MODULE 2:

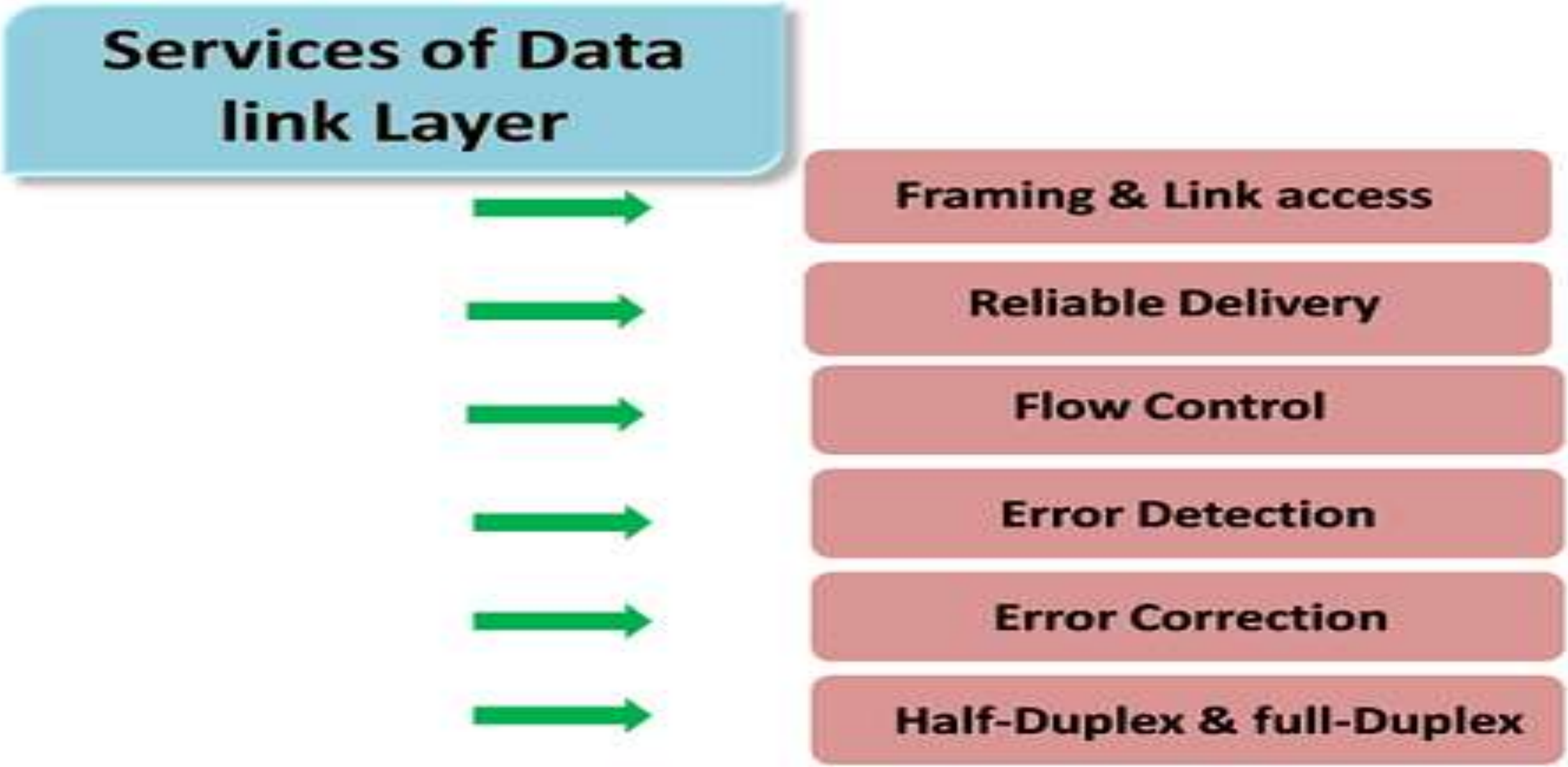
Data Link Layer

Links, Access Networks, and LANs- Introduction to the Link Layer, The Services Provided by the Link Layer, Types of errors, Redundancy, Detection vs Correction, Forward error correction Versus Retransmission Error-Detection and Correction Techniques, Parity Checks, Check summing Methods, Cyclic Redundancy Check (CRC) , Framing, Flow Control and Error Control protocols , Noisy less Channels and Noisy Channels, HDLC, Multiple Access Protocols, Random Access, ALOHA, Controlled access, Channelization Protocols. 802.11 MAC Protocol, IEEE 802.11 Frame.

Introduction to the Link Layer

- In the OSI model, the data link layer is a 6th layer from the top and 2nd layer from the bottom.
- The communication channel that **connects the adjacent nodes is known as links**, and in order to move the datagram from source to the destination, the datagram must be moved across an individual link.
- The main responsibility of the Data Link Layer is to transfer the datagram across an individual link.
- The Data link layer protocol defines the format of the packet exchanged across the nodes as well as the actions such as Error detection, retransmission, flow control, and random access.
- The Data Link Layer protocols are Ethernet, token ring, FDDI and PPP.
- An important characteristic of a Data Link Layer is that datagram can be handled by different link layer protocols on different links in a path.
- For example, the datagram is handled by Ethernet on the first link, PPP on the second link.

Services are provided by the Data Link Layer:



•**Framing & Link access:**

- Data Link Layer protocols encapsulate each network frame within a Link layer frame before the transmission across the link.
- A frame consists of a data field in which network layer datagram is inserted and a number of data fields.
- It specifies the structure of the frame as well as a channel access protocol by which frame is to be transmitted over the link.

•**Reliable delivery:**

- Data Link Layer provides a reliable delivery service, i.e., transmits the network layer datagram without any error.
- A reliable delivery service is accomplished with transmissions and acknowledgements.
- A data link layer mainly provides the reliable delivery service over the links as they have higher error rates and they can be corrected locally, link at which an error occurs rather than forcing to retransmit the data.

- **Flow control:**
  - A receiving node can receive the frames at a faster rate than it can process the frame.
  - Without flow control, the receiver's buffer can overflow, and frames can get lost.
  - To overcome this problem, the data link layer uses the flow control to prevent the sending node on one side of the link from overwhelming the receiving node on another side of the link.

- **Error detection:**
  - Errors can be introduced by signal attenuation and noise.
  - Data Link Layer protocol provides a mechanism to detect one or more errors.
  - This is achieved by adding error detection bits in the frame and then receiving node can perform an error check.

•Error correction:

Error correction is similar to the Error detection, except that receiving node not only detect the errors but also determine where the errors have occurred in the frame.

•Half-Duplex & Full-Duplex:

In a Full-Duplex mode, both the nodes can transmit the data at the same time.

In a Half-Duplex mode, only one node can transmit the data at the same time.

Data-link layer uses error control techniques to ensure that frames, i.e. bit streams of data, are transmitted from the source to the destination with a certain extent of accuracy.

Errors

When bits are transmitted over the computer network, they are subject to get corrupted due to interference and network problems.

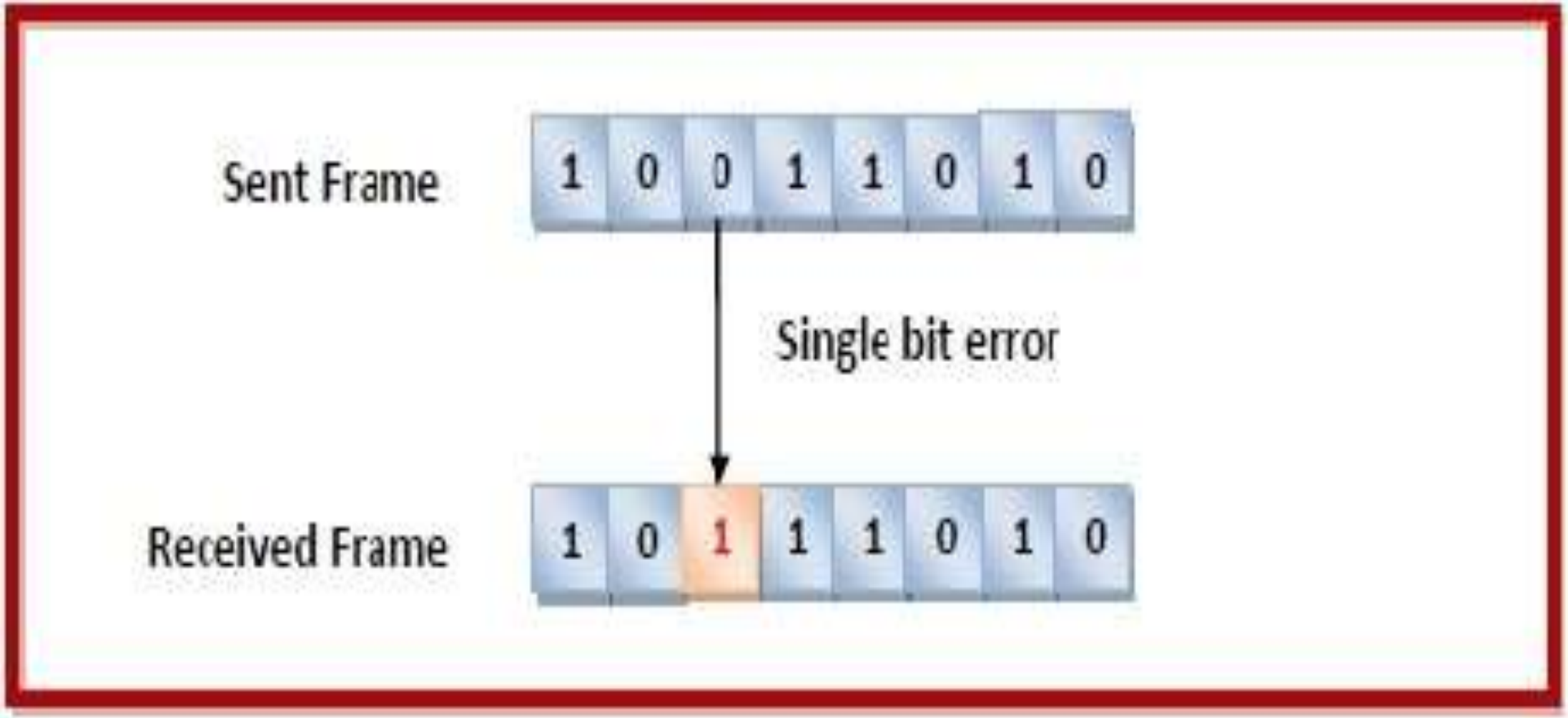
The corrupted bits leads to spurious data being received by the destination and are called errors.

Types of Errors

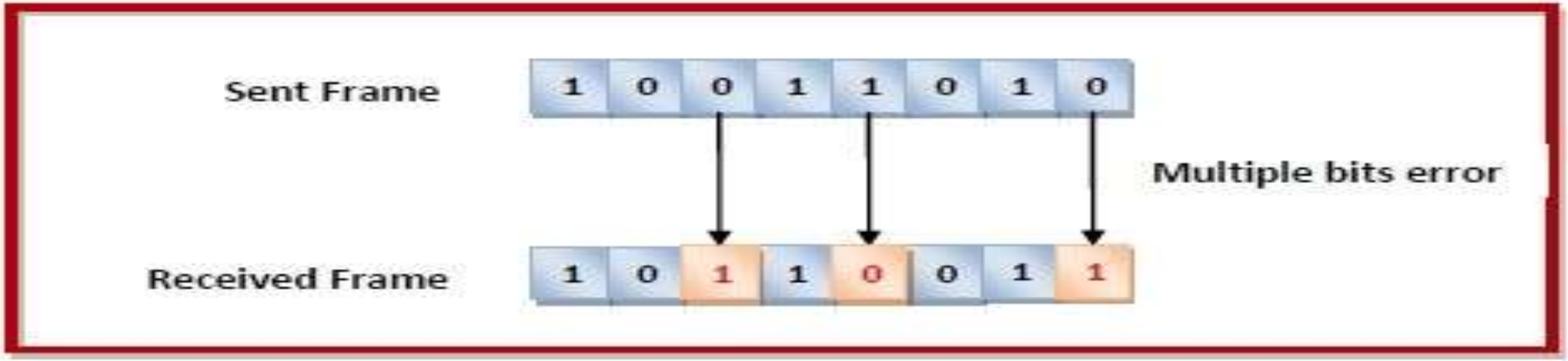
Errors can be of three types, namely

- ✓ single bit errors,
- ✓ multiple bit errors,
- ✓ and burst errors.

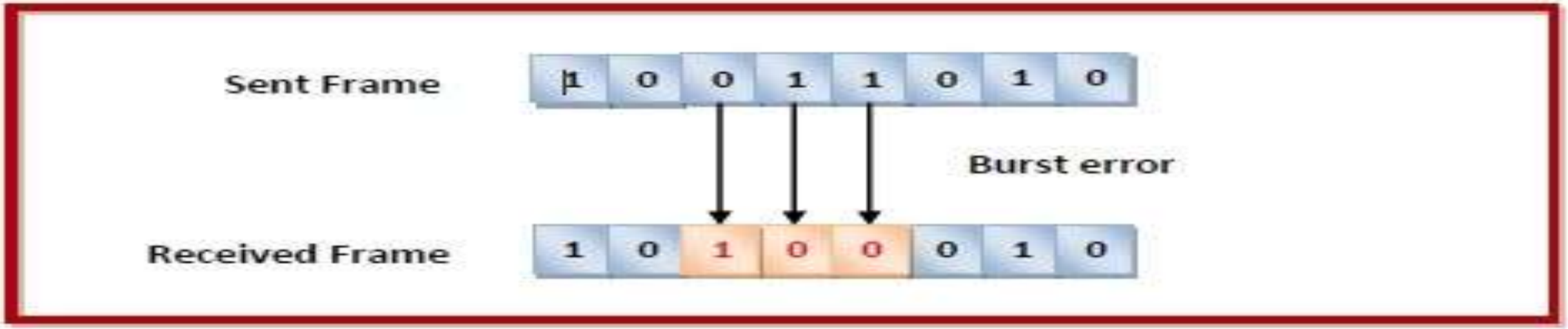
Single bit error − In the received frame, only one bit has been corrupted, i.e. either changed from 0 to 1 or from



Multiple bits error − In the received frame, more than one bits are corrupted.

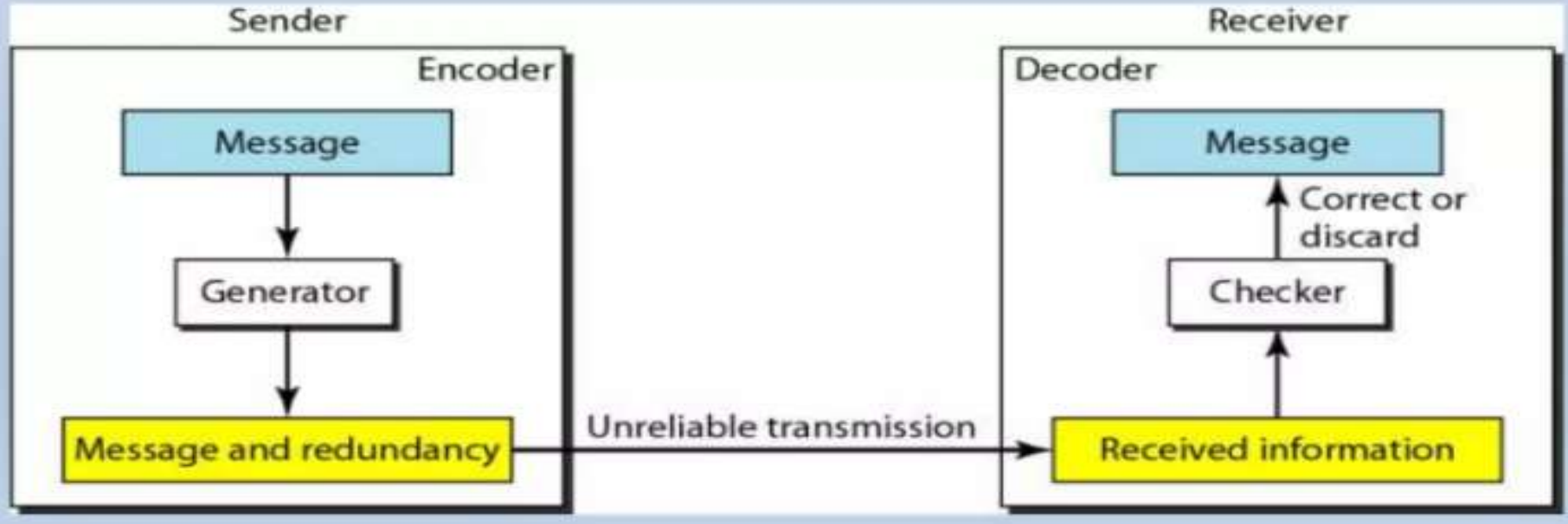


Burst error − In the received frame, more than one consecutive bits are corrupted.



Redundancy

- Error detection uses the concept of ***redundancy***, which means adding extra (redundant) bits for detecting errors at the destination
- These redundant bits are added by the sender and removed by the receiver



Error control

- **Detection VS Correction**
  - Detection: error ? yes or no
  - Correction: Need to know the exact number of bits that are corrupted, and their location in the message
- **Forward Error Correction VS Retransmission(backward error correction)**

- Forward error correction is the process in which the receiver tries to guess the message by using redundant bits
- if the number of **errors is small.** Correction by retransmission is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message**. Resending** is repeated until a message arrives that the receiver believes is **error-free**

Forward Error Correction versus Retransmission

- Retransmission (Backward Error Correction) is a technique in which the receiver detects the occurrence of an error and asks the sender to resend the message.
- In error correction by retransmission, when an error is discovered, the receiver will inform the sender to retransmit the entire data unit. In many cases, receiver sends an acknowledgement (ACK) of correctly received bits and the sender re-sends anything not acknowledged with a reasonable time period.

Forward Error Correction versus Retransmission

- Forward error correction is the process in which the receiver tries to guess the message by using redundant bits.
- An error-correcting code (ECC) or forward error correction (FEC) code is redundant data that is added to the message on the sender side. If the number of error is within the capability of the code being used, the receiver can use the extra information to discover the location of errors and detect them.
- This type of error correction method is best suitable in simplex communication such as broadcasting.

Error-Detection and Correction Techniques

To detect errors, a common technique is to introduce redundancy bits that provide additional information.

Various techniques for error detection include:

- Simple Parity Check
- Two-dimensional Parity Check
- Checksum
- Cyclic Redundancy Check (CRC)

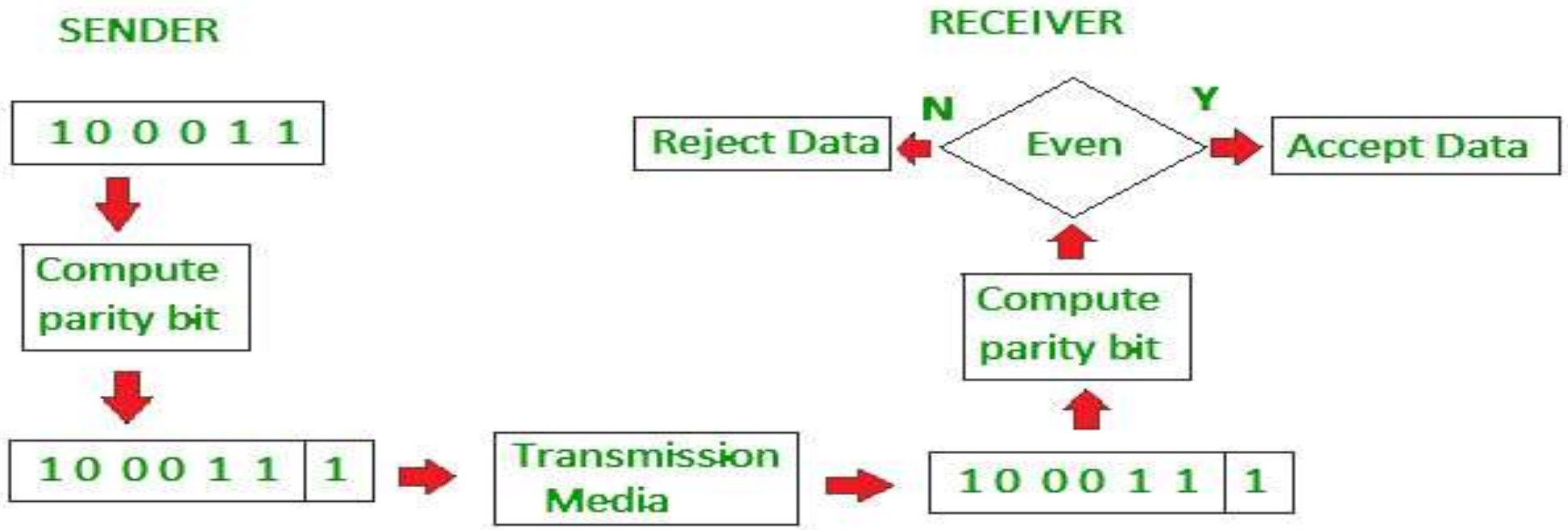
Simple Parity Check

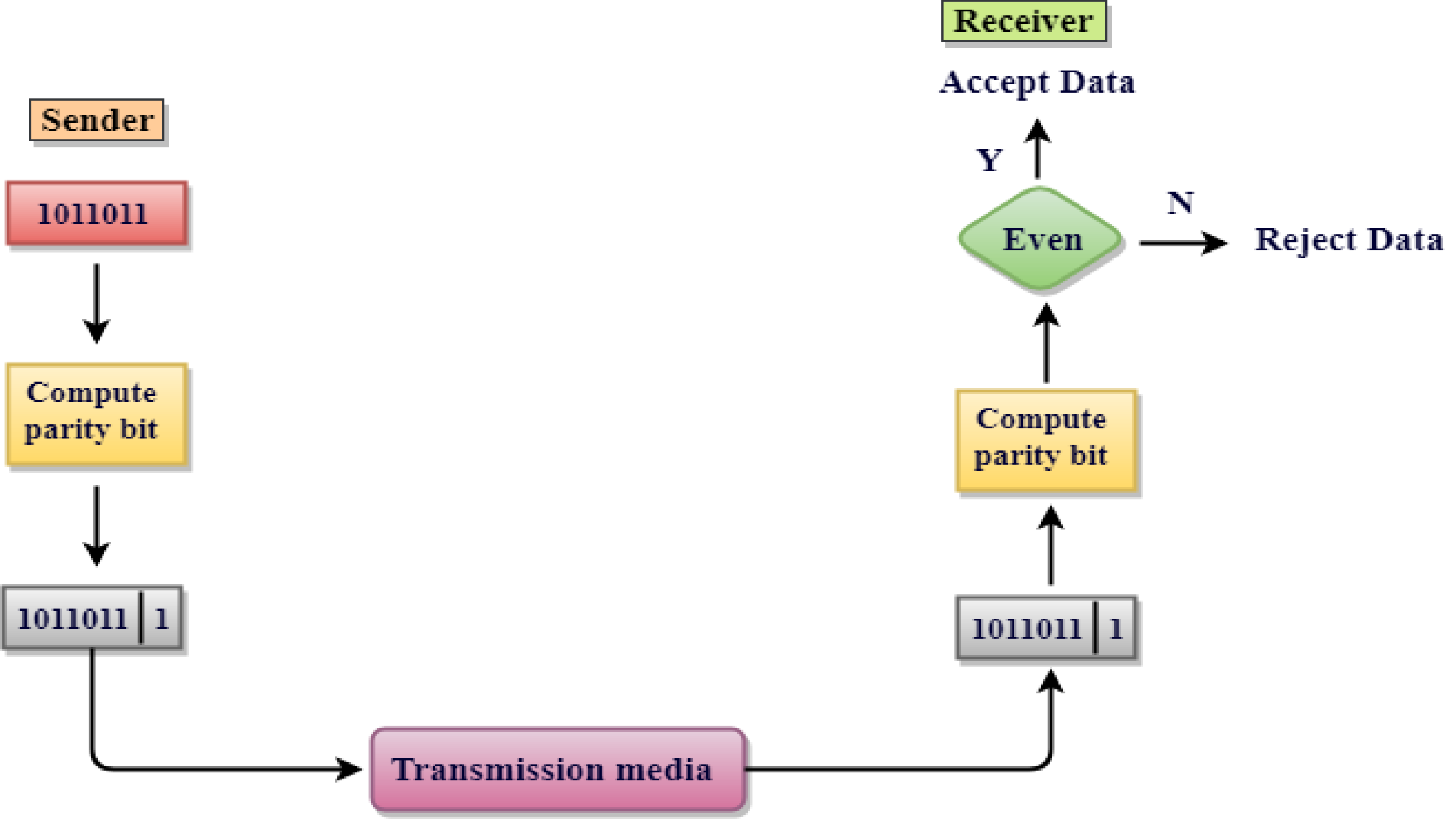
Simple-bit parity is a simple error detection method that involves adding an extra bit to a data transmission.

It works as:

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

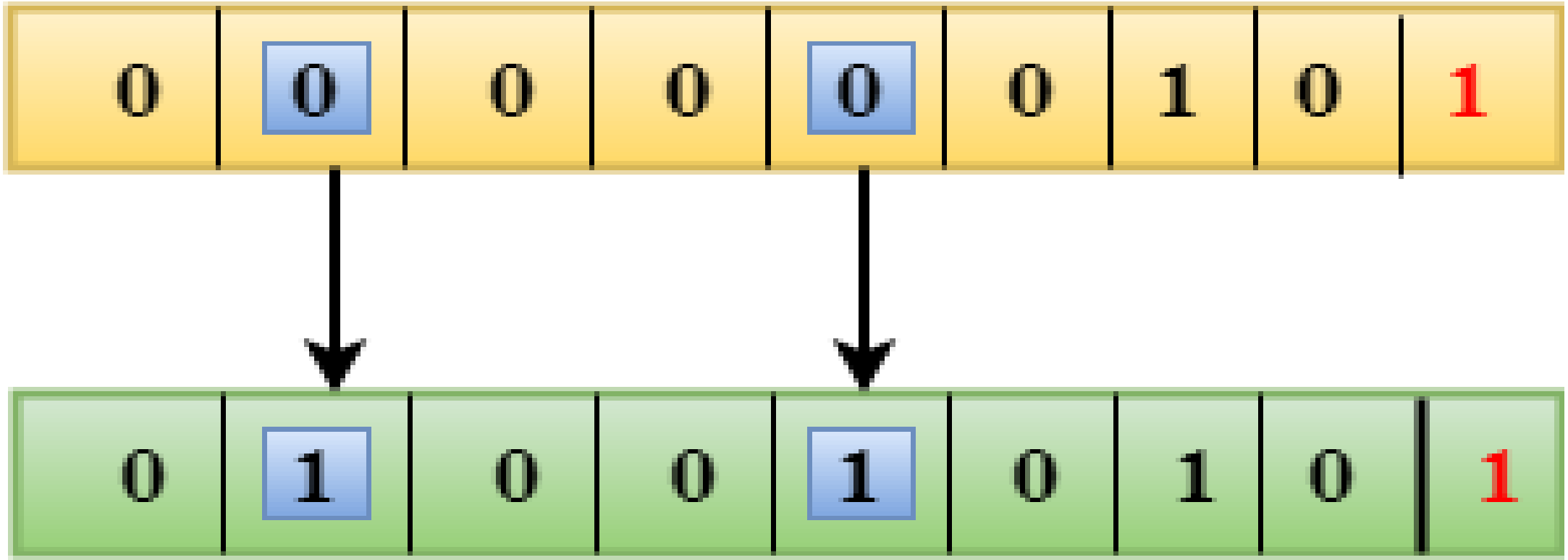
This scheme makes the total number of 1's even, that is why it is called even parity checking.




Sender
1011011
Compute parity bit
1011011 | 1
Transmission media
Receiver
Accept Data
Y
Even
N
Reject Data
Compute parity bit
1011011 | 1

Drawbacks Of Single Parity Checking

- It can only detect single-bit errors which are very rare.
- If two bits are interchanged, then it cannot detect the errors.



Note

A simple parity-check code is a single-bit error-detecting code in which $n = k + 1$ with $d_{min} = 2$.
Even parity (ensures that a codeword has an even number of 1's) and odd parity (ensures that there are an odd number of 1's in the codeword)

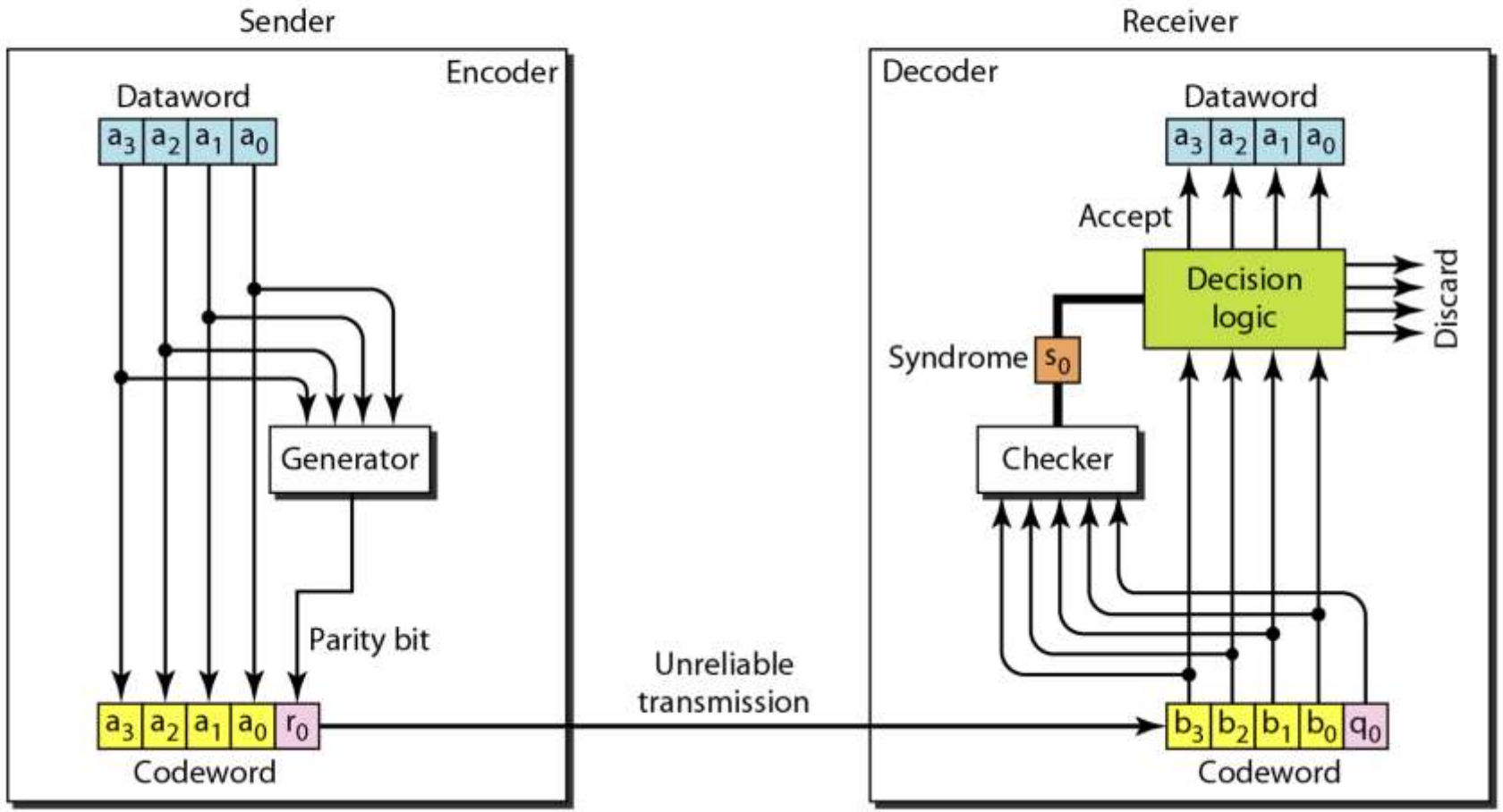
Note

A simple parity-check code can detect an odd number of errors.

Table 10.3 ***Simple parity-check code C(5, 4)***

Datawords	*Codewords*	*Datawords*	*Codewords*
0000	00000	1000	10001
0001	00011	1001	10010
0010	00101	1010	10100
0011	00110	1011	10111
0100	01001	1100	11000
0101	01010	1101	11011
0110	01100	1110	11101
0111	01111	1111	11110

Figure 10.10 *Encoder and decoder for simple parity-check code*



Let us look at some transmission scenarios. Assume the sender sends the dataword 1011. The codeword created from this dataword is 10111, which is sent to the receiver. We examine five cases:

1. *No error occurs; the received codeword is 10111. The syndrome is 0. The dataword 1011 is created.*
2. *One single-bit error changes a_1. The received codeword is 10011. The syndrome is 1. No dataword is created.*
3. *One single-bit error changes r_0. The received codeword is 10110. The syndrome is 1. No dataword is created.*
4. *An error changes r_0 and a second error changes a_3. The received codeword is 00110. The syndrome is 0. The dataword 0011 is created at the receiver. Note that here the dataword is wrongly created due to the syndrome value.*
5. *Three bits—a_3, a_2, and a_1—are changed by errors. The received codeword is 01011. The syndrome is 1. The dataword is not created. This shows that the simple parity check, guaranteed to detect one single error, can also find any odd number of errors.*

Two-dimensional Parity Check

- Two-dimensional 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.

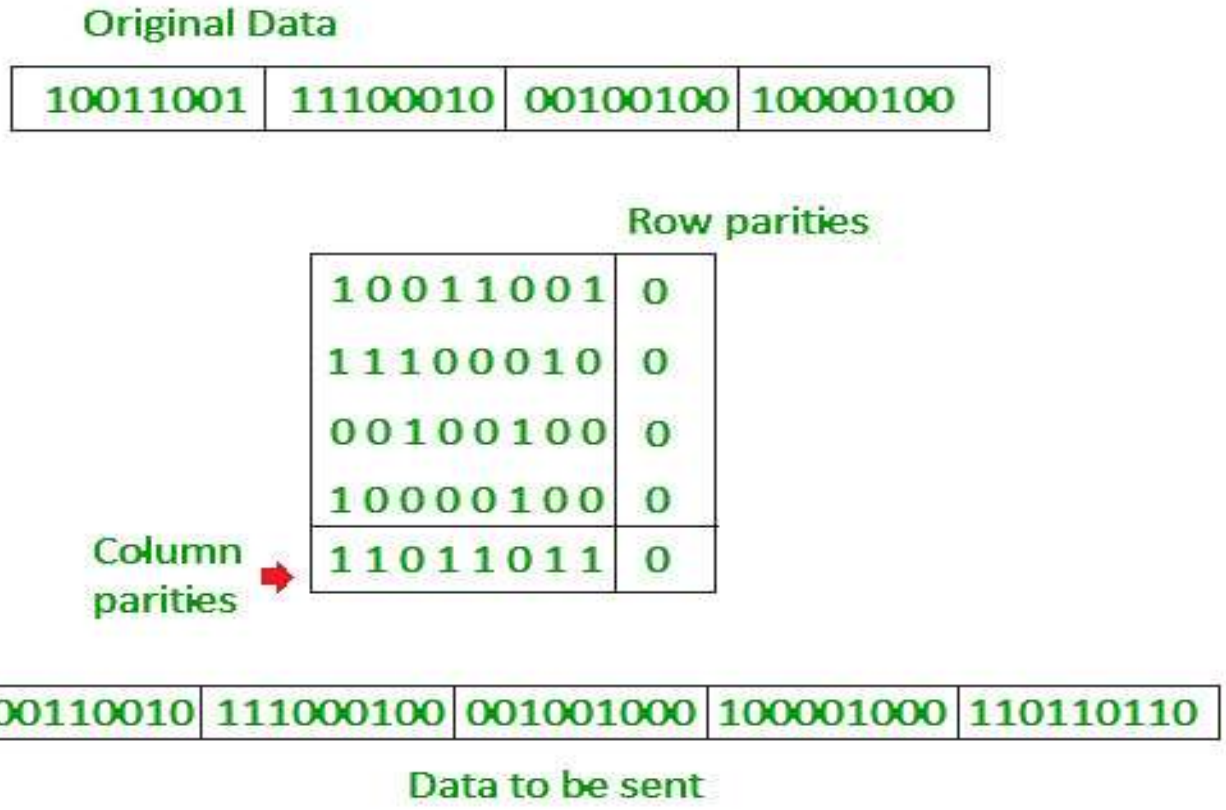
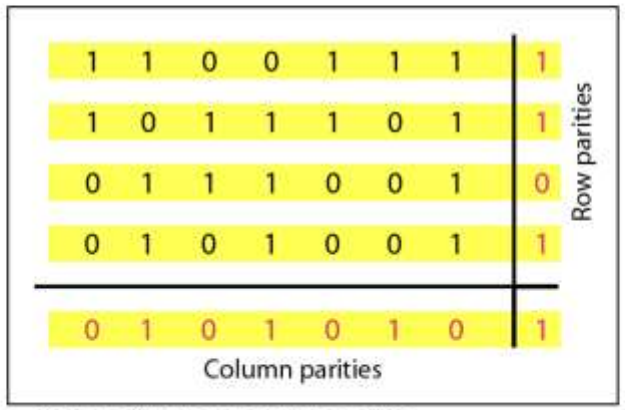
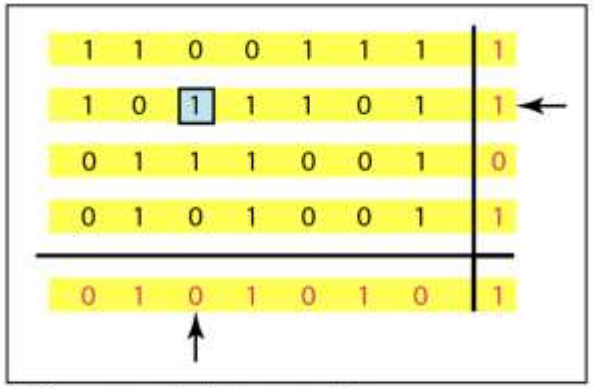


Figure 10.11 *Two-dimensional parity-check code*

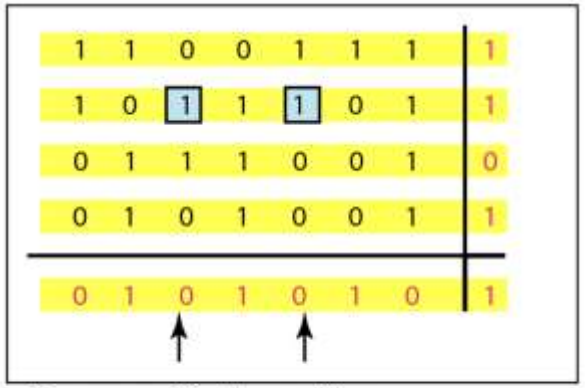
Figure 10.11 *Two-dimensional parity-check code*



a. Design of row and column parities

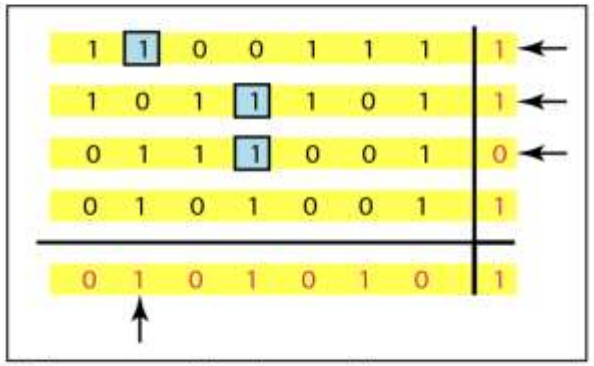


b. One error affects two parities

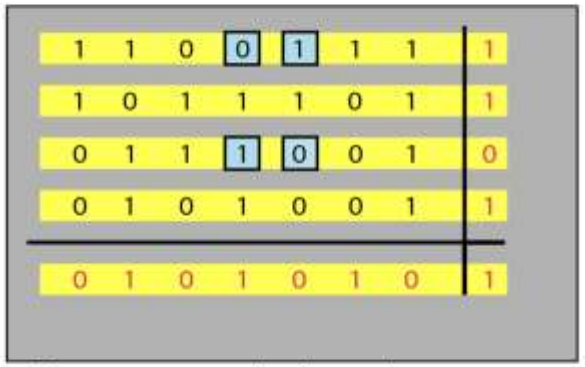


c. Two errors affect two parities

Figure 10.11 *Two-dimensional parity-check code*



d. Three errors affect four parities



e. Four errors cannot be detected

10-5 CHECKSUM

The last error detection method we discuss here is called the checksum. The checksum is used in the Internet by several protocols although not at the data link layer. However, we briefly discuss it here to complete our discussion on error checking

Topics discussed in this section:

Idea
One's Complement
Internet Checksum

Checksum

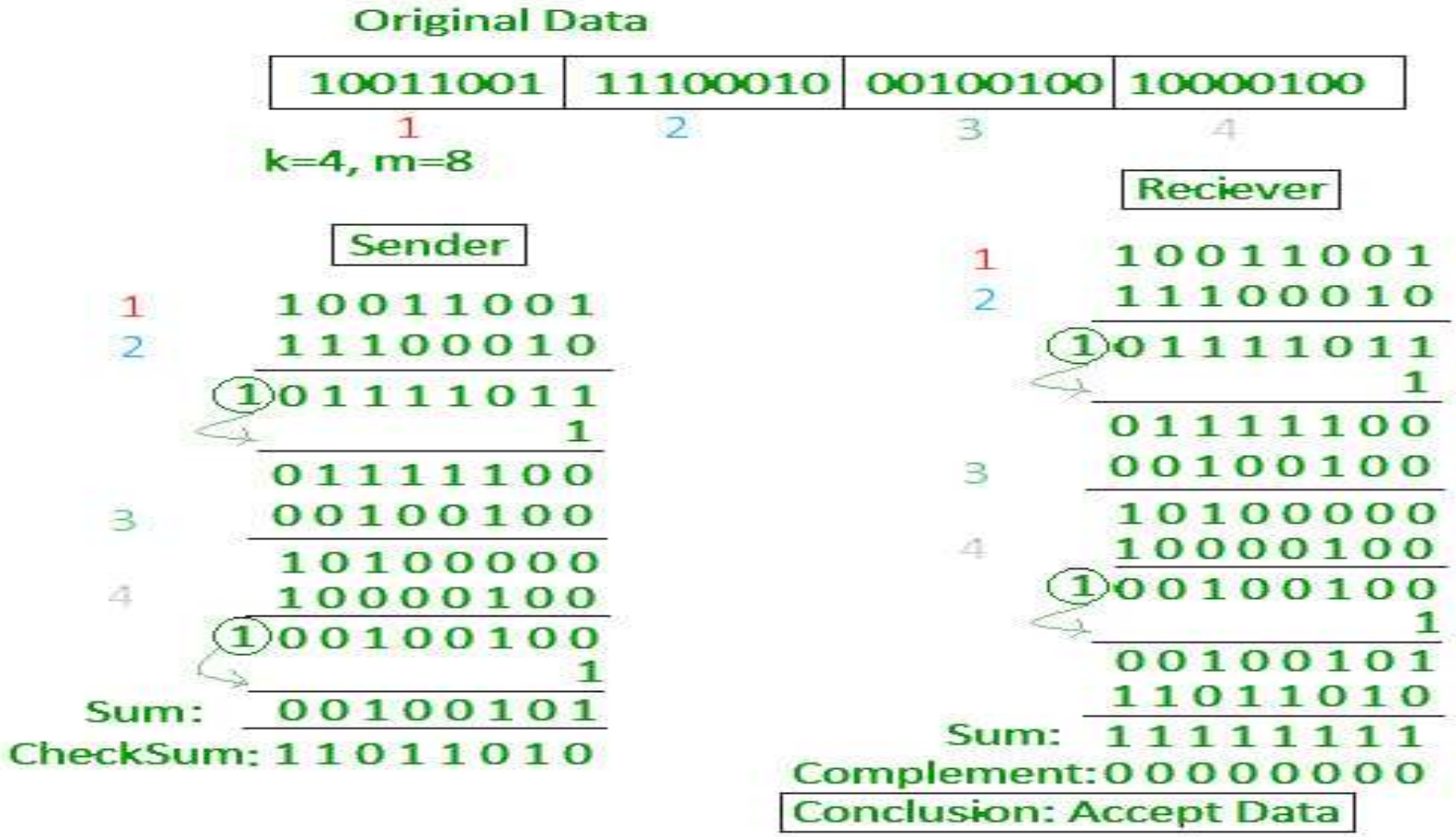
- Checksum error detection is a method used to identify errors in transmitted data.
- The process involves dividing the data into equally sized segments and using a 1’s complement to calculate the sum of these segments.
- The calculated sum is then sent along with the data to the receiver.
- At the receiver’s end, the same process is repeated and if all zeroes are obtained in the sum, it means that the data is correct.

Checksum – Operation at Sender’s Side

- Firstly, the data is divided into k segments each of m bits.
- On 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.

Checksum – Operation at Receiver's Side

- 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.



10-4 CYCLIC CODES

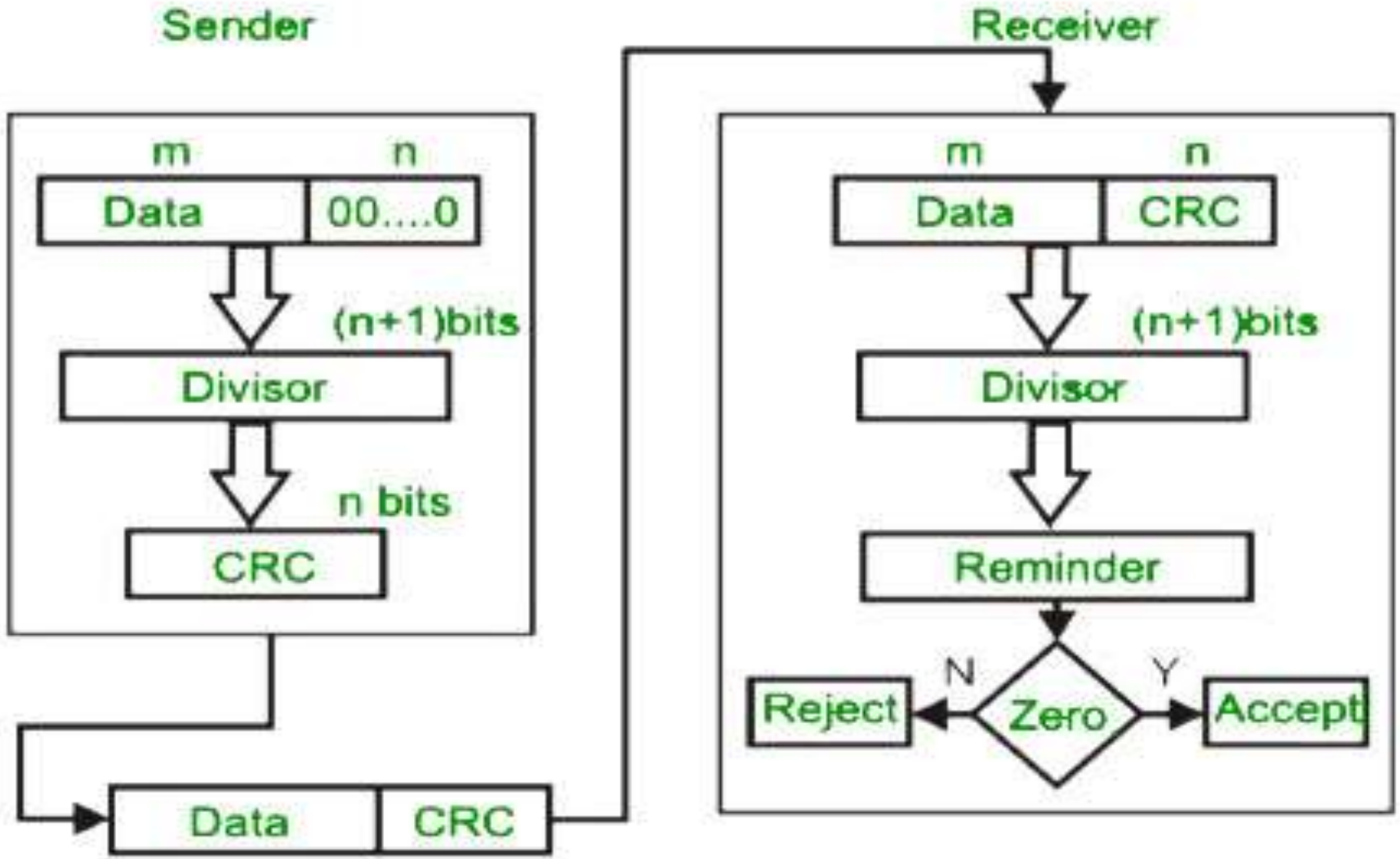
***Cyclic codes** are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.*

Topics discussed in this section:

Cyclic Redundancy Check
Hardware Implementation
Polynomials
Cyclic Code Analysis
Advantages of Cyclic Codes
Other Cyclic Codes

Cyclic Redundancy Check (CRC)

- Unlike the 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 the 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.


Sender
m
n
Data
00....0
(n+1)bits
Divisor
n bits
CRC
Data
CRC
Receiver
m
n
Data
CRC
(n+1)bits
Divisor
Reminder
N
Y
Reject
Zero
Accept

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
```

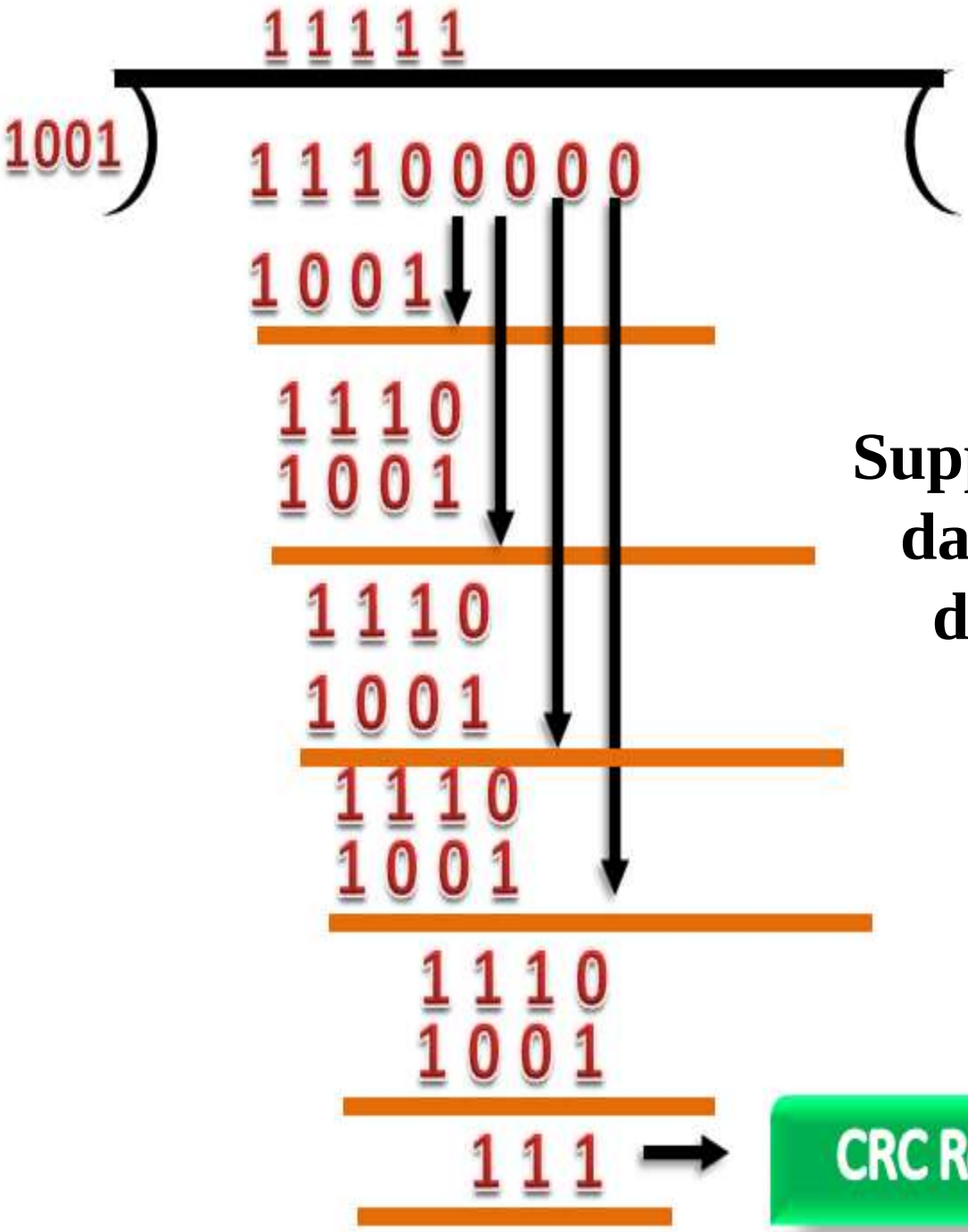
Message to be transmitted

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

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

Receiver

Zero means data is accepted



Suppose the original data is 11100 and divisor is 1001.

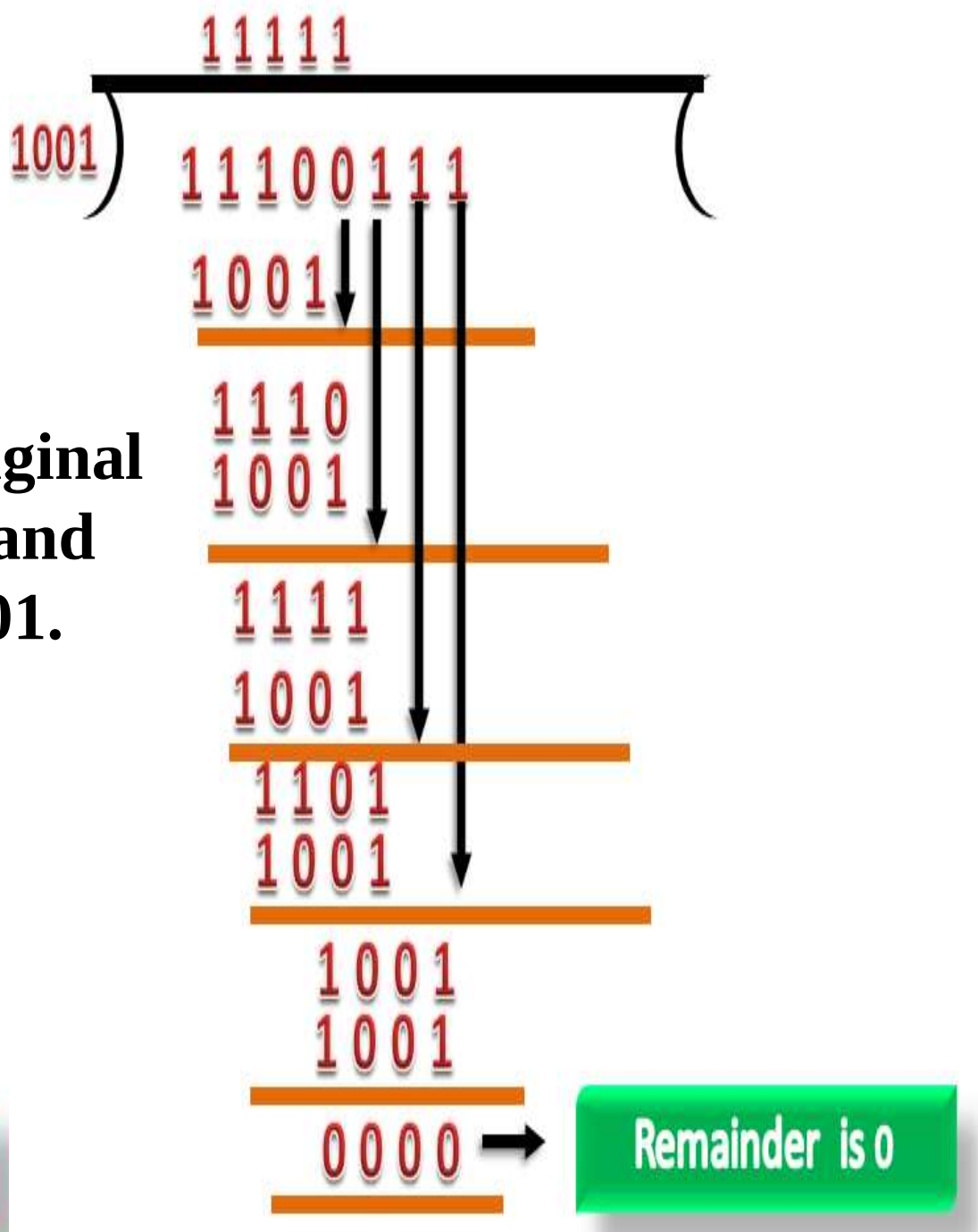


Table 10.6 *A CRC code with C(7, 4)*

Dataword	*Codeword*	*Dataword*	*Codeword*
0000	0000000	1000	1000101
0001	0001011	1001	1001110
0010	0010110	1010	1010011
0011	0011101	1011	1011000
0100	0100111	1100	1100010
0101	0101100	1101	1101001
0110	0110001	1110	1110100
0111	0111010	1111	1111111

Figure 10.14 *CRC encoder and decoder*

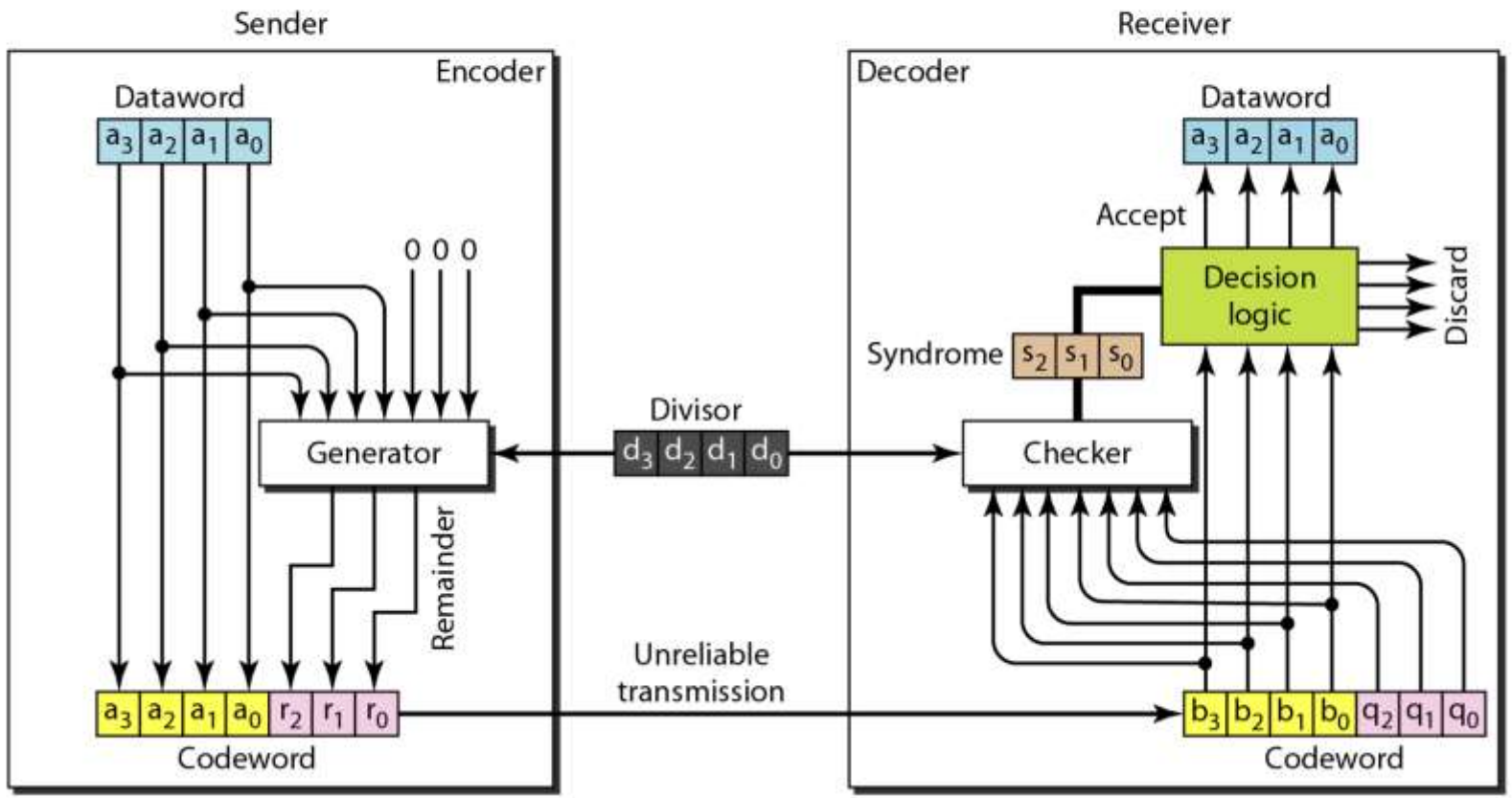


Figure 10.15 *Division in CRC encoder*

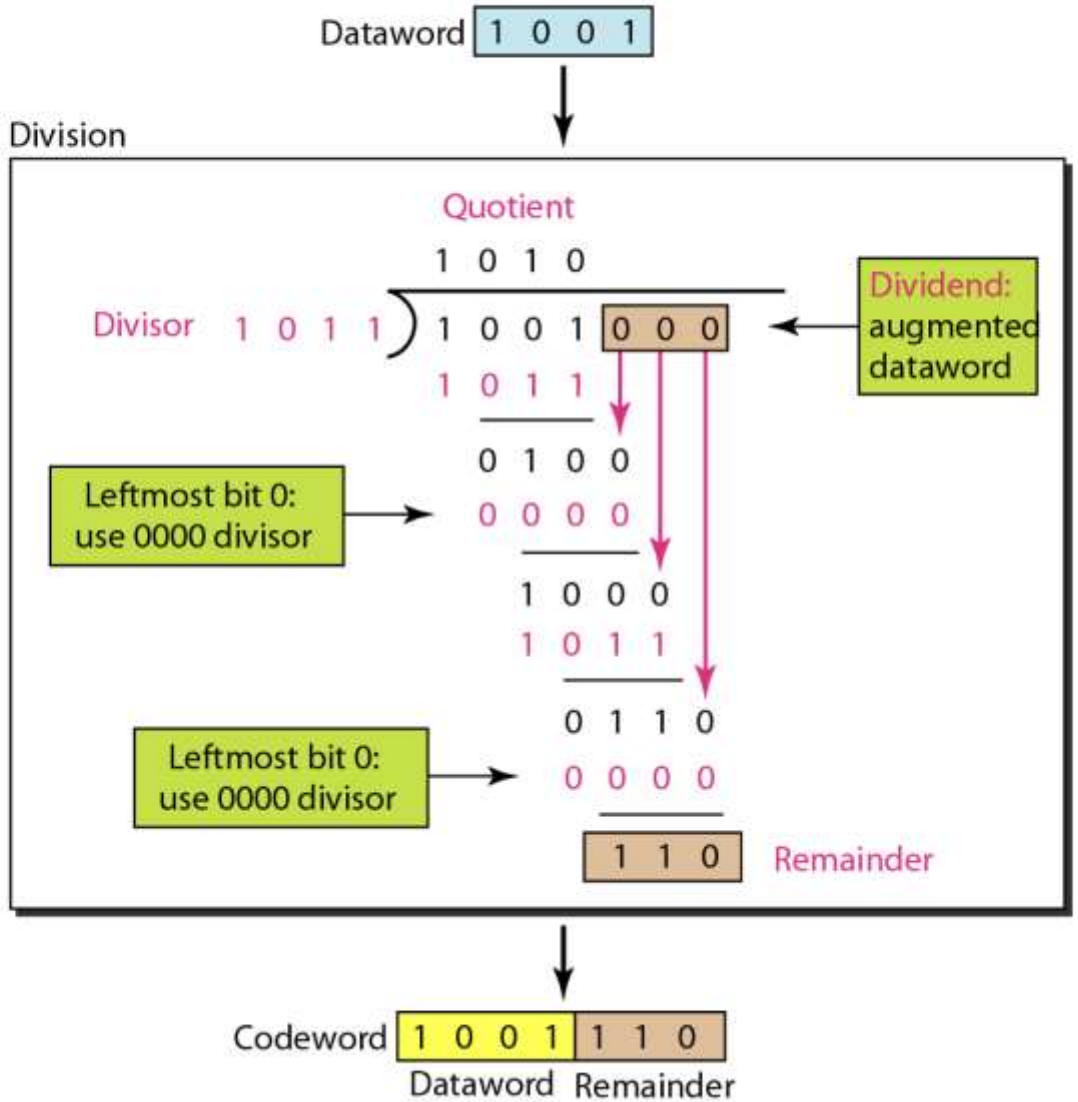


Figure 10.16 ***Division in the CRC decoder for two cases***

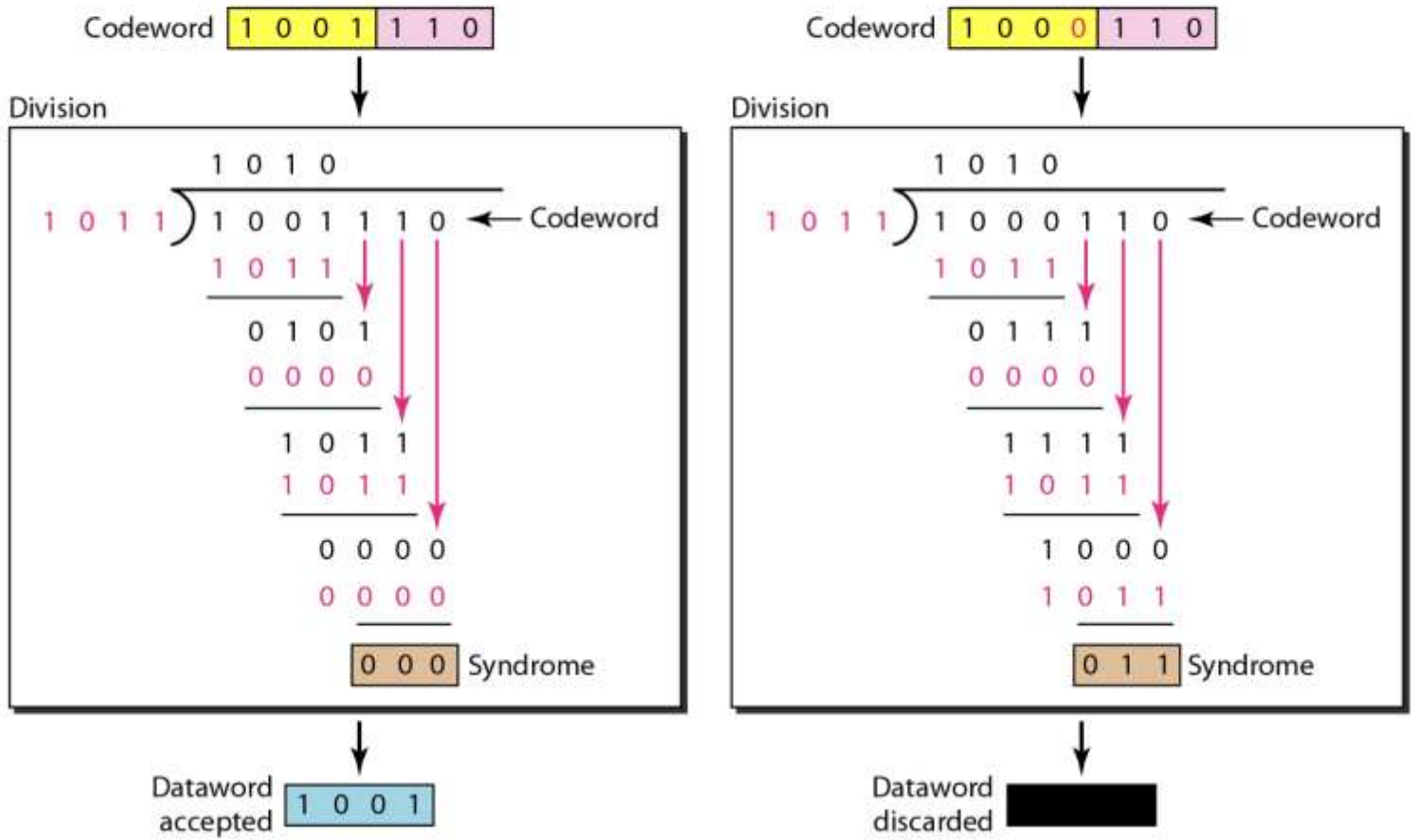
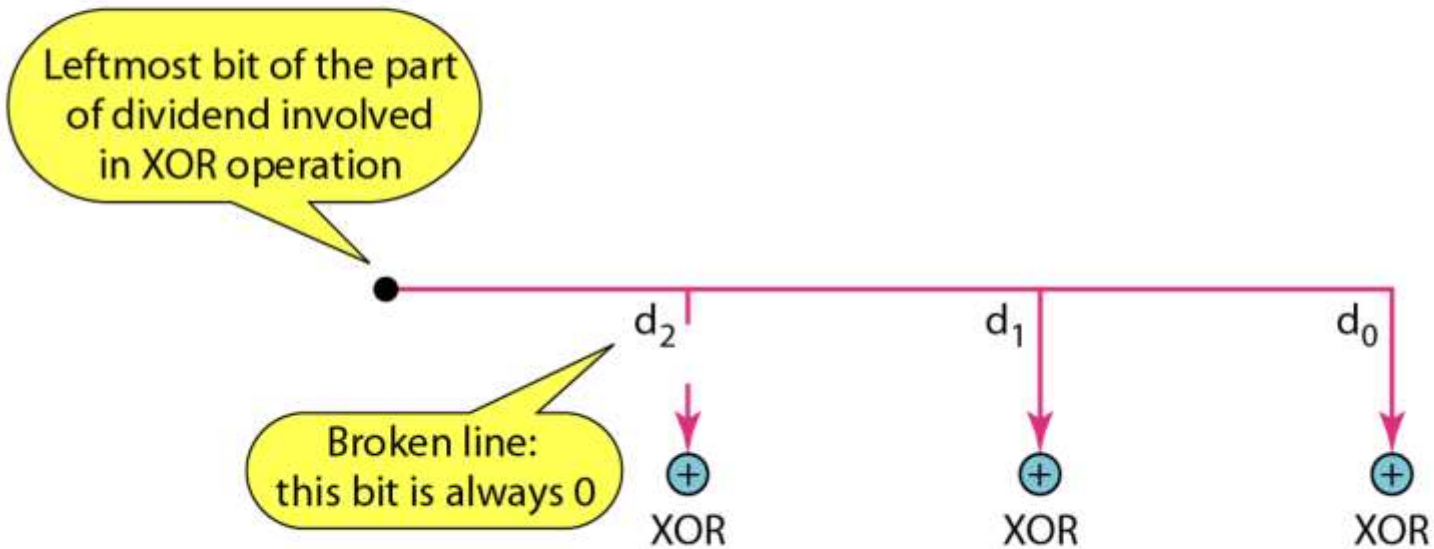


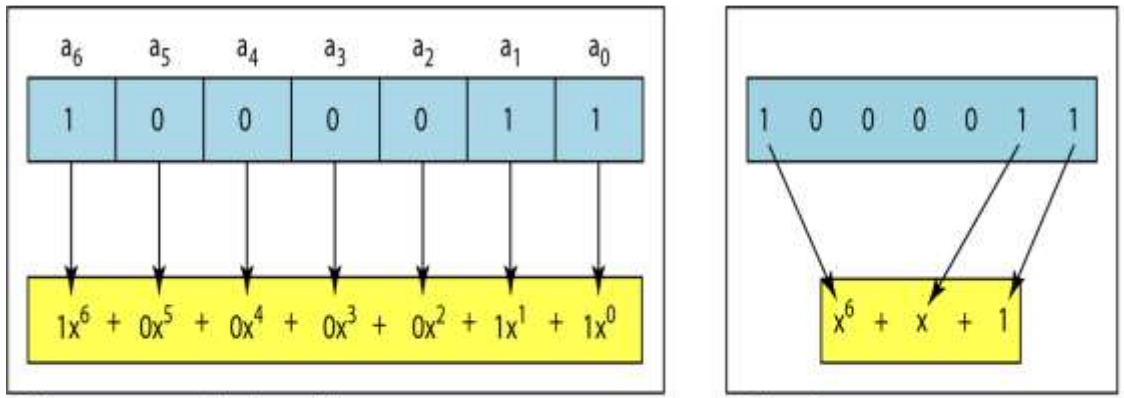
Figure 10.17 *Hardwired design of the divisor in CRC*



Using Polynomials

- We can use a polynomial to represent a binary word.
- Each bit from right to left is mapped onto a power term.
- The rightmost bit represents the “0” power term. The bit next to it the “1” power term, etc.
- If the bit is of value zero, the power term is deleted from the expression.

Figure 10.21 *A polynomial to represent a binary word*



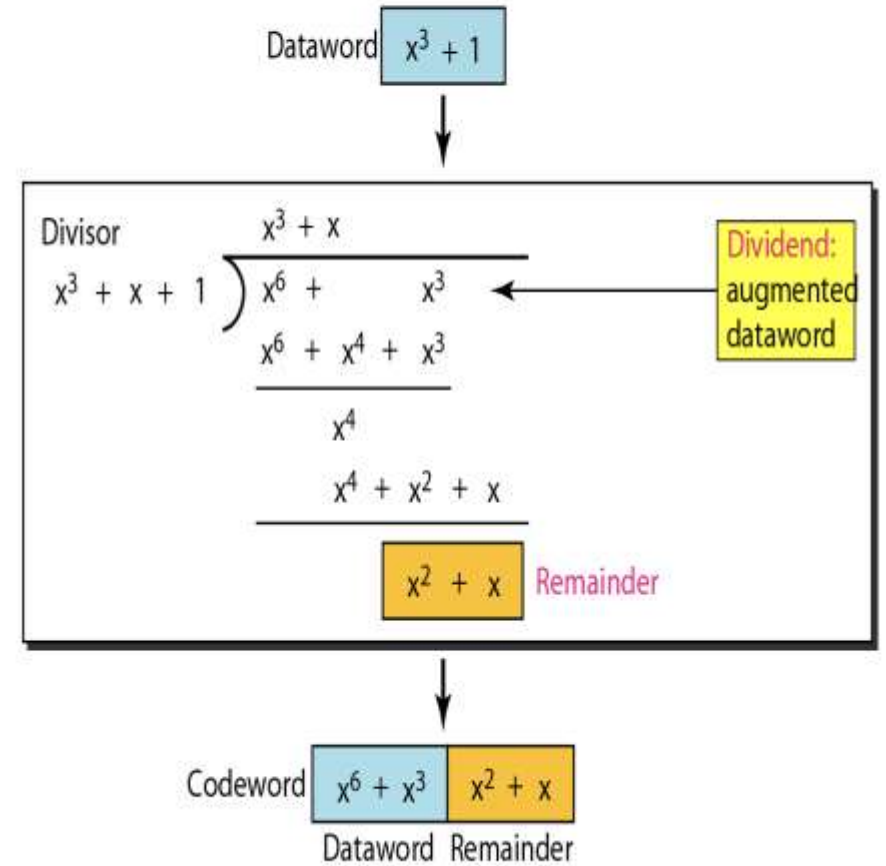
a. Binary pattern and polynomial

b. Short form

Note

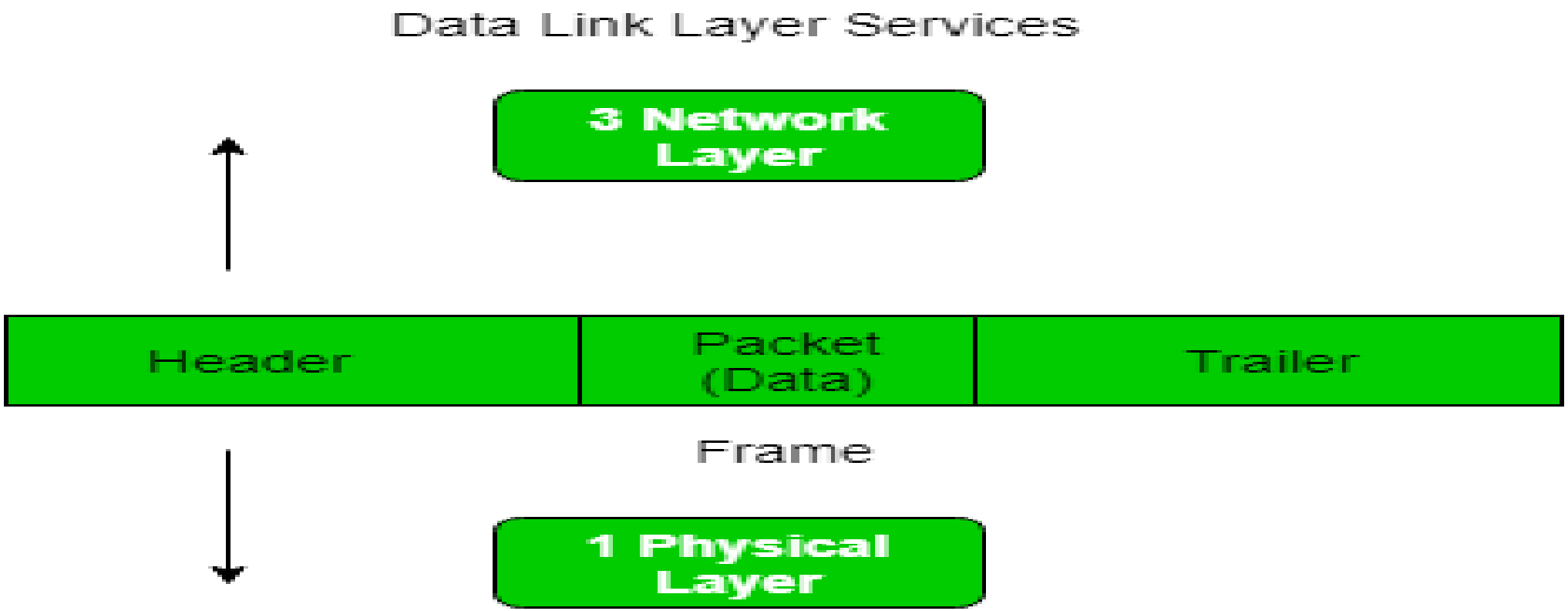
The divisor in a cyclic code is normally called the generator polynomial or simply the generator.

Figure 10.22 *CRC division using polynomials*



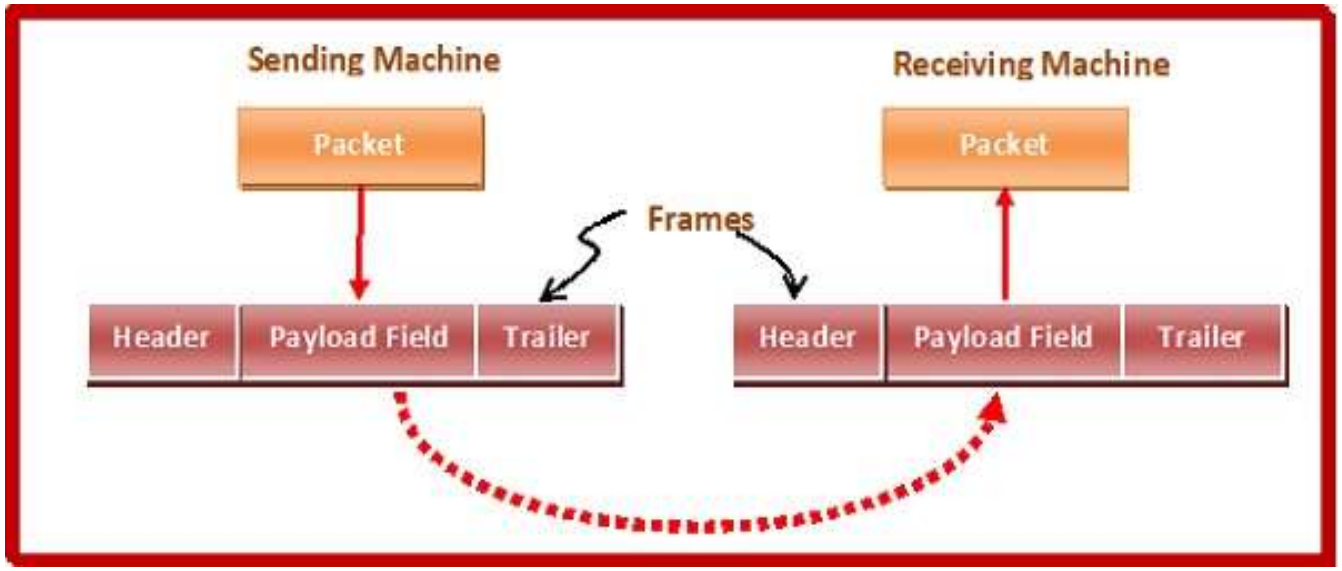
Framing in Data Link Layer

- Frames are the units of digital transmission, particularly in computer networks and telecommunications.
- Frames are comparable to **the packets of energy called photons in the case of light energy.**
- Frame is **continuously used in Time Division Multiplexing process.**
- Framing is a point-to-point connection between two computers or devices consisting of a wire in which data is transmitted as a stream of bits.
- However, these bits must be framed into discernible blocks of information.
- Framing is a function of the data link layer.
- It provides a way for a sender to transmit a set of bits that are meaningful to the receiver.
- Ethernet, token ring, frame relay, and other data link layer technologies have their own frame structures.
- Frames have headers that contain information such as error-checking codes.



- At the data link layer, it extracts the message from the sender and provides it to the receiver by providing the sender's and receiver's addresses.
- The advantage of using frames is that data is broken up into recoverable chunks that can easily be checked for corruption.
- The process of dividing the data into frames and reassembling it is transparent to the user and is handled by the data link layer.

- Framing is an important aspect of data link layer protocol design because it allows the transmission of data to be organized and controlled, ensuring that the data is delivered accurately and efficiently.



Parts of a Frame

A frame has the following parts −

Frame Header − It contains the source and the destination addresses of the frame.

Payload field − It contains the message to be delivered.

Trailer − It contains the error detection and error correction bits.

Flag − It marks the beginning and end of the frame.

Types of Framing

Framing can be of two types, fixed sized framing and variable sized framing.

1. Fixed-size: The frame is of fixed size and there is no need to provide boundaries to the frame, the length of the frame itself acts as a delimiter.

Drawback: It suffers from internal fragmentation if the data size is less than the frame size

Solution: Padding

2. Variable size: In this, there is a need to define the end of the frame as well as the beginning of the next frame to distinguish. This can be done in two ways:

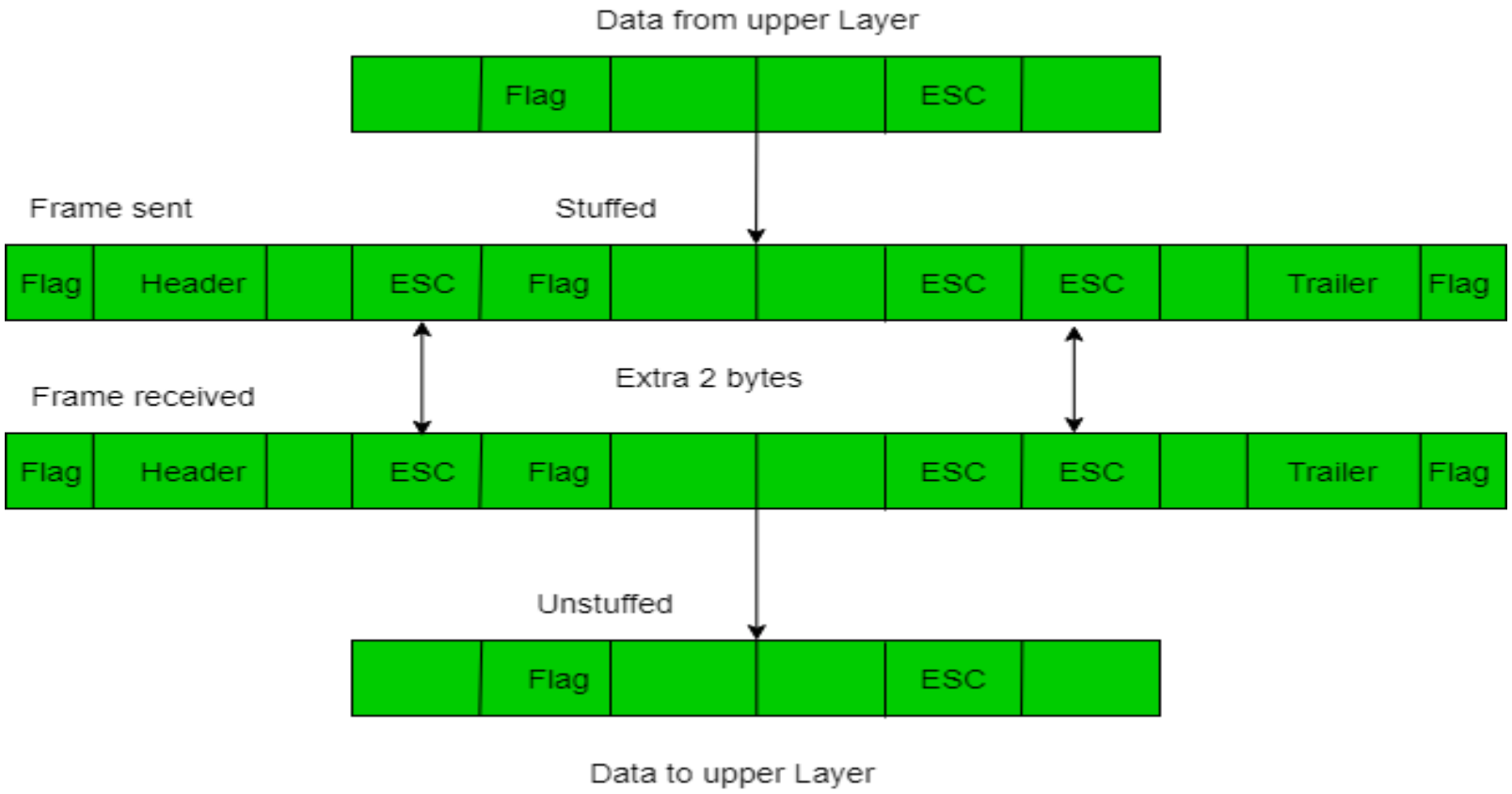
Length field – We can introduce a length field in the frame to indicate the length of the frame. Used in Ethernet(802.3). The problem with this is that sometimes the length field might get corrupted.

End Delimiter (ED) – We can introduce an ED(pattern) to indicate the end of the frame.

Used in Token Ring. The problem with this is that ED can occur in the data.

Character/Byte Stuffing: Used when frames consist of characters.
If data contains ED then, a byte is stuffed into data to differentiate it from ED.

Let ED = “$” –> if data contains ‘$’ anywhere, it can be escaped using ‘\O’ character.
–> if data contains ‘\O$’ then, use ‘\O\O\O$'($ is escaped using \O and \O is escaped using \O).

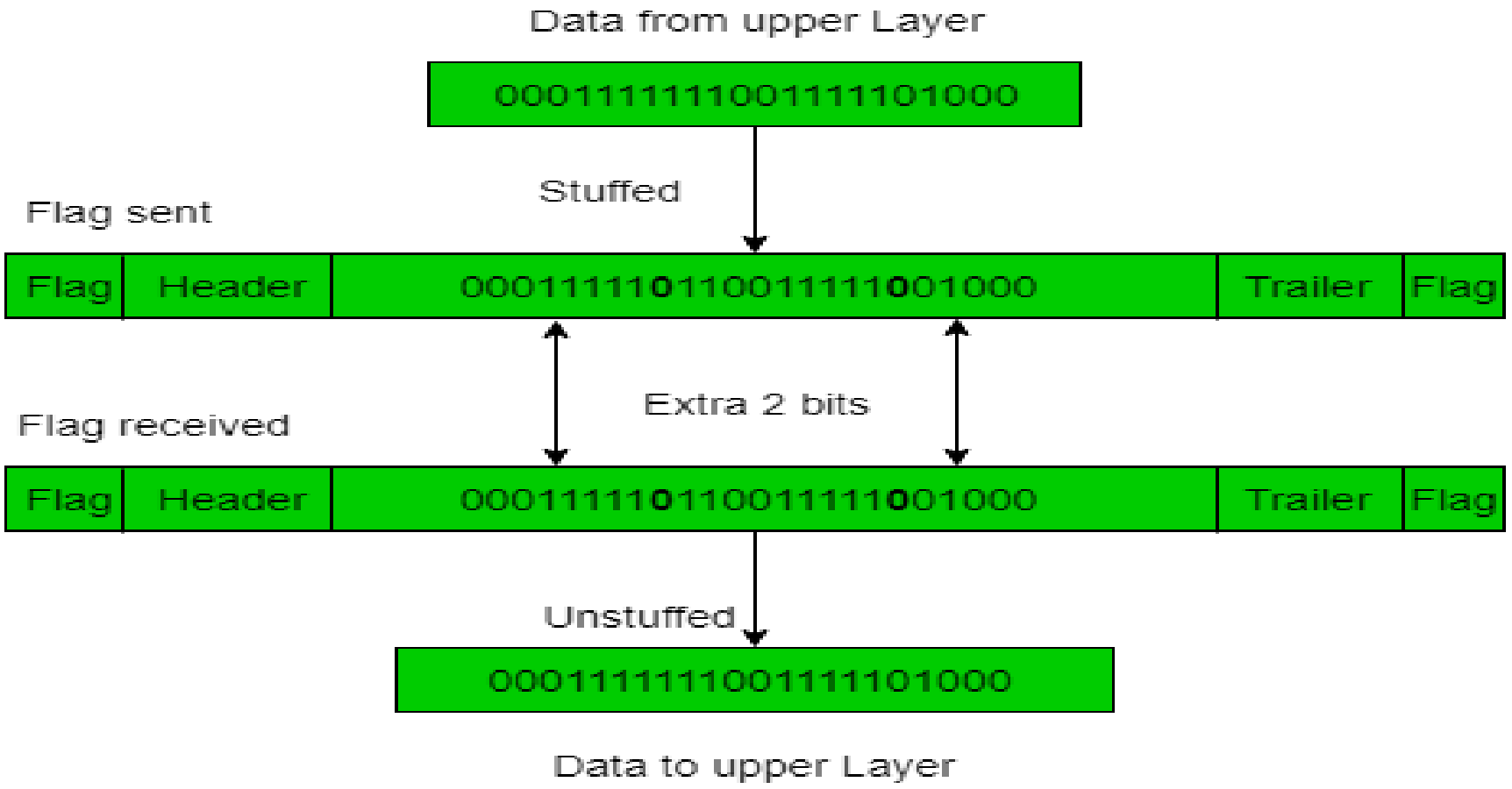


Bit Stuffing: Let ED = 01111 and if data = 01111

–> Sender stuffs a bit to break the pattern i.e. here appends a 0 in data = 011101.

–> Receiver receives the frame.

–> If data contains 011101, receiver removes the 0 and reads the data.



Examples:

If Data –> 011100011110 and ED –> 011 then, find data after bit stuffing.

→ 011**0**10001**1**11**0**0

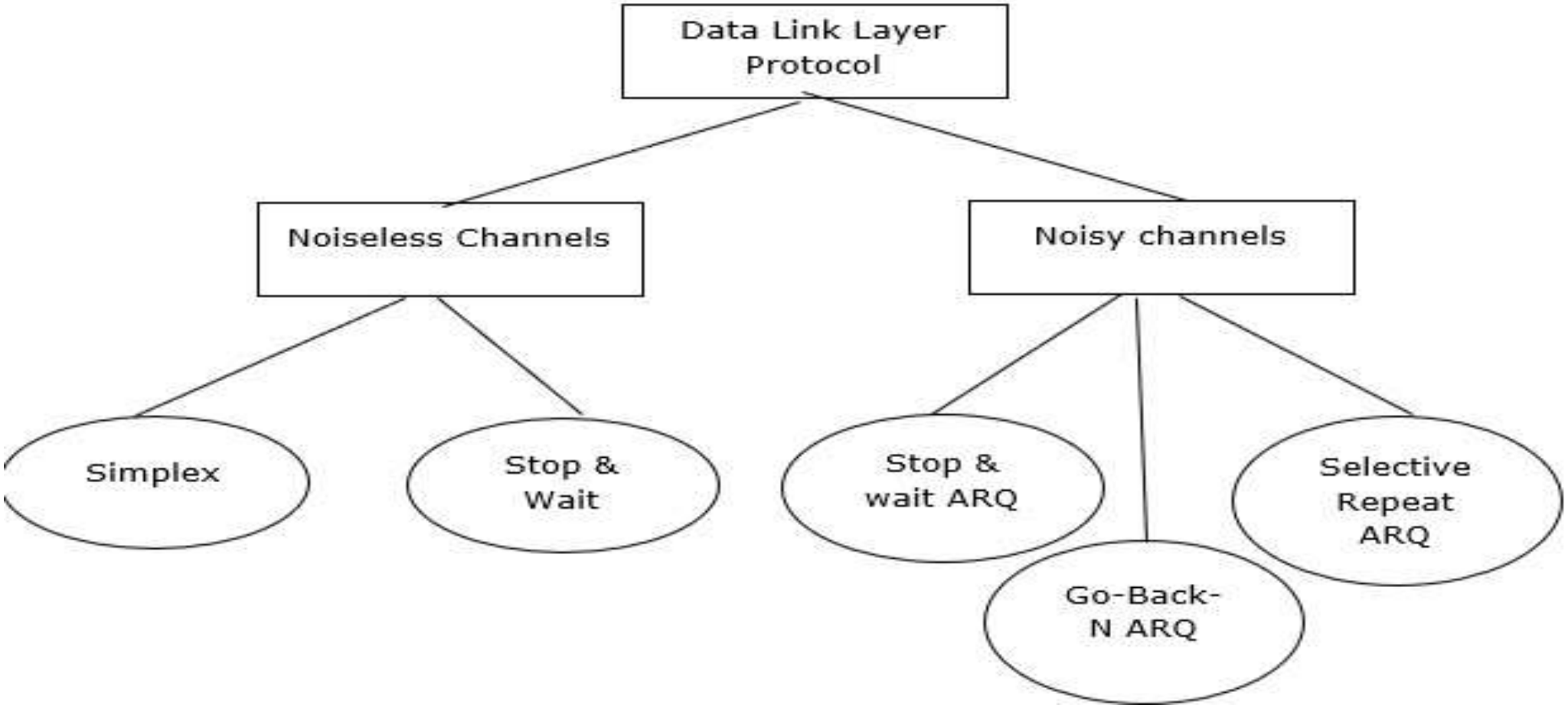
If Data –> 110001001 and ED –> 100 then, find data after bit stuffing?

→ 1100**1**0100**1**1

What are noiseless and noisy channels?

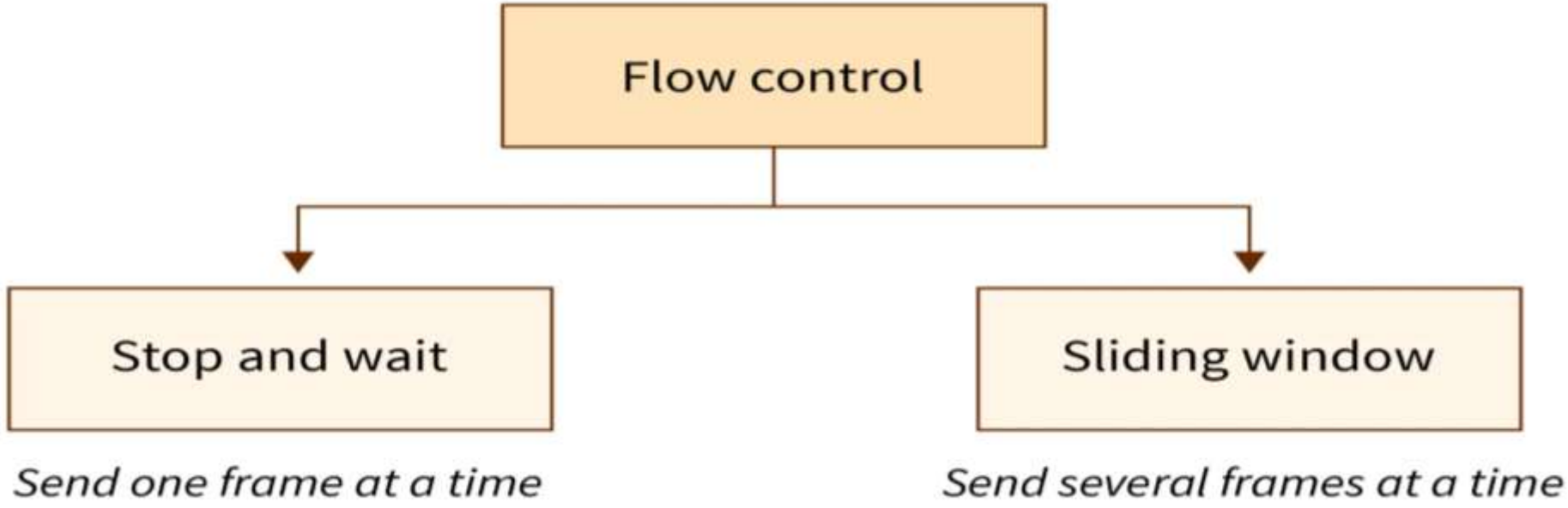
Data link layer protocols are divided into two categories based on whether the transmission channel is noiseless or noisy.

The data link layer protocol is diagrammatically represented below −



Flow Control

- When a data frame (Layer-2 data) is sent from one host to another over a single medium, it is required that the sender and receiver should work at the same speed.
- That is, sender sends at a speed on which the receiver can process and accept the data.
- if the speed (hardware/software) of the sender or receiver differs.
- If sender is sending too fast the receiver may be overloaded, (swamped) and data may be lost.



11-2 FLOW AND ERROR CONTROL

The most important responsibilities of the data link layer are flow control and error control. Collectively, these functions are known as data link control.

Topics discussed in this section:

Flow Control
Error Control

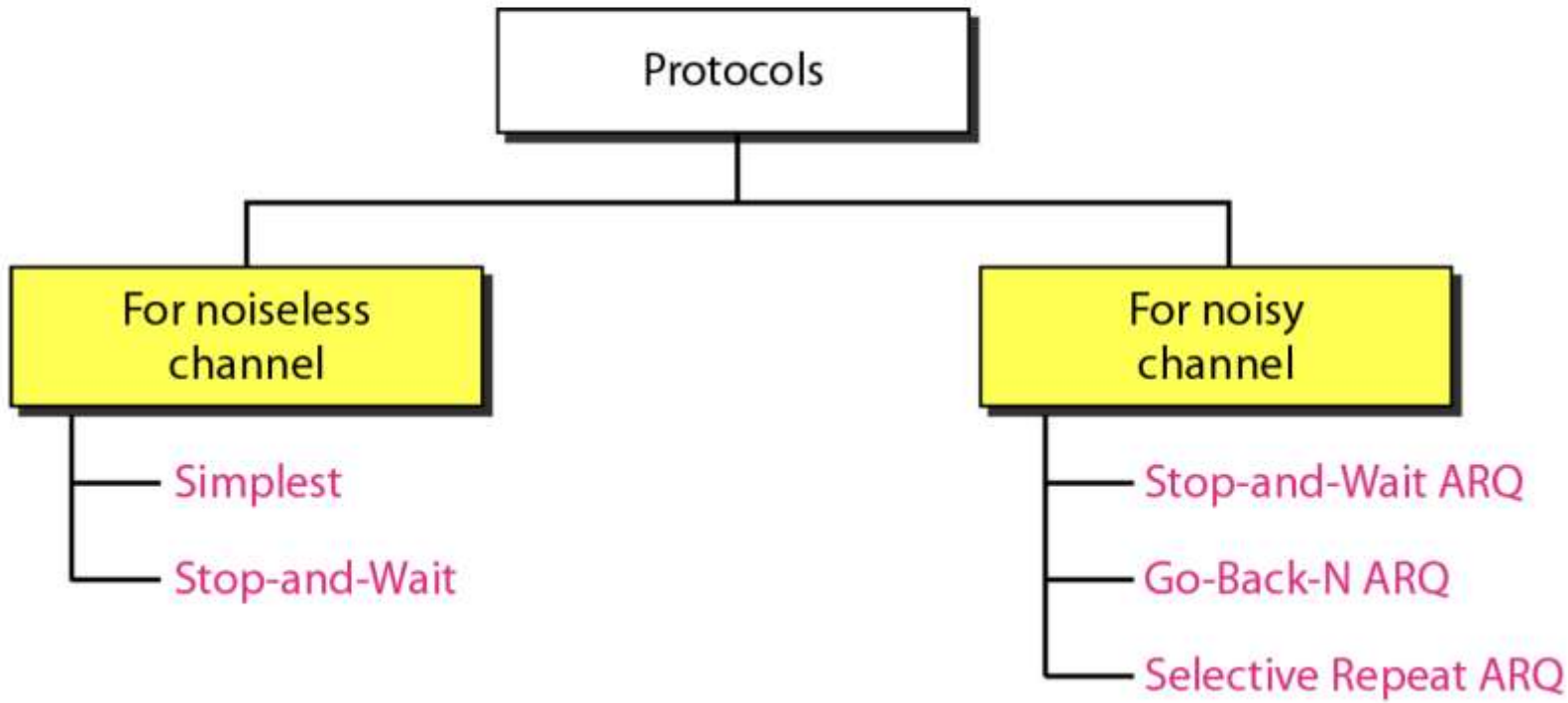
Note

Error control in the data link layer is based on automatic repeat request, which is the retransmission of data.

Note

Flow control refers to a set of procedures used to restrict the amount of data that the sender can send before waiting for acknowledgment.

Figure 11.5 *Taxonomy of protocols discussed in this chapter*



11-4 NOISELESS CHANNELS

Let us first assume we have an ideal channel in which no frames are lost, duplicated, or corrupted. We introduce two protocols for this type of channel.

Topics discussed in this section:

Simplest Protocol

Stop-and-Wait Protocol

Figure 11.6 *The design of the simplest protocol with no flow or error control*

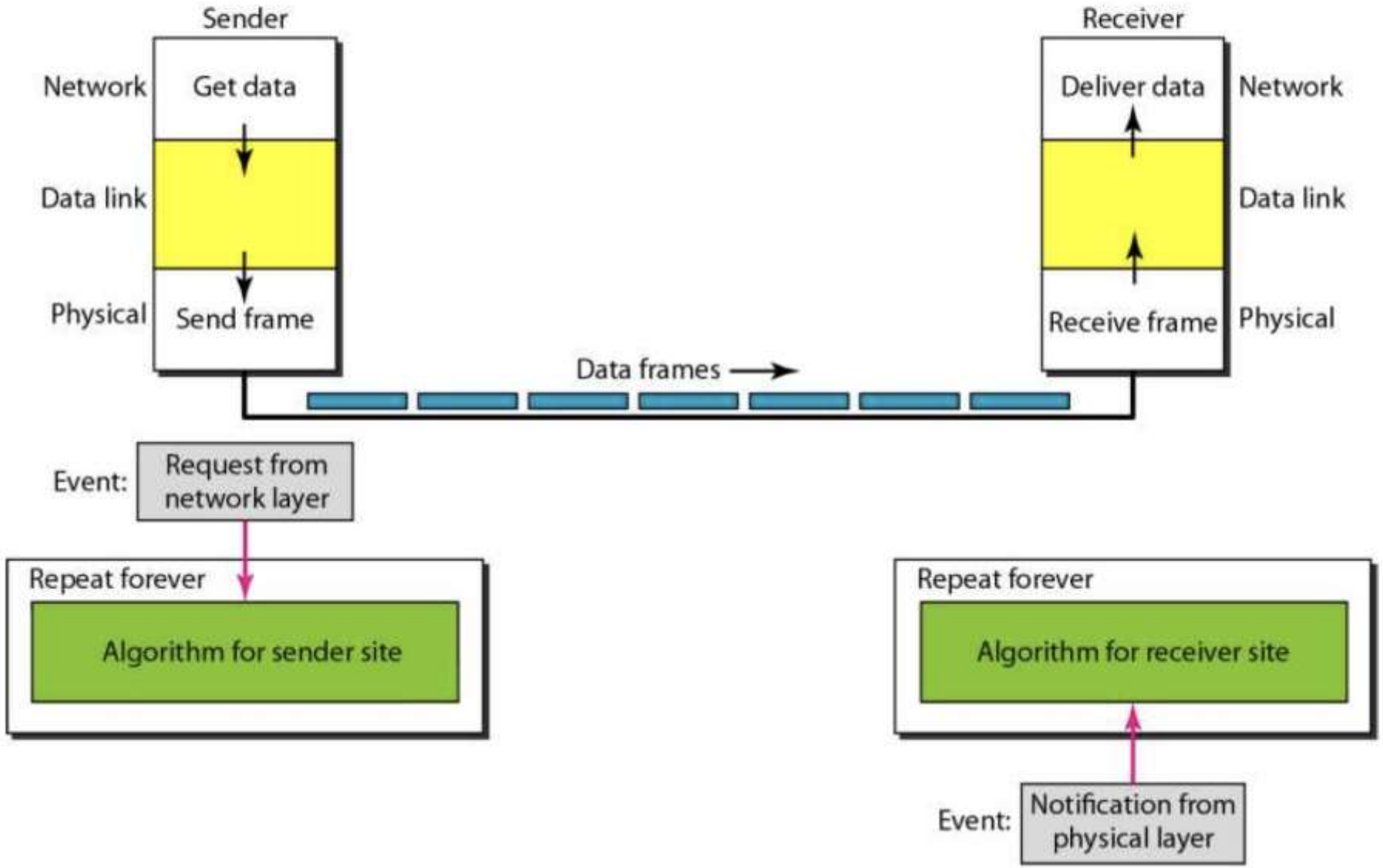
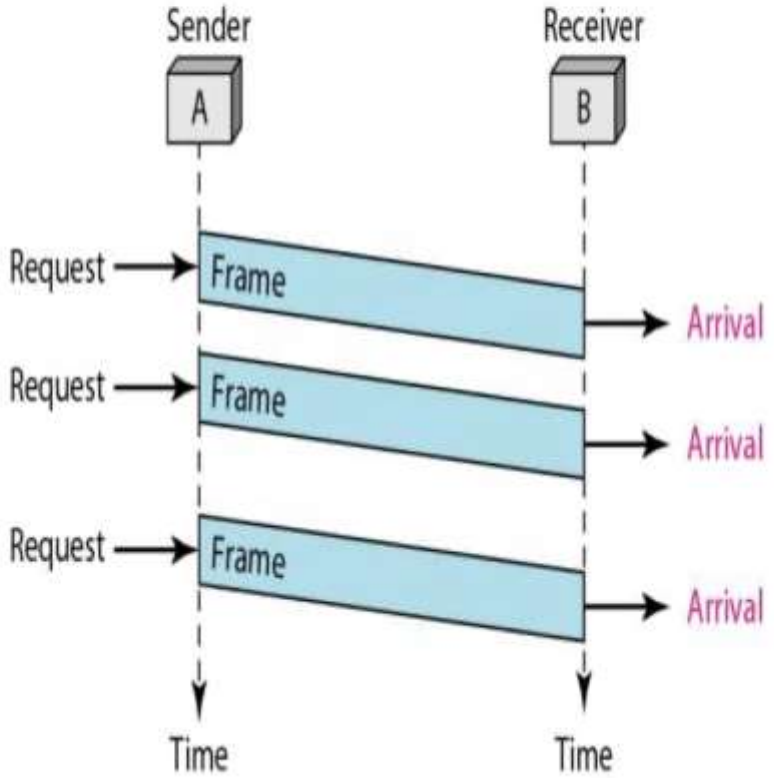


Figure 11.7 *Flow diagram for Example 11.1*



Example 11.1

Figure 11.7 shows an example of communication using this protocol. It is very simple. The sender sends a sequence of frames without even thinking about the receiver. To send three frames, three events occur at the sender site and three events at the receiver site. Note that the data frames are shown by tilted boxes; the height of the box defines the transmission time difference between the first bit and the last bit in the frame.

Algorithm 11.1 *Sender-site algorithm for the 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 }
```

Algorithm 11.2 *Receiver-site algorithm for the 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 }
```

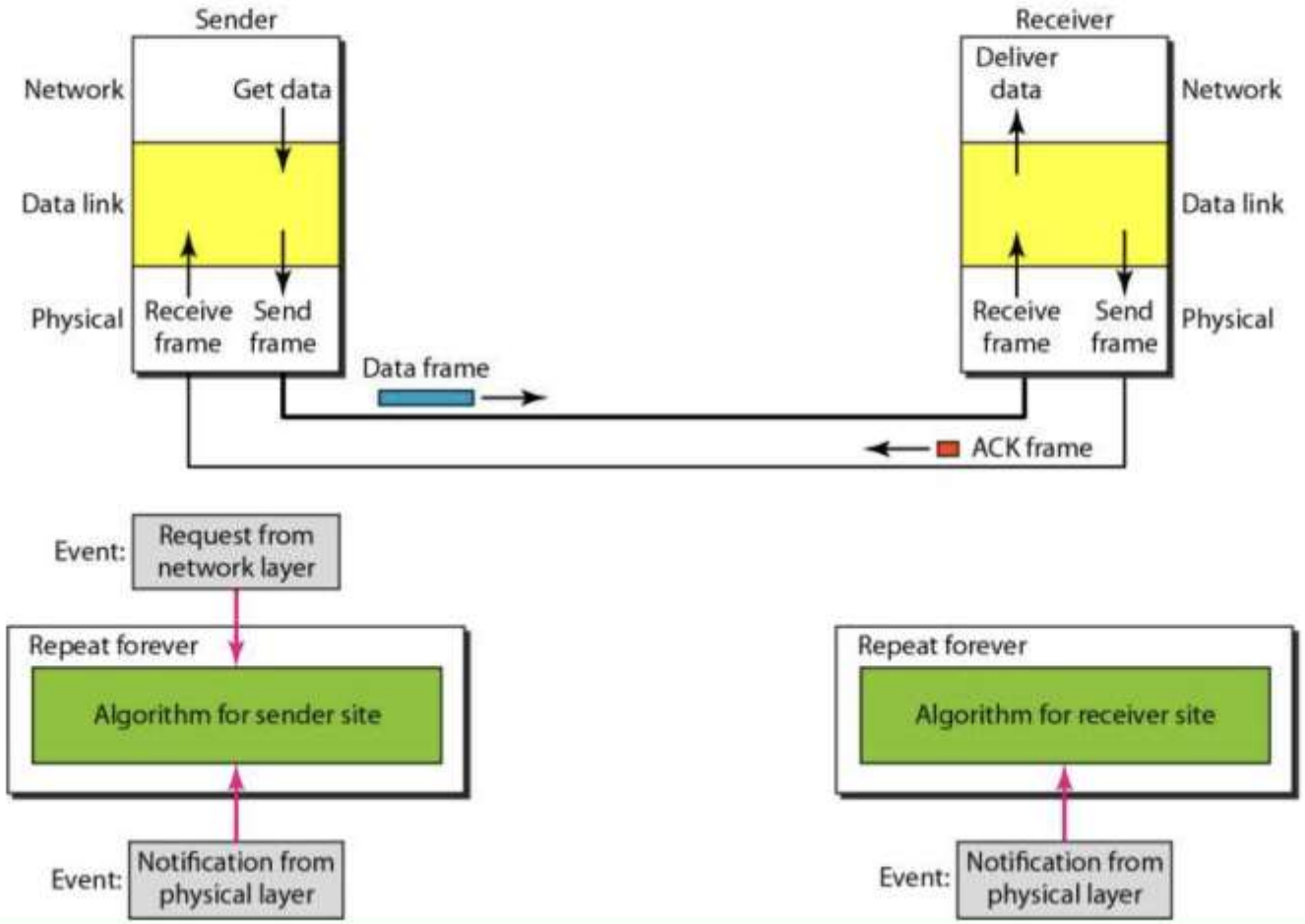
Stop and Wait

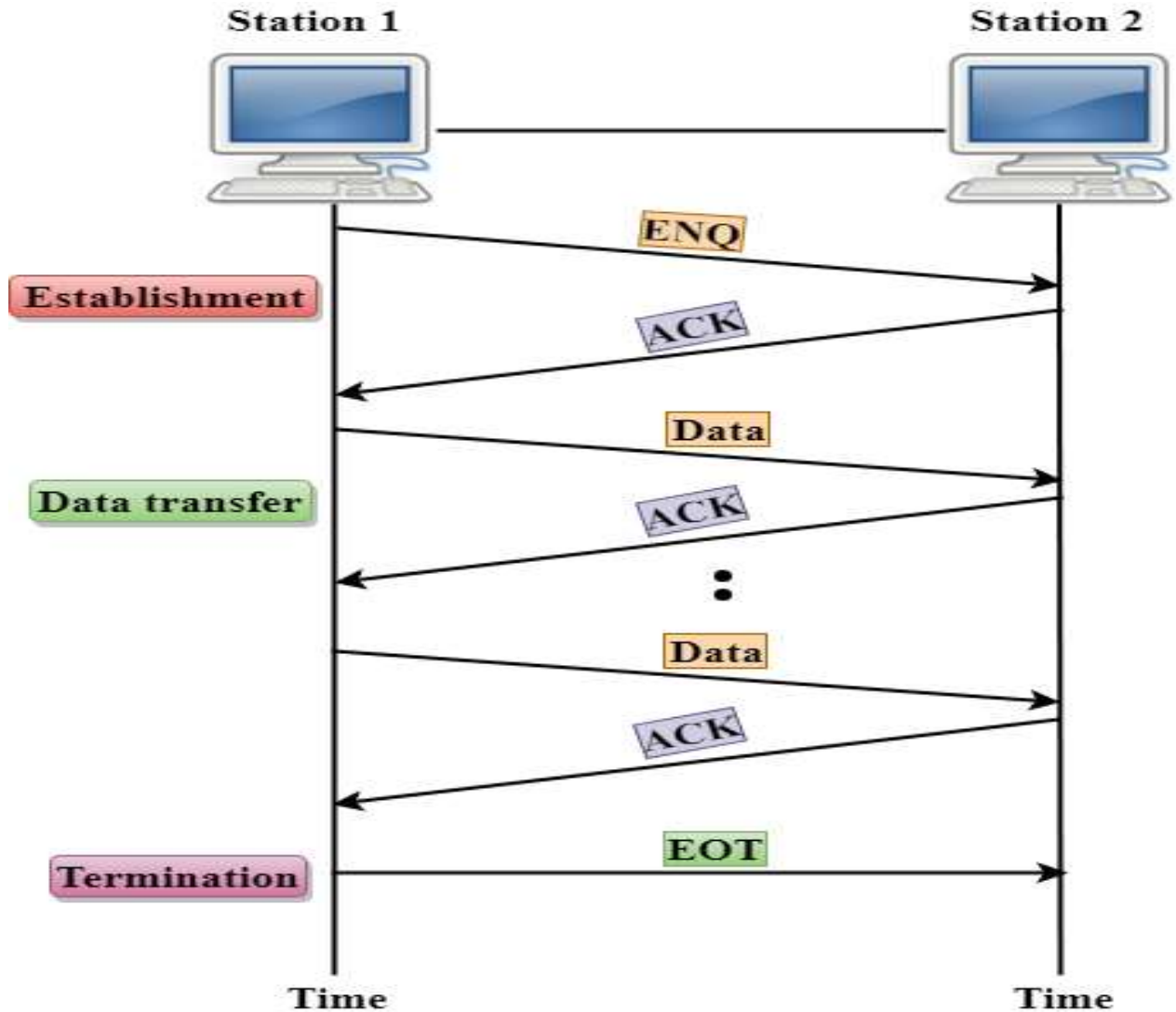
Stop-and-wait protocol works under the assumption that the communication channel is **noiseless** and transmissions are **error-free**.

Working:

- ✓ The sender sends data to the receiver.
- ✓ The sender stops and waits for the acknowledgment.
- ✓ The receiver receives the data and processes it.
- ✓ The receiver sends an acknowledgment for the above data to the sender.
- ✓ The sender sends data to the receiver after receiving the acknowledgment of previously sent data.
- ✓ The process is unidirectional and continues until the sender sends the **End of Transmission (EoT)** frame.

Figure 11.8 *Design of Stop-and-Wait Protocol*

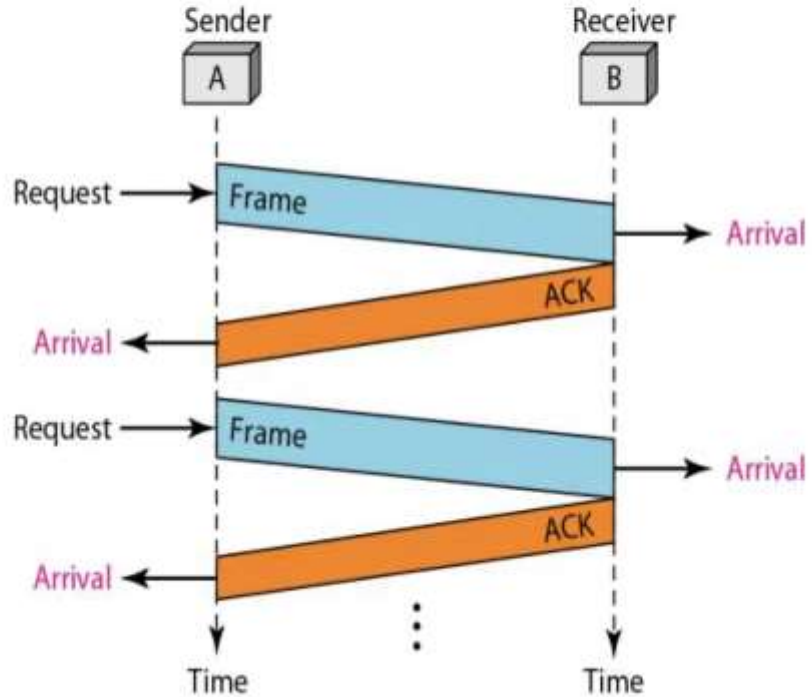



Station 1
Station 2
ENQ
Establishment
ACK
Data
Data transfer
ACK
Data
ACK
EOT
Termination
Time
Time

Example 11.2

Figure 11.9 shows an example of communication using this protocol. It is still very simple. The sender sends one frame and waits for feedback from the receiver. When the ACK arrives, the sender sends the next frame. Note that sending two frames in the protocol involves the sender in four events and the receiver in two events.

Figure 11.9 *Flow diagram for Example 11.2*



Algorithm 11.3 *Sender-site algorithm for Stop-and-Wait Protocol*

```
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 }
```

Algorithm 11.4 *Receiver-site algorithm for Stop-and-Wait Protocol*

```
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 }
```

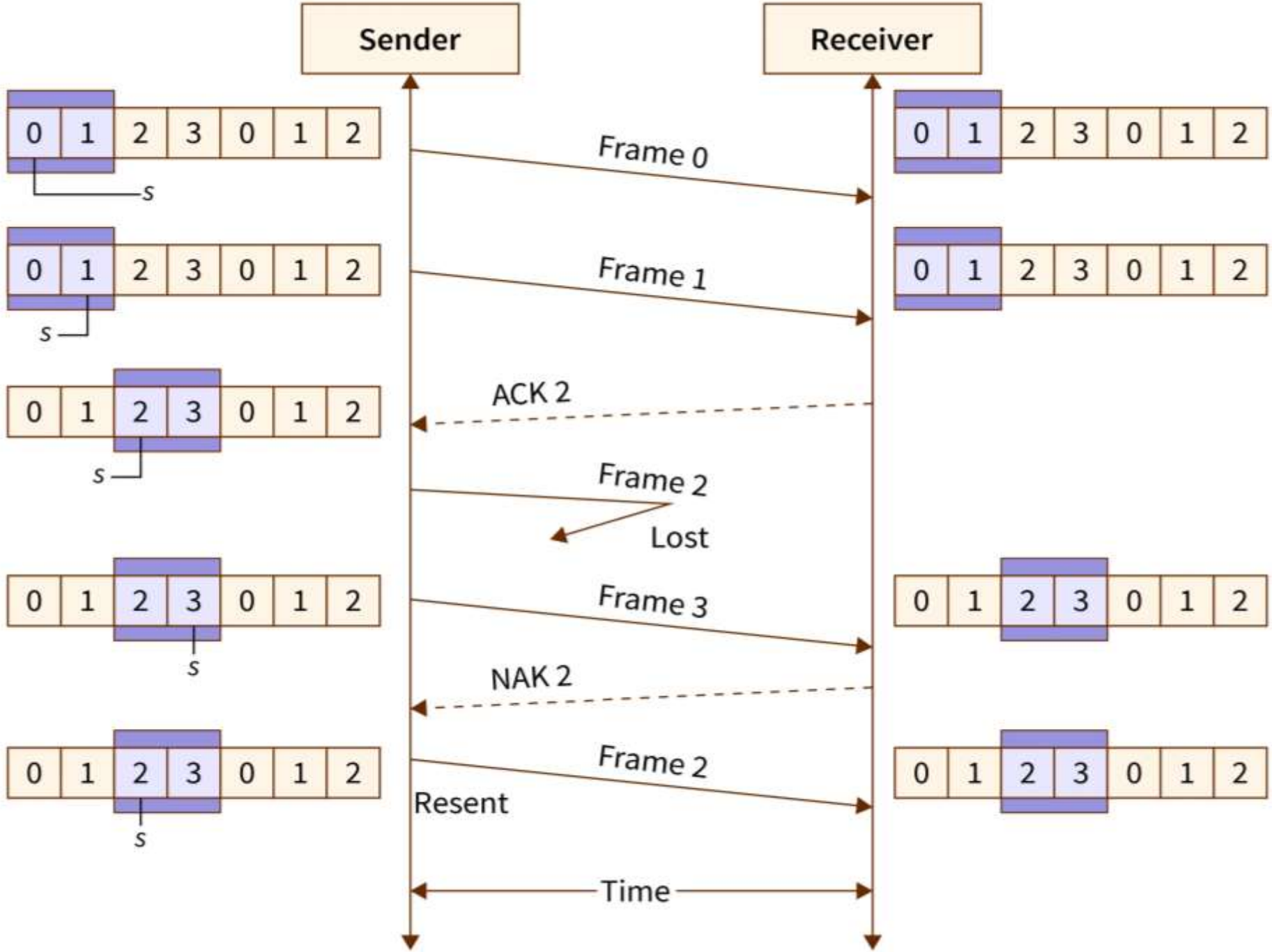
Sliding Window Protocol

The sliding window protocol is the flow control protocol for noisy channels that allows the sender to send multiple frames even before acknowledgments are received.

It is called a Sliding window because the sender slides its window upon receiving the acknowledgments for the sent frames.

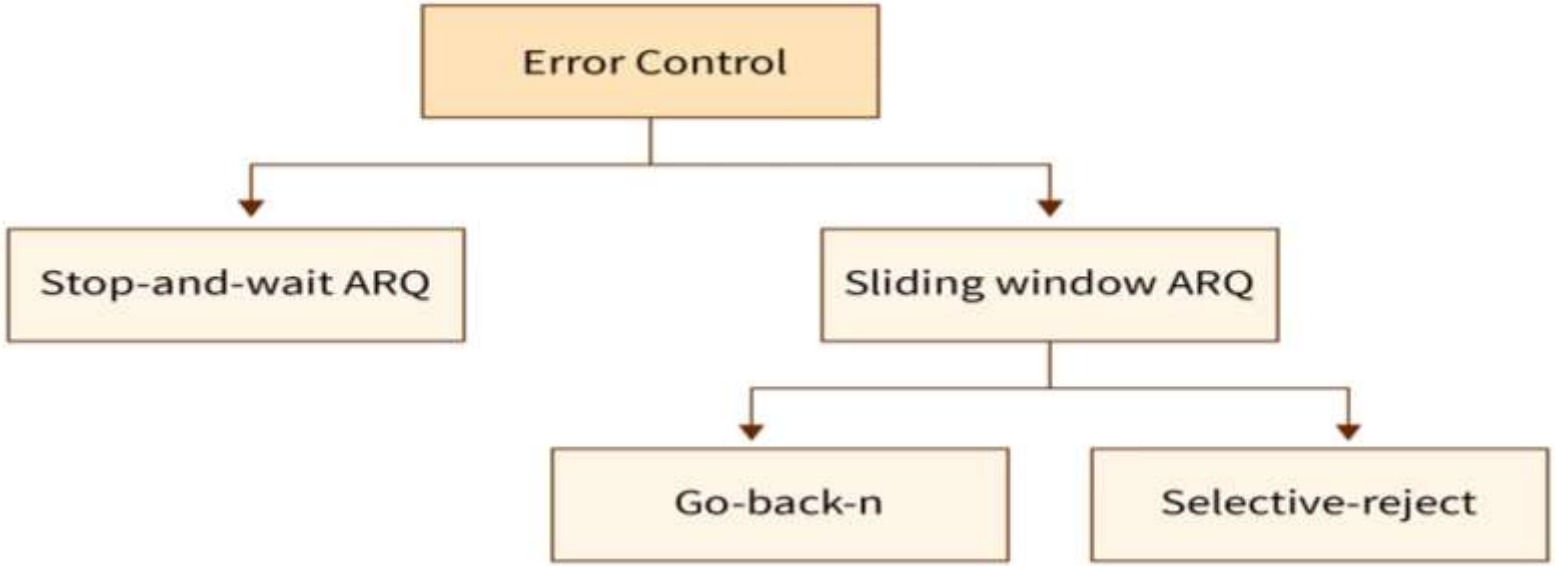
Working:

- ✓ The sender and receiver have a “window” of frames.
- ✓ A window is a space that consists of multiple bytes.
- ✓ The size of the window on the receiver side is always 1.
- ✓ Each frame is sequentially numbered from 0 to n - 1, where n is the window size at the sender side.
- ✓ The sender sends as many frames as would fit in a window.
- ✓ After receiving the desired number of frames, the receiver sends an acknowledgment.
- ✓ The acknowledgment (ACK) includes the number of the next expected frame.


Sender
Receiver
0 1 2 3 0 1 2
s
Frame 0
Frame 1
ACK 2
Frame 2
Lost
Frame 3
NAK 2
Frame 2
Resent
Time

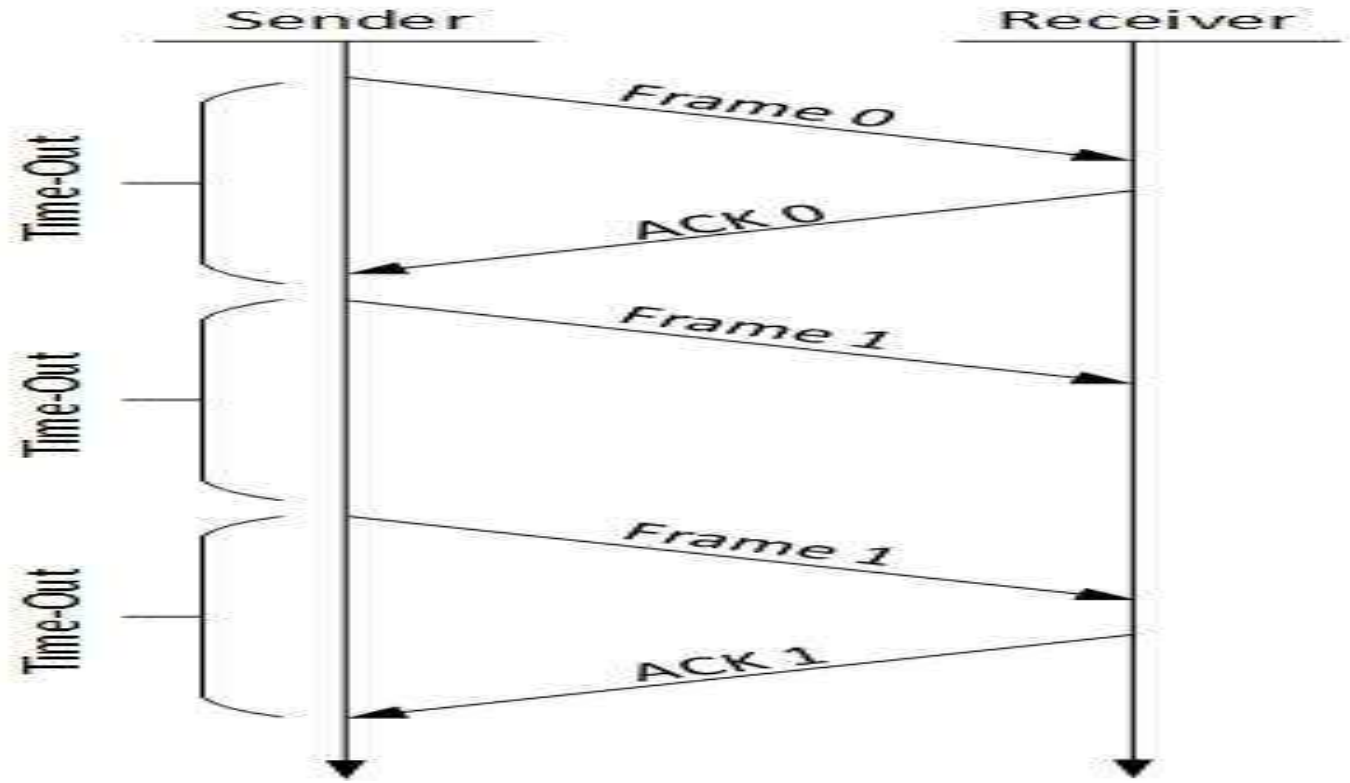
What is Error Control in the Data Link Layer?

- **Error Control** is a combination of both error detection and error correction.
  It ensures that the data received at the receiver end is the same as the one sent by the sender.
- **Error detection** is the process by which the receiver informs the sender about any erroneous frame (damaged or lost) sent during transmission.
- **Error correction** refers to the retransmission of those frames by the sender.



Stop-and-wait ARQ

- In the case of stop-and-wait ARQ **(Automatic Repeat Request)**. after the frame is sent, the sender maintains a timeout counter.
- If acknowledgment of the frame comes in time, the sender transmits the next frame in the queue.
- Else, the sender retransmits the frame and starts the timeout counter.
- In case the receiver receives a negative acknowledgment, the sender retransmits the frame.

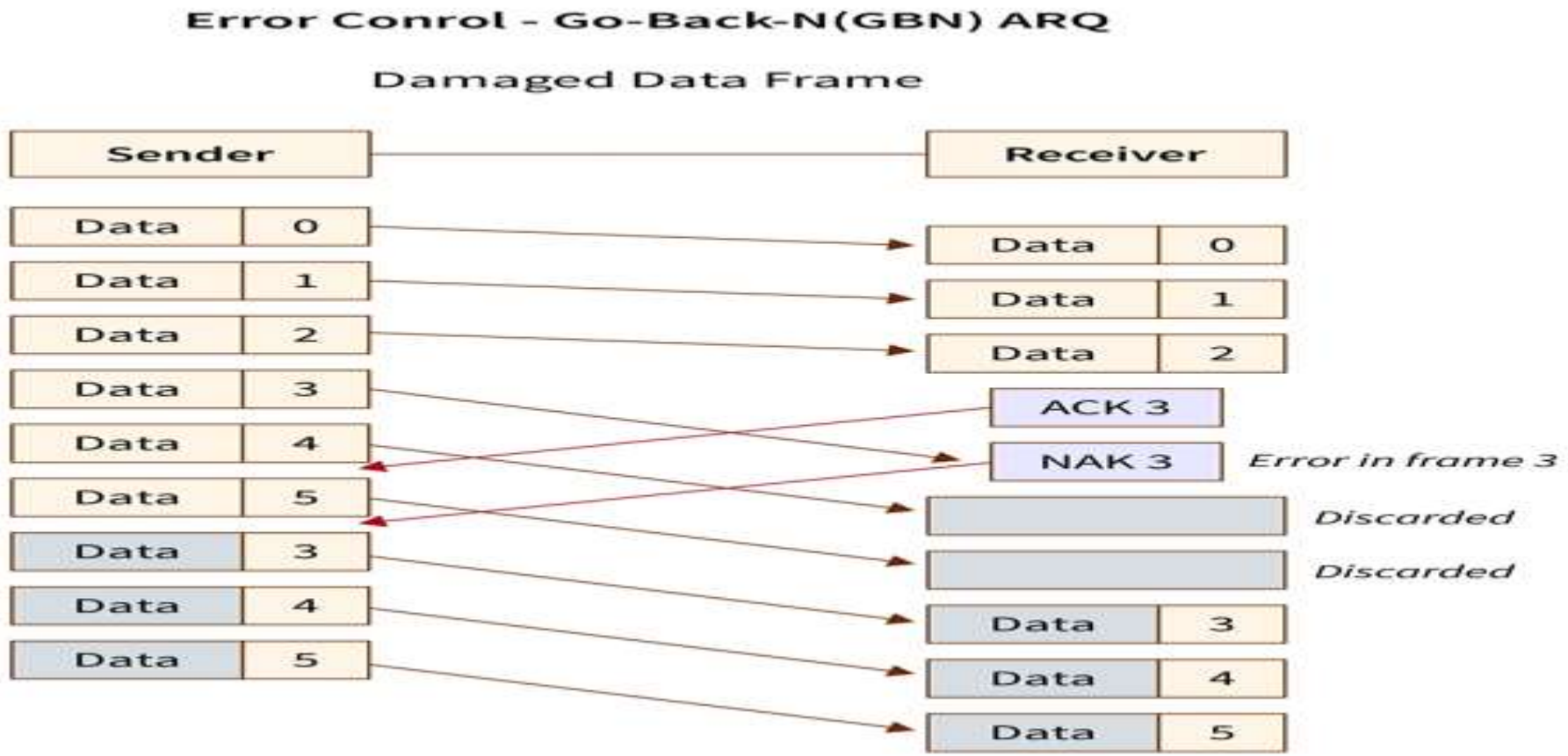


Sliding Window ARQ

To deal with the retransmission of lost or damaged frames, a few changes are made to the sliding window mechanism used in flow control.

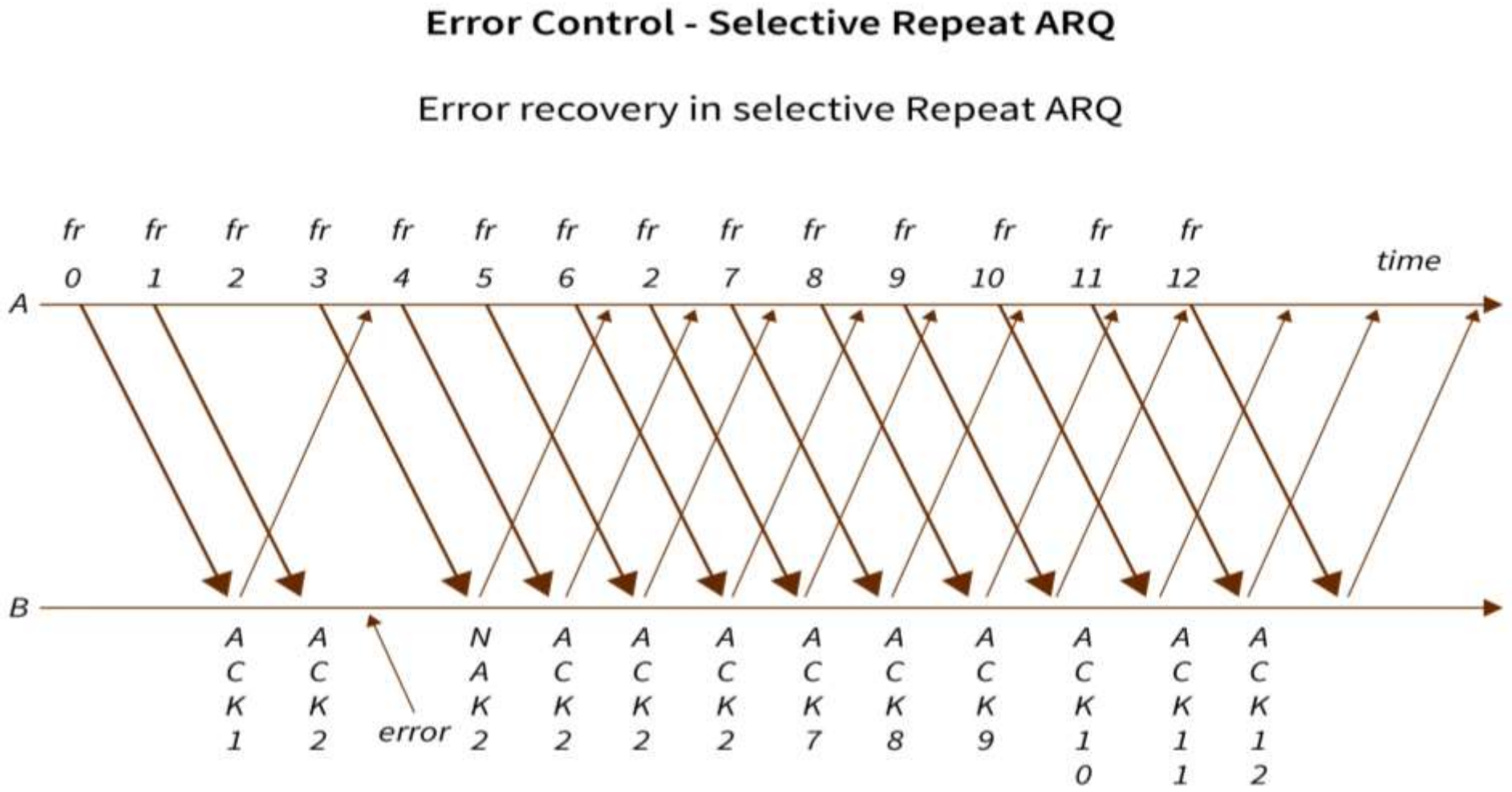
Go-Back-N ARQ :

In Go-Back-N ARQ, if the sent frames are suspected or damaged, all the frames are re-transmitted from the lost packet to the last packet transmitted.



Selective Repeat ARQ:

Selective repeat ARQ/ Selective Reject ARQ is a type of Sliding Window ARQ in which only the suspected or damaged frames are re-transmitted.



Differences between Flow Control and Error Control

Flow control	Error control
Flow control refers to the transmission of data frames from sender to receiver.	Error control refers to the transmission of error-free and reliable data frames from sender to receiver.
Approaches for Flow Control : Feedback-based Flow Control and Rate-based Flow Control.	Approaches for error detection are Checksum, Cyclic Redundancy Check, and Parity Checking. Approaches for error correction are Hamming code, Binary Convolution codes, Reed-Solomon code, and Low-Density Parity-Check codes.
Flow control focuses on the proper flow of data and data loss prevention.	Error control focuses on the detection and correction of errors.
Examples of Flow Control techniques are : 1. Stop and Wait for Protocol, 2. Sliding Window Protocol.	Examples of Error Control techniques are : 1. Stop and Wait for ARQ, 2. Sliding Window ARQ.

How to calculate Data Rate in any channel?

The data rate in a channel can be calculated by considering the amount of information that is transmitted over the channel in a given amount of time.

The formula for data rate, also known as bandwidth, is given by:

Data Rate (R) = Total number of bits transmitted / Total time taken to transmit the bits

High-level Data Link Control (HDLC)

- High-level Data Link Control (HDLC) is a group of communication protocols of the data link layer for transmitting data between network points or nodes.

- Since it is a data link protocol, data is organized into frames.

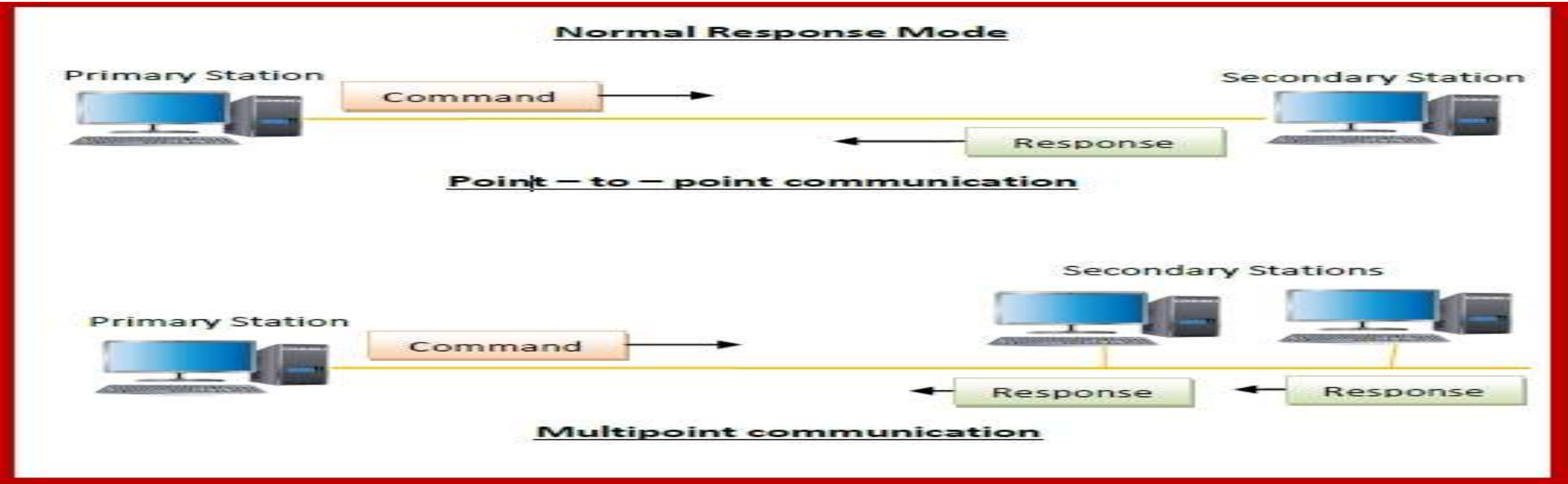
- A frame is transmitted via the network to the destination that verifies its successful arrival.

- It is a bit - oriented protocol that is applicable for both point - to - point and multipoint communications.

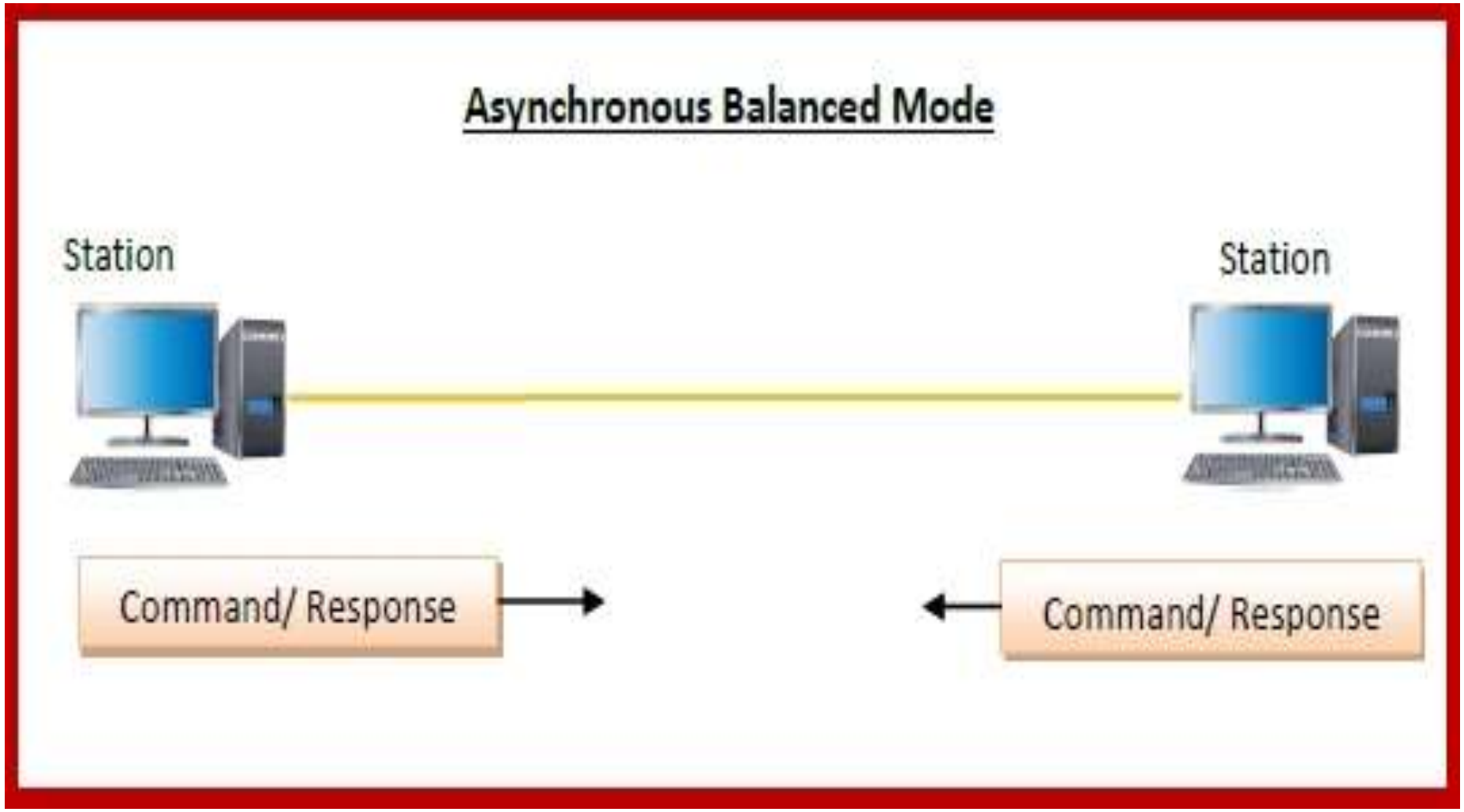
Transfer Modes

HDLC supports two types of transfer modes, normal response mode and asynchronous balanced mode.

•**Normal Response Mode (NRM)** − Here, two types of stations are there, a primary station that send commands and secondary station that can respond to received commands. It is used for both point - to - point and multipoint communications.



Asynchronous Balanced Mode (ABM) − Here, the configuration is balanced, i.e. each station can both send commands and respond to commands. It is used for only point - to - point communications.



HDLC Frame

HDLC is a bit - oriented protocol where each frame contains up to six fields.
The structure varies according to the type of frame.

The fields of a HDLC frame are −

- **Flag −** It is an 8-bit sequence that marks the beginning and the end of the frame. The bit pattern of the flag is 01111110.
- **Address −** It contains the address of the receiver. If the frame is sent by the primary station, it contains the address(es) of the secondary station(s). If it is sent by the secondary station, it contains the address of the primary station. The address field may be from 1 byte to several bytes.
- **Control −** It is 1 or 2 bytes containing flow and error control information.
- **Payload −** This carries the data from the network layer. Its length may vary from one network to another.
- **FCS −** It is a 2 byte or 4 bytes frame check sequence for error detection. The standard code used is CRC (cyclic redundancy code)

HDLC Frame


Flag
Address
Control
Payload
FCS
Flag
1 byte
1 byte
1 byte
variable
2 or 4 bytes
1 byte
(01111110)
(01111110)

Types of HDLC Frames

There are three types of HDLC frames.

The type of frame is determined by the control field of the frame −

- **I-frame** − I-frames or Information frames carry user data from the network layer. They also include flow and error control information that is piggybacked on user data. The first bit of control field of I-frame is 0.

- **S-frame** − S-frames or Supervisory frames do not contain information field. They are used for flow and error control when piggybacking is not required. The first two bits of control field of S-frame is 10.

- **U-frame** − U-frames or Un-numbered frames are used for myriad miscellaneous functions, like link management. It may contain an information field, if required. The first two bits of control field of U-frame is 11.

HDLC Frame

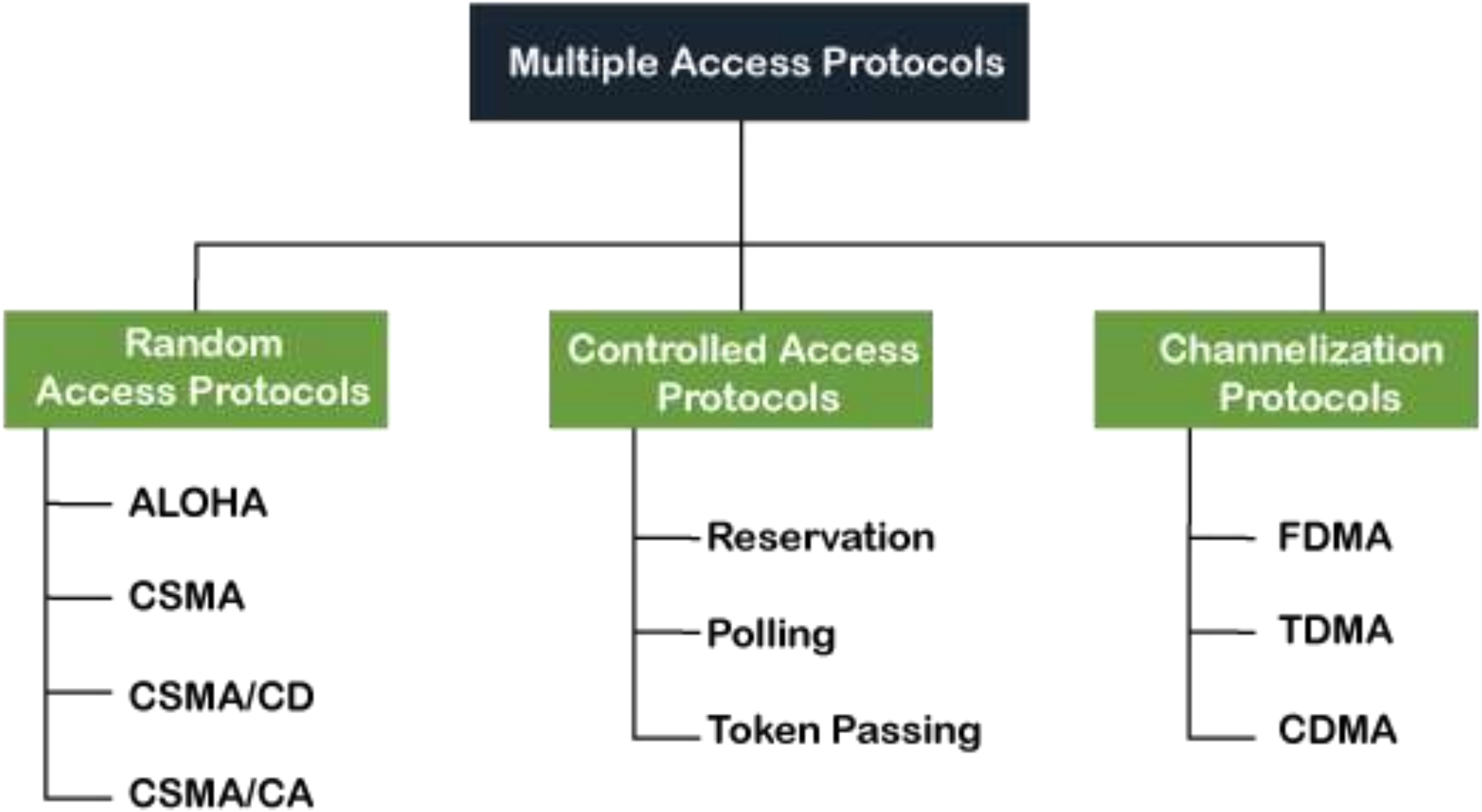
I – Frame

S – Frame

U – Frame

Multiple Access Protocols

- When a sender and receiver have a dedicated link to transmit data packets, the data link control is enough to handle the channel.
- Suppose there is no dedicated path to communicate or transfer the data between two devices.
- In that case, multiple stations access the channel and simultaneously transmits the data over the channel.
- It may create collision and cross talk.
- Hence, the multiple access protocol is required to reduce the collision and avoid crosstalk between the channels.


Multiple Access Protocols
Random Access Protocols
ALOHA
CSMA
CSMA/CD
CSMA/CA
Controlled Access Protocols
Reservation
Polling
Token Passing
Channelization Protocols
FDMA
TDMA
CDMA

A. Random Access Protocol

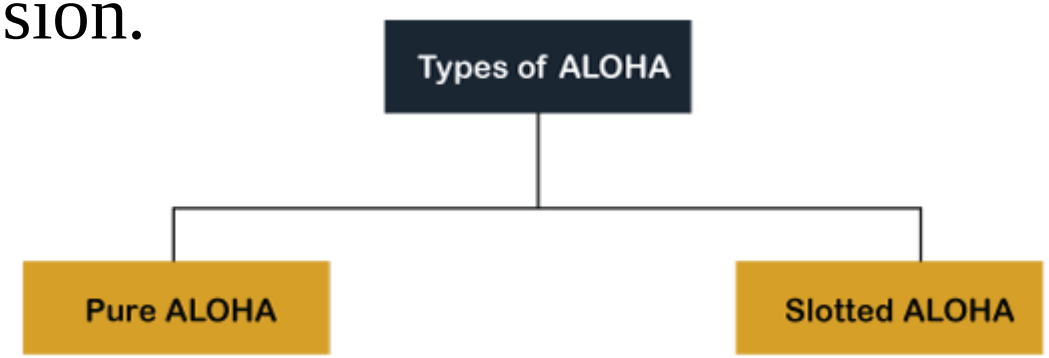
- In this protocol, all the station has the equal priority to send the data over a channel.
- In random access protocol, one or more stations cannot depend on another station nor any station control another station.
- Depending on the channel's state (idle or busy), each station transmits the data frame.
- However, if more than one station sends the data over a channel, there may be a collision or data conflict.
- Due to the collision, the data frame packets may be lost or changed. And hence, it does not receive by the receiver end.

  - ✓ Aloha
  - ✓ CSMA
  - ✓ CSMA/CD
  - ✓ CSMA/CA

ALOHA Random Access Protocol

It is designed for wireless LAN (Local Area Network) but can also be used in a shared medium to transmit data.

Using this method, any station can transmit data across a network simultaneously when a data frameset is available for transmission.



Aloha Rules

- Any station can transmit data to a channel at any time.
- It does not require any carrier sensing.
- Collision and data frames may be lost during the transmission of data through multiple stations.
- Acknowledgment of the frames exists in Aloha.
- Hence, there is no collision detection.
- It requires retransmission of data after some random amount of time.

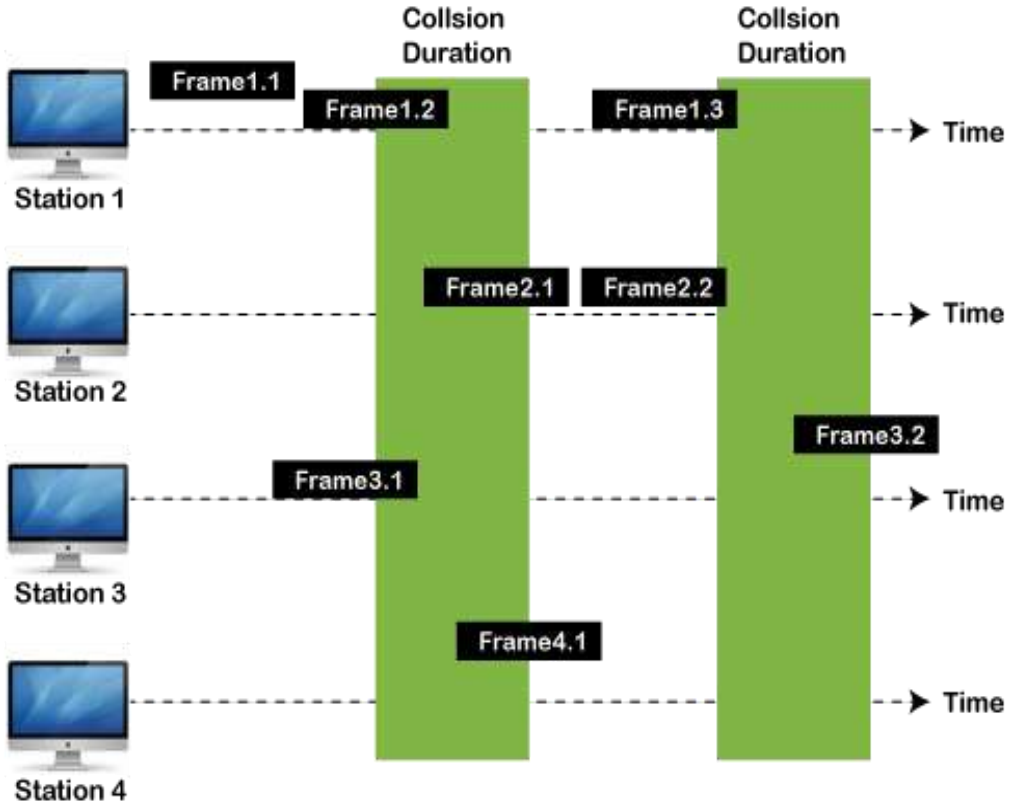
Pure Aloha:

- When a station sends data it waits for an acknowledgement.
- If the acknowledgement doesn't come within the allotted time then the station waits for a random amount of time called back-off time (Tb) and re-sends the data.
- Since different stations wait for different amount of time, the probability of further collision decreases.

Vulnerable Time = 2* Frame transmission time

Throughput = G exp{-2*G}

Maximum throughput = 0.184 for G=0.5



Frames in Pure ALOHA

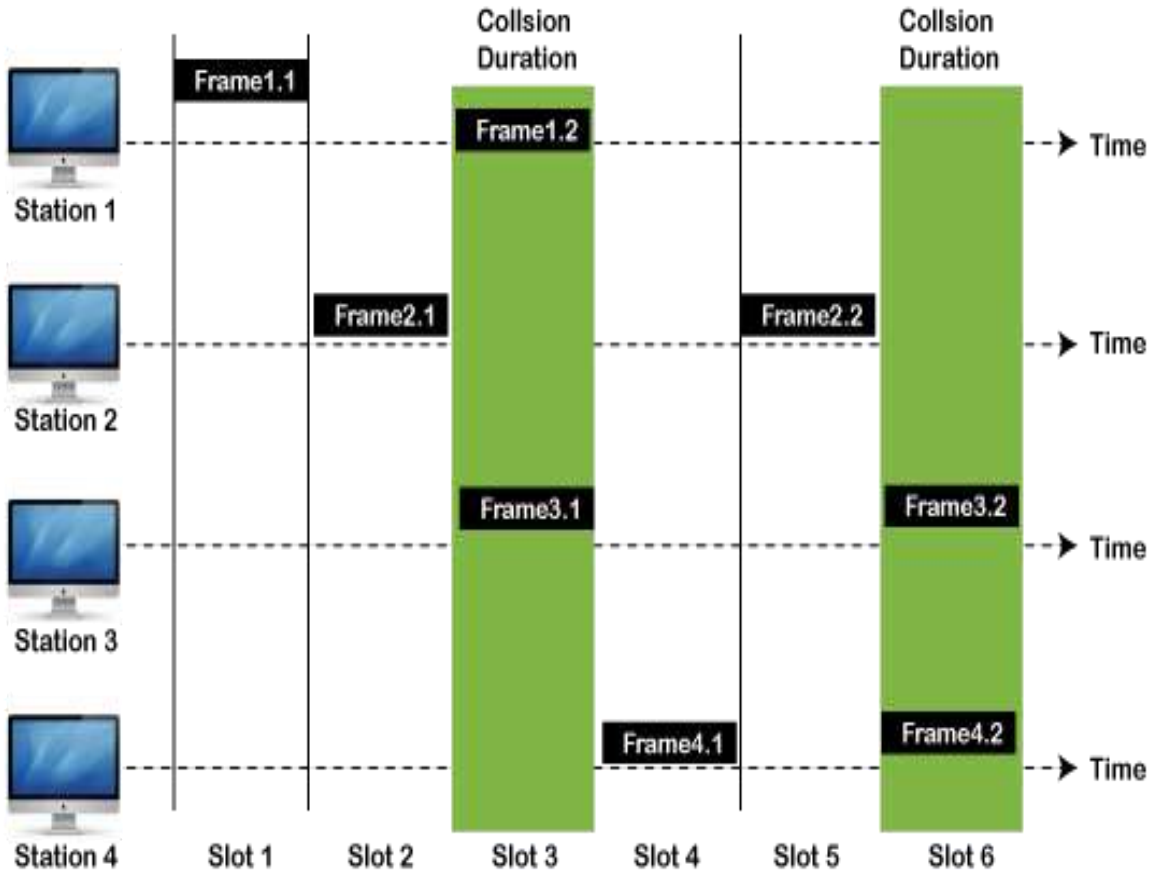
•**Slotted Aloha:**

- It is similar to pure aloha, except that we divide time into slots and sending of data is allowed only at the beginning of these slots.
- If a station misses out the allowed time, it must wait for the next slot.
- This reduces the probability of collision.

Vulnerable Time = Frame transmission time

Throughput = G exp{-*G}

Maximum throughput = 0.368 for G=1



Frames in Slotted ALOHA

CSMA – Carrier Sense Multiple Access ensures fewer collisions as the station is required to first sense the medium (for idle or busy) before transmitting data.
If it is idle then it sends data, otherwise it waits till the channel becomes idle. However there is still chance of collision in CSMA due to propagation delay.

CSMA access modes-

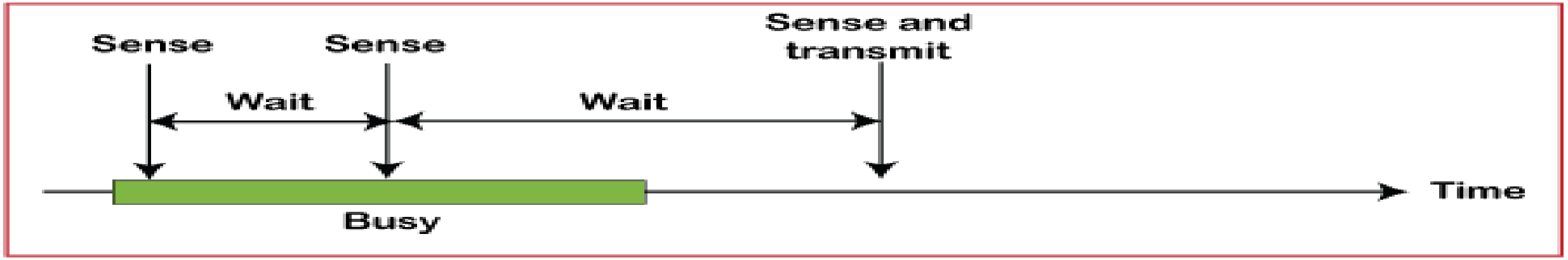
1-persistent: The node senses the channel, if idle it sends the data, otherwise it continuously keeps on checking the medium for being idle and transmits unconditionally(with 1 probability) as soon as the channel gets idle.

Non-Persistent: The node senses the channel, if idle it sends the data, otherwise it checks the medium after a random amount of time (not continuously) and transmits when found idle.

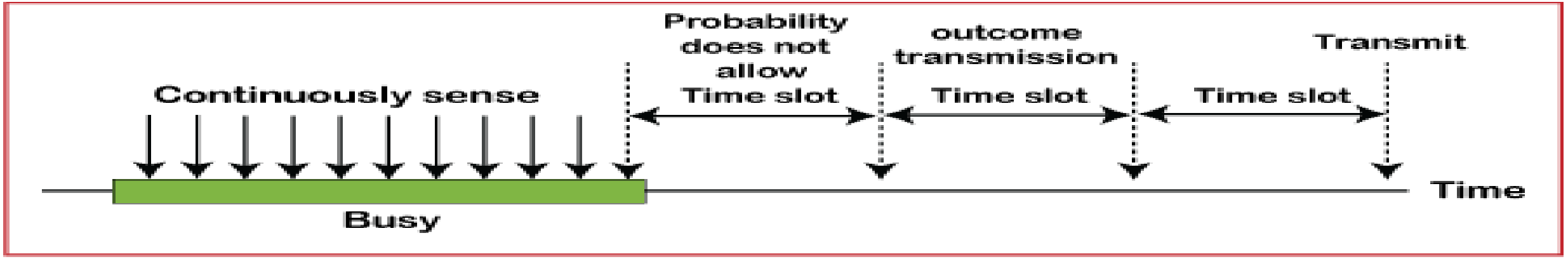
P-persistent: The node senses the medium, if idle it sends the data with p probability. If the data is not transmitted ((1-p) probability) then it waits for some time and checks the medium again, now if it is found idle then it send with p probability. This repeat continues until the frame is sent. It is used in Wifi and packet radio systems.

O-persistent: Superiority of nodes is decided beforehand and transmission occurs in that order. If the medium is idle, node waits for its time slot to send data.

a. 1-persistent



b. Nonpersistent



c. p-persistent

CSMA/ CD

- It is a carrier sense multiple access/ collision detection network protocol to transmit data frames.
- The CSMA/CD protocol works with a medium access control layer.
- Therefore, it first senses the shared channel before broadcasting the frames, and if the channel is idle, it transmits a frame to check whether the transmission was successful.
- If the frame is successfully received, the station sends another frame.
- If any collision is detected in the CSMA/CD, the station sends a jam/ stop signal to the shared channel to terminate data transmission.
- After that, it waits for a random time before sending a frame to a channel.

CSMA/ CA

- It is a carrier sense multiple access/collision avoidance network protocol for carrier transmission of data frames.
- It is a protocol that works with a medium access control layer.
- When a data frame is sent to a channel, it receives an acknowledgment to check whether the channel is clear.
- If the station receives only a single (own) acknowledgments, that means the data frame has been successfully transmitted to the receiver.
- But if it gets two signals (its own and one more in which the collision of frames), a collision of the frame occurs in the shared channel.
- Detects the collision of the frame when a sender receives an acknowledgment signal.

Following are the methods used in the CSMA/ CA to avoid the collision:

- **Interframe space –** Station waits for medium to become idle and if found idle it does not immediately send data (to avoid collision due to propagation delay) rather it waits for a period of time called Interframe space or IFS.
  After this time it again checks the medium for being idle.
  The IFS duration depends on the priority of station.
- **Contention Window –** It is the amount of time divided into slots.
  If the sender is ready to send data, it chooses a random number of slots as wait time which doubles every time medium is not found idle.
  If the medium is found busy it does not restart the entire process, rather it restarts the timer when the channel is found idle again.
- **Acknowledgement –** The sender re-transmits the data if acknowledgement is not received before time-out.

Controlled Access Protocol

- It is a method of reducing data frame collision on a shared channel.
- In the controlled access method, each station interacts and decides to send a data frame by a particular station approved by all other stations.
- It means that a single station cannot send the data frames unless all other stations are not approved.
- It has three types of controlled access: Reservation, Polling, and Token Passing.

Channelization Protocols

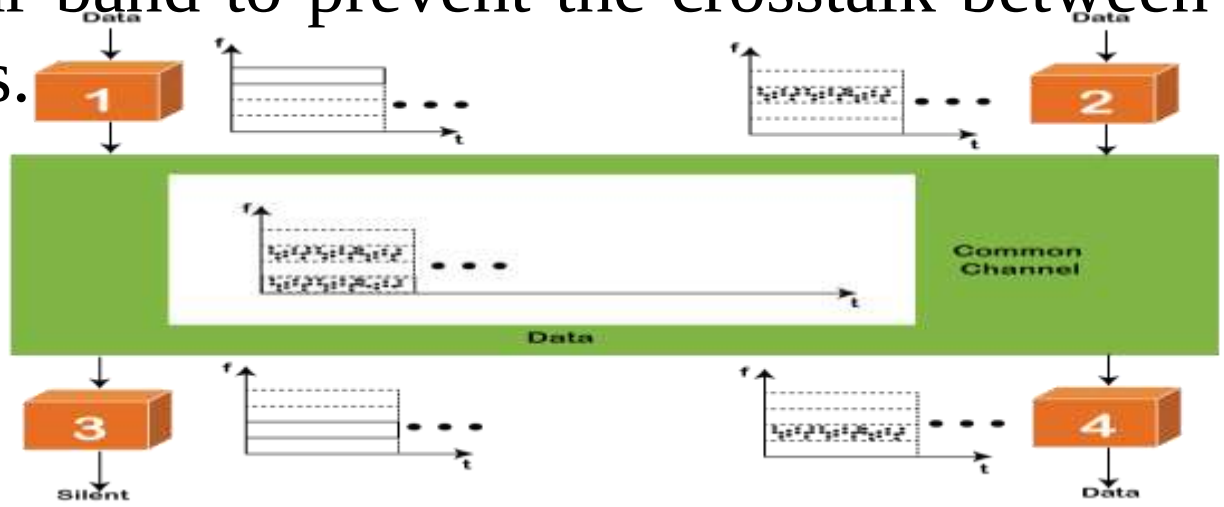
- It is a channelization protocol that allows the total usable bandwidth in a shared channel to be shared across multiple stations based on their time, distance and codes.
- It can access all the stations at the same time to send the data frames to the channel.

Following are the various methods to access the channel based on their time, distance and codes:

- ✓ FDMA (Frequency Division Multiple Access)
- ✓ TDMA (Time Division Multiple Access)
- ✓ CDMA (Code Division Multiple Access)

FDMA

- ➢ It is a frequency division multiple access (FDMA) method used to divide the available bandwidth into equal bands so that multiple users can send data through a different frequency to the sub-channel.
- ➢ Each station is reserved with a particular band to prevent the crosstalk between the channels and interferences of stations.



TDMA

- Time Division Multiple Access (TDMA) is a channel access method.
- It allows the same frequency bandwidth to be shared across multiple stations.
- to avoid collisions in the shared channel, it divides the channel into different frequency slots that allocate stations to transmit the data frames.
- The same frequency bandwidth into the shared channel by dividing the signal into various time slots to transmit it.
- However, TDMA has an overhead of synchronization that specifies each station's time slot by adding synchronization bits to each slot.

Code Division Multiple Access (CDMA) –

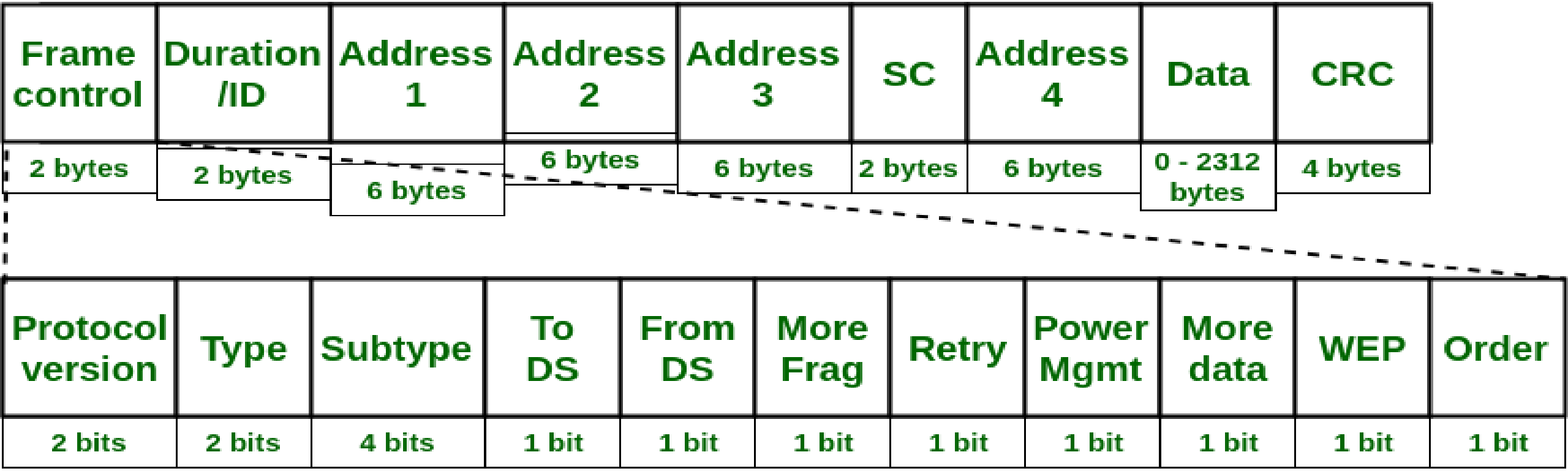
- One channel carries all transmissions simultaneously.

- There is neither division of bandwidth nor division of time.

For example, if there are many people in a room all speaking at the same time, then also perfect reception of data is possible if only two person speak the same language.
Similarly, data from different stations can be transmitted simultaneously in different code languages.

Frame Format of IEEE 802.11

IEEE 802.11 MAC layer data frame consists of 9 fields:



IEEE 802.11 MAC Frame Structure

Frame Control

It is 2 bytes long and defines type of frame and control information.

The types of fields present in FC are:

- ✓ **Version:** Indicates the current protocol version.
- ✓ **Type:** Determines the function of frame i.e. management(00), control(01) or data(10).
- ✓ **Subtype:** Indicates subtype of frame like 0000 for association request, 1000 for beacon.
- ✓ **To DS:** When set indicates that the destination frame is for DS(distribution system).
- ✓ **From DS:** When set indicates frame coming from DS.
- ✓ **More frag (More fragments):** When set to 1 means frame is followed by other fragments.
- ✓ **Retry:** If the current frame is a re-transmission of an earlier frame, this bit is set to 1.
- ✓ **Power Mgmt (Power Management):** It indicates the mode of a station after successful transmission of a frame. Set to '1' field indicates that the station goes into power-save mode. If the field is set to 0, the station stays active.
- ✓ **More data:** It is used to indicate to the receiver that a sender has more data to send than the current frame.
- ✓ **WEP:** It indicates that the standard security mechanism of 802.11 is applied.
- ✓ **Order:** If this bit is set to 1 the received frames must be processed in strict order.

- **Duration / ID**

  It contains the value indicating the period of time in which the medium is occupied (in μs).
- **Address 1 to 4**

  These fields contain standard IEEE 802 MAC addresses (48 bit each).

  The meaning of each address is defined by DS bits in the frame control field.
- **SC (Sequence Control)**

  It consists of 2 sub-fields i.e. sequence number (12 bits) and fragment number (4 bits).

  Sequence number is used to filter duplicate frames.
- **Data**

  It is a variable length field which contains information specific to individual frames which is transferred transparently from a sender to the receiver.
- **CRC** (Cyclic Redundancy Check)

  It contains 32 bit CRC error detection sequence to ensure error free frame.

Features of the IEEE 802.11 MAC frame:

- ✓ **Frame Control Field:** The frame control field contains information about the type of frame, the data rate, and the power management status.
- ✓ **Duration Field:** The duration field specifies the length of time that the channel will be occupied by the transmission.
- ✓ **Address Fields:** The address fields specify the source and destination MAC addresses of the Wi-Fi devices involved in the communication.
- ✓ **Sequence Control Field:** The sequence control field is used to identify and manage the transmission sequence of the frames.
- ✓ **Frame Body:** The frame body contains the actual data being transmitted between Wi-Fi devices, such as IP packets, TCP segments, or UDP datagrams.
- ✓ **Frame Check Sequence:** The frame check sequence (FCS) is used to check the integrity of the data transmitted in the frame and to detect any transmission errors.
- ✓ **Management, Control, and Data Frames:** The IEEE 802.11 MAC frame defines three types of frames: management frames, control frames, and data frames. Management frames are used for network management, control frames are used for coordination between Wi-Fi devices, and data frames are used for the transmission of actual data.
- ✓ **Fragmentation:** The IEEE 802.11 MAC frame supports fragmentation, which allows large data packets to be divided into smaller fragments for transmission.
- ✓ **Acknowledgments:** The IEEE 802.11 MAC frame uses acknowledgments to confirm the successful transmission of frames and to request the retransmission of any frames that were not successfully received.