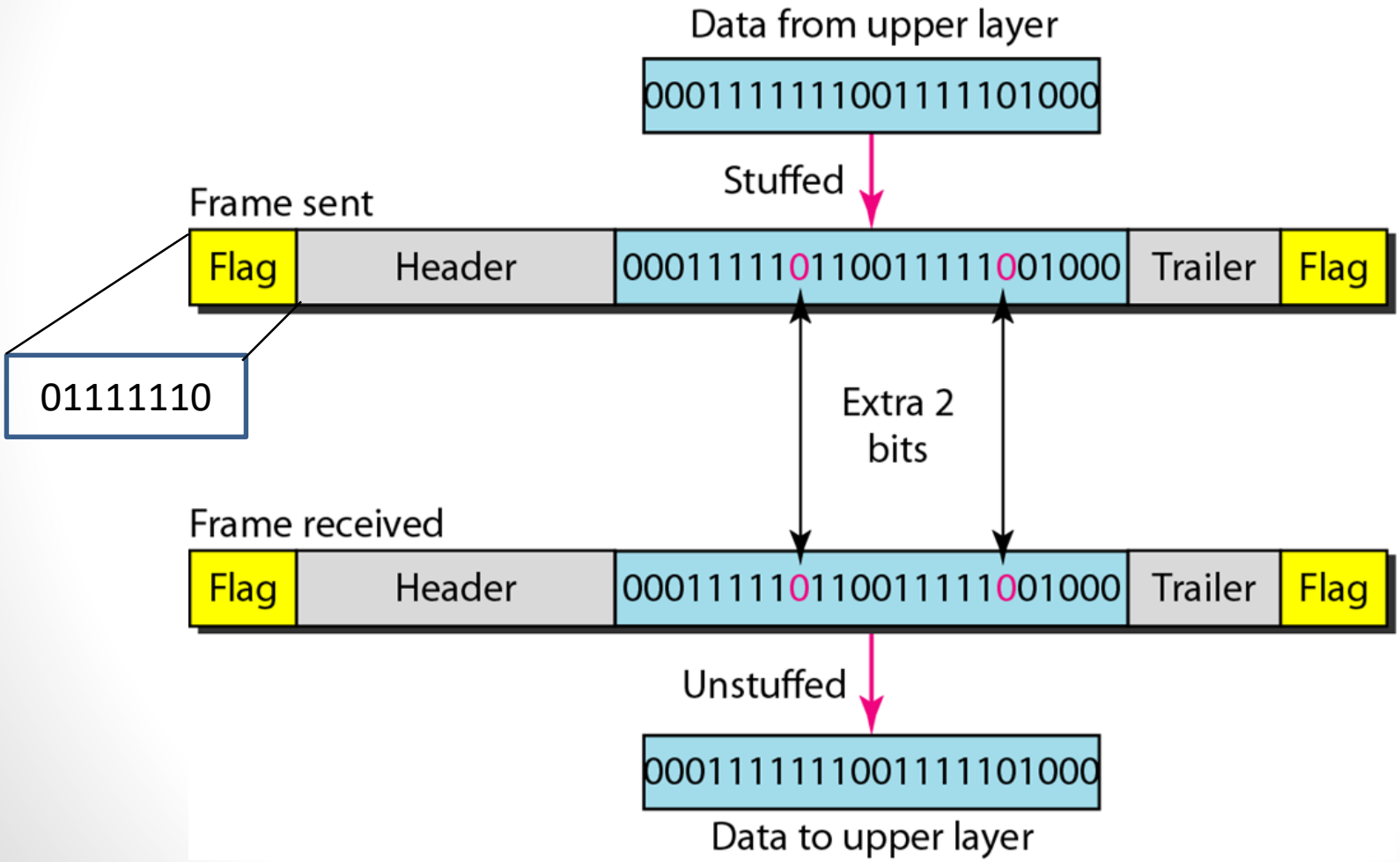
Data Link Control

2. A frame in a bit-oriented protocol

- A bit stuffing technique is used to prevent the receiver from detecting the special flag byte in user data.
- Bit stuffing is the process of adding one extra **0** whenever five consecutive 1s follow a 0 in the data, so that the receiver does not mistake the pattern 0111110 for a flag.

Data Link Control

Bit stuffing and unstuffing



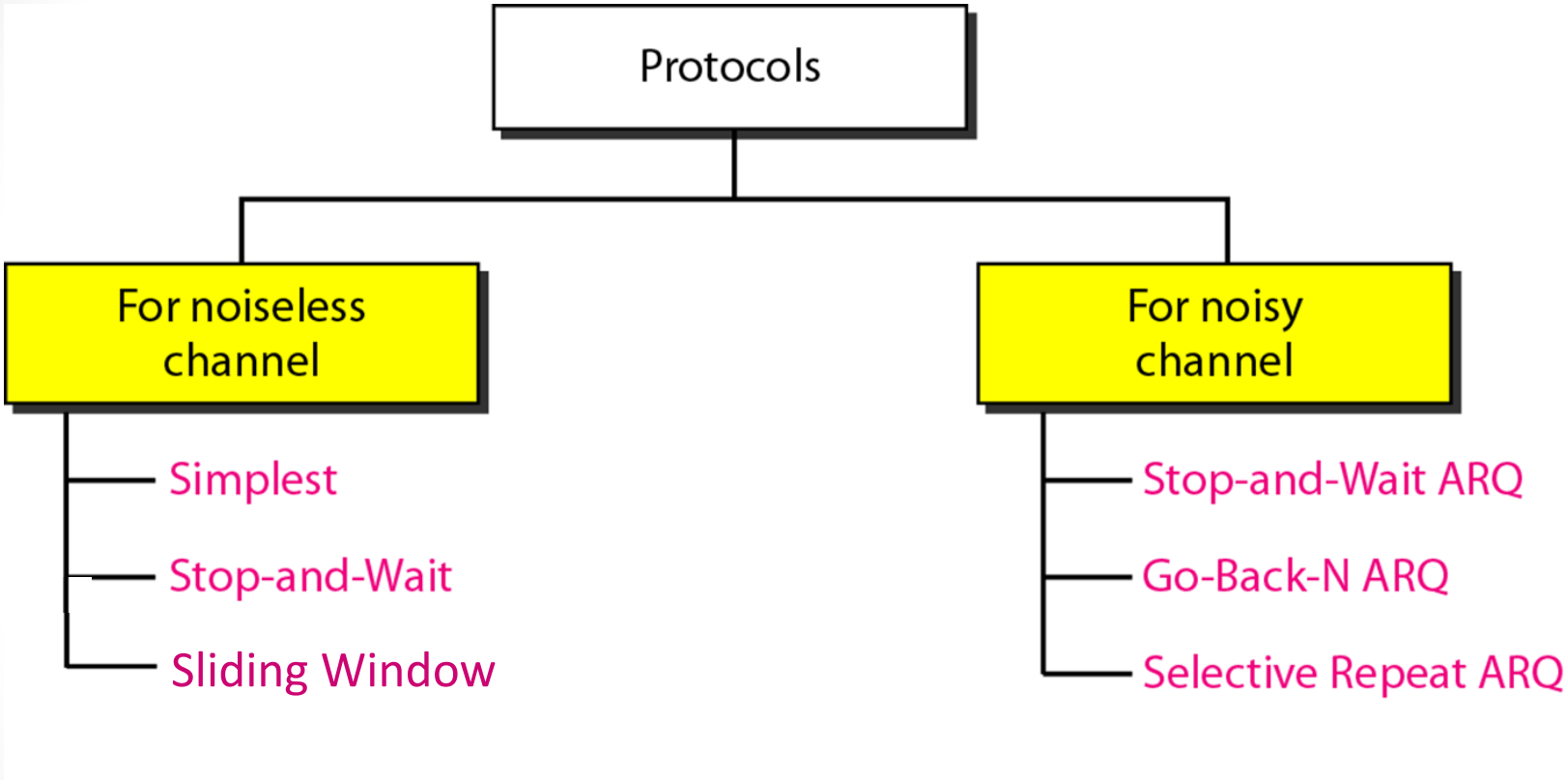
Flow and Error Control

- Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for Acknowledgment.
- Error Control
  - Includes both error detection and correction
  - Allows receiver to inform sender of lost or duplicate frames
  - Mostly based on Automatic Repeat Request (ARQ)

Data Link Protocols

- Now let us see how the data link layer can combine framing, flow control, and error control to achieve the delivery of data from one node to another.
- The protocols are normally implemented in software by using one of the common programming languages.

Data Link Protocols

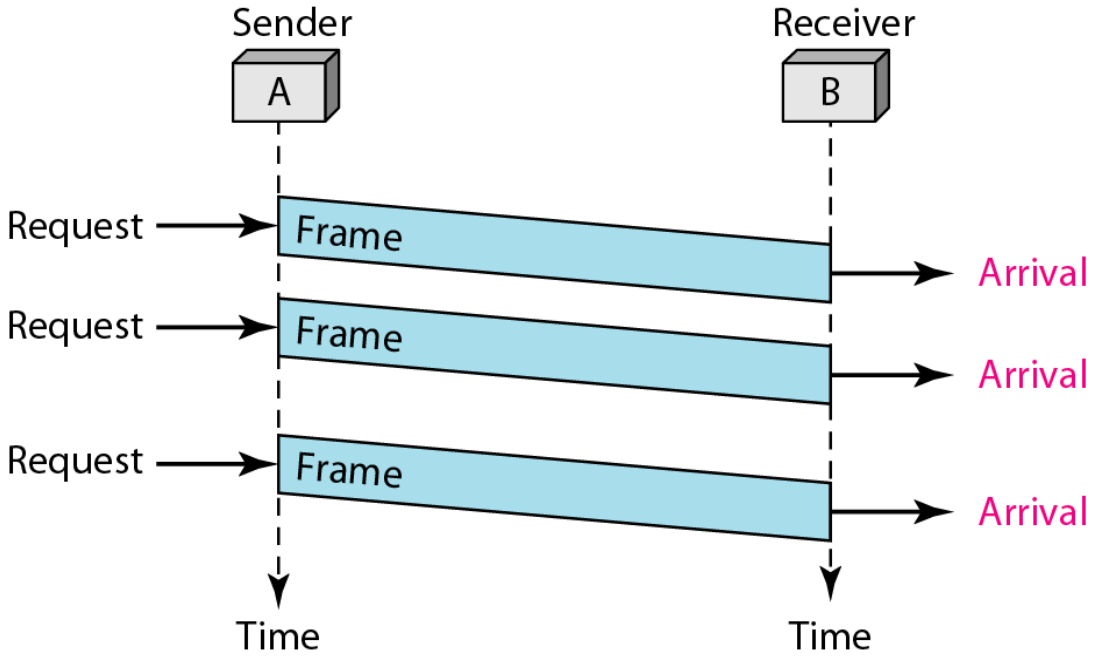


Noiseless Channel

- Assuming channel is error free
- No need error control
- Three types of mechanisms can be deployed to control the flow such as:
  1. Simplest
  2. Stop and Wait
  3. Sliding Window

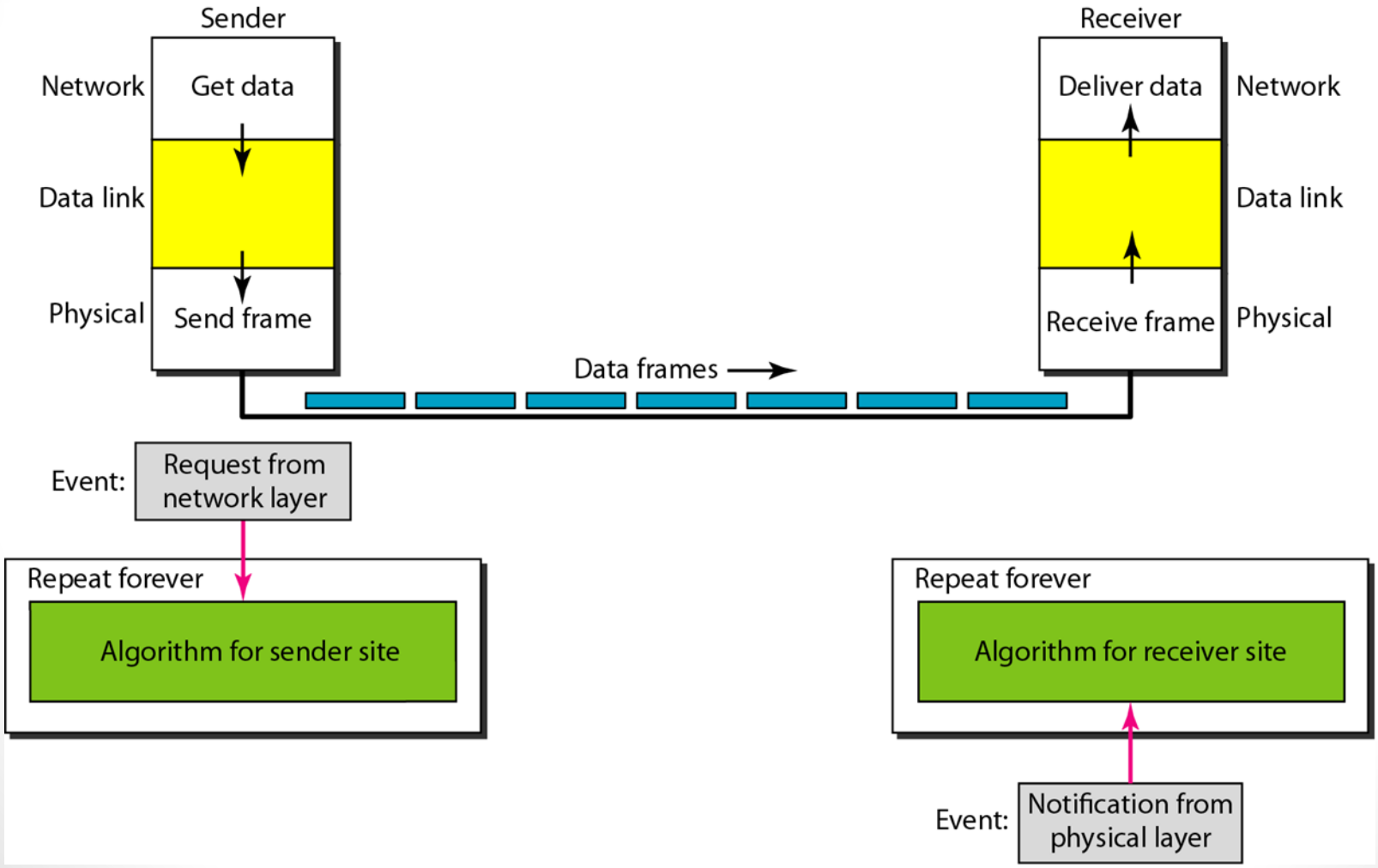
Noiseless Channel

1. Simplest Protocol: it is very simple, since the sender sends a sequence of frame without even thinking about the receiver.



Noiseless Channel

1. Simplest Protocol



Noiseless Channel

1. Simplest Protocol

Pseudo code: Sender-site algorithm for simplest protocol

```
1  while(true)                          // Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(RequestToSend))           //There is a packet to send
5    {
6       GetData();
7       MakeFrame();
8       SendFrame();                    //Send the frame
9    }
10 }
```

Noiseless Channel

1. Simplest Protocol

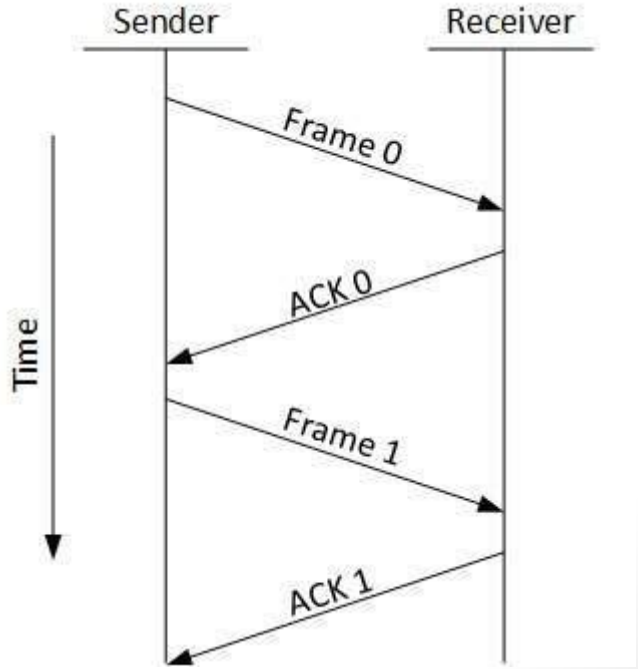
Pseudo code: Receiver-site algorithm for simplest protocol

```
1  while(true)                          // Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(ArrivalNotification))     //Data frame arrived
5    {
6       ReceiveFrame();
7       ExtractData();
8       DeliverData();                  //Deliver data to network layer
9    }
10 }
```

Noiseless Channel

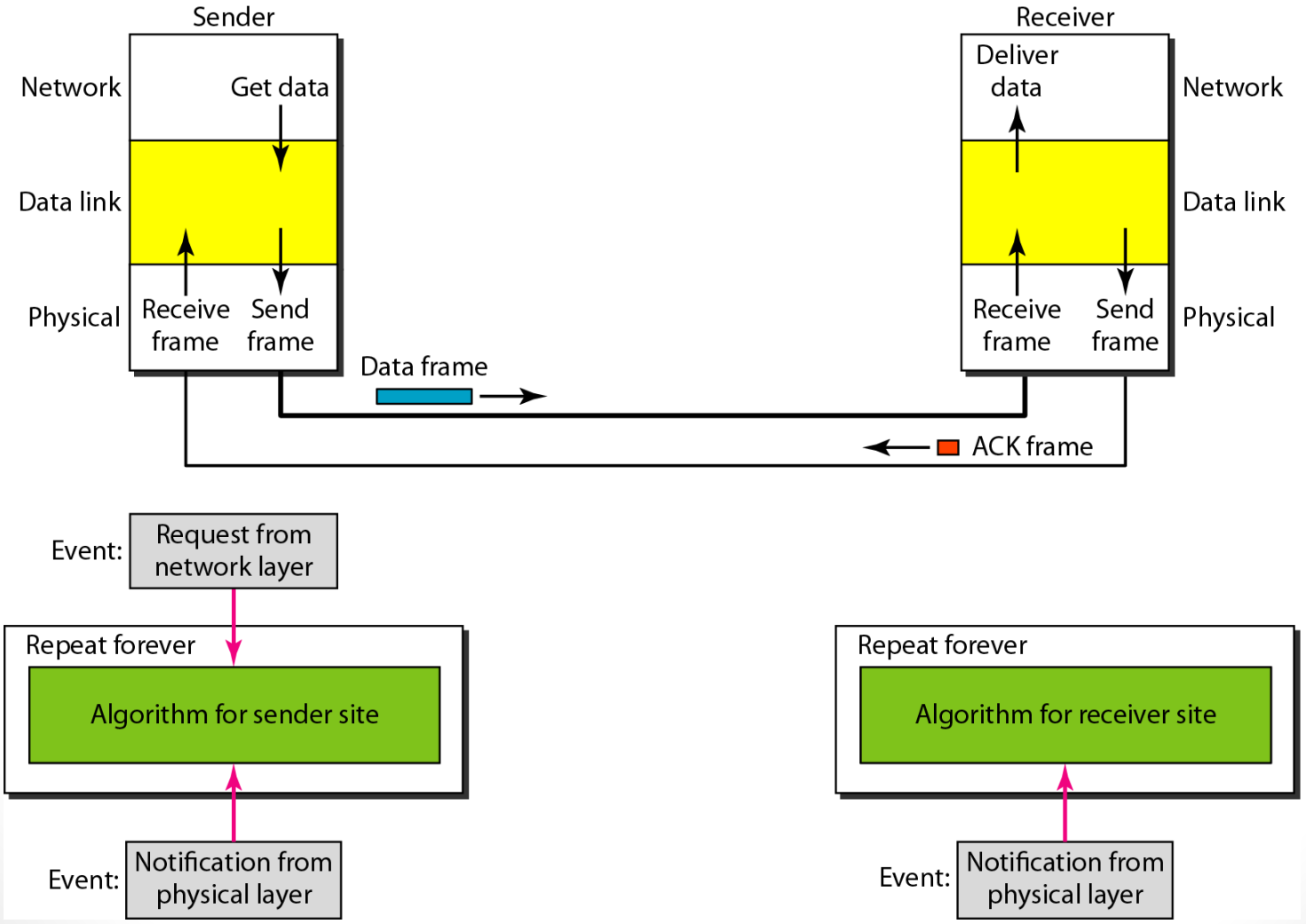
2. Stop and Wait: this flow control mechanism forces the sender after transmitting a data frame to stop and wait until the acknowledgement of the data-frame sent is receiver.

- Source transmits frame
- Destination receivers frame and replies with ACK
- Source waits for ACK before sending next frame
- Destination can stop flow by not sending ACK



Noiseless Channel

2. Stop and Wait



Noiseless Channel

2. Stop and Wait

Pseudo code: Sender-site algorithm for Stop and Wait

```
1  while(true)                          //Repeat forever
2  canSend = true                       //Allow the first frame to go
3  {
4    WaitForEvent();                    // Sleep until an event occurs
5    if(Event(RequestToSend) AND canSend)
6    {
7       GetData();
8       MakeFrame();
9       SendFrame();                    //Send the data frame
10      canSend = false;                //Cannot send until ACK arrives
11   }
12   WaitForEvent();                    // Sleep until an event occurs
13   if(Event(ArrivalNotification)      // An ACK has arrived
14    {
15      ReceiveFrame();                 //Receive the ACK frame
16      canSend = true;
17    }
18 }
```

Noiseless Channel

2. Stop and Wait

Pseudo code: Receiver-site algorithm for Stop and Wait

```
1  while(true)                          //Repeat forever
2  {
3    WaitForEvent();                    // Sleep until an event occurs
4    if(Event(ArrivalNotification))     //Data frame arrives
5    {
6       ReceiveFrame();
7       ExtractData();
8       Deliver(data);                  //Deliver data to network layer
9       SendFrame();                    //Send an ACK frame
10   }
11 }
```

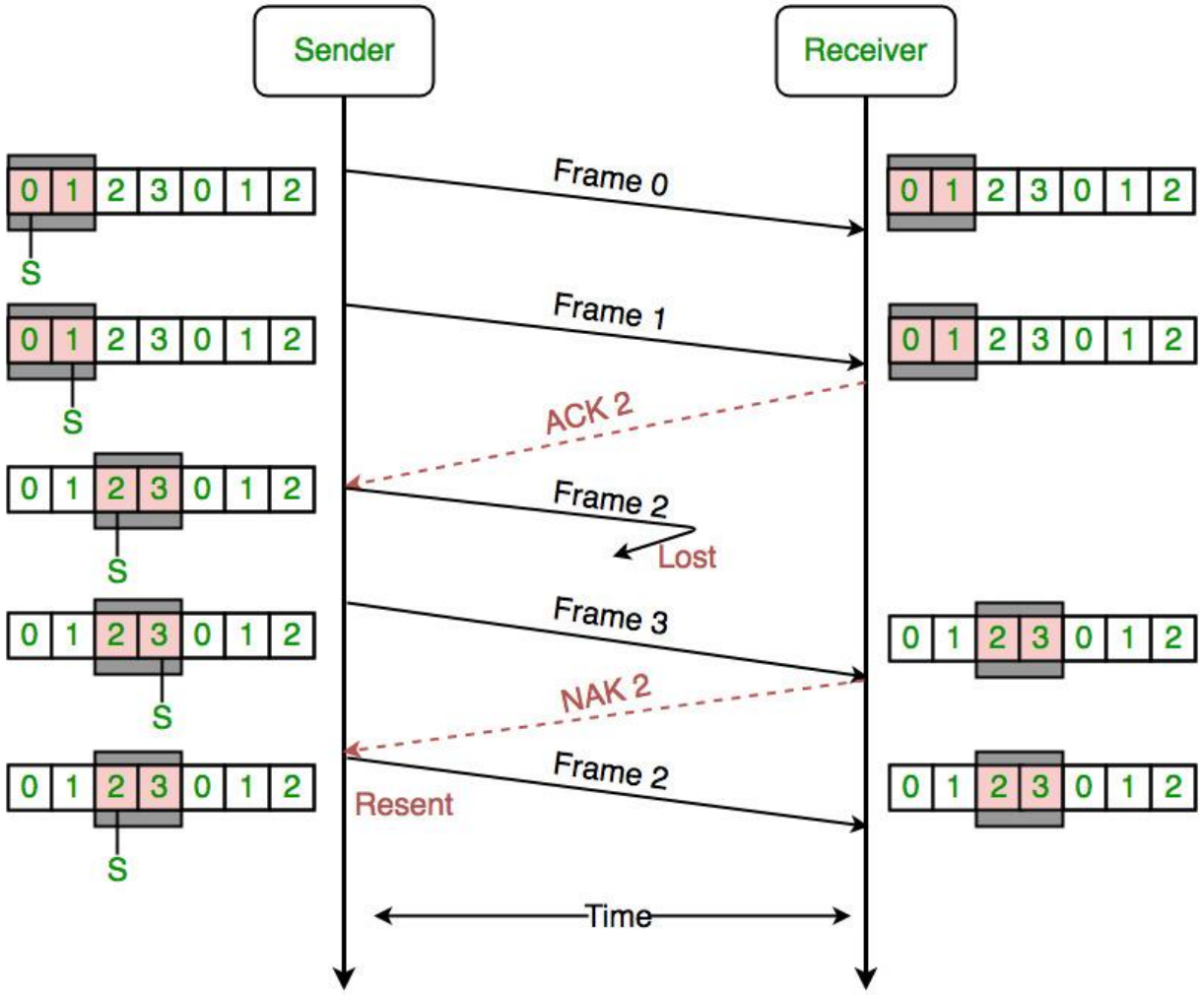
Noiseless Channel

3. Sliding Window: this flow control mechanism , both sender and receiver agree on the number of data frames after which the acknowledgement should be sent.

- As we learnt, stop and wait flow control mechanism wastes resources.
- This protocol tries to make use of underlying resource as much as possible.

Noiseless Channel

3. Sliding Window



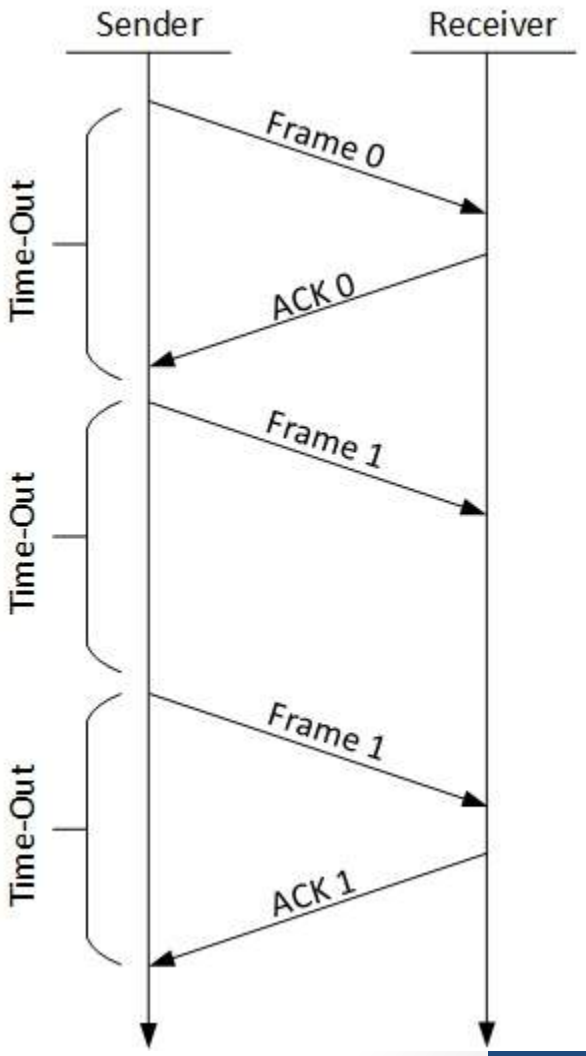
Noisy Channel

- In realistic, error can be happen during transmitting of each frame so that we might need error control.
- Mostly based on Automatic Repeat Request (ARQ)
- There are 3 types of techniques to control error:
    1. Stop and Wait ARQ
    2. Go Back N ARQ
    3. Selective Repeat ARQ

Noisy Channel

1. Stop and Wait ARQ

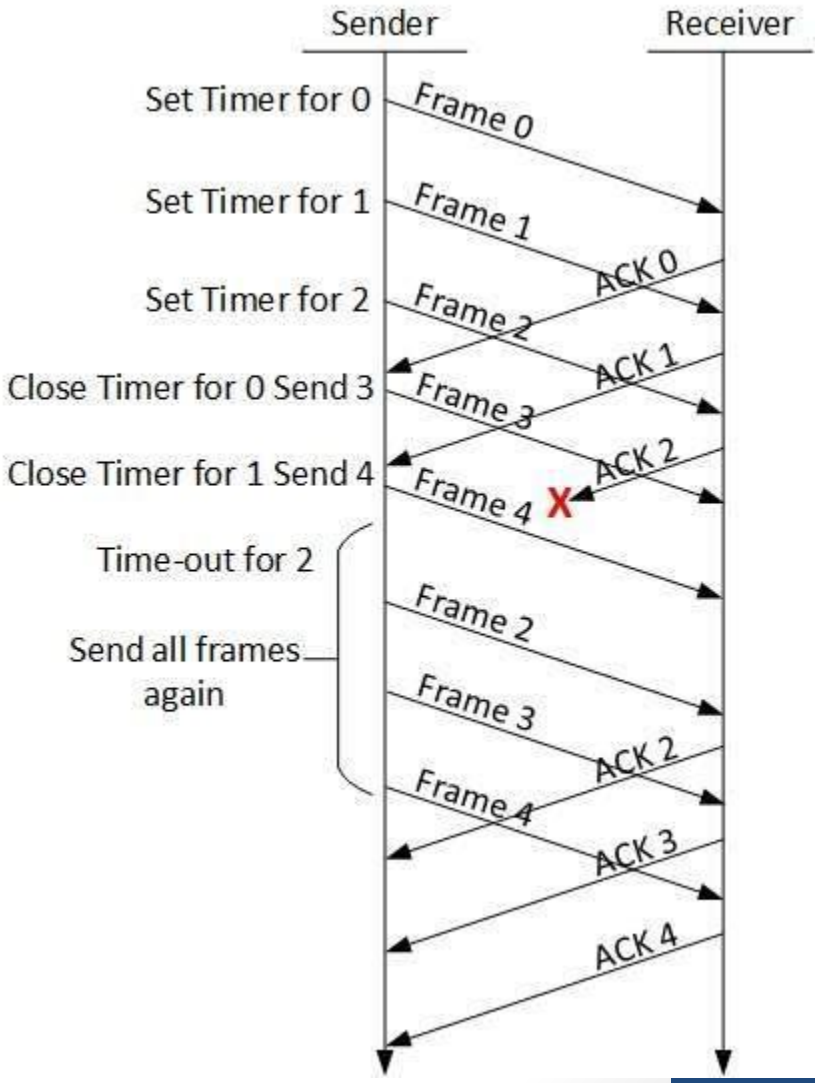
- The sender maintains a timeout counter.
- When a frame is sent, the sender starts the timeout counter.
- If acknowledgement of frame comes in time, the sender transmits the next frame in queue.
- If acknowledgement does not come in time, the sender assumes that either the frame or its acknowledgement is lost in transit. Sender retransmits the frame and starts the timeout counter.
- If a negative acknowledgement is received, the sender retransmits the frame.



Noisy Channel

2. Go-Back-N ARQ

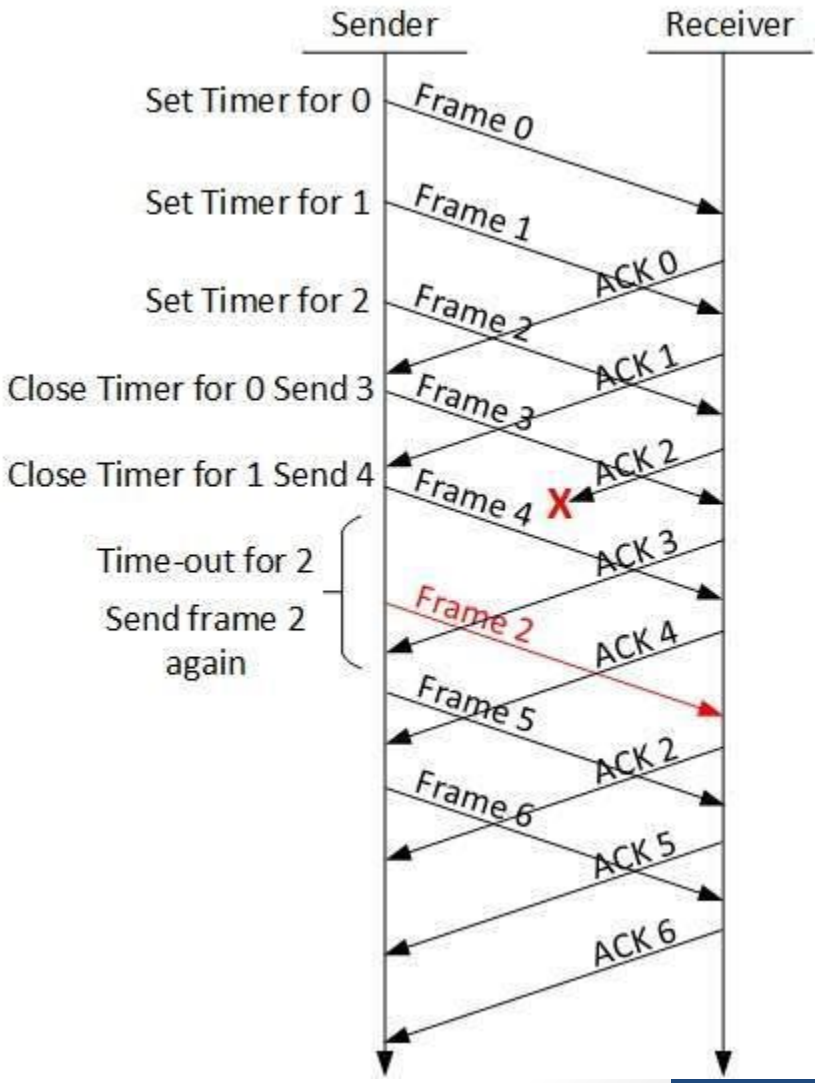
- Uses sliding window flow control
- When receiver detects error, it sends Negative Acknowledgement (NACK)
- Sender must begin transmitting again from rejected frame.
- Transmitter must keep a copy of all transmitted frames



Noisy Channel

3. Selective Repeat ARQ

- Also called Selective retransmission
- Only rejected frames are retransmitted
- Subsequent frames are accepted by the receiver and buffered
- Minimizes retransmissions
- Receiver must maintain large enough buffer
- Complex system



HDLC

- *High-level Data Link Control (HDLC)* is a bit oriented protocol for communication over point-to-point and multipoint links.
- It's supports full-duplex communication
- It's is most widely accepted protocol. It's offer a high level of flexibility, adaptability, reliability and efficiency.
- Flow control adjust window sized based on receiver capability.

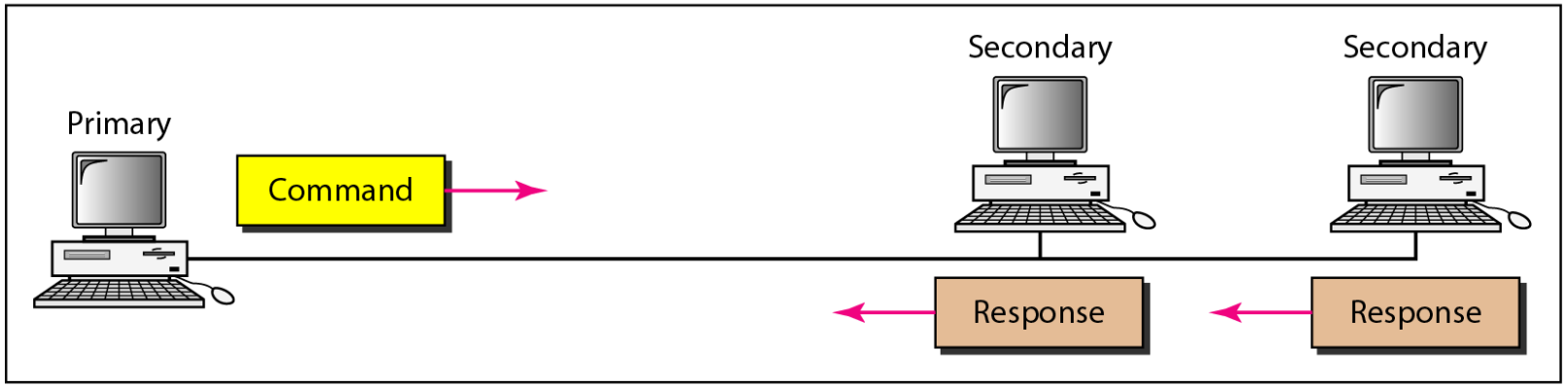
HDLC

To make HDLC protocol applicable to various network configurations, three types of station have been defined:

1. Primary Station (Normal Response Mode)
2. Secondary Station (Asynchronous Response Mode)
3. Combined Station (Asynchronous Balance Mode)

HDLC

a. Point-to-point



b. Multipoint

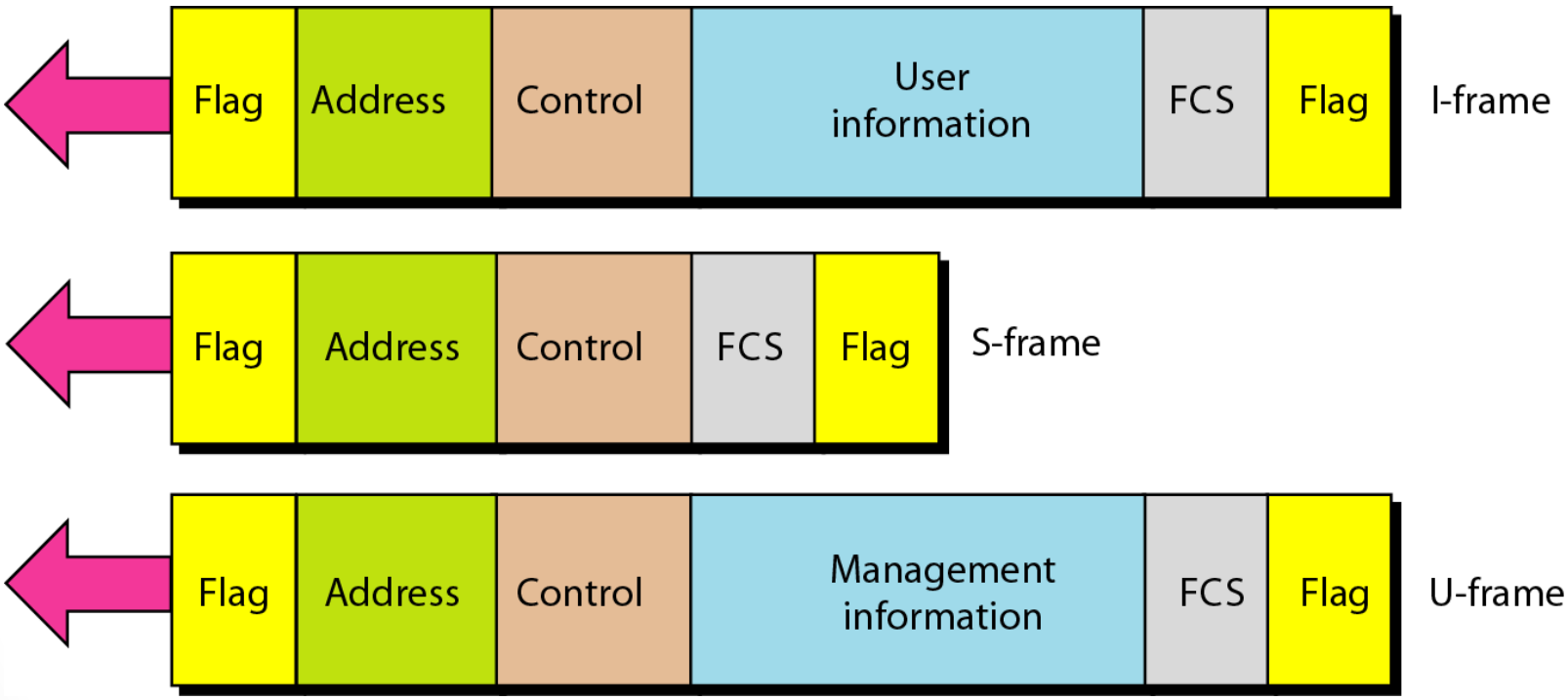
HDLC

HDLC Frames

- There are three types of frames in HDLC:
  1. Unnumbered or U-Frames, used for exchanging session management and control information between communicating devices.
  2. Information or I-Frames, which is carry actual information. If the first bit in control field is 0 it is identified as I-Frame.
  3. Supervisory or S-Frames, which is used for error and flow control purpose. If the two bits of control field are 1 and 0 it is identified as S-Frame.

HDLC

HDLC Frames



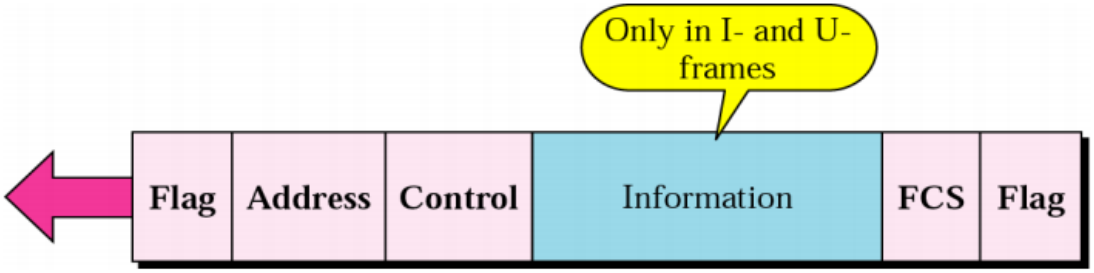
HDLC

HDLC Frames

- Flag: The flag filed of an HDLC frame is an 8-bit sequence with the bit pattern 01111110 that identifies both the beginning and the end of a frame.
- Address: The second field of an HDLC contains the address of the secondary station.
- Control: the control field is a 1- or 2 bytes segment of the frame uses for flow and error control. The interpretation of bits in this field depends on the frame type.
- Information: the information field contains the user's data from the network layer.
- FCS: Frame Check Sequence is the HDLC error detection field.

HDLC

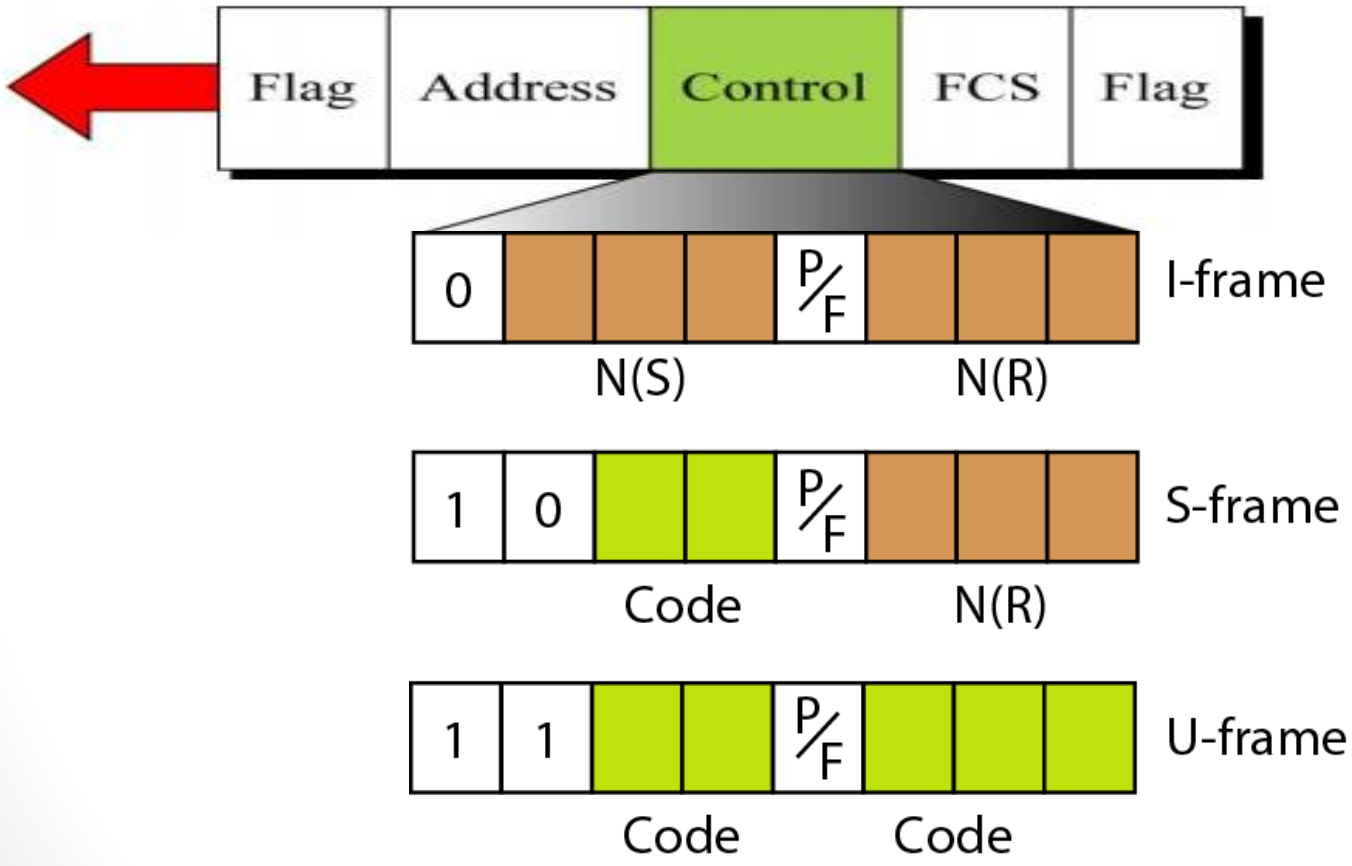
HDLC Frames



Field Name	Size (in Bits)
Flag Field (F)	8 bits
Address Field (A)	8 bits
Control Field (C)	8 bits or 16 bits
Information Field (I) or Data	-
Frame Check Sequence (FCS)	16 bits or 32 bits
Closing Flag Field (F)	8 bits

HDLC

Control field format for the difference frame types



HDLC

Control field format for the difference frame types

- N(S): Send sequence number
- N(R): Receive sequence number
- S: Supervisory function bits
- M: Unnumbered function bits
- P/F: Poll/final bit
  a) When P/F =1, it means poll, frame is send by primary
  b) When P/F = 0, it means final, frame is send by secondary station

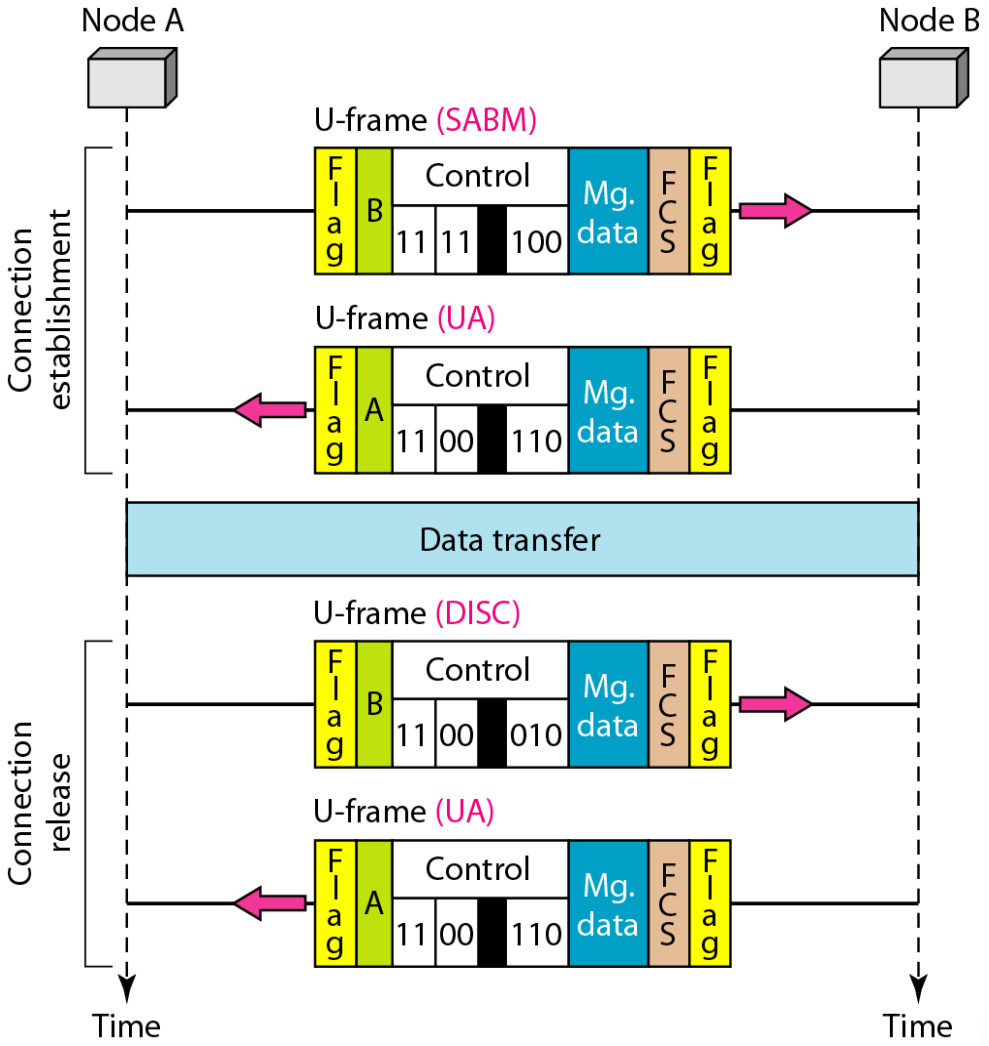
HDLC

U-frame control command and response

Code	*Command*	*Response*	*Meaning*
00 001	SNRM		Set normal response mode
11 011	SNRME		Set normal response mode, extended
11 100	SABM	**DM**	Set asynchronous balanced mode or **disconnect mode**
11 110	SABME		Set asynchronous balanced mode, extended
00 000	UI	**UI**	Unnumbered information
00 110		**UA**	**Unnumbered acknowledgment**
00 010	DISC	**RD**	Disconnect or **request disconnect**
10 000	SIM	**RIM**	Set initialization mode or **request information mode**
00 100	UP		Unnumbered poll
11 001	RSET		Reset
11 101	XID	**XID**	Exchange ID
10 001	FRMR	**FRMR**	Frame reject

HDLC

U-frame: Connection and Disconnection



HDLC

U-frame Example

- U-frame can be used for connection establishment and connection release.
- Node A asks for a connection with a set asynchronous balanced mode (SABM) frame; node B gives a positive response with an unnumbered acknowledgment (UA) frame.
- After these two exchanges, data can be transferred between the two nodes (not shown in the figure).
- After data transfer, node A sends a DISC (disconnect) frame to release the connection; it is confirmed by node B responding with a UA (unnumbered acknowledgment).

Q&A

More explanation please visit the following link:

https://www.youtube.com/watch?v=BOQlBBedtcE