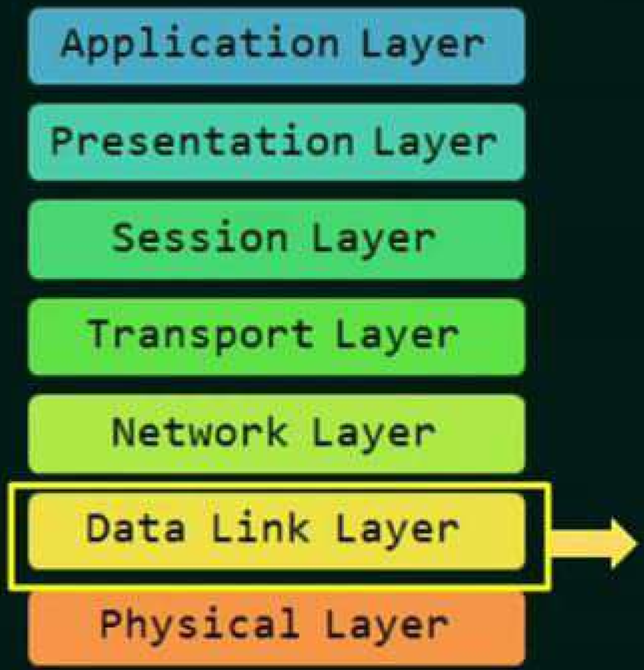
Data Link Layer



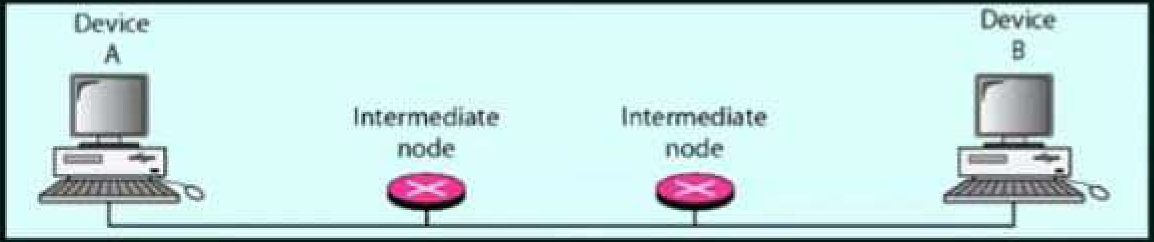
It is responsible for moving data(frames) from one node to another node.

SERVICES PROVIDED BY DATA LINK LAYER

- ★ Framing.
- ★ Physical Addressing.
- ★ Flow Control.
- ★ Error Control.
- ★ Access Control.

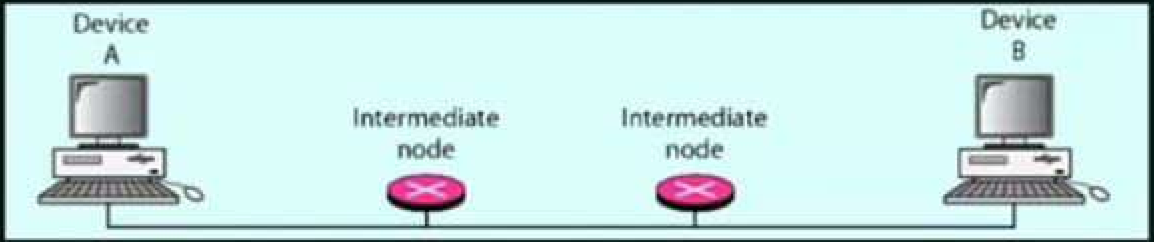
FRAMING

- ★ The data link layer needs to pack bits into frames, so that each frame is distinguishable from another.
- ★ Our postal system practices a type of framing.
- ★ The simple act of inserting a letter into an envelope separates one piece of information from another; the envelope serves as the delimiter.



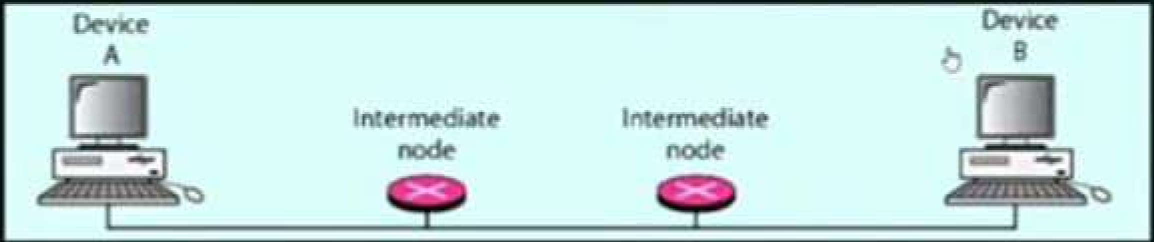
Physical Addressing

- ★ A Frame is the encapsulation of the header and trailer information with the packet.
- ★ In the header, the source and the destination MAC address are dealt.



Flow control

- Flow Control is one of the duties of data link control sublayer.
- The flow control in data link layer is end to end flow control.
- Speed matching mechanism.
- Flow control coordinates the amount of data that can be sent before receiving an acknowledgment.

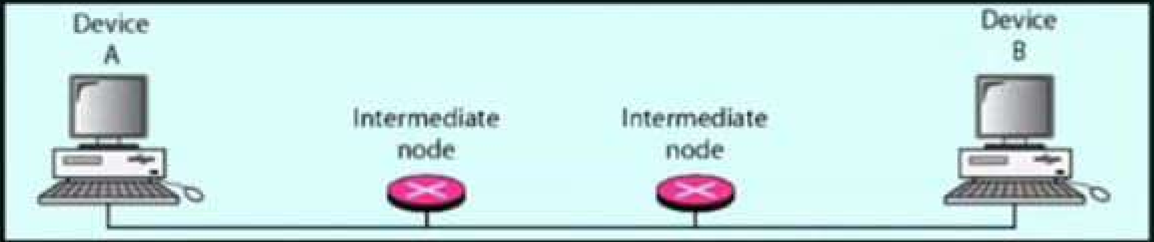


Access control

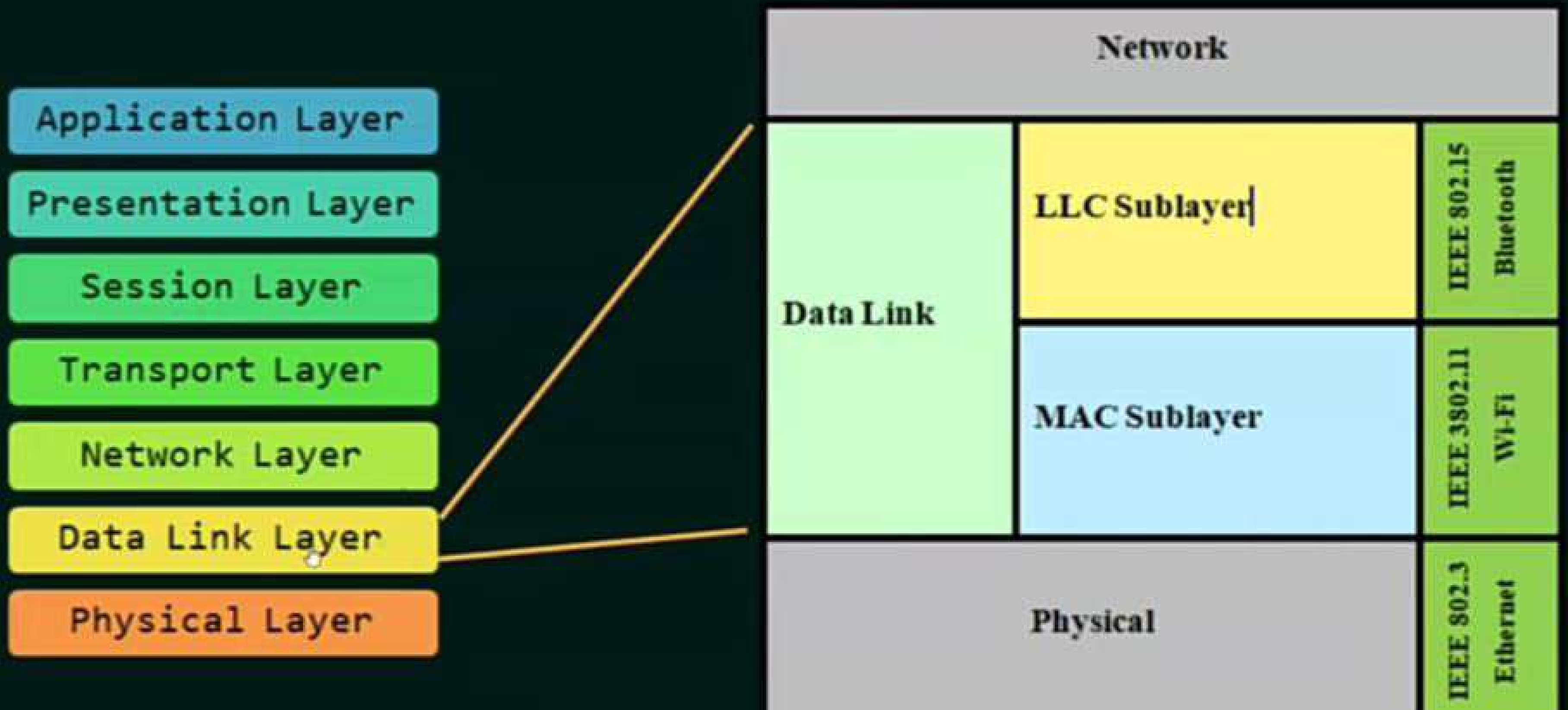
★ Media Access control.

Error control

- ★ Error Detection.
- ★ Error Correction.



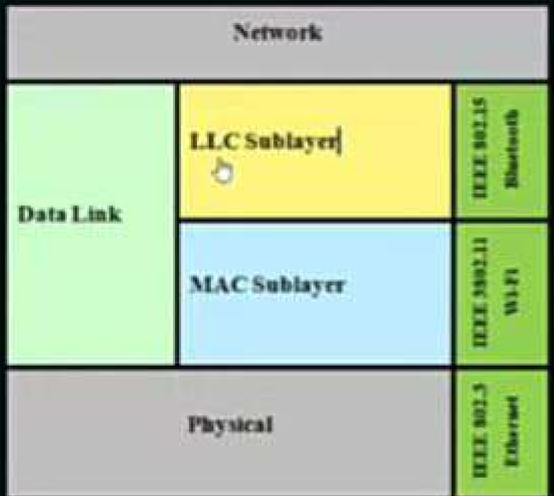
Data Link Layer



Data Link sublayers

Logical Link Control (LLC) or Data Link Control (DLC) Sublayer

- ★ Handles communication between upper and lower layers.
- ★ Takes the network protocol data and adds control information to help deliver the packet to the destination. (Flow control)



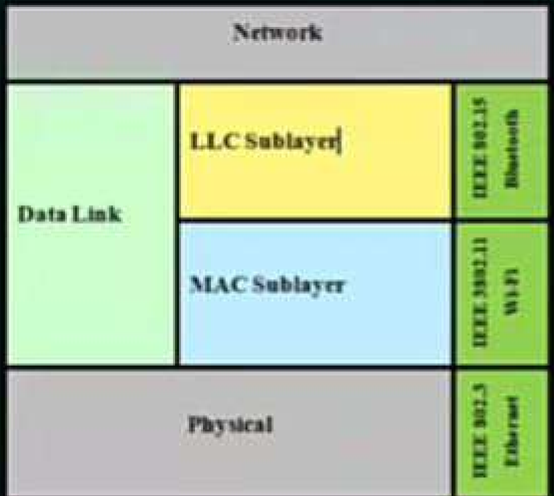
Data Link sublayers

Logical Link Control (LLC) or Data Link Control (DLC) Sublayer

- ★ Handles communication between upper and lower layers.
- ★ Takes the network protocol data and adds control information to help deliver the packet to the destination. (Flow control)

MAC Sublayer

- ★ Constitutes the lower sublayer of the data link layer.
- ★ Implemented by hardware, typically in the computer NIC.
- ★ Two primary responsibilities:
    - ★ Data encapsulation
    - ★ Media access control



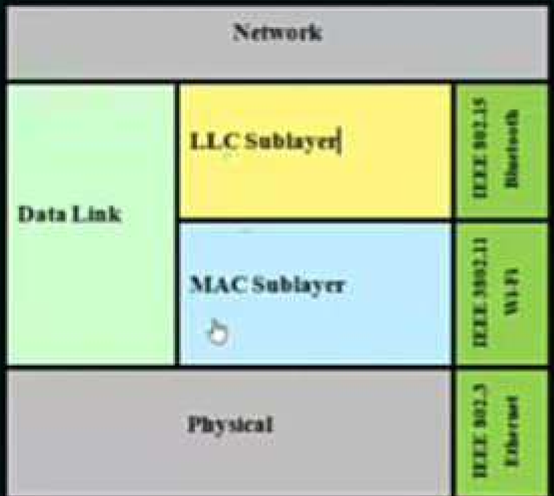
MAC SUBLAYER

Data encapsulation

- ★ Frame assembly before transmission and frame disassembly upon reception of a frame.
- ★ MAC layer adds a header and trailer to the network layer PDU.

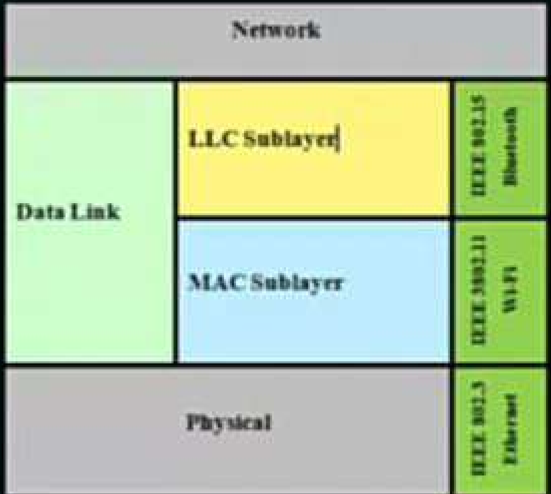
Provides three primary functions:

- ★ Framing.
- ★ Physical Addressing or MAC Addressing.
- ★ Error control.



MAC SUBLAYER

- ★ Responsible for the placement of frames on the media and the removal of frames from the media
- ★ Communicates directly with the physical layer.



Activity Time

Mention the sublayer that is responsible for the service shown in the table.

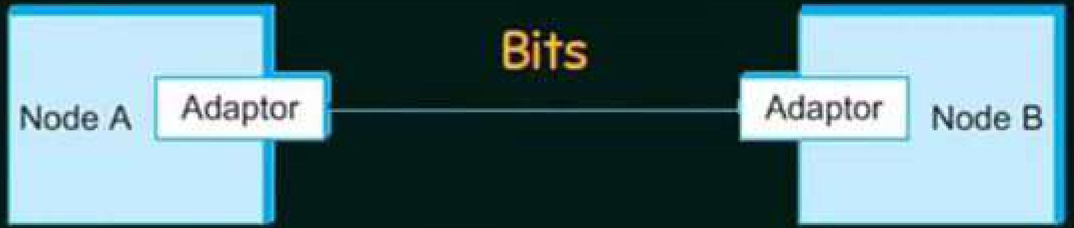
Service	Sublayer
Flow Control	LLC or DLC
Framing	MAC
Physical Addressing	MAC
Error Control	MAC
Access Control	MAC

OUTCOMES

Upon the completion of this session, the learner will be able to

- ★ Understand the sublayers of data link layer.
- ★ Know the responsibilities of LLC or DLC.
- ★ Know the responsibilities of MAC sublayer.

Framing

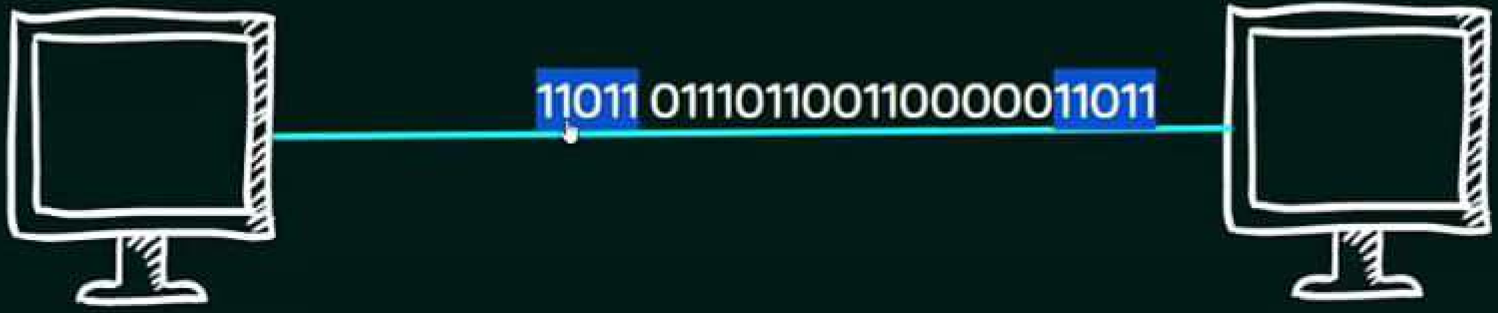


Bits flow between adaptors, frames between hosts

Framing

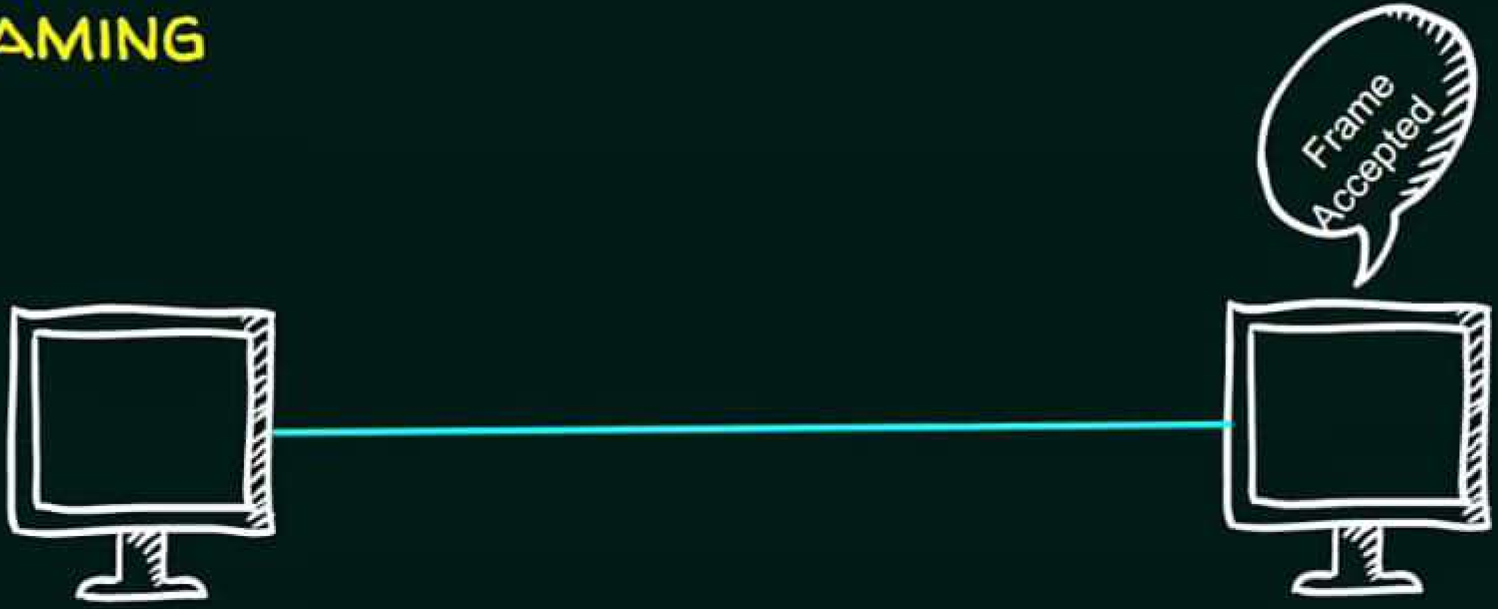
Protocol: Let the start of frame and end of frame be 11011

Framing



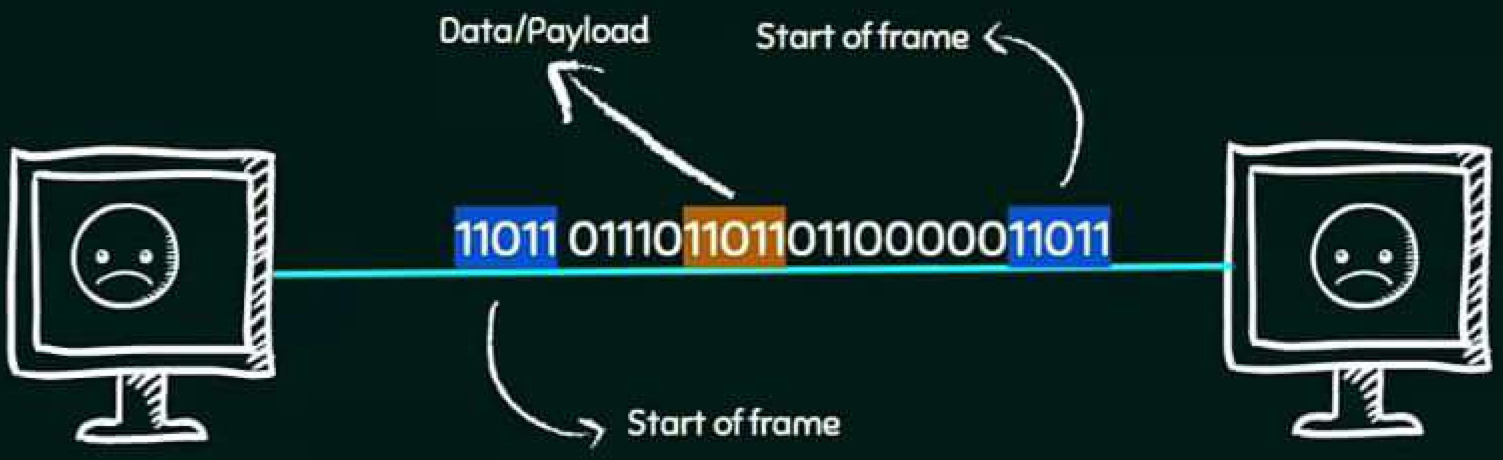
Let the start of frame and end of frame be 11011

FRAMING



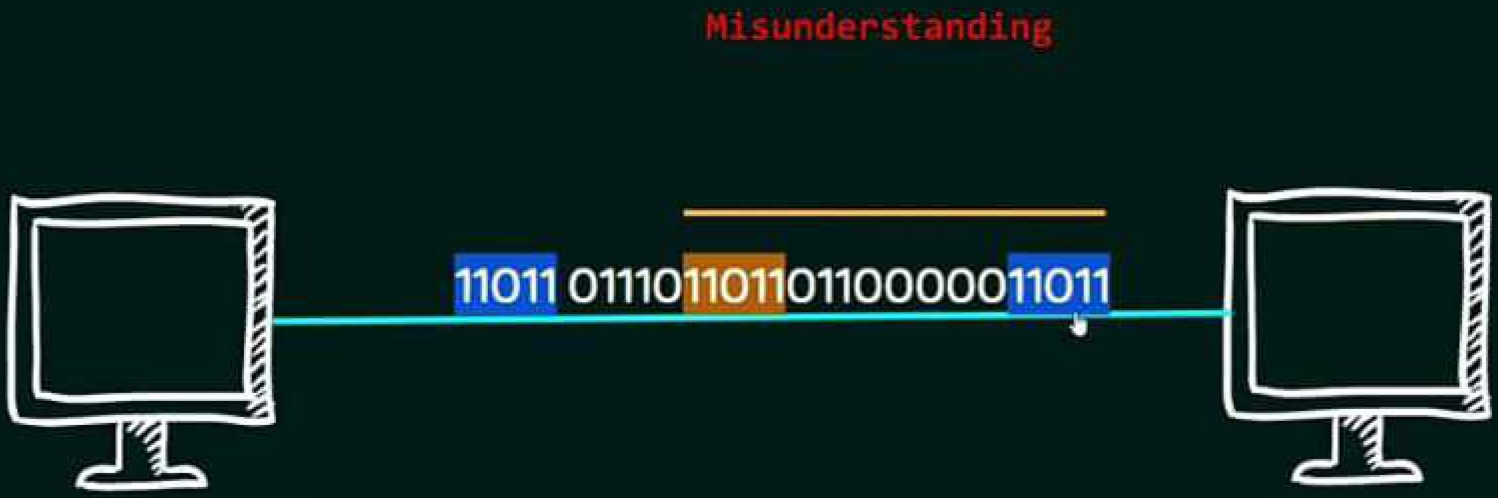
Let the start of frame and end of frame be 11011

FRAMING ERROR



Let the start of frame and end of frame be 11011
Data = Start of Frame or End of Frame

Framing error



Let the start of frame and end of frame be 11011

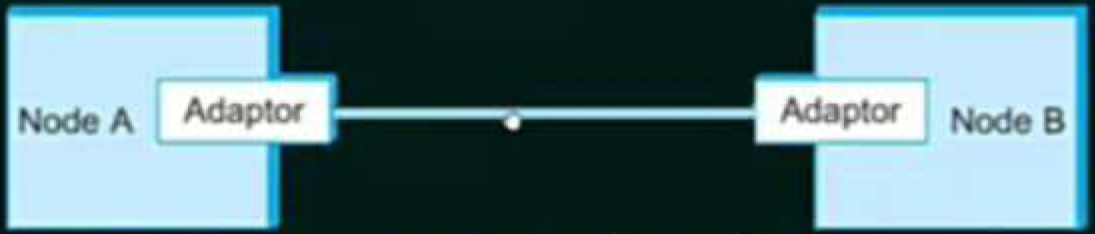
Outcomes

Upon the completion of this session, the learner will be able to

- ★ Understand framing.
- ★ Understand framing errors.

FRAMING

- ★ Framing in the data link layer separates a frame distinguishable from another frame.
- ★ Frame = Header + Network Layer PDU + Trailer.
- ★ In packet switched networks, the block of data called frames are exchanged between nodes, not bits streams.



FRAMING

- When node A wishes to transmit a frame to node B, it tells its adaptor to transmit a frame from the node's memory.
- This results in a sequence of bits being sent over the link.
- The adaptor on node B then collects together the sequence of bits arriving on the link and deposits the corresponding frame in B's memory.

Types of framing

1. Fixed-size framing.
2. Variable-size framing.

TYPES OF FRAMING

1. Fixed-size framing.

- ★ Here the size of the frame is fixed and so the frame length acts as delimiter of the frame.
- ★ Consequently, it does not require additional boundary bits to identify the start and end of the frame.

2. Variable-size framing.

- ★ Here, the size of each frame to be transmitted may be different.
- ★ So additional mechanisms are kept to mark the end of one frame and the beginning of the next frame.

Various framing approaches

OUTCOMES

Upon the completion of this session, the learner will be able to

- ★ Understand the basics of framing.
- ★ Understand the types of framing.
- ★ Know various framing approaches.

Bit oriented approach

- It simply views the frame as a collection of bits.
- In bit-oriented framing, data is transmitted as a sequence of bits that can be interpreted in the upper layers both as text as well as multimedia data.

BIT ORIENTED PROTOCOL

★ HDLC <-> High-Level Data Link Control

BYTE ORIENTED APPROACH

- One of the oldest approaches to framing.
- Here each frame is viewed as a collection of bytes (characters) rather than bits.
- a.k.a Character Oriented Approach.

Byte oriented protocols

- ★ BISYNC <-> Binary Synchronous Communication Protocol.
- ★ DDCMP <-> Digital Data Communication Message Protocol.
- ★ PPP <-> Point-to-Point Protocol

Outcomes

Upon the completion of this session, the learner will be able to

- ★ Know bit oriented approach and bit oriented protocols.
- ★ Know byte oriented approach and byte oriented protocols.
- ★ Know clock based framing.

BIT ORIENTED APPROACH

- ★ It simply views the frame as a collection of bits.

Bit Oriented Protocol

HDLC <-> High-Level Data Link Control

HDLC

- The Synchronous Data Link Control (SDLC) protocol developed by IBM is an example of a bit-oriented protocol.
- SDLC was later standardized by the ISO as the High-Level Data Link Control (HDLC) protocol.
- Bit Oriented Protocol.

HDLC – Frame format

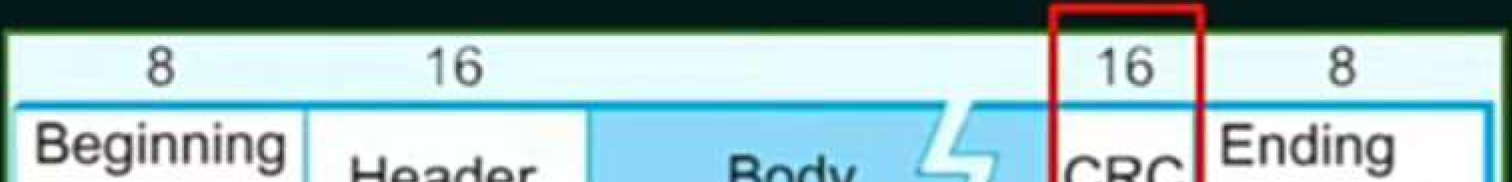
Beginning and Ending Sequences: 01111110

This sequence is also transmitted during any times that the link is idle so that the sender and receiver can keep their clocks synchronized.

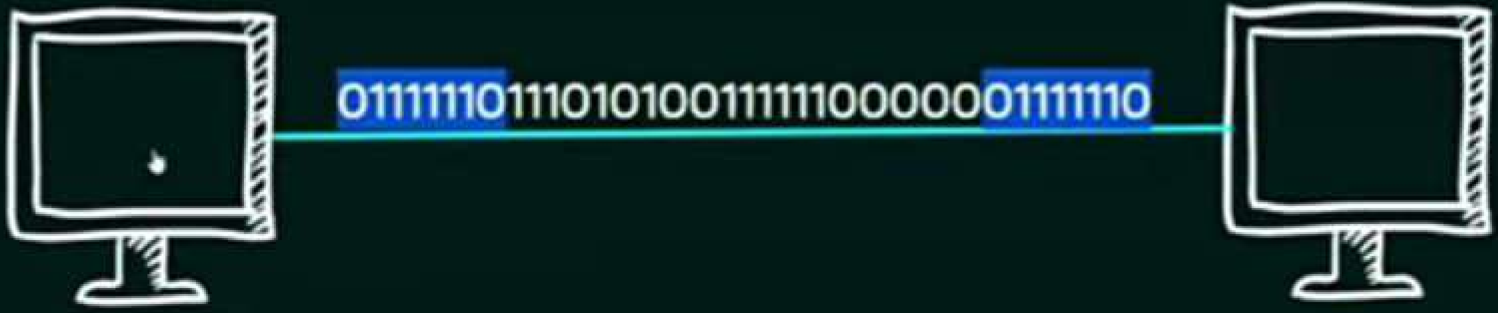
Header: Address and Control Field.

Body: Payload (Variable size)

CRC: Cyclic Redundancy check – Error Detection

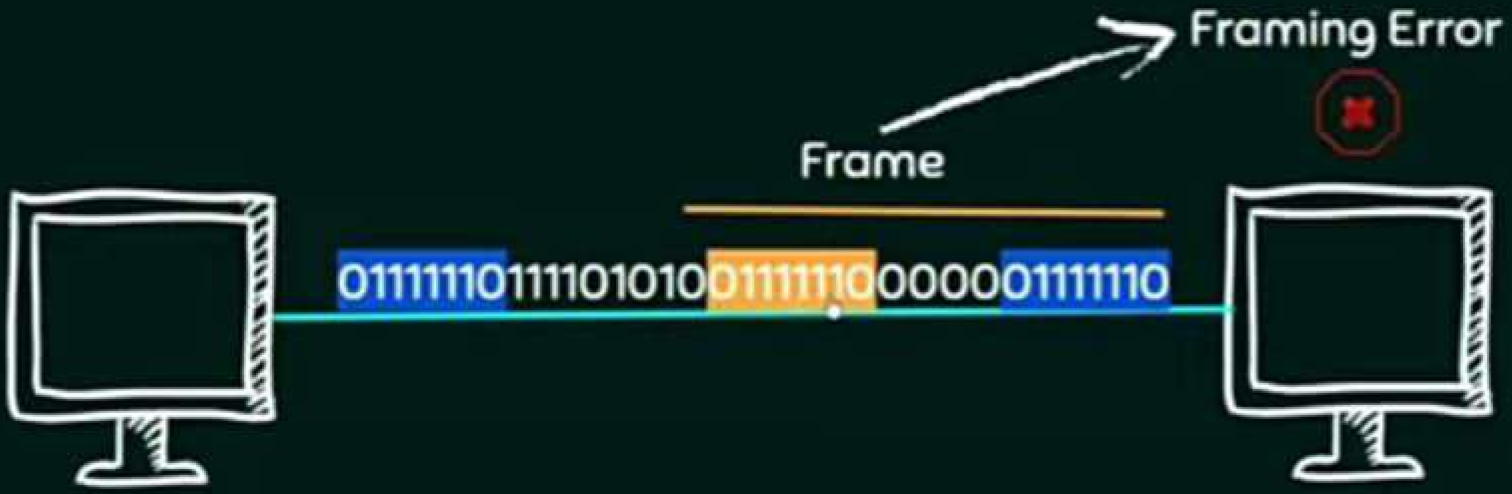


Problem here...



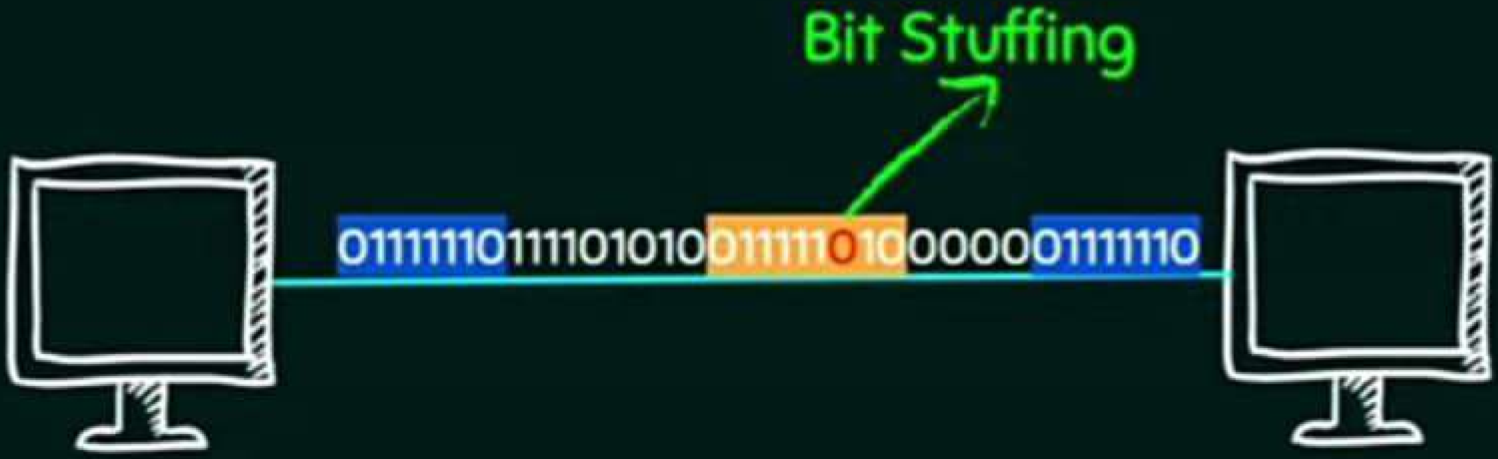
HDLC Protocol: Beginning and Ending Sequence is 01111110

PROBLEM



HDLC Protocol: Beginning and Ending Sequence is 01111110

Bit stuffing



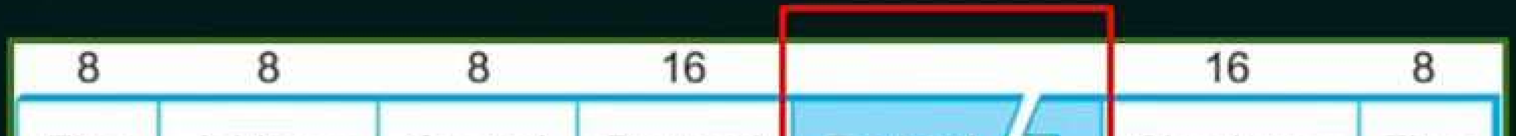
HDLC Protocol: Beginning and Ending Sequence is 01111110

PPP

- PPP is a data link layer protocol.
- PPP is a WAN protocol and which is commonly run over Internet links.
- It is widely used in broadband communications having heavy loads and high speeds.
- It is used to transmit multiprotocol data between two directly connected (point-to-point) computers.

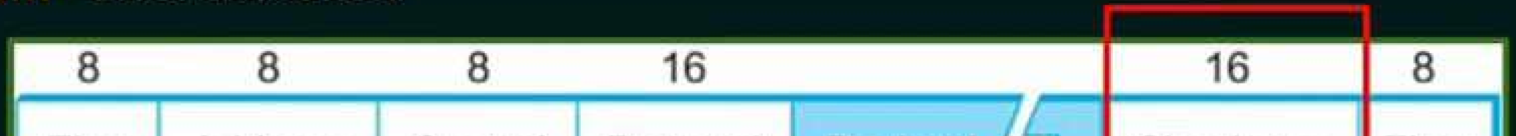
PPP – Frame format

- ★ Flag – 1 byte that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
- ★ Address – 1 byte which is set to 11111111 in case of broadcast.
- ★ Control – 1 byte set to a constant value of 11000000.
- ★ Protocol – 1 or 2 bytes that define the type of data contained in the payload field.
- ★ Payload – This carries the data from the network layer. The maximum length of the payload field is 1500 bytes. However, this may be negotiated between the endpoints of communication.



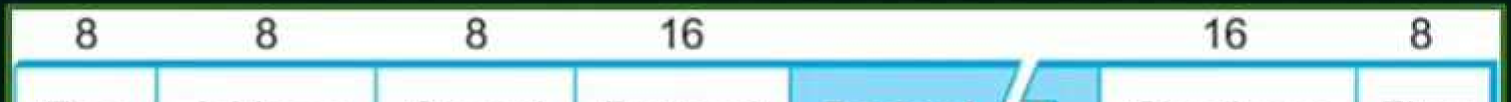
PPP – Frame format

- ★ Flag – 1 byte that marks the beginning and the end of the frame. The bit pattern of the flag is 0111110.
- ★ Address – 1 byte which is set to 1111111 in case of broadcast.
- ★ Control – 1 byte set to a constant value of 11000000.
- ★ Protocol – 1 or 2 bytes that define the type of data contained in the payload field.
- ★ Payload – This carries the data from the network layer. The maximum length of the payload field is 1500 bytes. However, this may be negotiated between the endpoints of communication.
- ★ Checksum – Error detection.



CHARACTER STUFFING

Byte stuffing or Character stuffing is the process of adding one extra byte whenever there is a flag sequence appear in the payload.

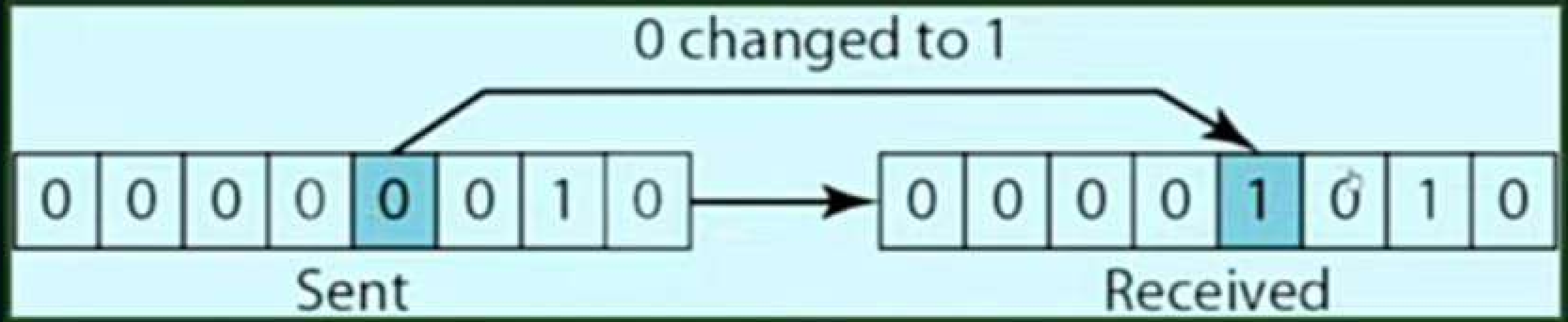


Error

- ★ Data are transmitted in the network.
- ★ The data can be corrupted during transmission.
- ★ Transmission error.
- ★ For reliable communication, errors must be detected and corrected.
- ★ Error detection and correction are implemented either at the data link layer or the transport layer of the OSI model.

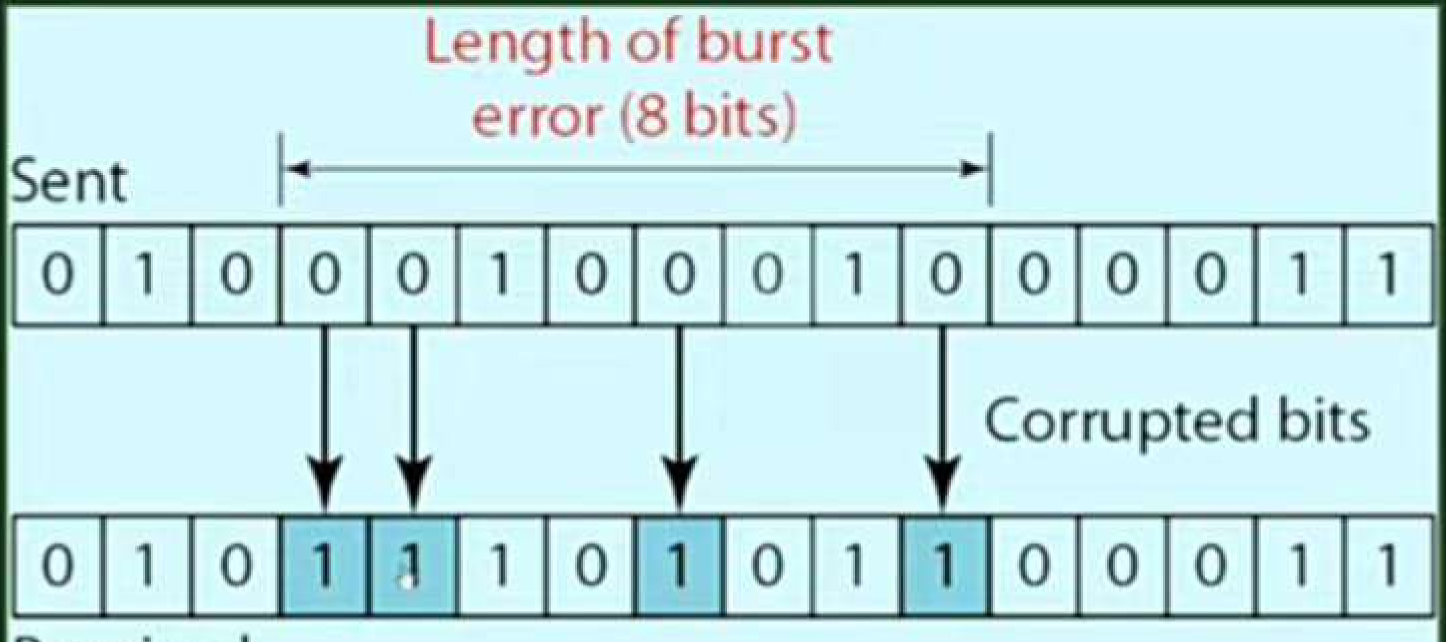
BIT ERROR

- a.k.a single bit error.
- In a single bit error, only 1 bit in the data unit has been changed.



Burst error

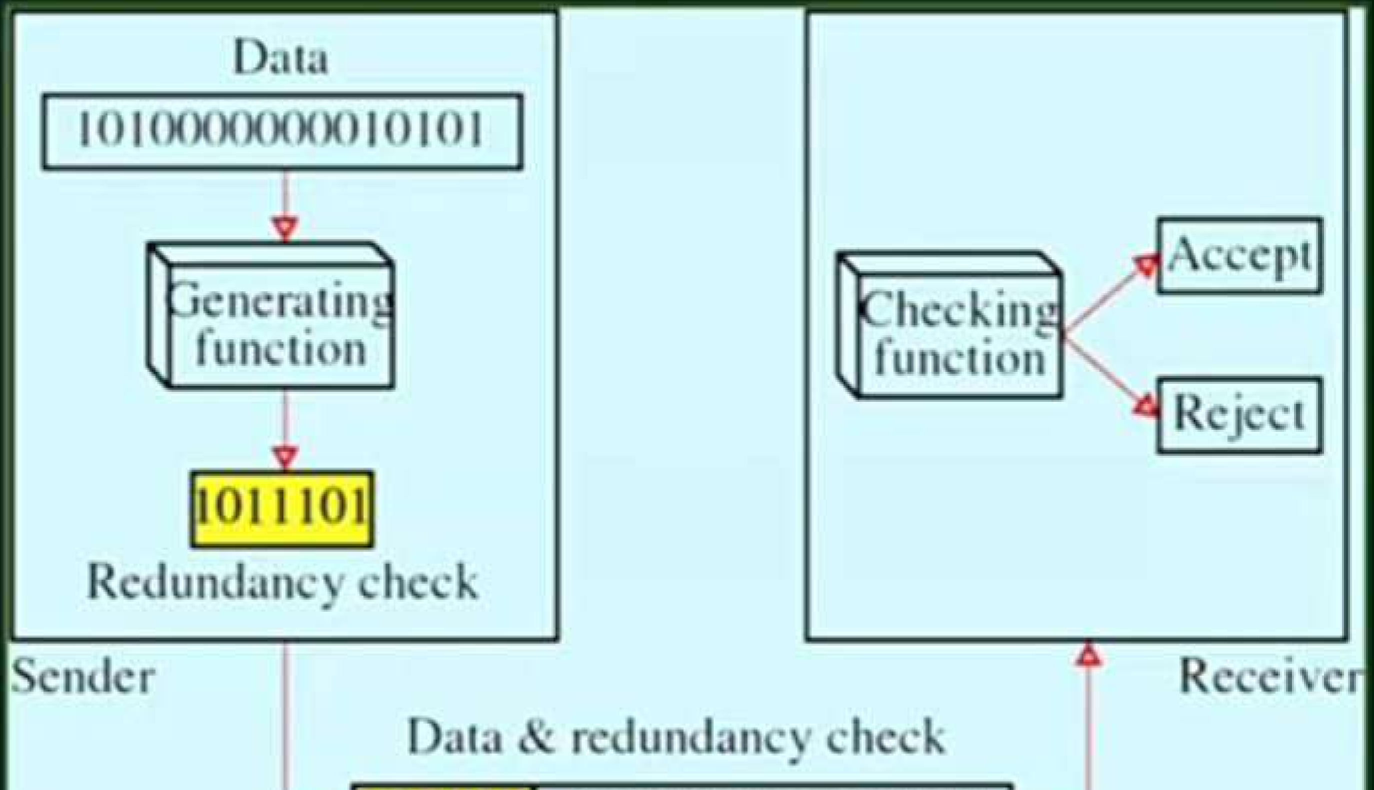
★ In burst error, 2 or more bits in the data unit have changed.



HOW TO DETECT THE ERRORS?

- ★ Error detection means to decide whether the received data is correct or not without having a copy of the original message.
- ★ To detect or correct errors, we need to send some extra bits with the data.
- ★ The extra bits are called as redundant bits.

Redundancy



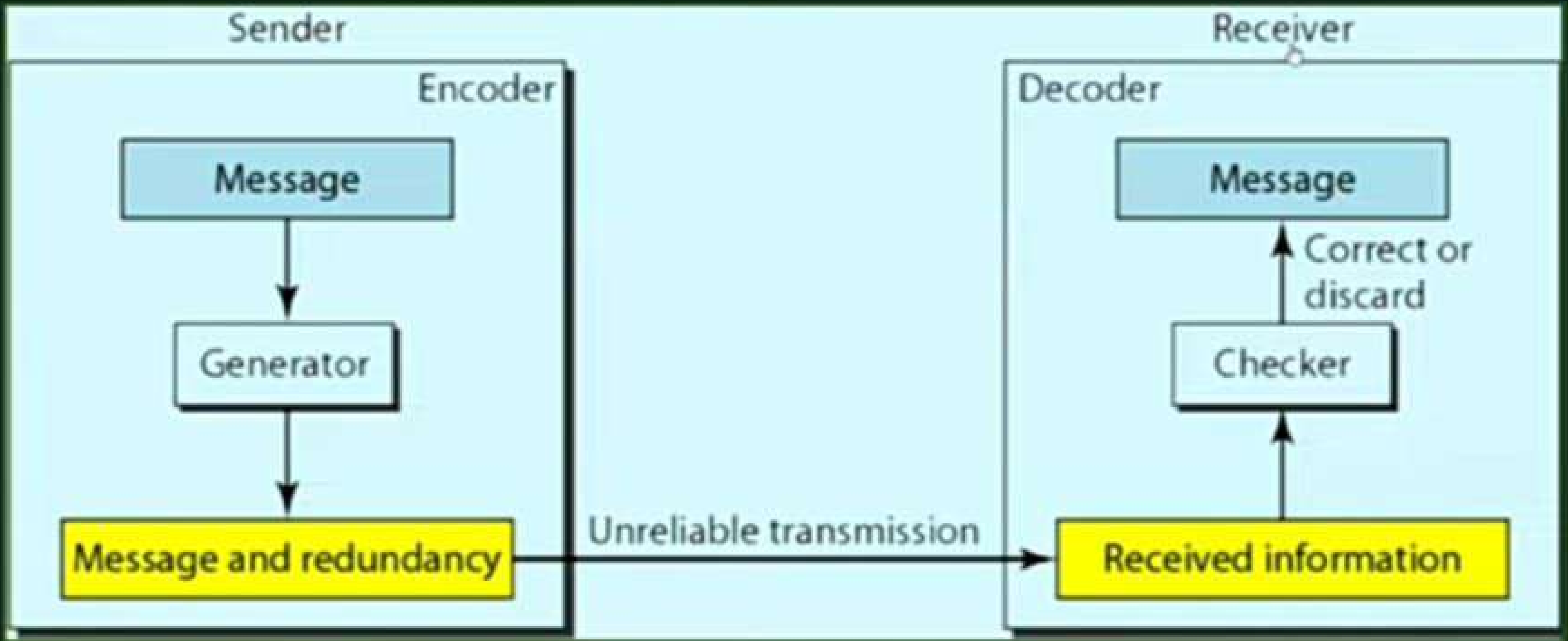
Data
1010000000010101
Generating function
1011101
Redundancy check
Sender
Checking function
Accept
Reject
Receiver
Data & redundancy check

Error correction

It can be handled in two ways:

1) Receiver can have the sender retransmit the entire data unit.
2) The receiver can use an error-correcting code, which automatically corrects certain errors.

Error detection/correction



Error detection techniques

Four types of redundancy checks are used in data communications. They are:

1. Vertical Redundancy Check (VRC)
2. Longitudinal Redundancy Check (LRC)
3. Checksum
4. Cyclic Redundancy Check (CRC)

Checksum

Checksum = Check + sum.

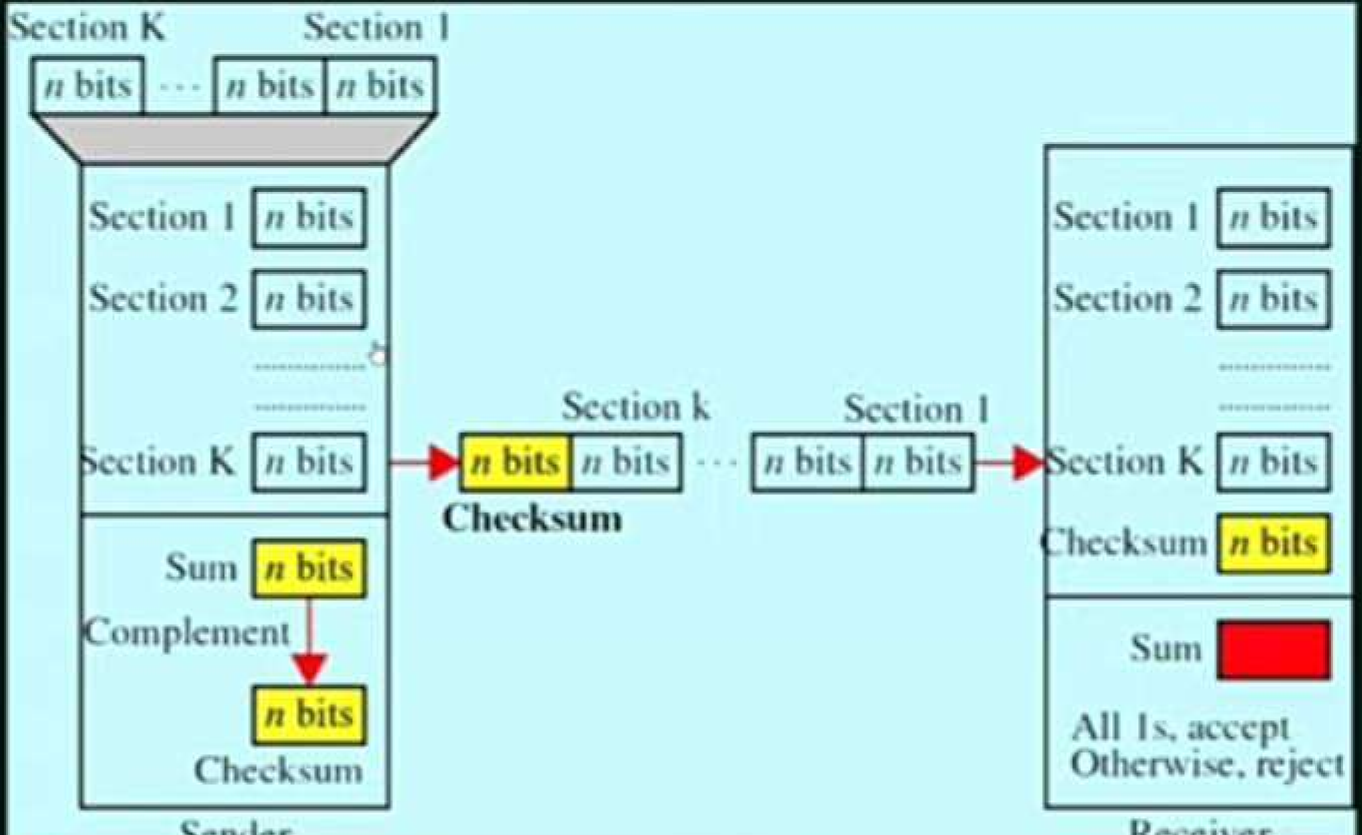
Sender side – Checksum Creation.

Receiver side – Checksum Validation.

CHECKSUM – OPERATION AT SENDER SIDE

1. Break the original message in to 'k' number of blocks with 'n' bits in each block.
2. Sum all the 'k' data blocks.
3. Add the carry to the sum, if any.
4. Do 1's complement to the sum = Checksum.

Checksum



Section K
Section 1
n bits
n bits
n bits
Section 1
n bits
Section 2
n bits
Section K
n bits
Sum
n bits
Complement
n bits
Checksum
Sender
Section k
Section 1
n bits
n bits
n bits
n bits
Checksum
Section 1
n bits
Section 2
n bits
Section K
n bits
Checksum
n bits
Sum
All 1s, accept
Otherwise, reject
Receiver

Checksum – Example

Consider the data unit to be transmitted is:

10011001111000100010010010000100

10011001	11100010	00100100	10000100

Checksum – Example

10011001   11100010   00100100   10000100

Carry		1	1	1	1	1			
		1	0	0	0	0	1	0	0
		0	0	1	0	0	1	0	0
		1	1	1	0	0	0	1	0
		1	0	0	1	1	0	0	1
1	0	0	0	1	0	0	0	1	1

Checksum – Example

10011001 11100010 00100100 10000100

Carry	1	1	1	1	1			
	1	0	0	0	0	1	0	0
	0	0	1	0	0	1	0	0
	1	1	1	0	0	0	1	0
	1	0	0	1	1	0	0	1
	0	0	1	0	0	0	1	1
							1	0
	0	0	1	0	0	1	0	1

CHECKSUM – EXAMPLE

10011001 11100010 00100100 10000100

Carry	1	1	1	1	1			
	1	0	0	0	0	1	0	0
	0	0	1	0	0	1	0	0
	1	1	1	0	0	0	1	0
	1	0	0	1	1	0	0	1
	0	0	1	0	0	0	1	1
							1	0
	0	0	1	0	0	1	0	1

CHECKSUM – EXAMPLE

11011010 10011001 11100010 00100100 10000100

Carry	1	1	1	1	1			
	1	0	0	0	0	1	0	0
	0	0	1	0	0	1	0	0
	1	1	1	0	0	0	1	0
	1	0	0	1	1	0	0	1
	0	0	1	0	0	0	1	1
							1	0
	0	0	1	0	0	1	0	1

Checksum – Example

11011010 10011001 11100010 00100100 10000100

CHECKSUM – OPERATION AT RECEIVER SIDE

- ★ Collect all the data blocks including the checksum.
- ★ Sum all the data blocks and checksum
- ★ If the result is all 1's, ACCEPT; Else, REJECT.

Checksum – Example

11011010 10011001 11100010 00100100 10000100

1	0	0	0	0	1	0	0
0	0	1	0	0	1	0	0
1	1	1	0	0	0	1	0
1	0	0	1	1	0	0	1
1	1	0	1	1	0	1	0

Checksum – Example

11011010 10011001 11100010 00100100 10000100

Carry			1	1	1	1	1	1		
			1	0	0	0	0	1	0	0
			0	0	1	0	0	1	0	0
			1	1	1	0	0	0	1	0
			1	0	0	1	1	0	0	1
			1	1	0	1	1	0	1	0
	1	0	1	1	1	1	1	1	0	1
									1	0

Checksum – Example

11011010 10011001 11100010 00100100 10000100

Carry	1	1	1	1	1	1		
	1	0	0	0	0	1	0	0
	0	0	1	0	0	1	0	0
	1	1	1	0	0	0	1	0
	1	0	0	1	1	0	0	1
	1	1	0	1	1	0	1	0
	1	1	1	1	1	1	0	1
							1	0

PERFORMANCE OF CHECKSUM

- The checksum detects all errors involving an odd number of bits.
- It detects most errors involving an even number of bits.
- If one or more bits of a segment are damaged and the corresponding bit or bits of opposite value in a second segment are also damaged, the sums of those columns will not change and the receiver will not detect the error(s).