Parul® University

Data Communication and Computer Networks

Dr. Kamini Solanki, Associate Professor
Faculty of IT & Computer Science

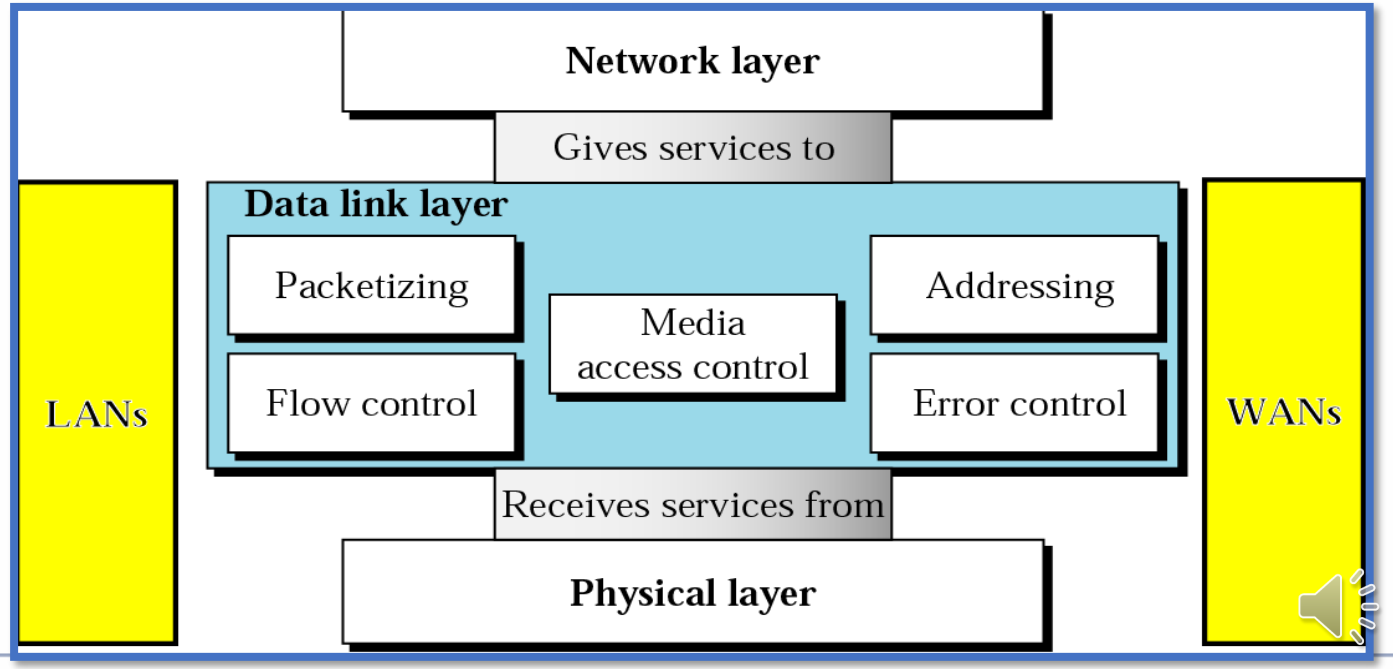
CHAPTER-4

The Data Link Layer

Overview of DLL

The data link layer transforms the physical layer, a raw transmission facility, to a link responsible for node-to-node (hop-to-hop) communication. Specific responsibilities of the data link layer include framing, addressing, flow control, error control, and media access control.

- Data link layer performs the most reliable node to node delivery of data.
- It forms frames from the packets that are received from network layer and gives it to physical layer.
- It also synchronizes the information which is to be transmitted over the data. Error controlling is easily done. The encoded data are then passed to physical.



Overview of DLL

- The main task of the **data link layer** is to transform a raw transmission facility into a line that appears free of undetected transmission errors to the network layer.
- It accomplishes this task by having the sender break up the input data into **data frames**(typically a few hundred or few thousand bytes) and transmit the frames sequentially.
- If the service is reliable, the receiver confirms correct receipt of each frame by send back an **acknowledgement frame**.
- Data link layer works between two hosts which are directly connected in some sense.
- This direct connection could be point to point or broadcast. Systems on broadcast network are said to be on same link.
- The work of data link layer tends to get more complex when it is dealing with multiple hosts on single collision domain.

Overview of DLL

- Data link layer is responsible for converting data stream to signals bit by bit and to send that over the underlying hardware.
- At the receiving end, Data link layer picks up data from hardware which are in the form of electrical signals, assembles them in a recognizable frame format, and hands over to upper layer.
- Data link layer has two sub-layers:
  - **Logical Link Control:** It deals with protocols, flow-control, and error control
  - **Media Access Control:** It deals with actual control of media

Functions of DLL

1. **Framing -** Data-link layer takes packets from Network Layer and encapsulates them into Frames. Then, it sends each frame bit-by-bit on the hardware. At receiver' end, data link layer picks up signals from hardware and assembles them into frames.
2. **Addressing -** Data-link layer provides layer-2 hardware addressing mechanism. Hardware address is assumed to be unique on the link. It is encoded into hardware at the time of manufacturing.
3. **Synchronization -** When data frames are sent on the link, both machines must be synchronized in order to transfer to take place.
4. **Error Control -** Sometimes signals may have encountered problem in transition and the bits are flipped. These errors are detected and attempted to recover actual data bits. It also provides error reporting mechanism to the sender.
5. **Flow Control -** Stations on same link may have different speed or capacity. Data-link layer ensures flow control that enables both machine to exchange data on same speed.

Data Link Layer Issues

The data link layer in the OSI (Open System Interconnections) Model, is in between the physical layer and the network layer. This layer converts the raw transmission facility provided by the physical layer to a reliable and error-free link.

The main functions and the design issues of this layer are

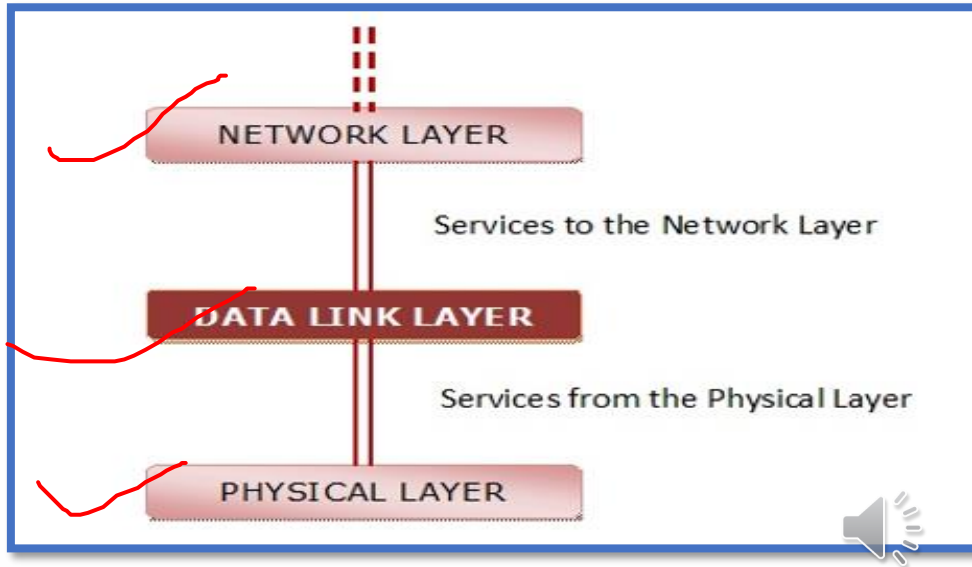
- Providing services to the network layer
- Framing
- Error Control
- Flow Control

Services to the Network Layer

- In the OSI Model, each layer uses the services of the layer below it and provides services to the layer above it. The data link layer uses the services offered by the physical layer.
- The primary function of this layer is to provide a well defined service interface to network layer above it.

The types of services provided can be of three types –

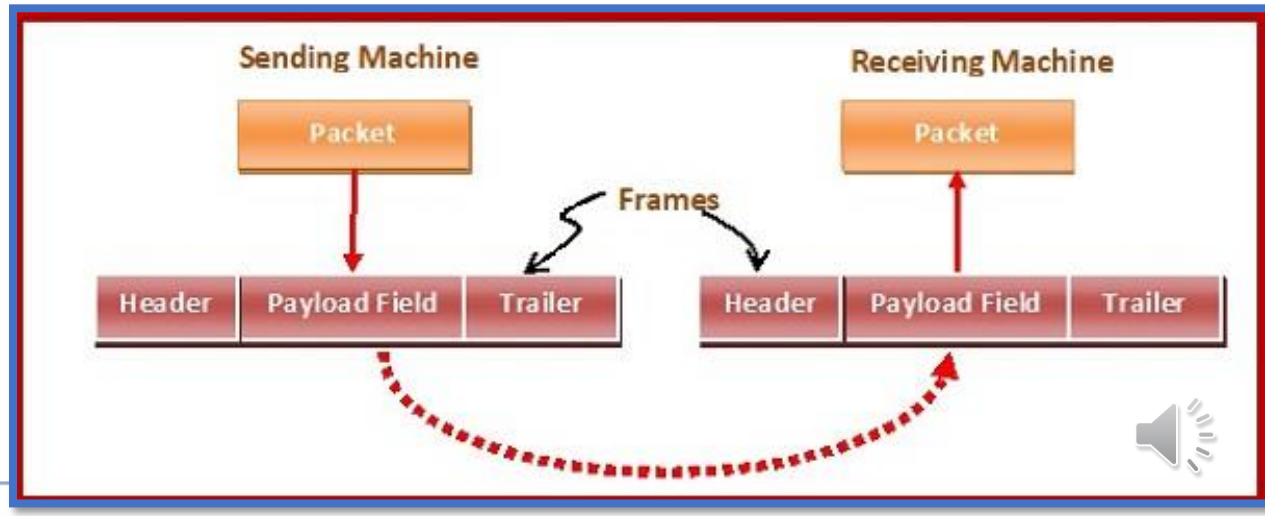
1. Unacknowledged connectionless service
2. Acknowledged connectionless service
3. Acknowledged connection - oriented service



Framing

- The data link layer encapsulates each data packet from the network layer into frames that are then transmitted.
- A frame has three parts, namely −

1. Frame Header
2. Payload field that contains the data packet from network layer
3. Trailer



Error Control

The data link layer ensures error free link for data transmission. The issues it caters to with respect to error control are –

- Dealing with transmission errors
- Sending acknowledgement frames in reliable connections
- Retransmitting lost frames
- Identifying duplicate frames and deleting them
- Controlling access to shared channels in case of broadcasting

Flow Control

The data link layer regulates flow control so that a fast sender does not drown a slow receiver. When the sender sends frames at very high speeds, a slow receiver may not be able to handle it. There will be frame losses even if the transmission is error-free. The two common approaches for flow control are −

- Feedback based flow control
- Rate based flow control

Parul®
University

DIGITAL LEARNING CONTENT

Data Link Layer – Revision

Data link layer Design Issues

Data Link Layer – Revision

Data link layer Design Issues

- Services provided to N/w layer
- Framing
- Error Control
- flow Control

Functions of DLL

Sending m/c

Packets

frame

Header	Payload field	Trailer

Receiving m/c

Packets

Header	Payload field

Error Detection

A condition when the receiver's information does not matches with the sender's information. During transmission, digital signals suffer from noise that can introduce errors in the binary bits travelling from sender to receiver. That means a 0 bit may change to 1 or a 1 bit may change to 0.

Error Detecting Codes (Implemented either at Data link layer or Transport Layer of OSI Model)
Whenever a message is transmitted, it may get scrambled by noise or data may get corrupted. To avoid this, we use error-detecting codes which are additional data added to a given digital message to help us detect if any error has occurred during transmission of the message.

Error Detection

Basic approach used for error detection is the use of redundancy bits, where additional bits are added to facilitate detection of errors.

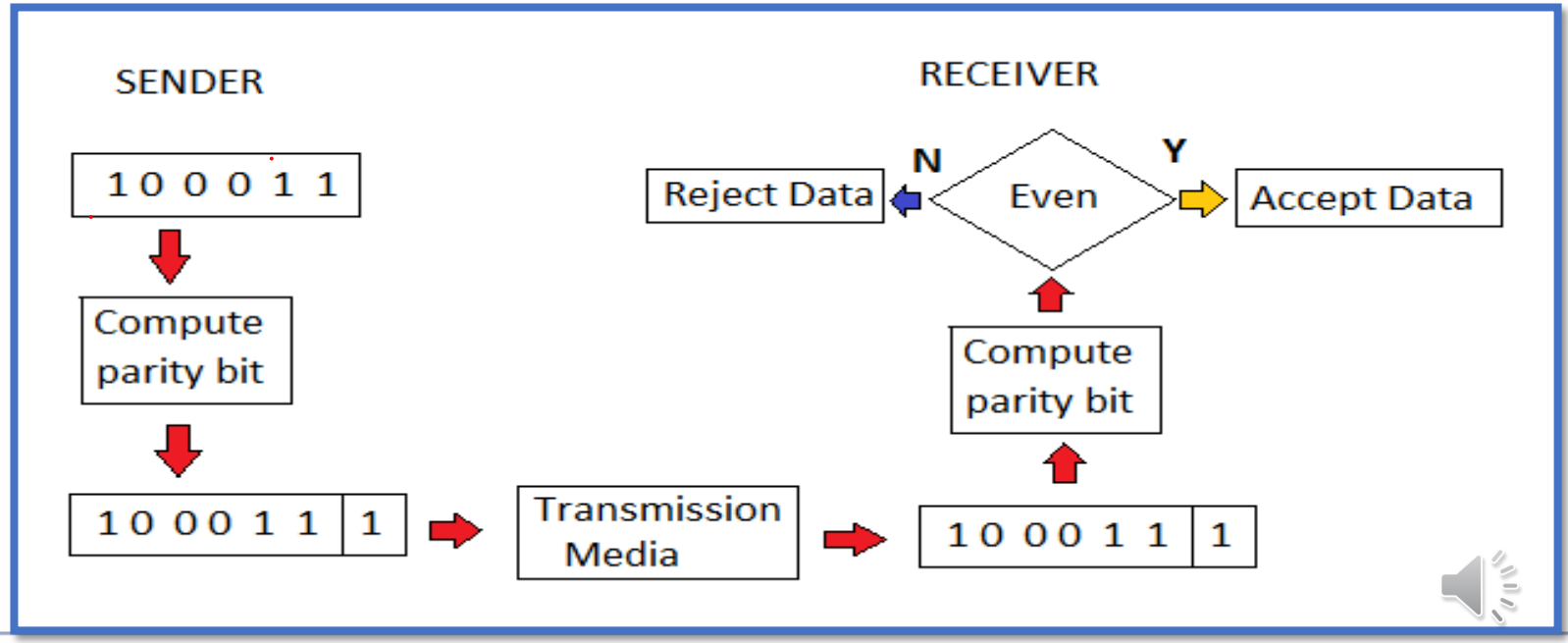
Some popular techniques for error detection are:
1. Simple Parity check
2. Two-dimensional Parity check
3. Checksum
4. Cyclic redundancy check

Error Detection

1. Simple Parity check

Blocks of data from the source are subjected to a check bit or parity bit generator form, where a parity of :

- 1 is added to the block if it contains odd number of 1's, and
- 0 is added if it contains even number of 1's



Error Detection

2. Two-dimensional Parity check

Parity check bits are calculated for each row, which is equivalent to a simple parity check bit. Parity check bits are also calculated for all columns, then both are sent along with the data. At the receiving end these are compared with the parity bits calculated on the received data.

Original Data

10011001	11100010	00100100	10000100

Row parities

10011001	0
11100010	0
00100100	0
10000100	0
11011011	0

Column parities

100110010	111000100	001001000	100001000	110110110

Data to be sent

Error Detection

3. Checksum

- In checksum error detection scheme, the data is divided into k segments each of m bits.
- In the sender's end the segments are added using 1's complement arithmetic to get the sum. The sum is complemented to get the checksum.
- The checksum segment is sent along with the data segments.
- At the receiver's end, all received segments are added using 1's complement arithmetic to get the sum. The sum is complemented.
- If the result is zero, the received data is accepted; otherwise discarded.

Error Detection - Checksum

Original Data

10011001	11100010	00100100	10000100
1	2	3	4

k=4, m=8

Sender

```
1        1 0 0 1 1 0 0 1
2        1 1 1 0 0 0 1 0
       (1)0 1 1 1 1 0 1 1
                       1
         0 1 1 1 1 1 0 0
3        0 0 1 0 0 1 0 0
         1 0 1 0 0 0 0 0
4        1 0 0 0 0 1 0 0
       (1)0 0 1 0 0 1 0 0
                       1
Sum:     0 0 1 0 0 1 0 1
CheckSum: 1 1 0 1 1 0 1 0
```

Reciever

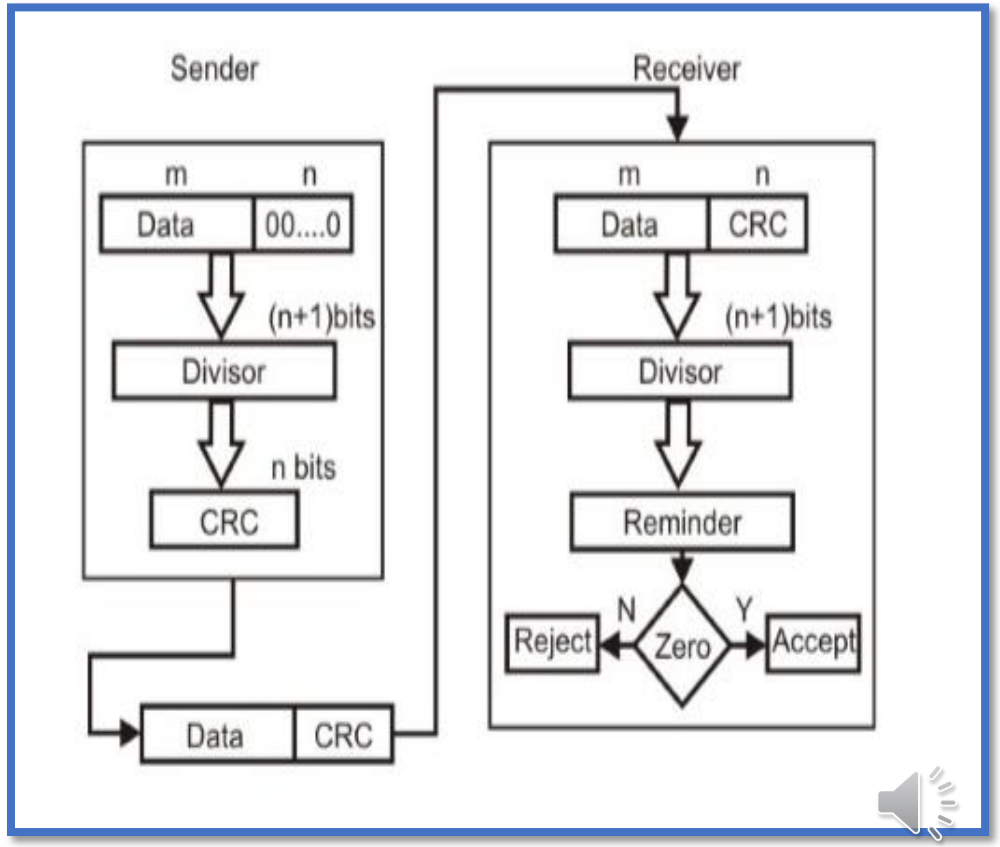
```
1        1 0 0 1 1 0 0 1
2        1 1 1 0 0 0 1 0
       (1)0 1 1 1 1 0 1 1
                       1
         0 1 1 1 1 1 0 0
3        0 0 1 0 0 1 0 0
         1 0 1 0 0 0 0 0
4        1 0 0 0 0 1 0 0
       (1)0 0 1 0 0 1 0 0
                       1
         0 0 1 0 0 1 0 1
         1 1 0 1 1 0 1 0
Sum:     1 1 1 1 1 1 1 1
Complement: 0 0 0 0 0 0 0 0
```

Conclusion: Accept Data

Error Detection - Cyclic redundancy check (CRC)

4. Cyclic redundancy check (CRC)

- Unlike checksum scheme, which is based on addition, CRC is based on binary division.
- In CRC, a sequence of redundant bits, called cyclic redundancy check bits, are appended to the end of data unit so that the resulting data unit becomes exactly divisible by a second, predetermined binary number.
- At the destination, the incoming data unit is divided by the same number. If at this step there is no remainder, the data unit is assumed to be correct and is therefore accepted.
- A remainder indicates that the data unit has been damaged in transit and therefore must be rejected.



Error Detection – Example

original message
1 0 1 0 0 0 0

@ means X-OR

Generator polynomial
x^3+1
$1.x^3+0.x^2+0.x^1+1.x^0$
CRC generator
1 0 0 1 4-bit

If CRC generator is of n bit then append (n-1) zeros in the end of original message

Sender

```
1 0 0 1 | 1 0 1 0 0 0 0 0 0 0
        @ 1 0 0 1
        -----------
          0 0 1 1 0 0 0 0 0 0
            @ 1 0 0 1
            ---------
              0 1 0 1 0 0 0 0
                @ 1 0 0 1
                ---------
                  0 0 1 1 0 0 0
                    @ 1 0 0 1
                    ---------
                      0 1 0 1 0
                        @ 1 0 0 1
                        ---------
                          0 0 1 1
```

Message to be transmitted

```
1 0 1 0 0 0 0 0 0 0 0
          + 0 1 1
-----------------------
1 0 1 0 0 0 0 0 1 1
```

```
1 0 0 1 | 1 0 1 0 0 0 0 0 1 1
        @ 1 0 0 1
        -----------
          0 0 1 1 0 0 0 0 1 1
            @ 1 0 0 1
            ---------
              0 1 0 1 0 0 1 1      <- Receiver
                @ 1 0 0 1
                ---------
                  0 0 1 1 0 1 1
                    @ 1 0 0 1
                    ---------
                      0 1 0 0 1
                        @ 1 0 0 1
                        ---------
                          0 0 0 0
```

Zero means data is accepted

Parul®
University

DIGITAL LEARNING CONTENT

Error Detection – Revision

COMPUTER NETWORKS

A Bottom up approach

Error Detection

NESO ACADEMY

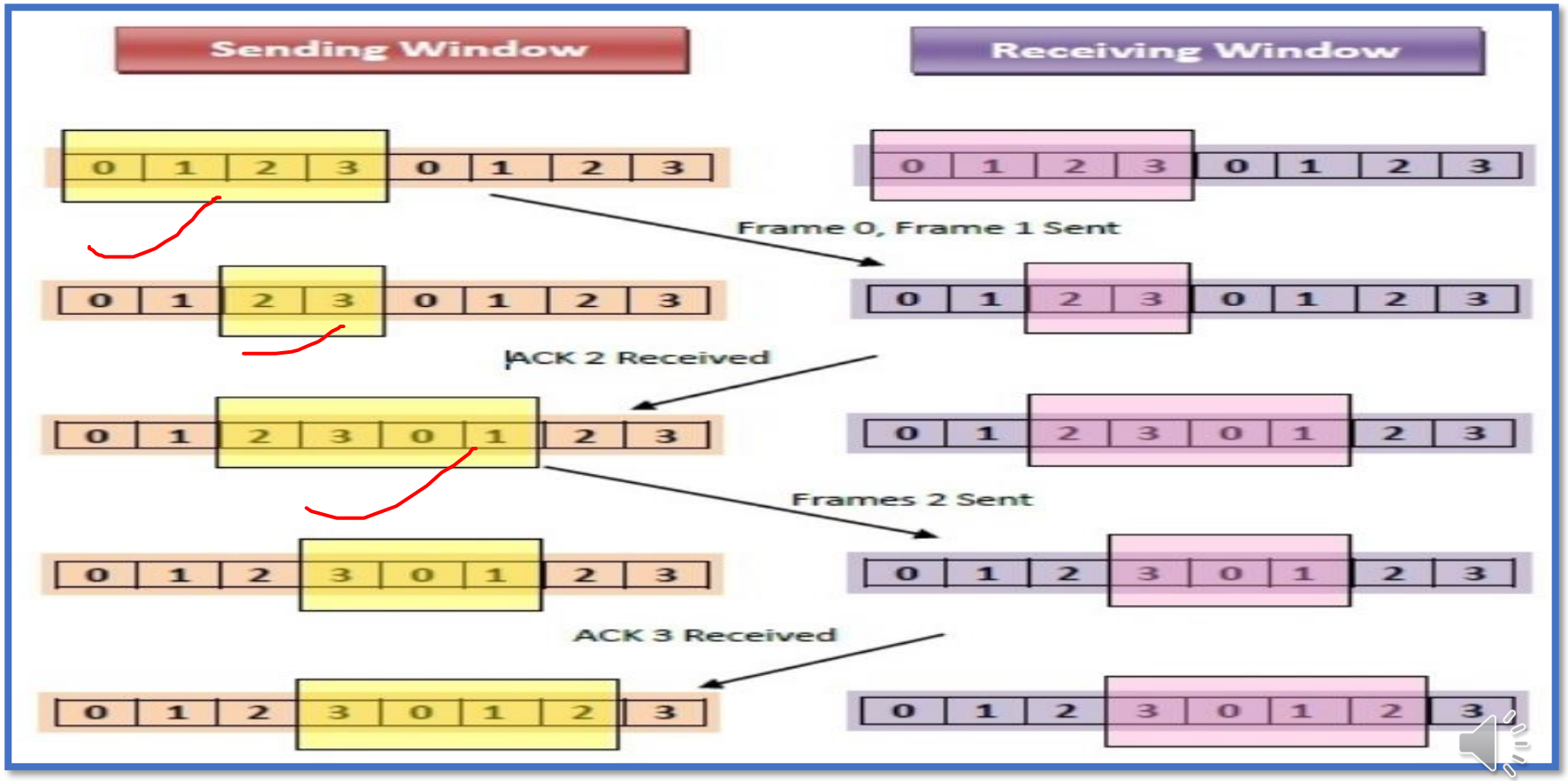
Sliding Window Protocol

The sliding window is a technique for sending multiple frames at a time. It controls the data packets between the two devices where reliable and gradual delivery of data frames is needed. It is also used in TCP (Transmission Control Protocol).

In this technique, each frame has sent from the sequence number. The sequence numbers are used to find the missing data in the receiver end. The purpose of the sliding window technique is to avoid duplicate data, so it uses the sequence number.

Sliding Window Protocol – Example

Suppose that we have sender window and receiver window each of size 4. So the sequence numbering of both the windows will be 0,1,2,3,0,1,2 and so on. The following diagram shows the positions of the windows after sending the frames and receiving acknowledgments



Parul®
University

DIGITAL LEARNING CONTENT

Sliding Window Protocol – Example

COMPUTER NETWORKS

A Bottom up approach

Sliding Window Protocol

NESO ACADEMY

DIGITAL LEARNING CONTENT

www.paruluniversity.ac.in