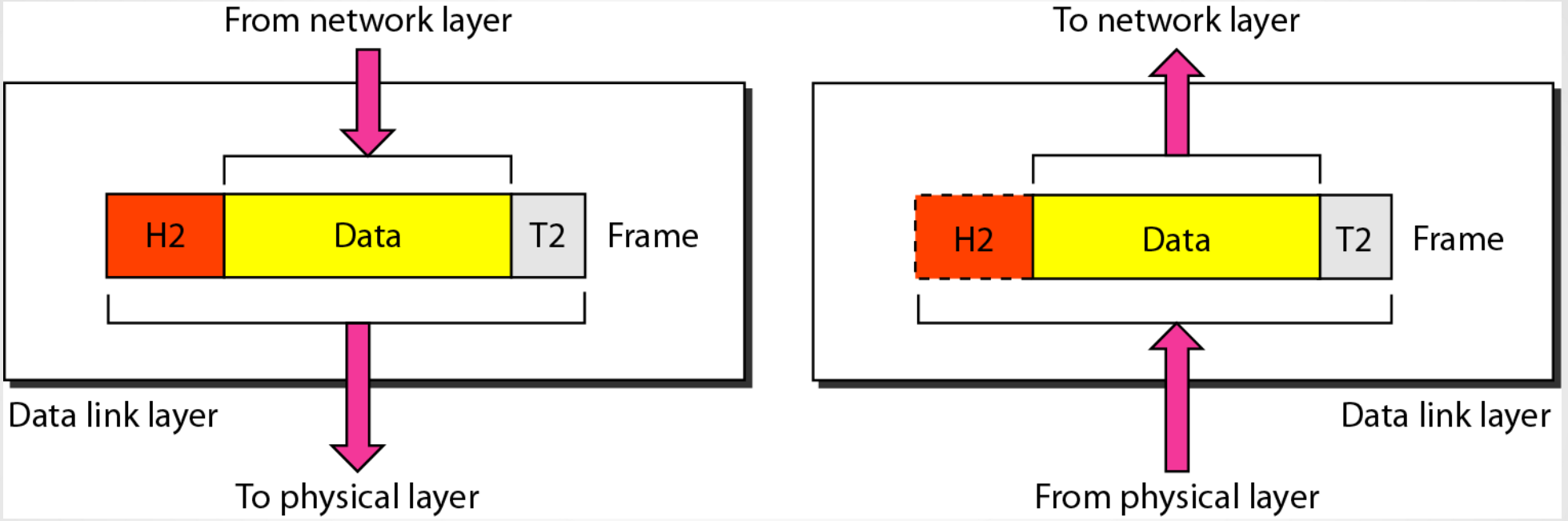
Unit -2

Logical Link Control Layer

Presentation Prepaired by: ***Prof. Atul G. Pawar,***
PCCOE, Pune.

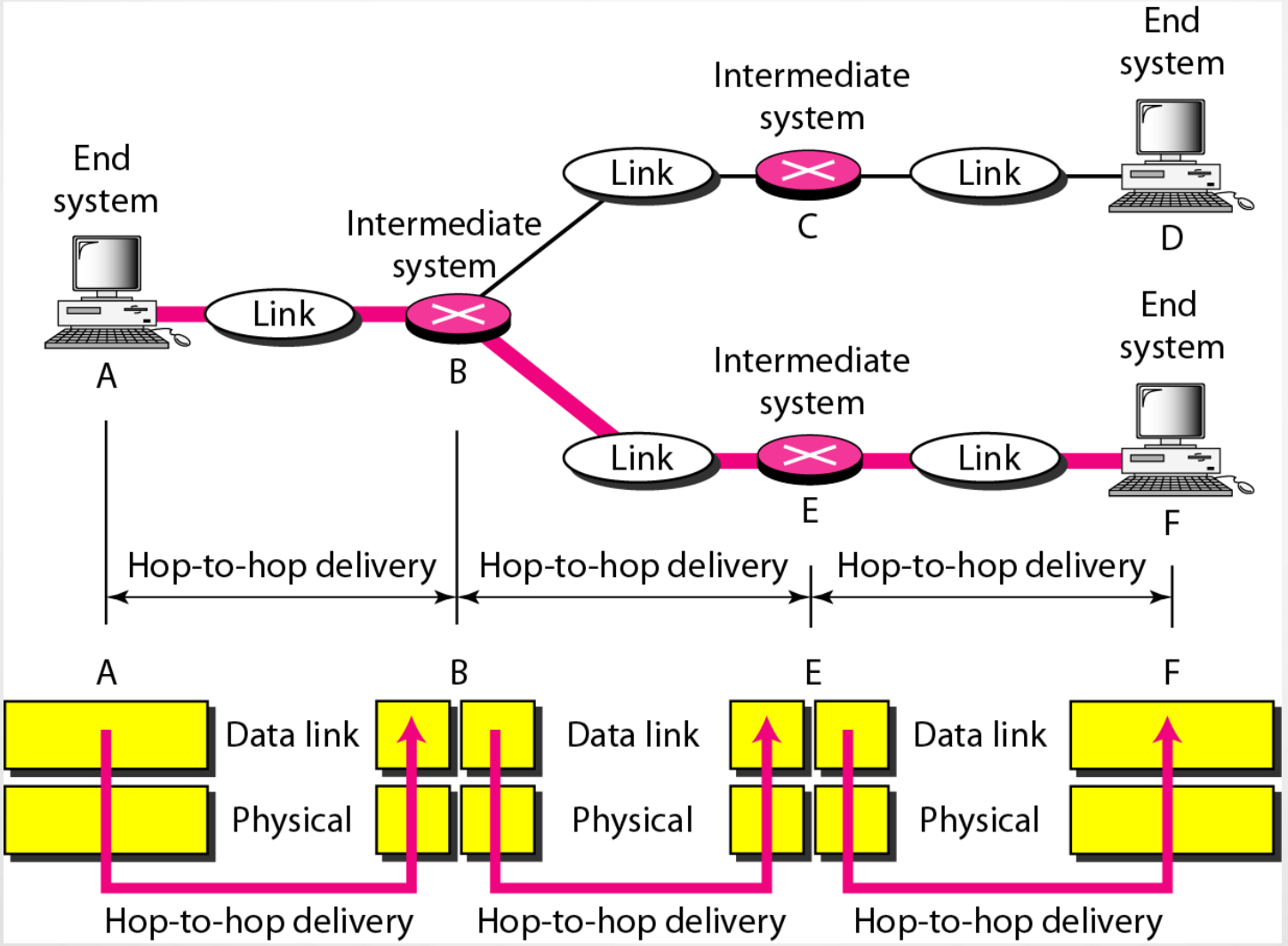
Data link layer



Note

The data link layer is responsible for moving frames from one hop (node) to the next.

Fig. *Hop-to-hop delivery*



Data Link Layer

- **1. Logical Link Control (LLC) Layer**
  - Error Control
  - Flow Control
  - Framing
  - Link Managment
- **2. MAC Layer**
  - Sharing of Medium using diff. Protocols
  - Packet transmission and reception service.

1. Logical Link Control (LLC)

- **Error Control**
  - - Error control is one of its most important tasks.
  - - Error-control schemes typically use redundancy,
  - retransmissions, choice of transmit parameters like packet
  - sizes and transmit powers.
  - - Increased error rates or increased reliability targets increas
  - the energy consumption.
- **Flow Control**

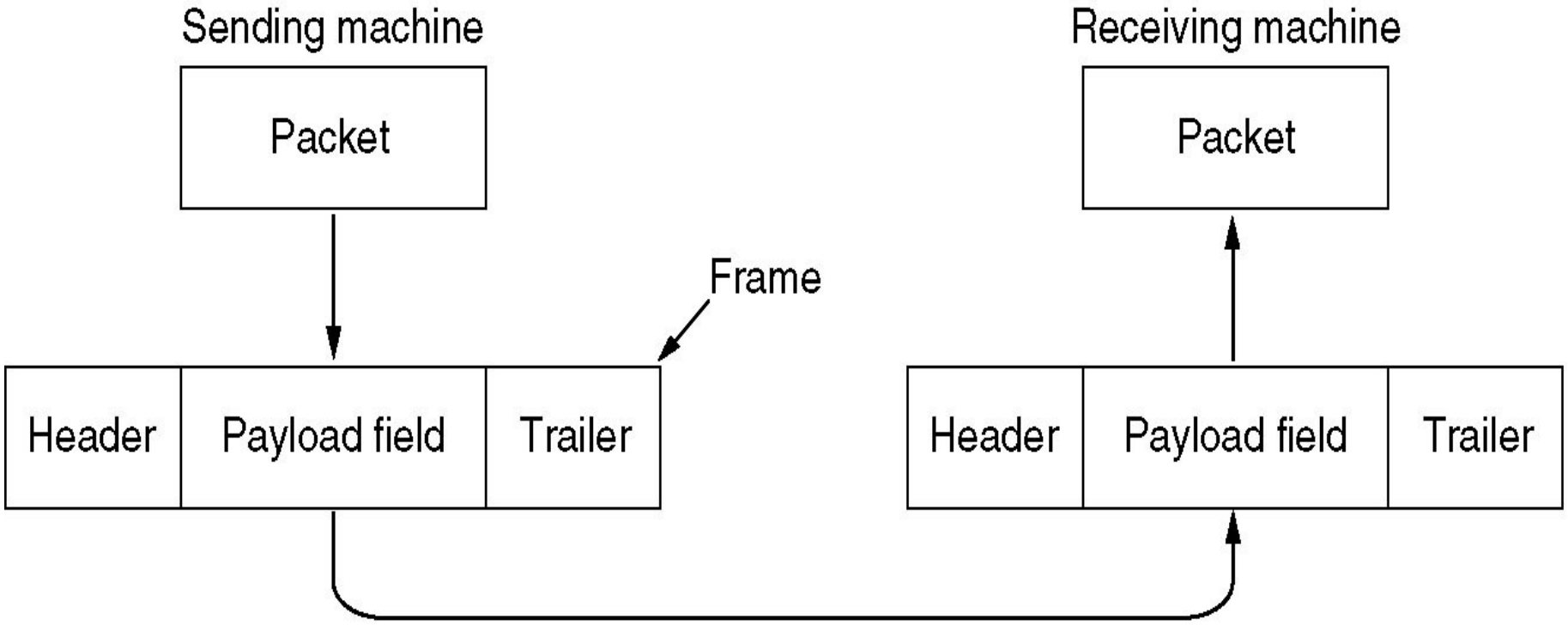
Flow-control mechanisms introduce some signaling to let the transmitter slow down transmission.

Link Managment

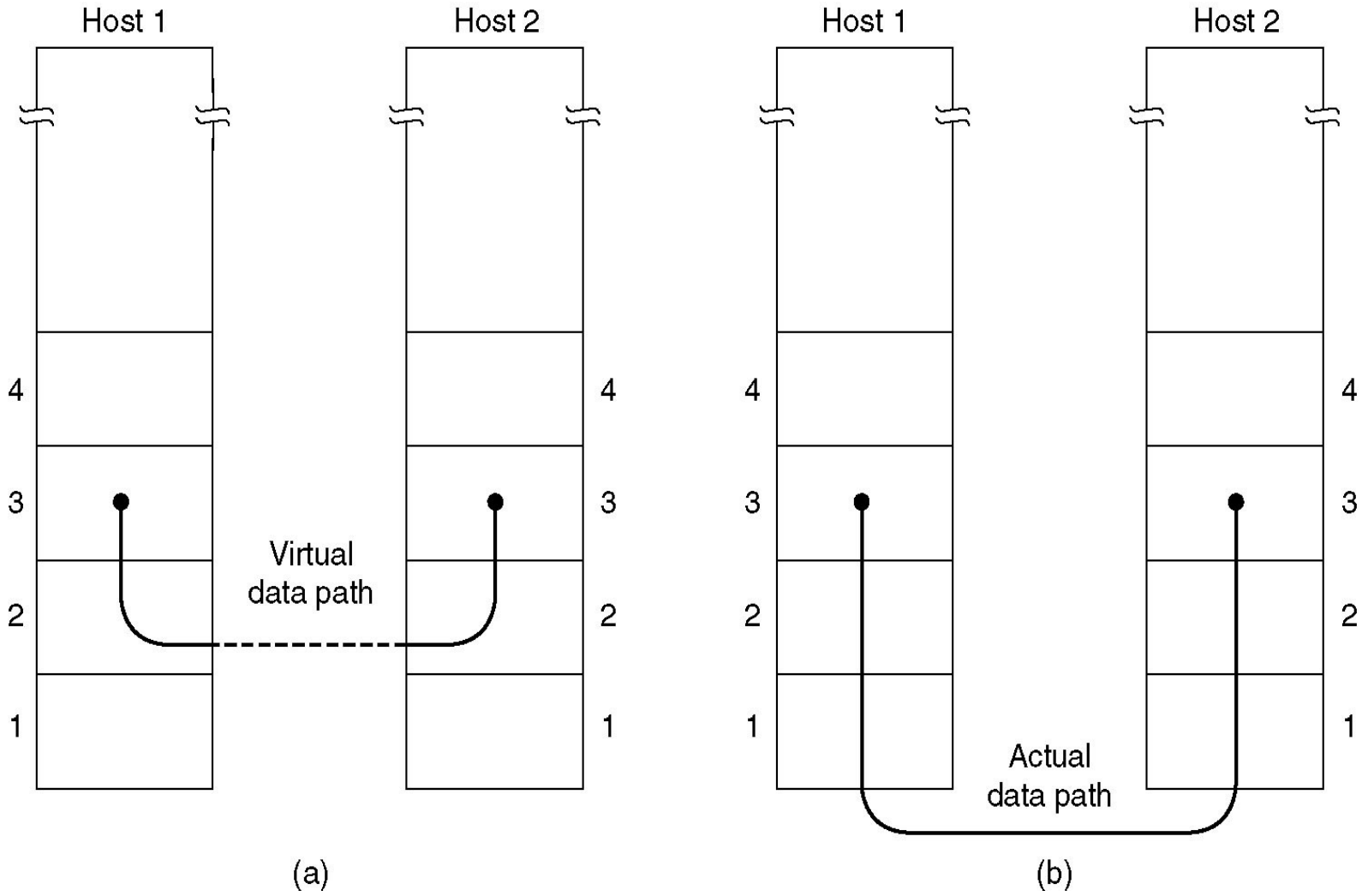
- This mechanism involves discovery, setup, maintenance,
- and teardown of links to neighbors

- **Framing**
  - User data is fragmented and formatted into packets or fram

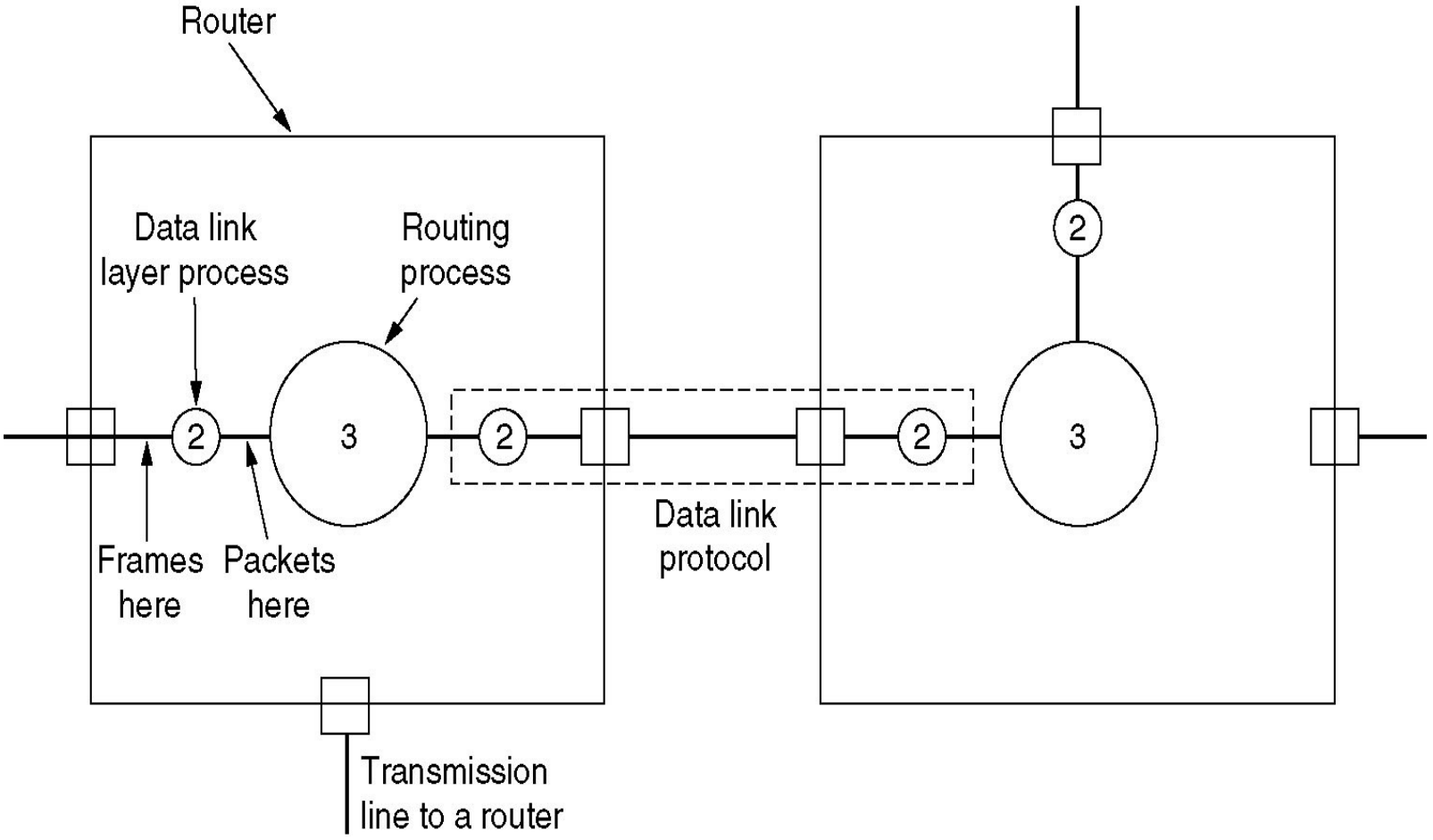
Functions of the Data Link Layer cont..



Services Provided to Network Layer



Placement of DLL



· Design Issues of DLL

- Services provided to Network Layer
  - Unacknowledged connectionless service.
  - Acknowledged connectionless service.
  - Acknowledged connection-oriented service.
- Framing
- Error Control
- Flow Control

Unacknowledged connectionless service (Best Effort).

- The receiver does not return acknowledgments to the sender, so the sender has no way of knowing if a frame has been successfully delivered.
- Losses are taken care of at higher layers
- Used on reliable medium like coax cables or optical fiber, where the error rate is low.
- Appropriate for voice, where delay is worse than bad data.

Acknowledged connectionless service.

- Useful on unreliable medium like wireless.
- Acknowledgements add delays.
- Adding ack in the DLL rather than in the NL is just an optimization and not a requirement.
- On reliable channels, like fiber, the overhead associated with the ack is not justified.

Acknowledged connection-oriented service.

- Most reliable,
- Guaranteed service –
  - Each frame sent is indeed received
  - Each frame is received exactly once
  - Frames are received in order
- Special care has to be taken to ensure this in connectionless services

• 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*

- *When a message is divided into smaller frames, a single-bit error affects only that small frame.*

Framing Continue....

Fixed-Size Framing:

Frames can be of fixed or variable size.

Variable-Size Framing:

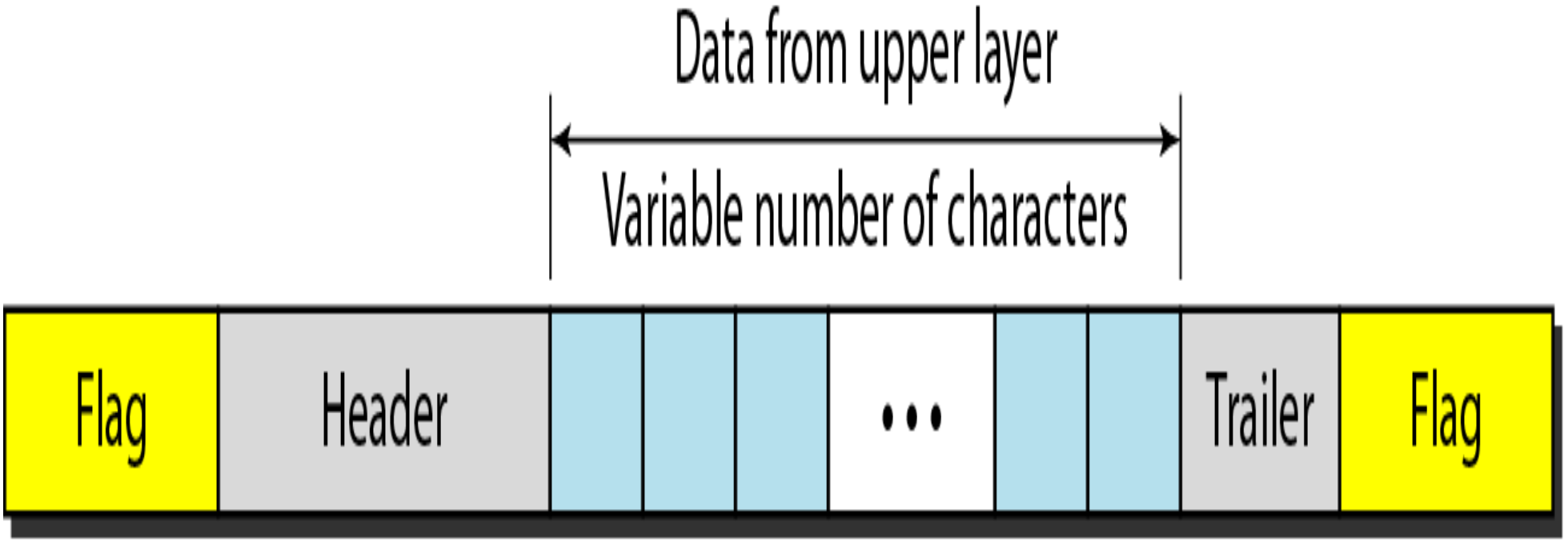
In variable-size framing, we need a way to define the end of the frame and the beginning of the next.

Two approaches were used for this purpose:

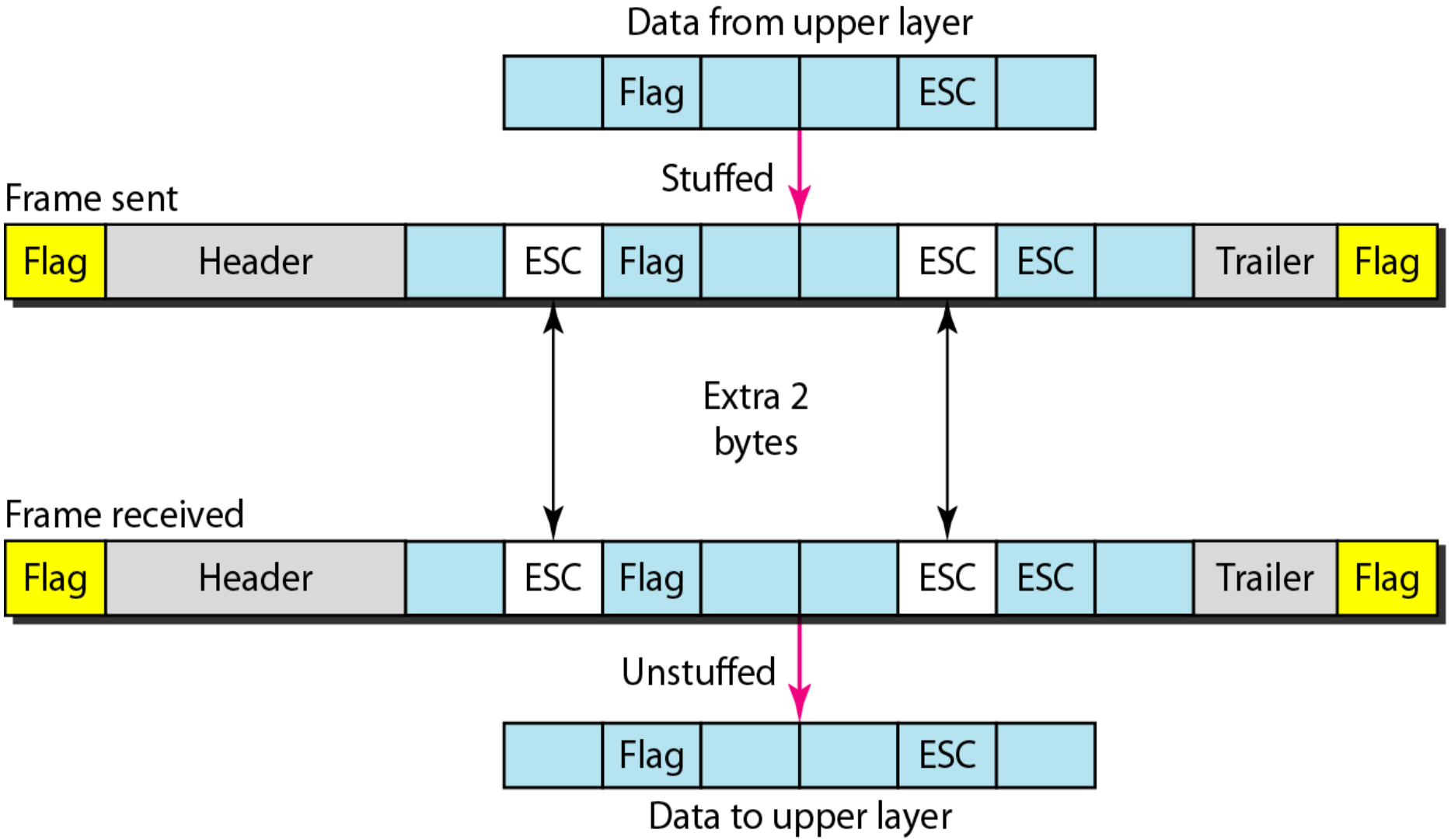
Character-oriented Protocol approach

Bit-oriented Protocol approach.

- Figure *A frame in a character-oriented protocol*



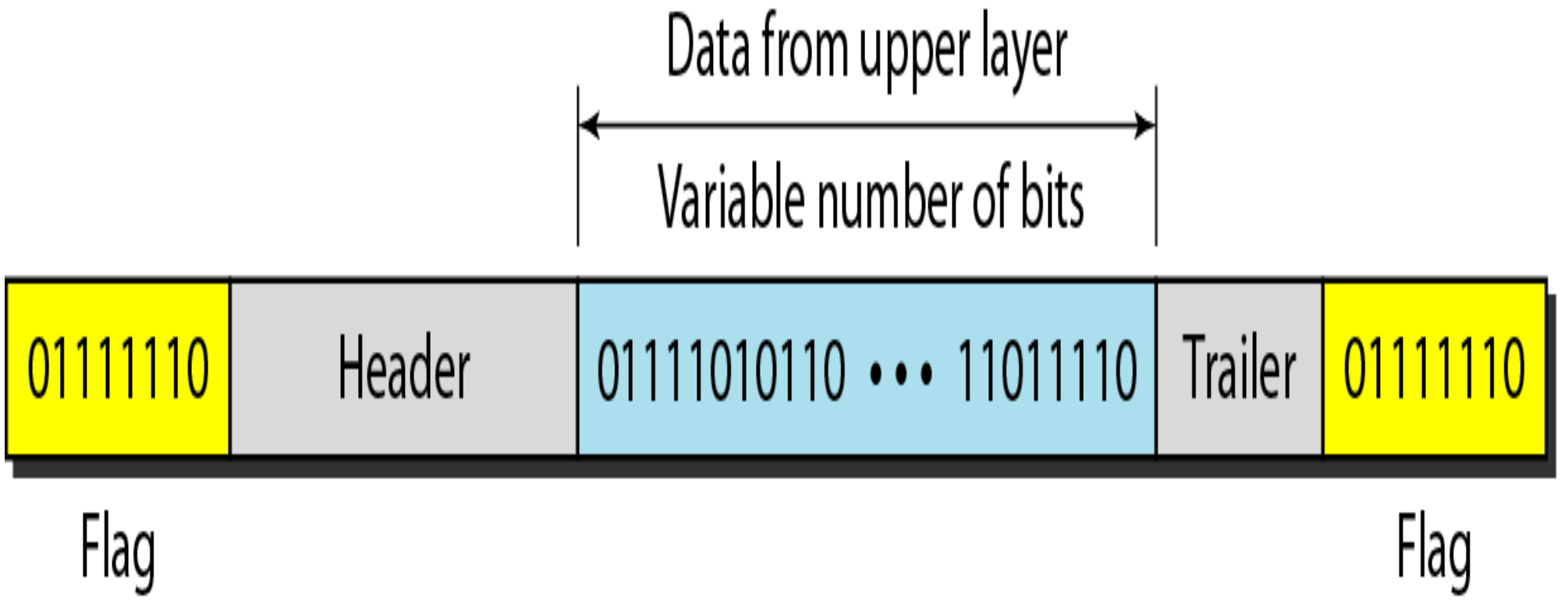
- *Byte stuffing and unstuffing*



- *Note*

Byte stuffing is the process of adding 1 extra byte whenever there is a flag or escape character in the text.

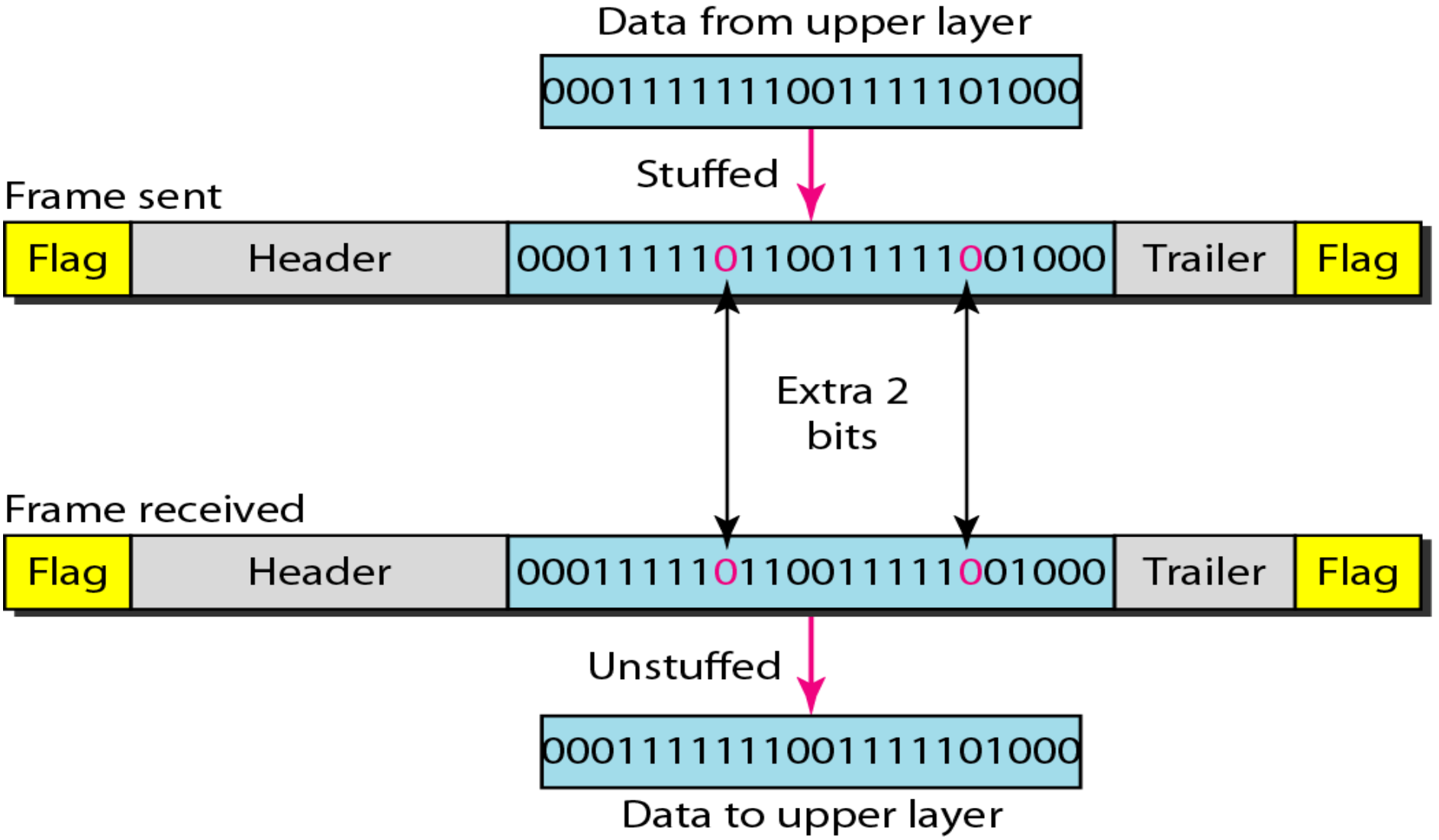
- Figure *A frame in a bit-oriented protocol*



- *Note*

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.

- Figure *Bit stuffing and unstuffing*



• 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.*

Flow Control

- *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.

Error Control

- *Note*

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

Note

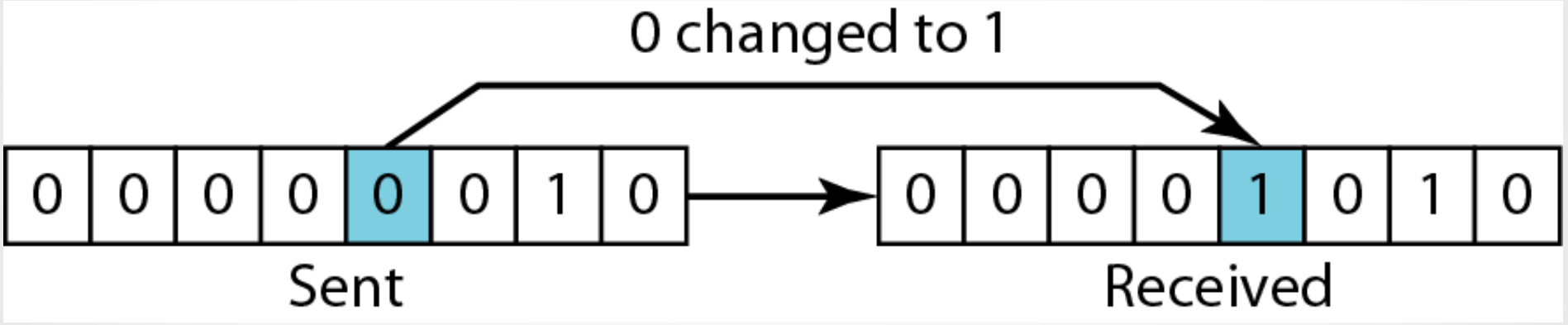
Data can be corrupted during transmission.

Some applications require that errors be detected and corrected.

Note

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

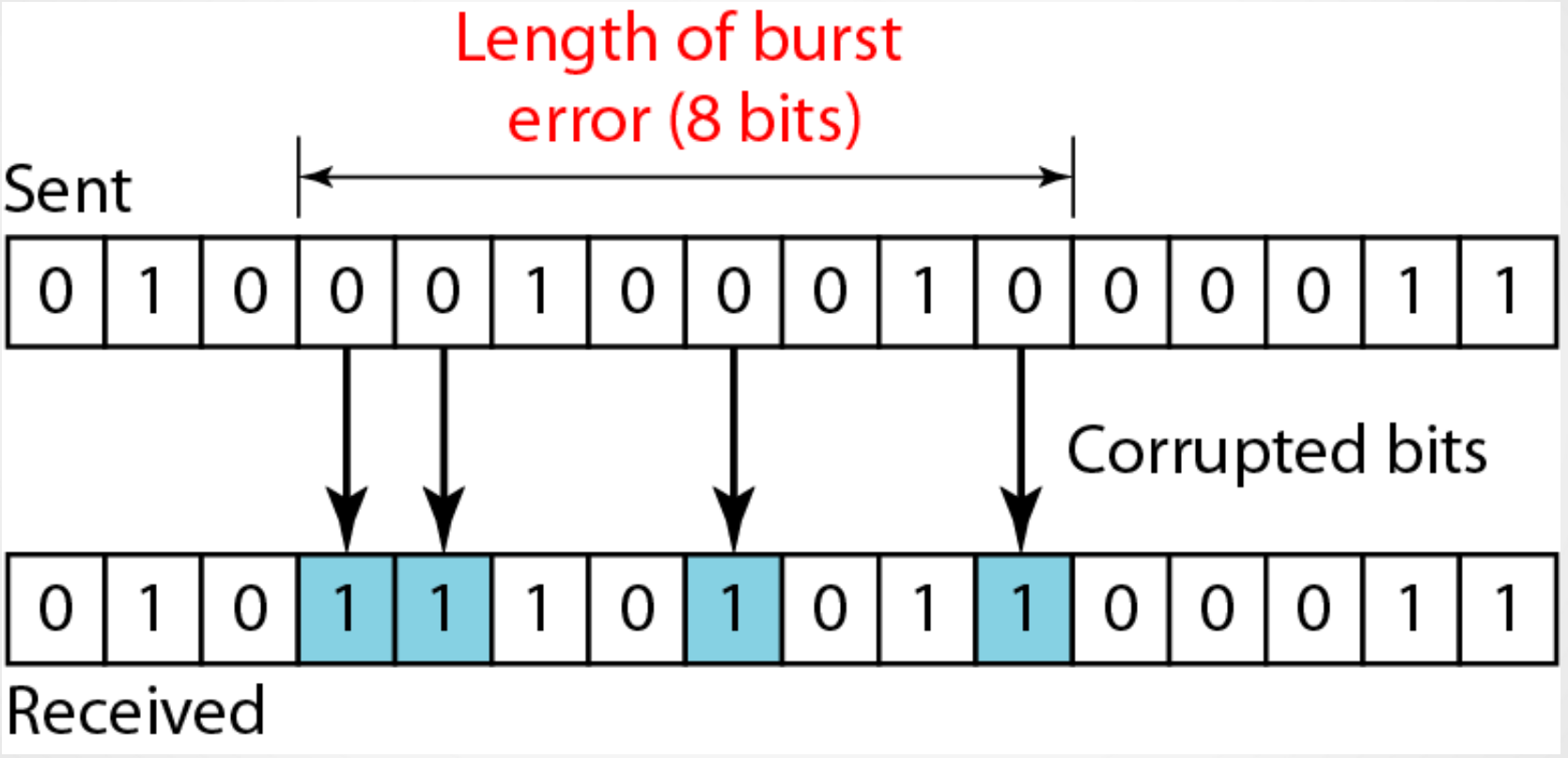
Figure *Single-bit error*



Note

A burst error means that 2 or more bits in the data unit have changed.

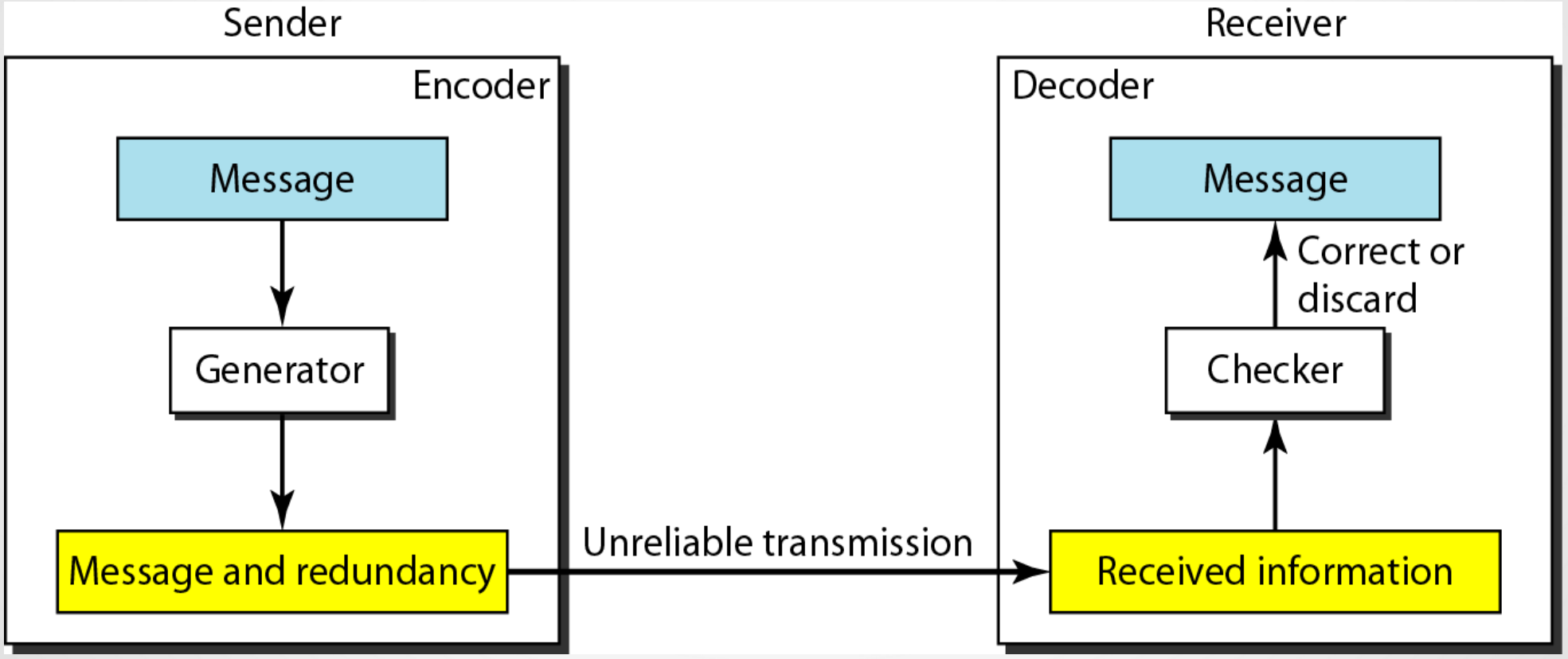
Burst error of length 8



Note

To detect or correct errors, we need to send extra (redundant) bits with data.

Figure : *The structure of encoder and decoder*



XORing of two single bits or two words

$0 \oplus 0 = 0$ $\quad$ $1 \oplus 1 = 0$

a. Two bits are the same, the result is 0.

$0 \oplus 1 = 1$ $\quad$ $1 \oplus 0 = 1$

b. Two bits are different, the result is 1.

$$\begin{array}{ccccccc} & 1 & 0 & 1 & 1 & 0 \\ \oplus & 1 & 1 & 1 & 0 & 0 \\ \hline & 0 & 1 & 0 & 1 & 0 \end{array}$$

c. Result of XORing two patterns

BLOCK CODING

- ***In block coding, we divide our message into blocks, each of k bits, called datawords. We add r redundant bits to each block to make the length n = k + r. The resulting n-bit blocks are called codewords.***

Eg. -1 *A code for error detection*

Datawords	*Codewords*
00	000
01	011
10	101
11	110

Note

An error-detecting code can detect only the types of errors for which it is designed; other types of errors may remain undetected.

Note

The Hamming distance between two words is the number of differences between corresponding bits.

Example 10.4

Let us find the Hamming distance between two pairs of words.

1. The Hamming distance d(000, 011) is 2 because

$$000 \oplus 011 \text{ is } 011 \text{ (two 1s)}$$

2. The Hamming distance d(10101, 11110) is 3 because

$$10101 \oplus 11110 \text{ is } 01011 \text{ (three 1s)}$$

Note

The minimum Hamming distance is the smallest Hamming distance between all possible pairs in a set of words.

Example 10.5

Find the minimum Hamming distance of the coding scheme in Table 10.1.

Solution

We first find all Hamming distances.

$d(000, 011) = 2$ $d(000, 101) = 2$ $d(000, 110) = 2$ $d(011, 101) = 2$
$d(011, 110) = 2$ $d(101, 110) = 2$

The d_{min} in this case is 2.

Example 10.6

Find the minimum Hamming distance of the coding scheme in Table 10.2.

Solution

We first find all the Hamming distances.

$d(00000, 01011) = 3$	$d(00000, 10101) = 3$	$d(00000, 11110) = 4$
$d(01011, 10101) = 4$	$d(01011, 11110) = 3$	$d(10101, 11110) = 3$

The d_{min} in this case is 3.

LINEAR BLOCK CODES

- ***Almost all block codes used today belong to a subset called linear block codes. A linear block code is a code in which the exclusive OR (addition modulo-2) of two valid codewords creates another valid codeword.***

Note

In a linear block code, the exclusive OR (XOR) of any two valid codewords creates another valid codeword.

Minimum Distance for Linear Block Codes

- ***It is simplke to find the minimum Hamming distance for a linear block code.***
- ***The minimum Hamming Distance is the number of 1s in the nonzero valid codeword with the smallest number of 1s.***

Table:

00 = 00
01 = 011
10 = 101
11 = 110

Here in nonzero valid codewords,the smallest number of 1s are 2. So d_{min}= 2

Parity-Check Code :

The most familiar error-detecting code is the simple parity-check code.

In this code, a k-bit dataword is changed to an n-bit codeword where n = k + 1. The extra bit, called the parity bit, is selected to make the total number of 1s in the codeword even.

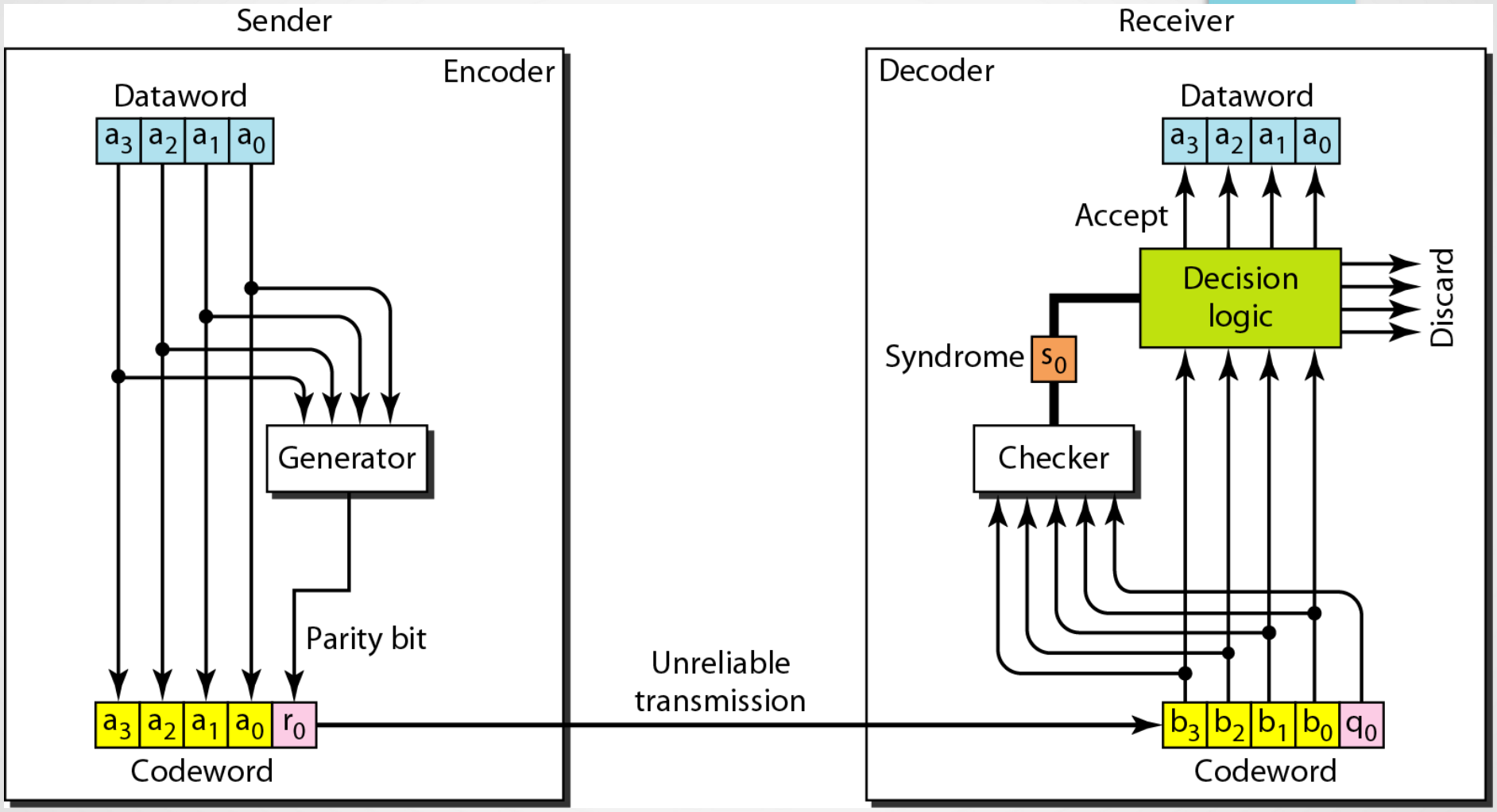
A simple parity-check code is a single-bit error-detecting code in which

$$n = k + 1 \text{ with } d_{min} = 2.$$

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*



Example

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.*

Example (continued)

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.

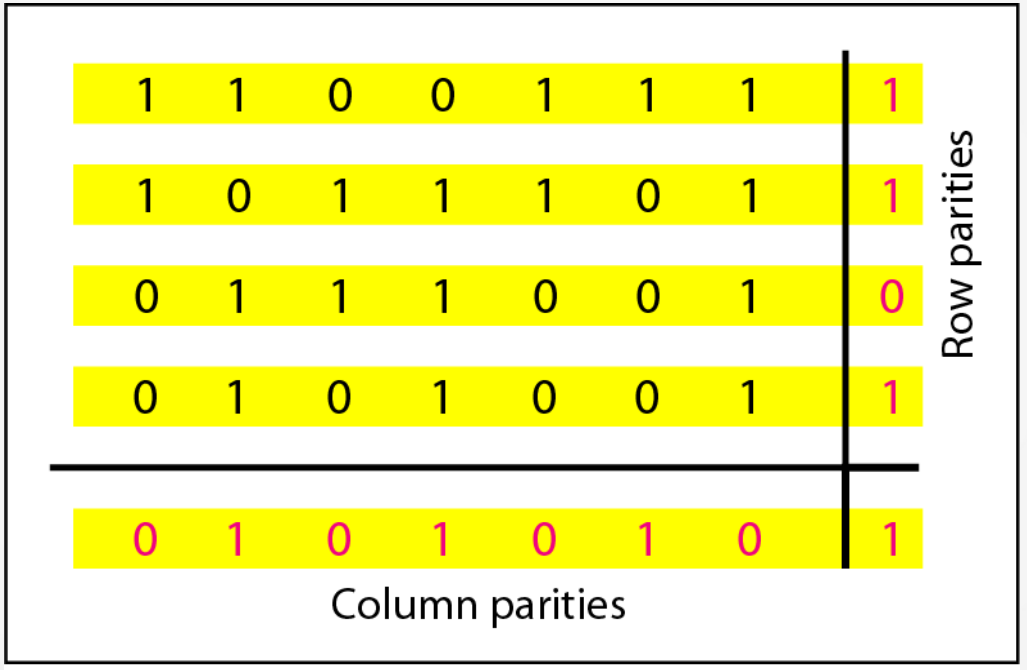
5. Three bits a_3, a_2, and a_1 are changed by errors. The received codeword is 01011. The syndrome is 1. The dataword is not created.

This shows that the simple parity check, guaranteed to detect one single error, can also find any odd number of errors.

Note

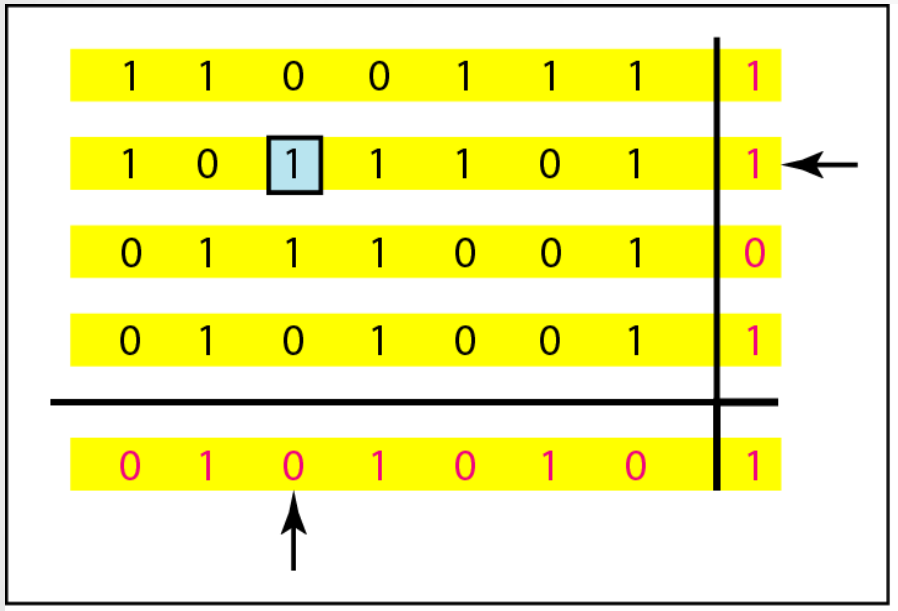
A simple parity-check code can detect an odd number of errors.

Two-dimensional parity-check code

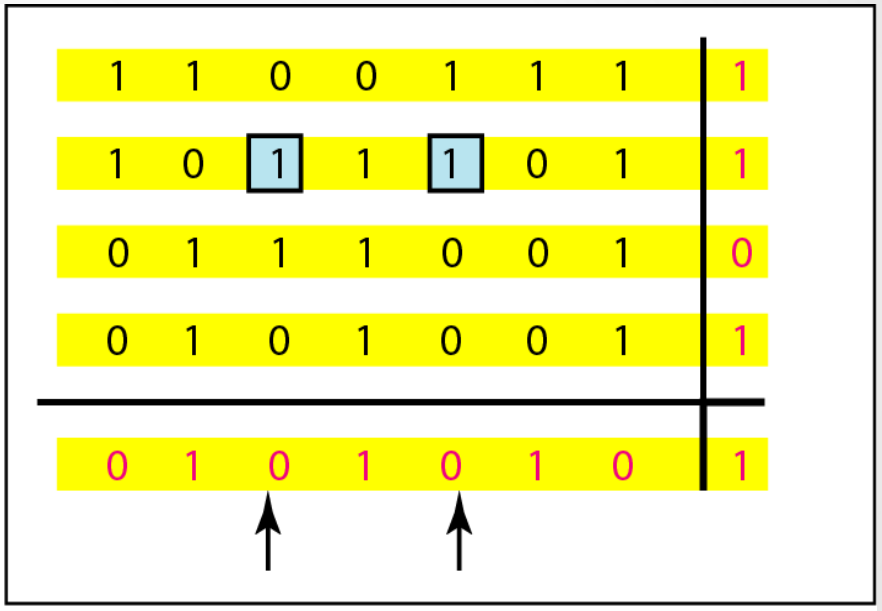


a. Design of row and column parities

Two-dimensional parity-check code

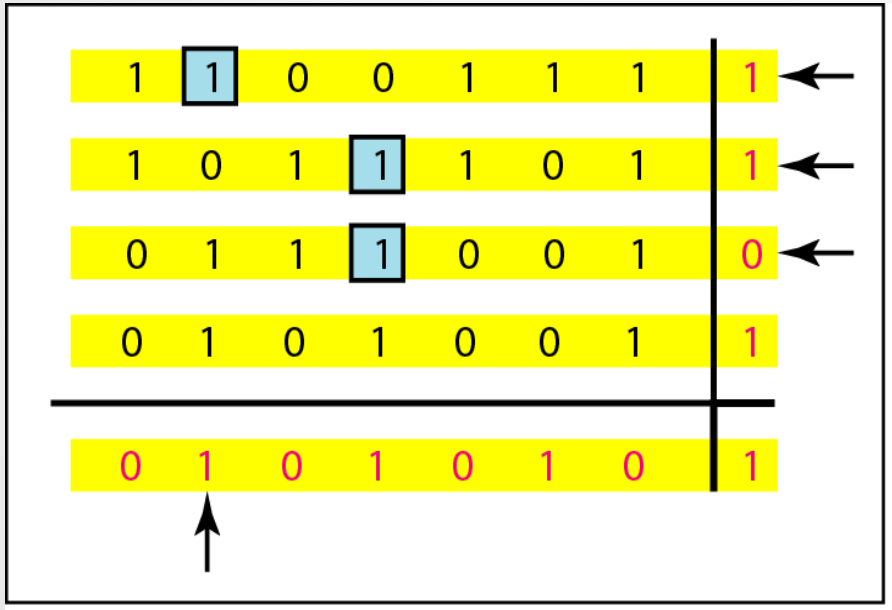


b. One error affects two parities

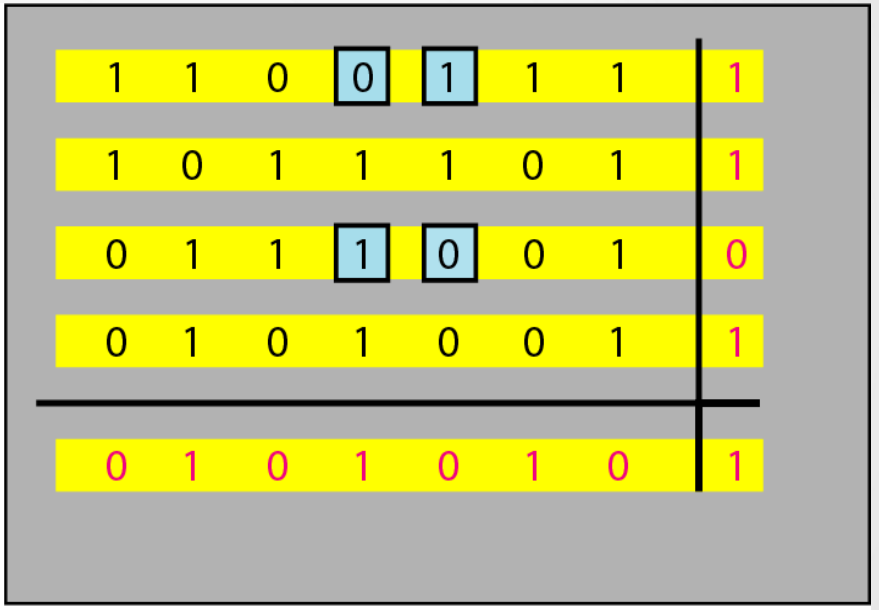


c. Two errors affect two parities

Two-dimensional parity-check code



d. Three errors affect four parities



e. Four errors cannot be detected

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.

For example:
if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword.
In this case, we call the bits in the first word ao to a6 and the bits in the second word bo to b6.

CYCLIC REDUNDANCY CHECK

In this section, we simply discuss a category of cyclic codes called the cyclic redundancy check (CRC) that is used in networks such as LANs and WANs.

Table： *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

CRC encoder and decoder

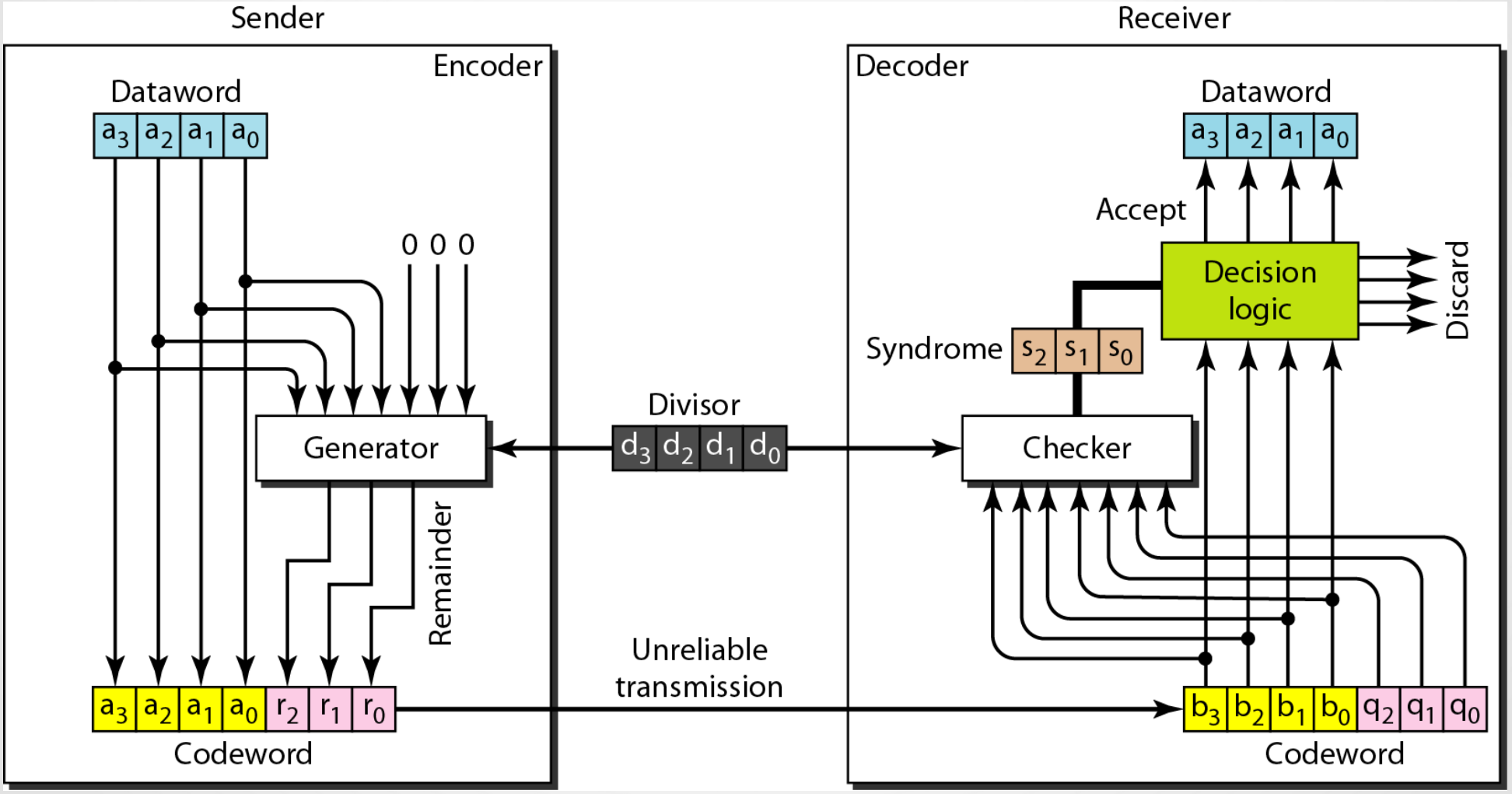


Figure 10.15 *Division in CRC encoder*

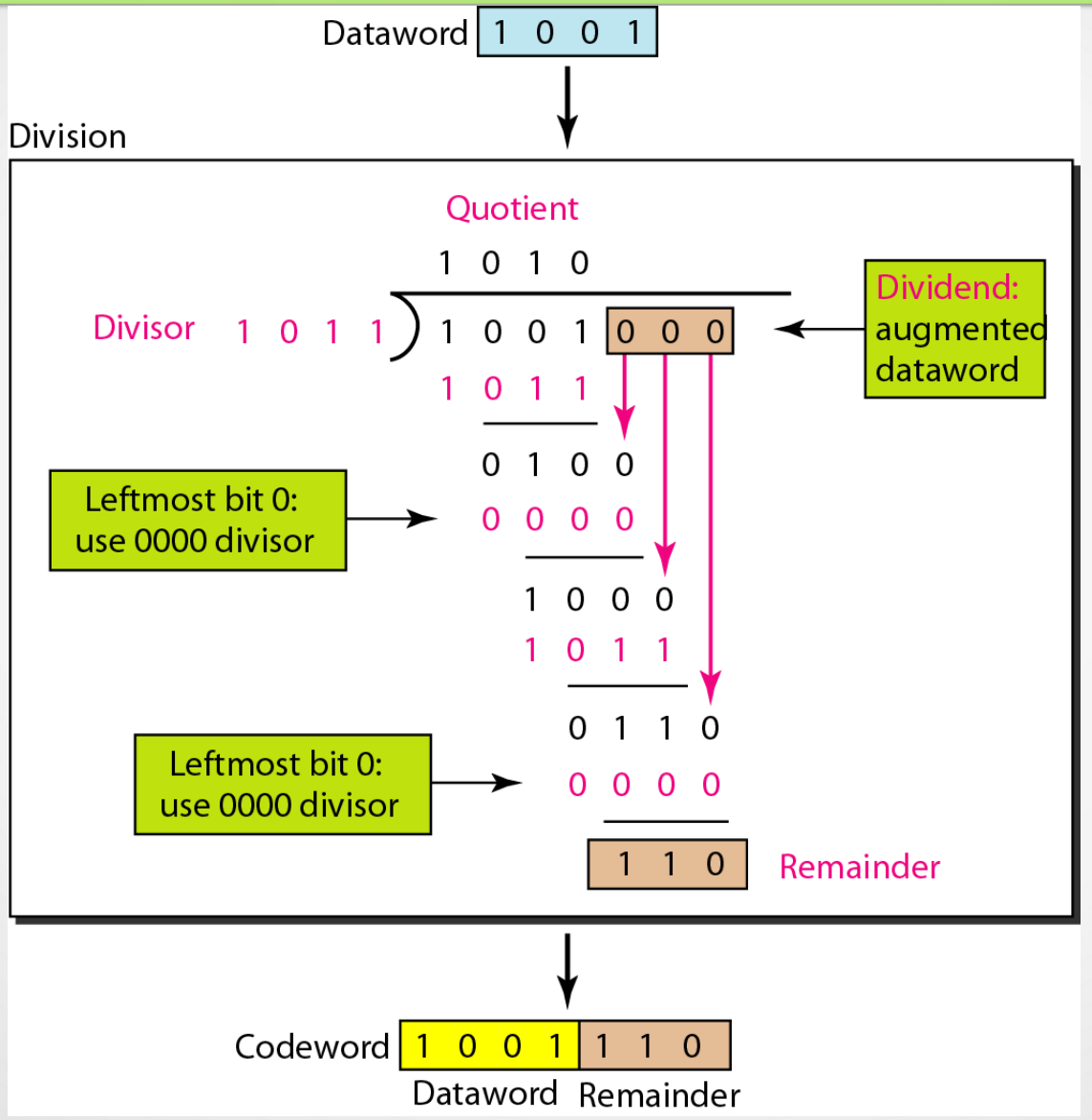
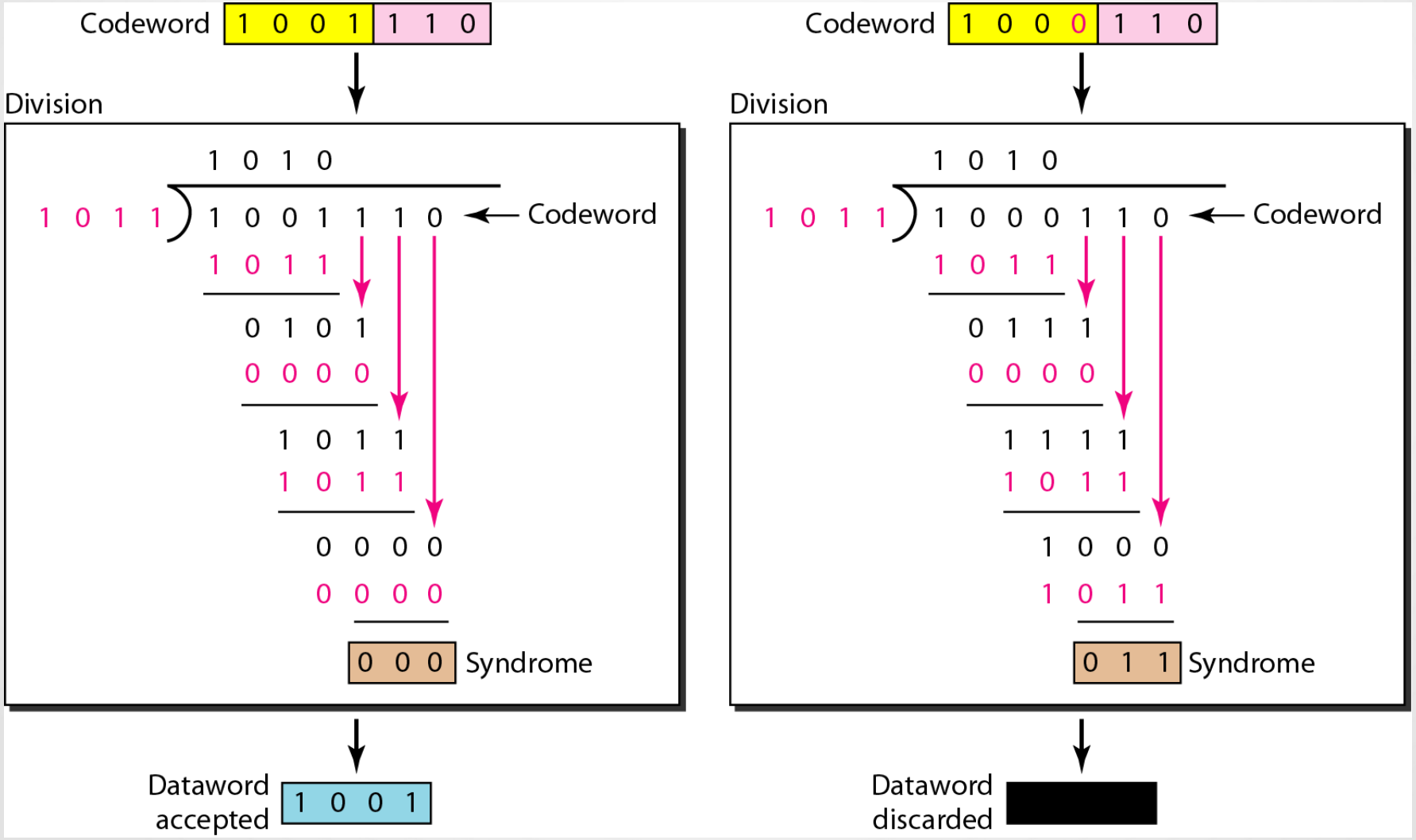


Figure 10.16 *Division in the CRC decoder for two cases*

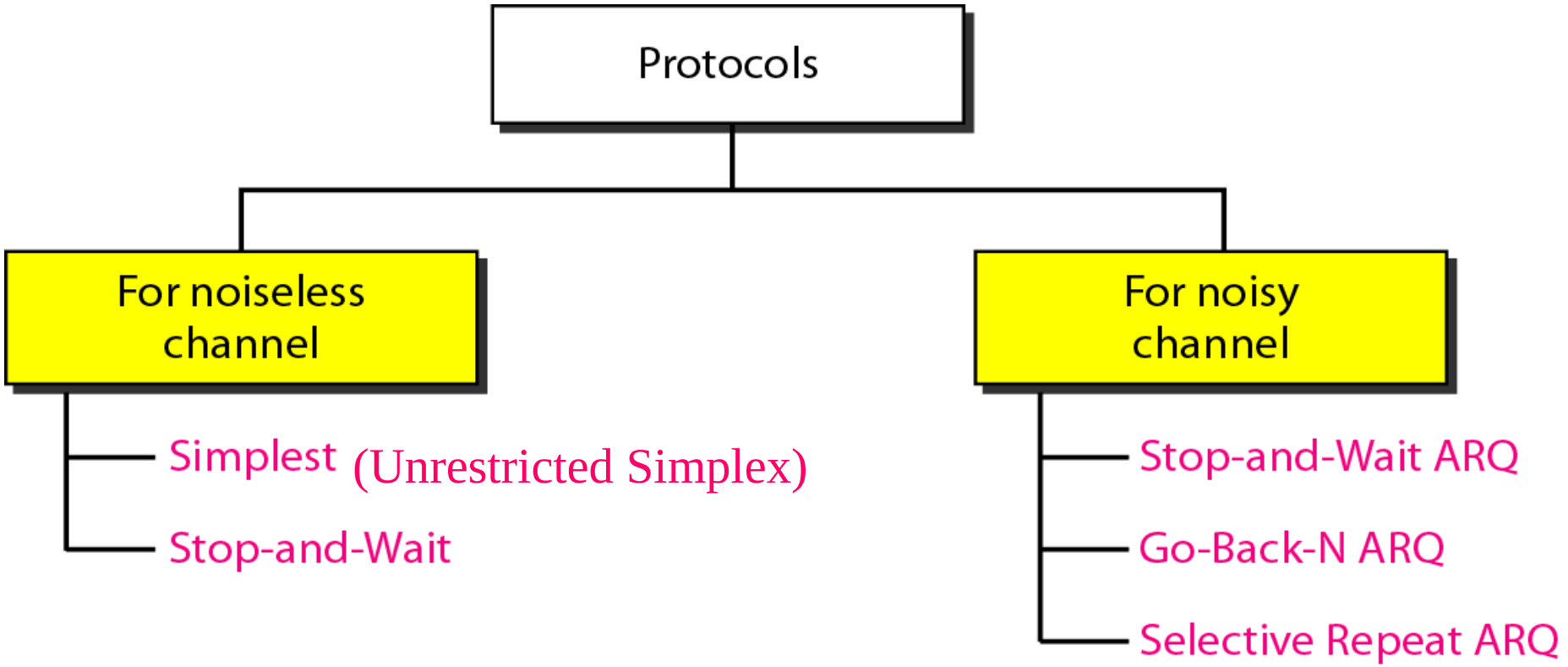


• PROTOCOLS

- *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.*

- Figure: Flow Control Protocols



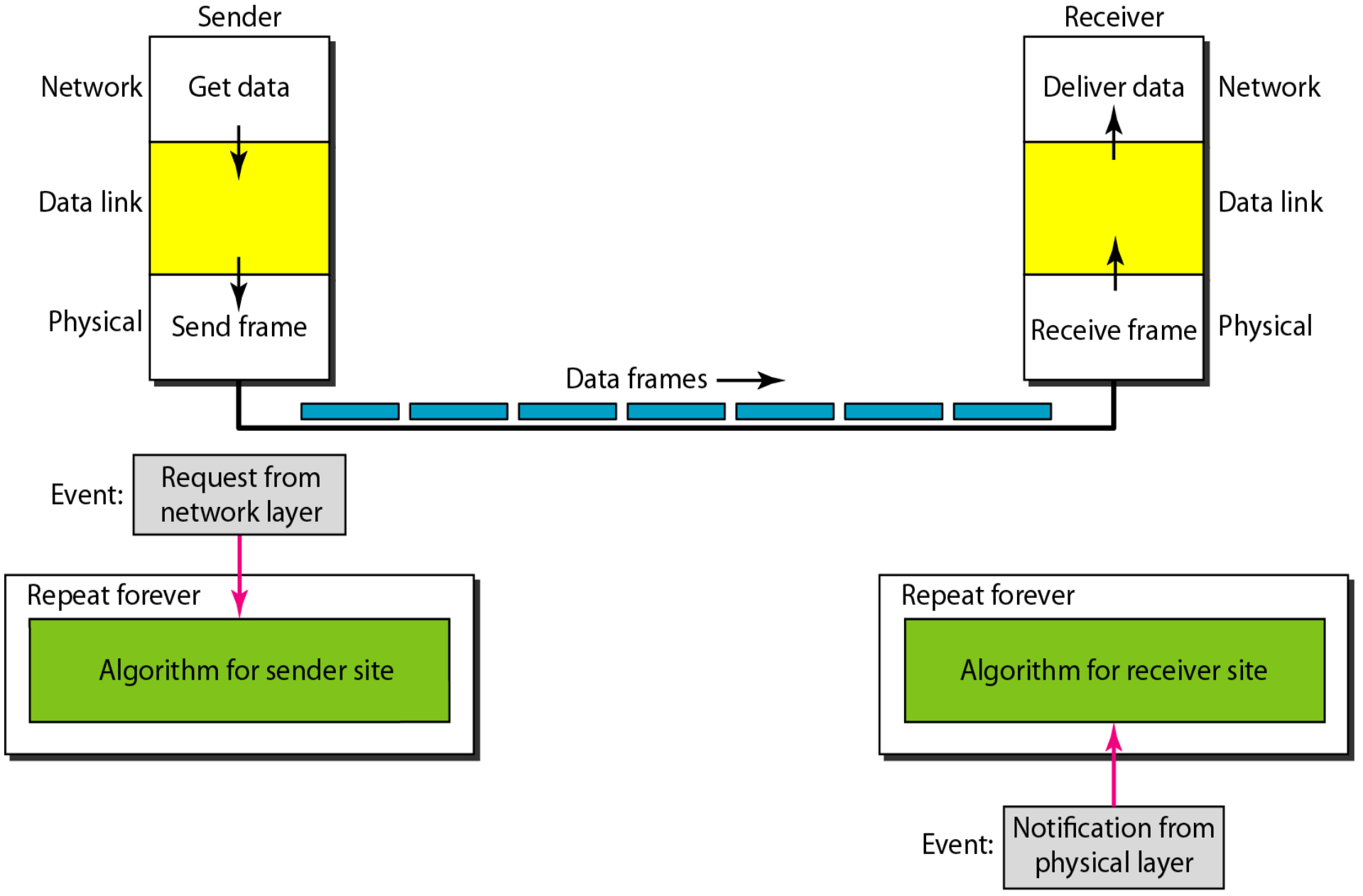
NOISELESS CHANNELS

- *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.*

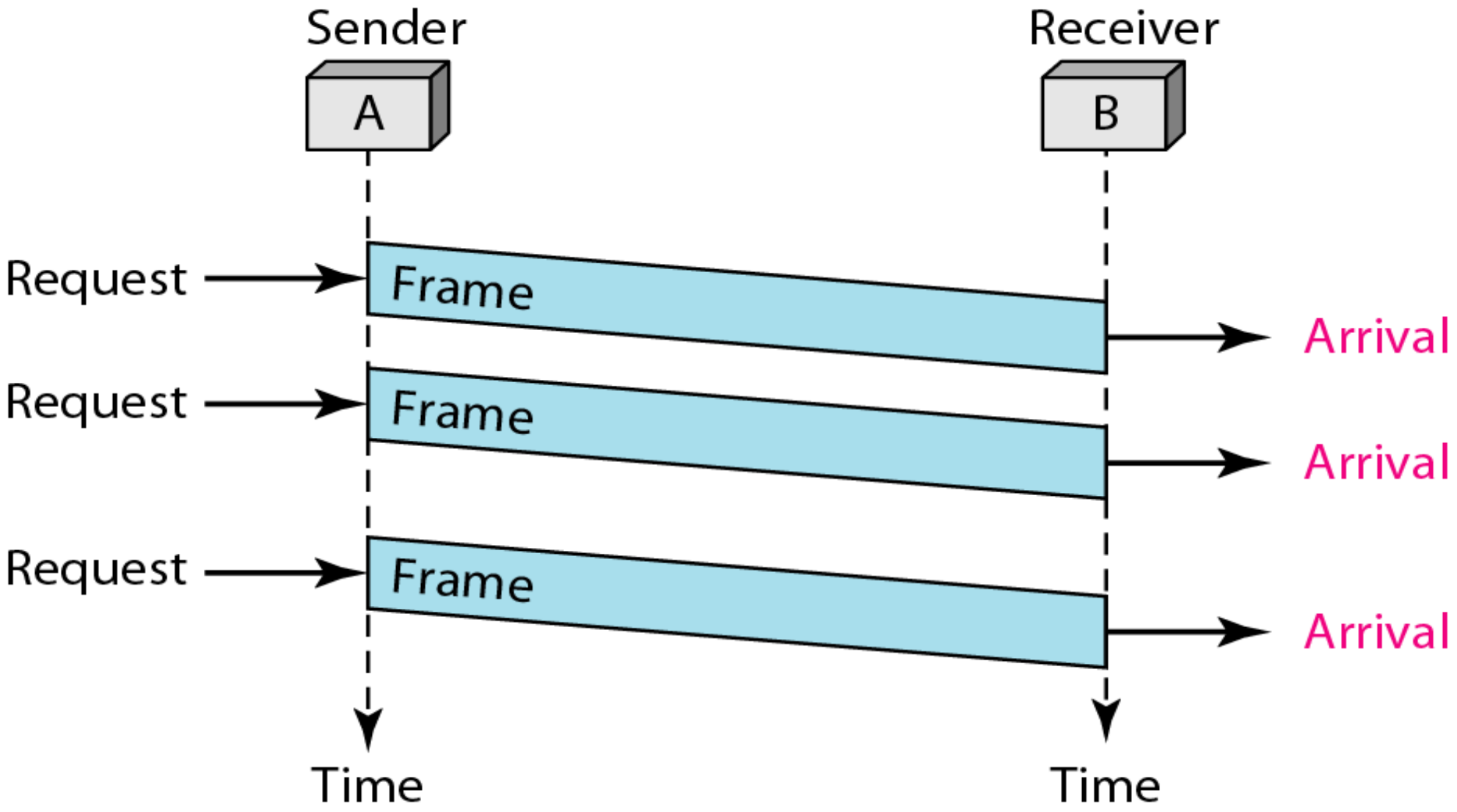
Simplest Protocol

- Stop-and-Wait Protocol

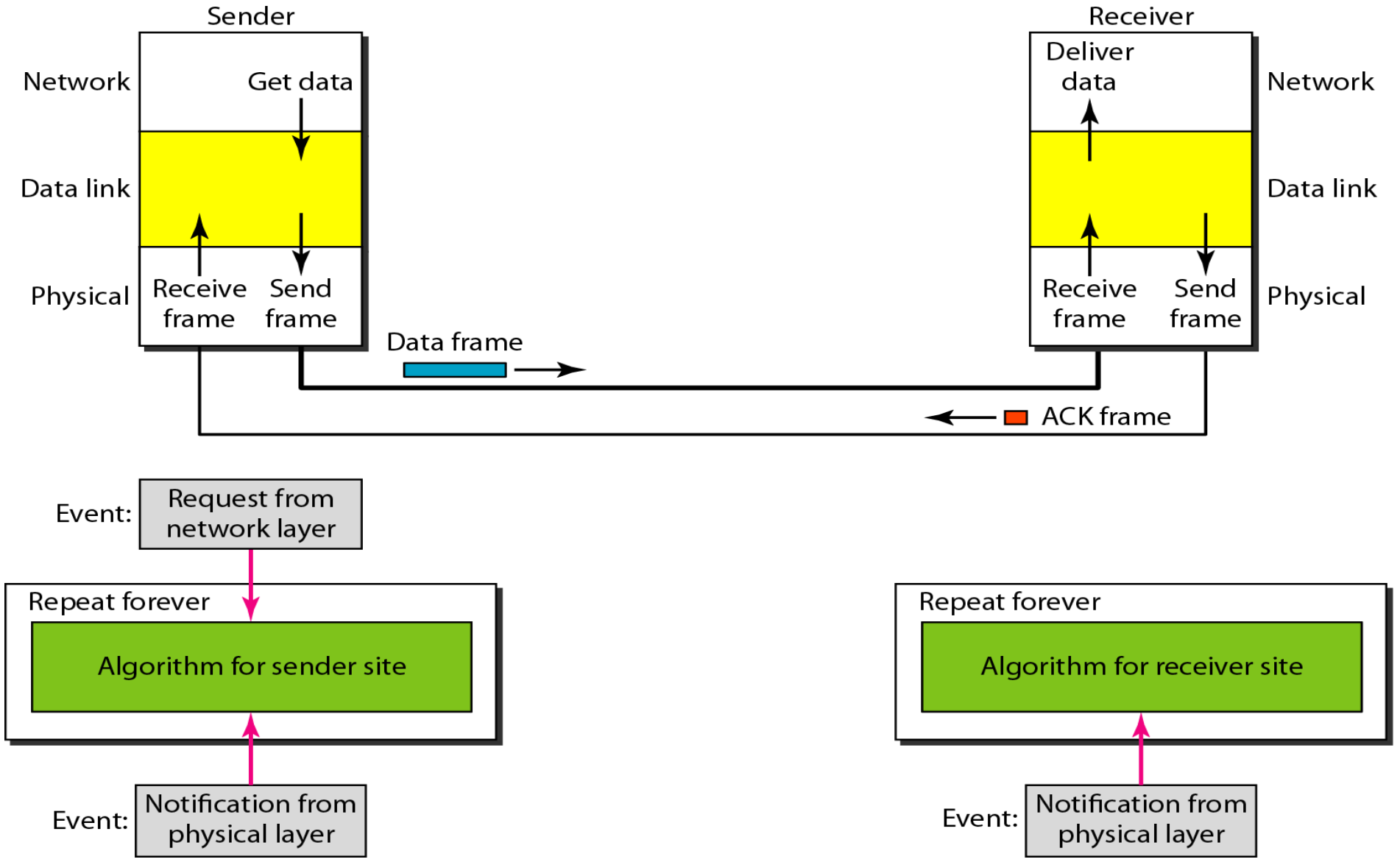
- Figure *The design of the simplest protocol with no flow or error control*



- Figure *Flow diagram*



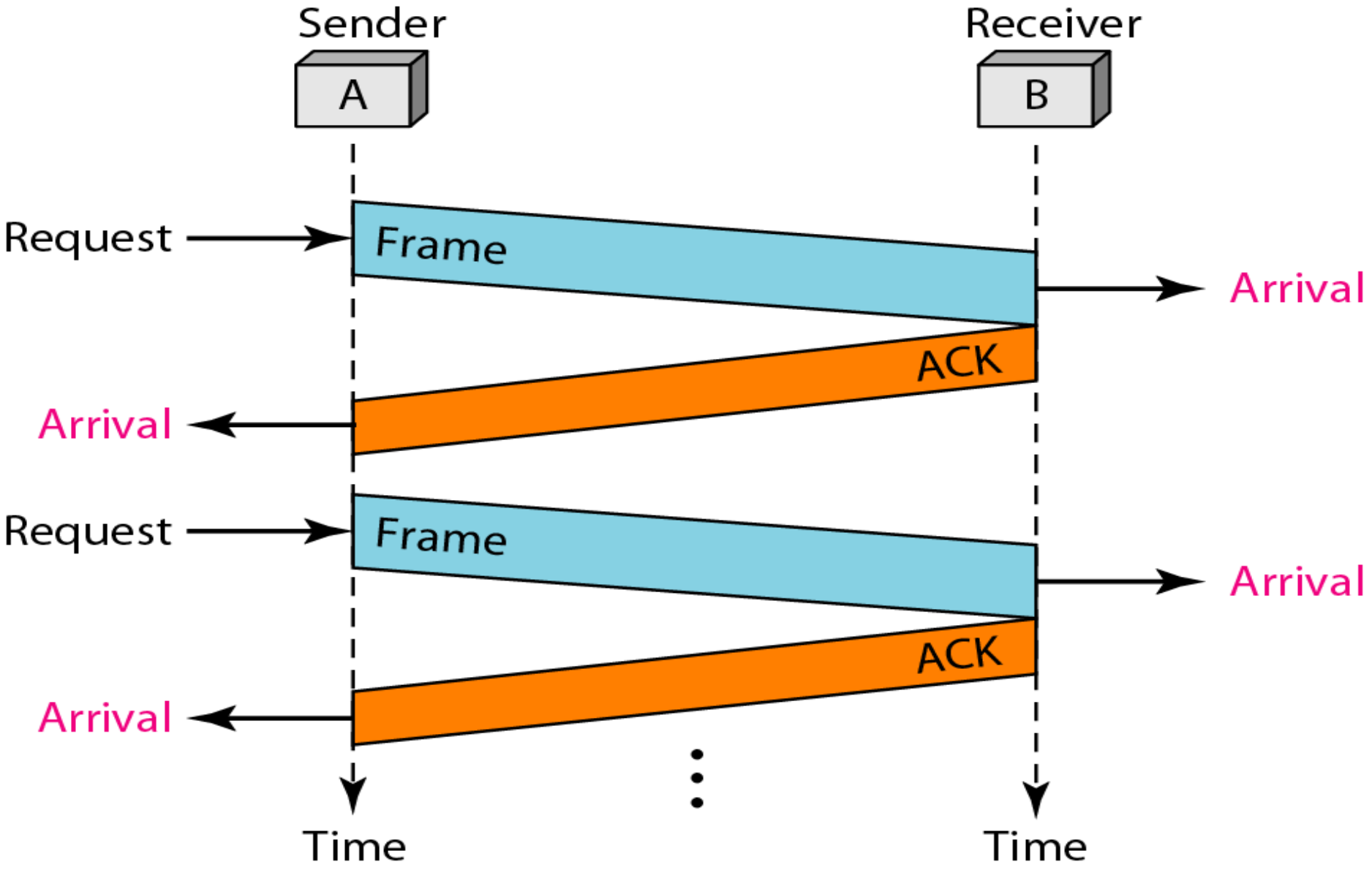
- Figure *Design of Stop-and-Wait Protocol*



• *Example*

- *Next figure 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.*

- Figure- *Flow diagram*



NOISY CHANNELS

- *Although the Stop-and-Wait Protocol gives us an idea of how to add flow control to its predecessor,noiseless channels are nonexistent.*

- *Three protocols in this section that use error control*
- *are as follows.*

- Stop-and-Wait Automatic Repeat Request
- Go-Back-N Automatic Repeat Request
- Selective Repeat Automatic Repeat Request

1. Stop-and-Wait Automatic Repeat Request

- *Note*

- Error correction in Stop-and-Wait ARQ is done by keeping a copy of the sent frame and retransmitting of the frame when the timer expires.

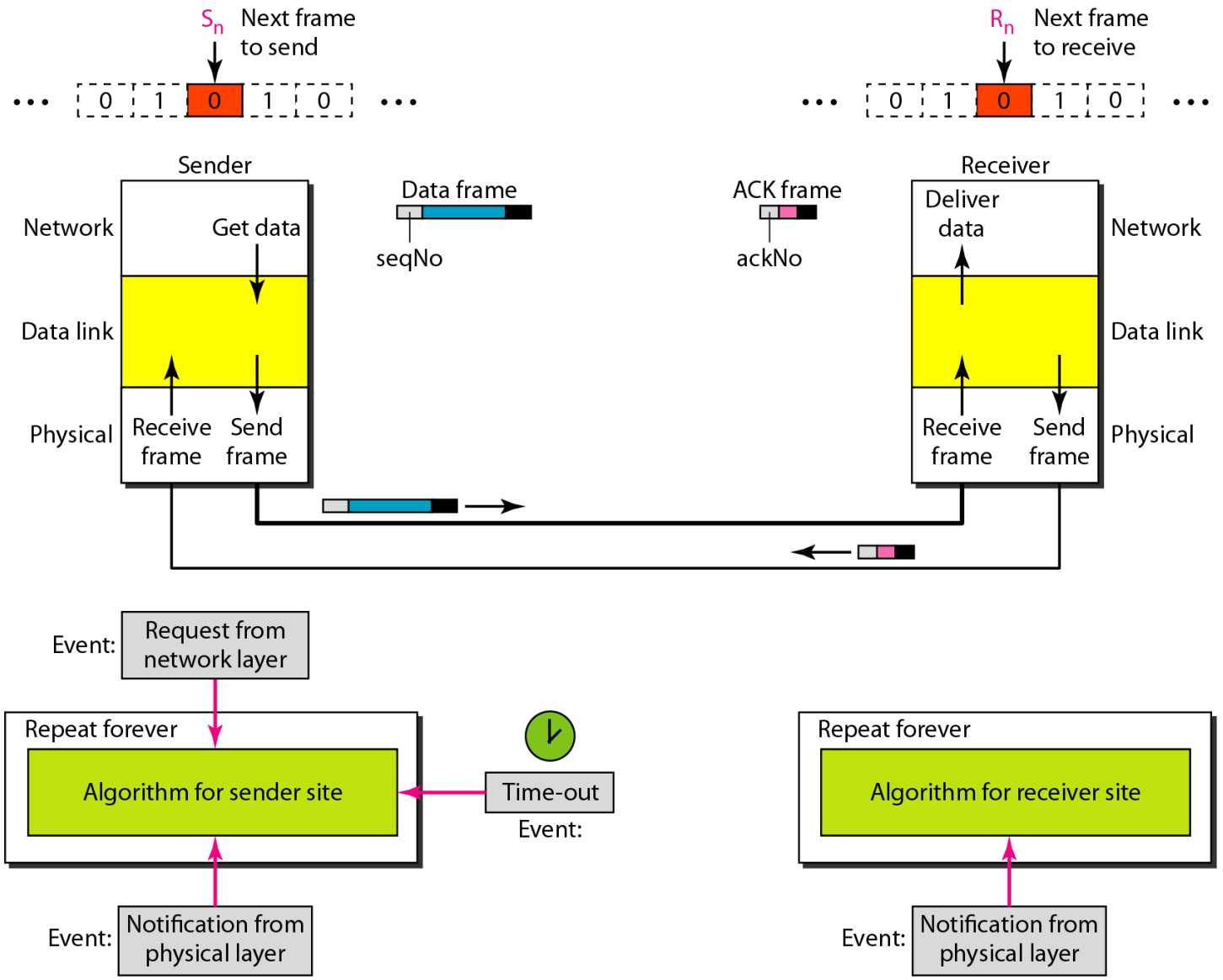
- *Note*

- In Stop-and-Wait ARQ, we use sequence numbers to number the frames.
  - The sequence numbers are based on modulo-2 arithmetic.

- *Note*

In Stop-and-Wait ARQ, the acknowledgment number always announces in modulo-2 arithmetic the sequence number of the next frame expected.

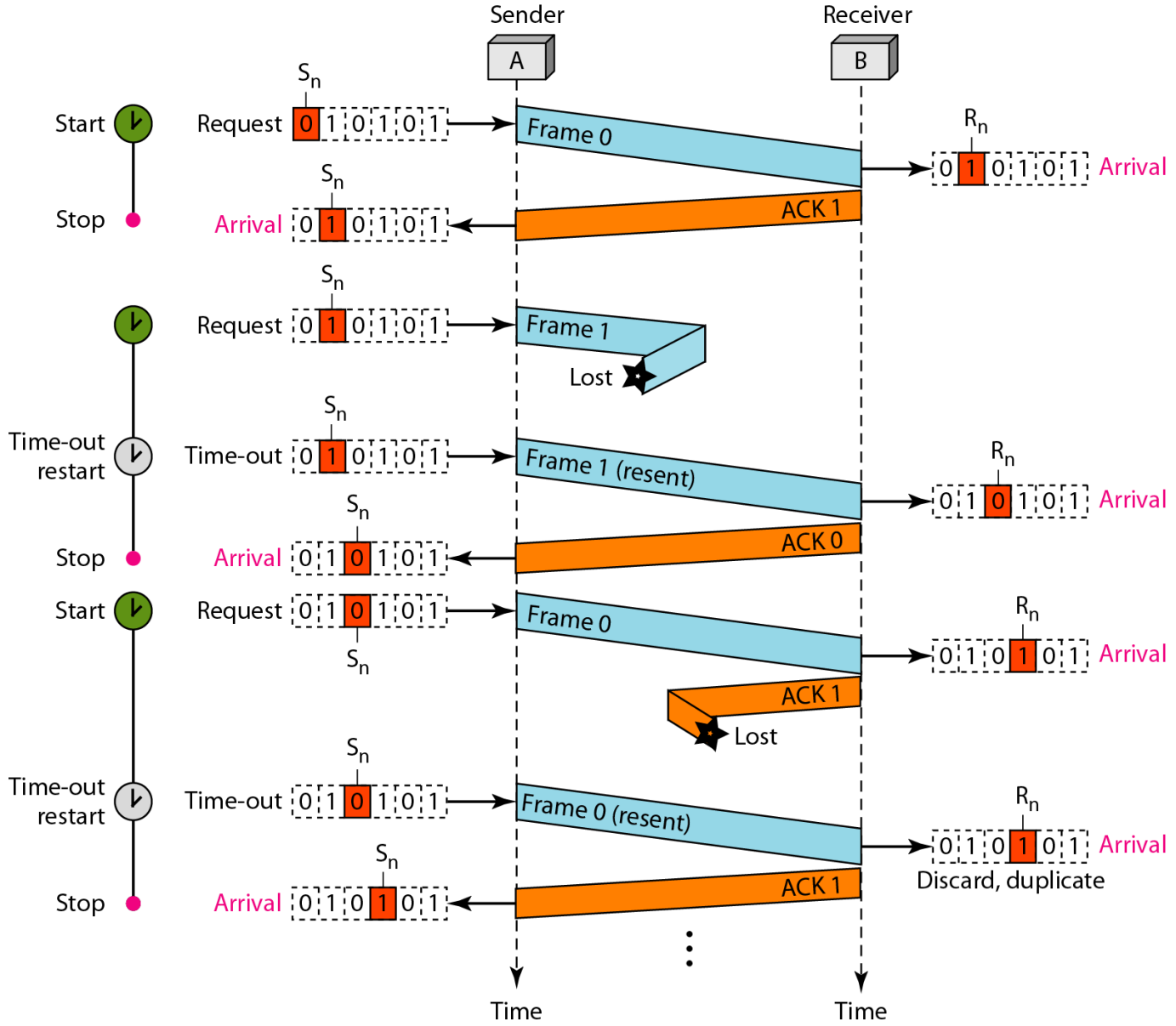
Figure : *Design of the Stop-and-Wait ARQ Protocol*



Example

Nest figure shows an example of Stop-and-Wait ARQ. Frame 0 is sent and acknowledged. Frame 1 is lost and resent after the time-out. The resent frame 1 is acknowledged and the timer stops. Frame 0 is sent and acknowledged, but the acknowledgment is lost. The sender has no idea if the frame or the acknowledgment is lost, so after the time-out, it resends frame 0, which is acknowledged.

Flow diagram Stop-and-Wait Automatic Repeat Request



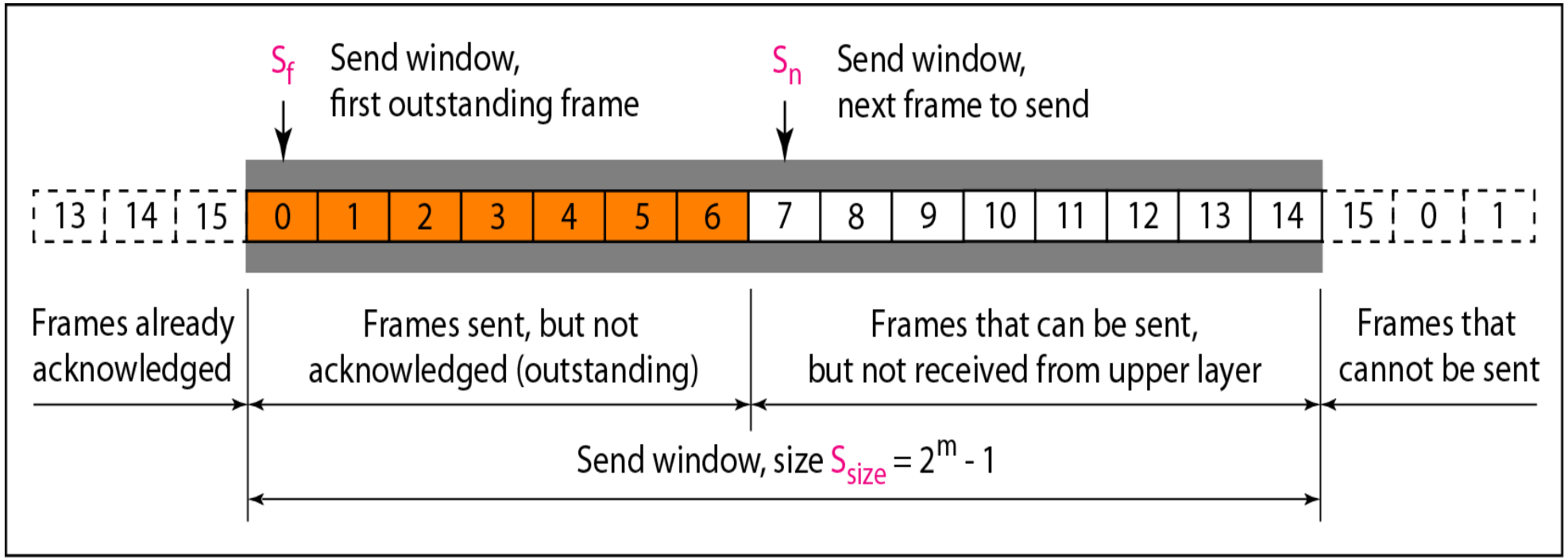
2. Go-Back-N Automatic Repeat Request

- *Note*

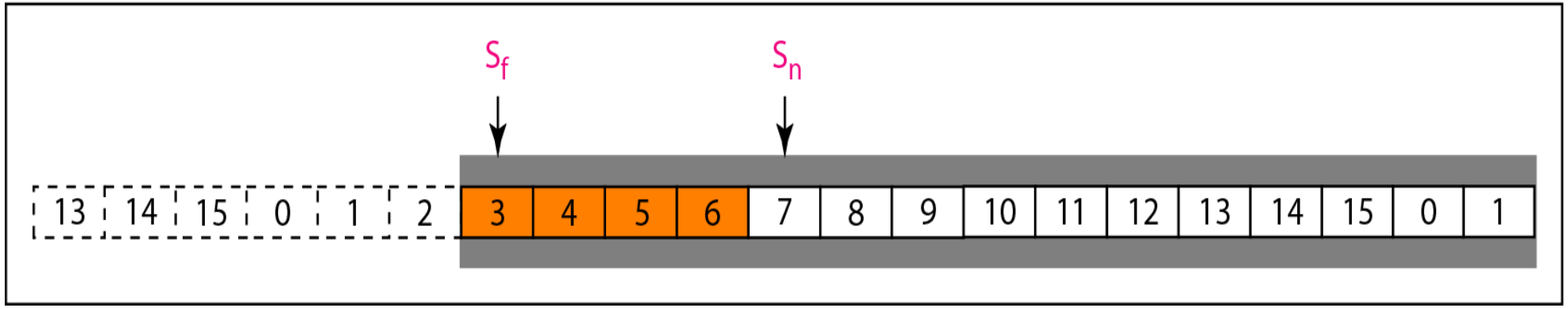
In the Go-Back-N Protocol, the sequence numbers are modulo 2^m,

- where m is the size of the sequence number field in bits.

- Figure *Send window for Go-Back-N ARQ*



a. Send window before sliding



b. Send window after sliding

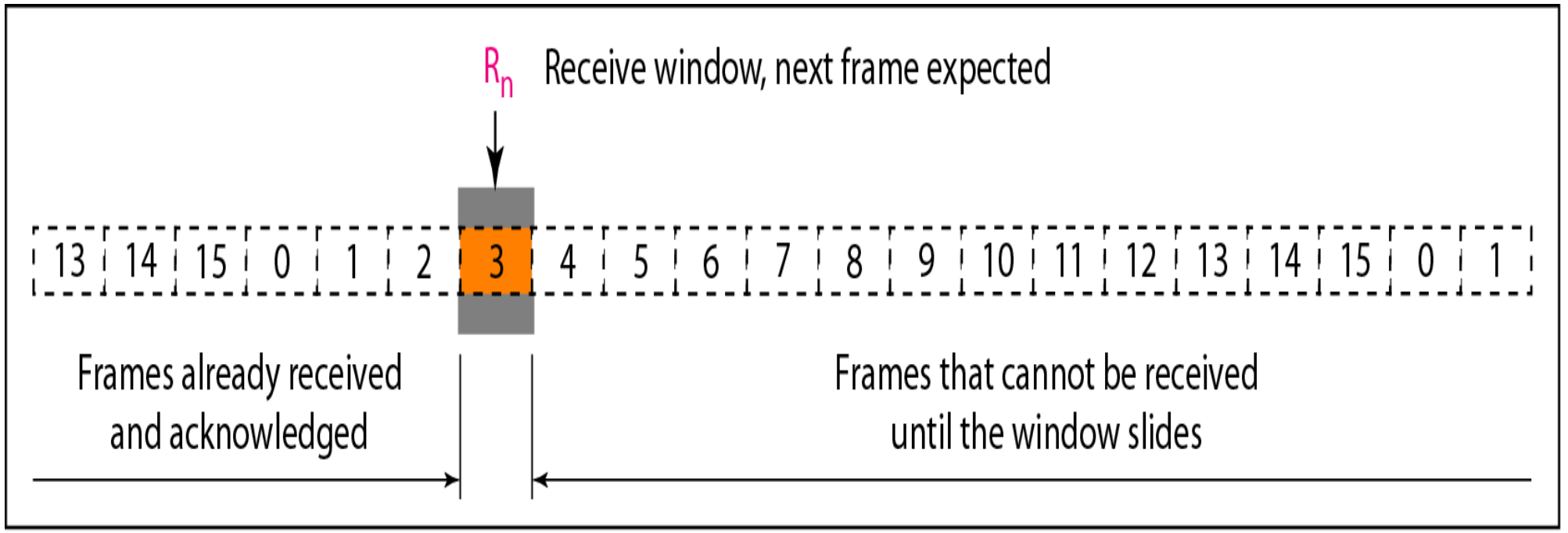
- *Note*

- The send window is an abstract concept defining an imaginary box of size $2^m - 1$ with three variables: Sf, S_n, and S_{size}.

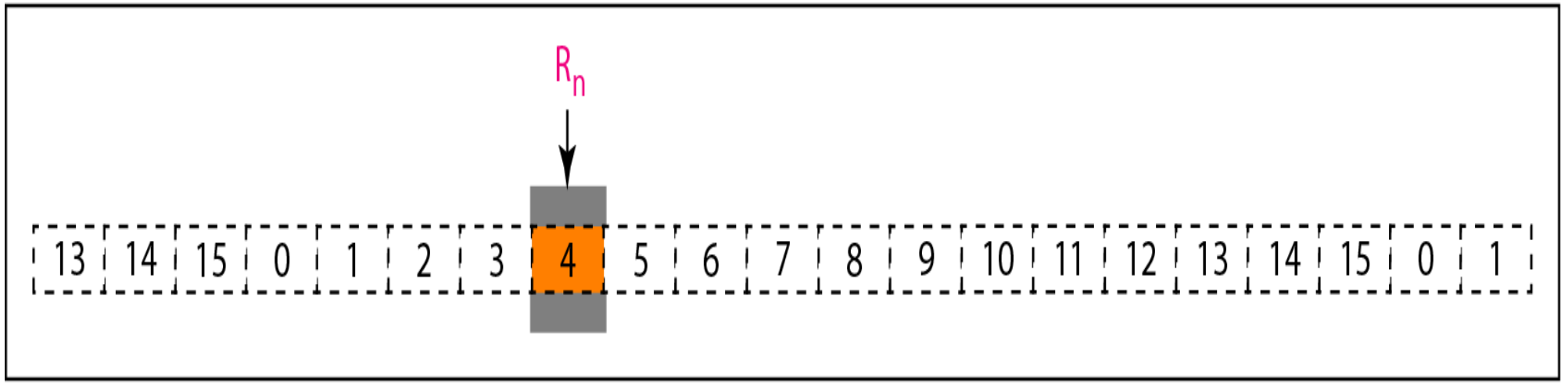
- *Note*

The send window can slide one or more slots when a valid acknowledgment arrives.

- Figure *Receive window for Go-Back-N ARQ*



a. Receive window

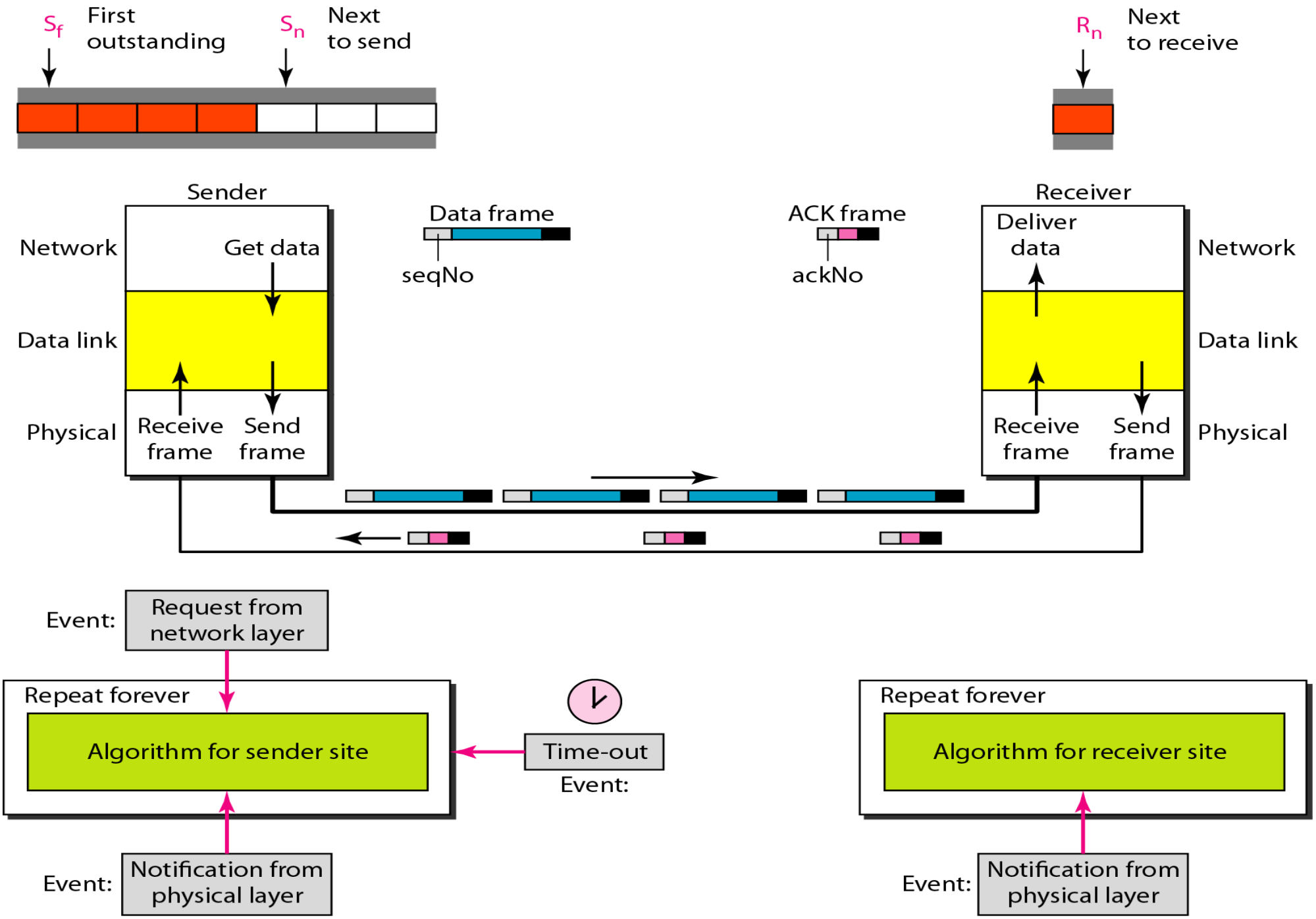


b. Window after sliding

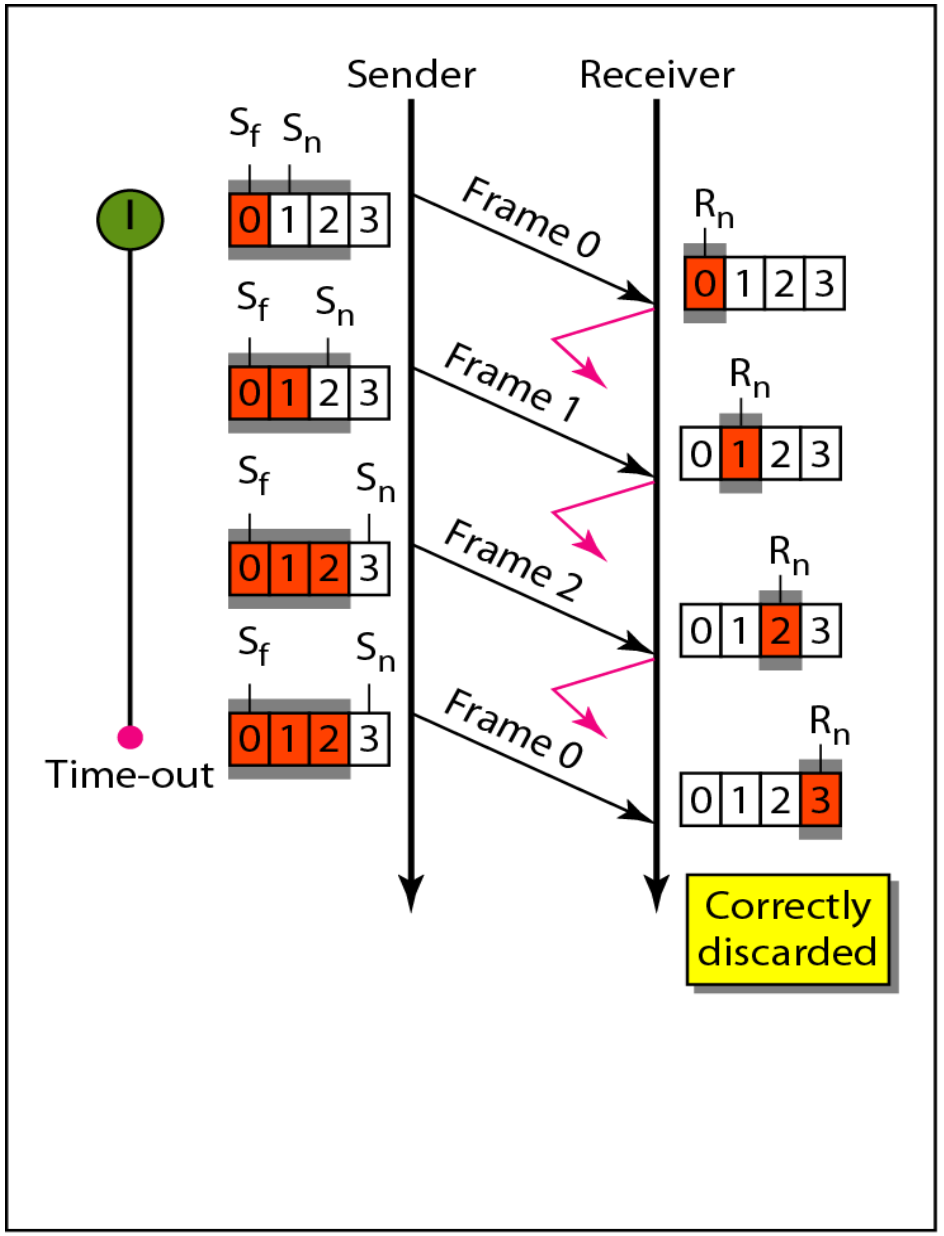
• *Note*

- The receive window is an abstract concept defining an imaginary box
  - of size 1 with one single variable R_n.
- The window slides
  - when a correct frame has arrived; sliding occurs one slot at a time.

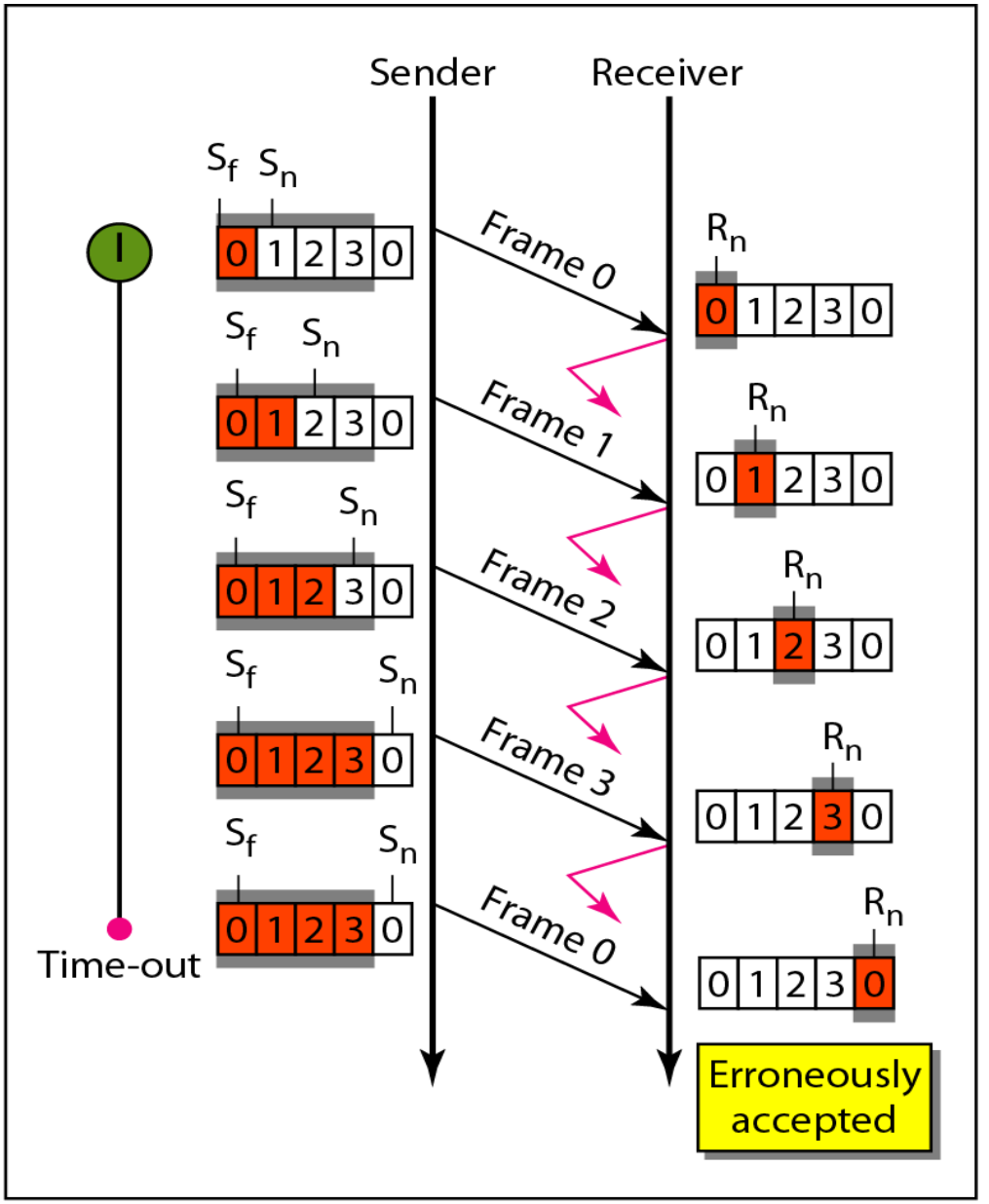
- Figure *Design of Go-Back-N ARQ*



- Figure *Window size for Go-Back-N ARQ*



a. Window size < 2^m

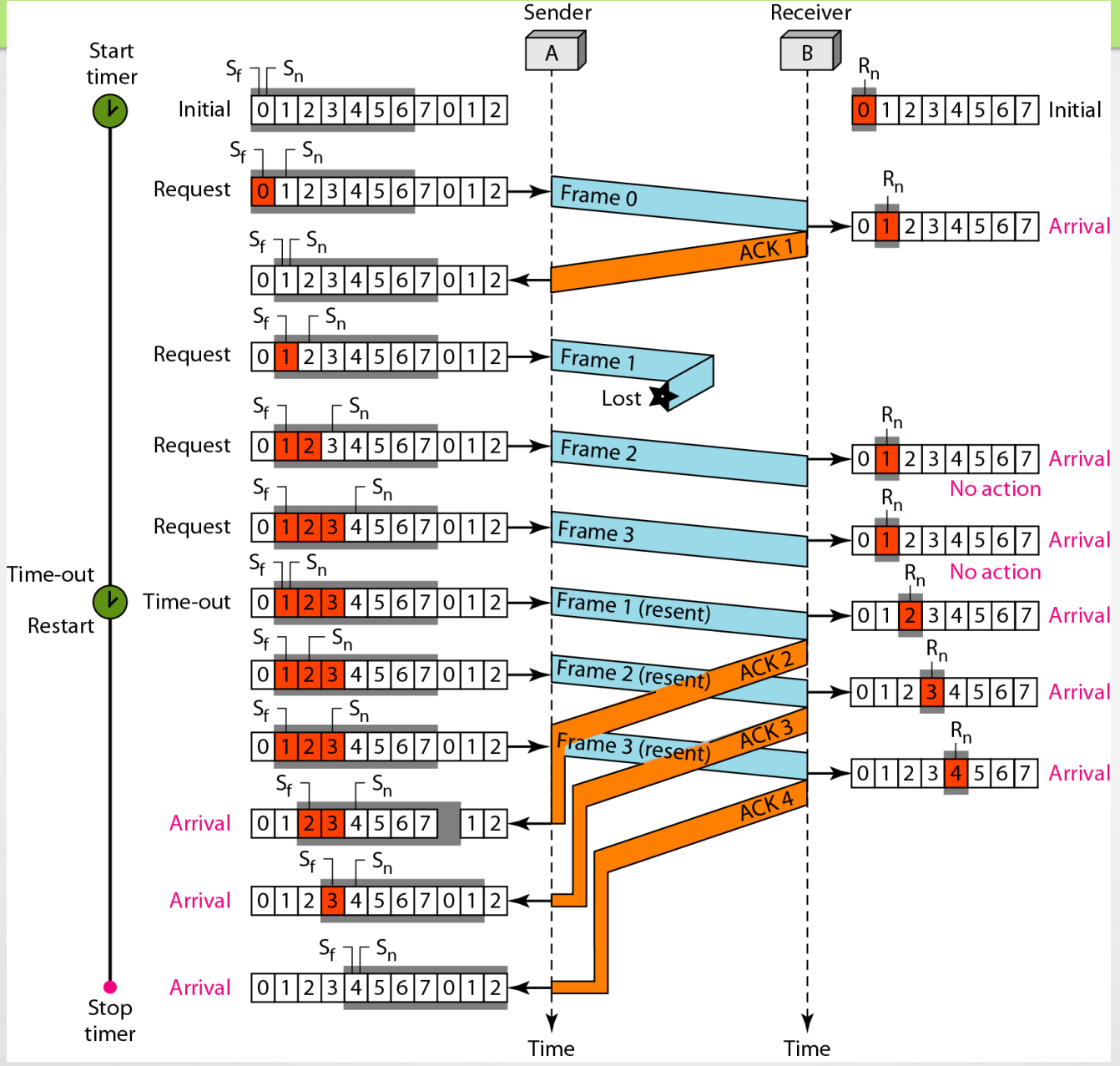


b. Window size = 2^m

- *Note*

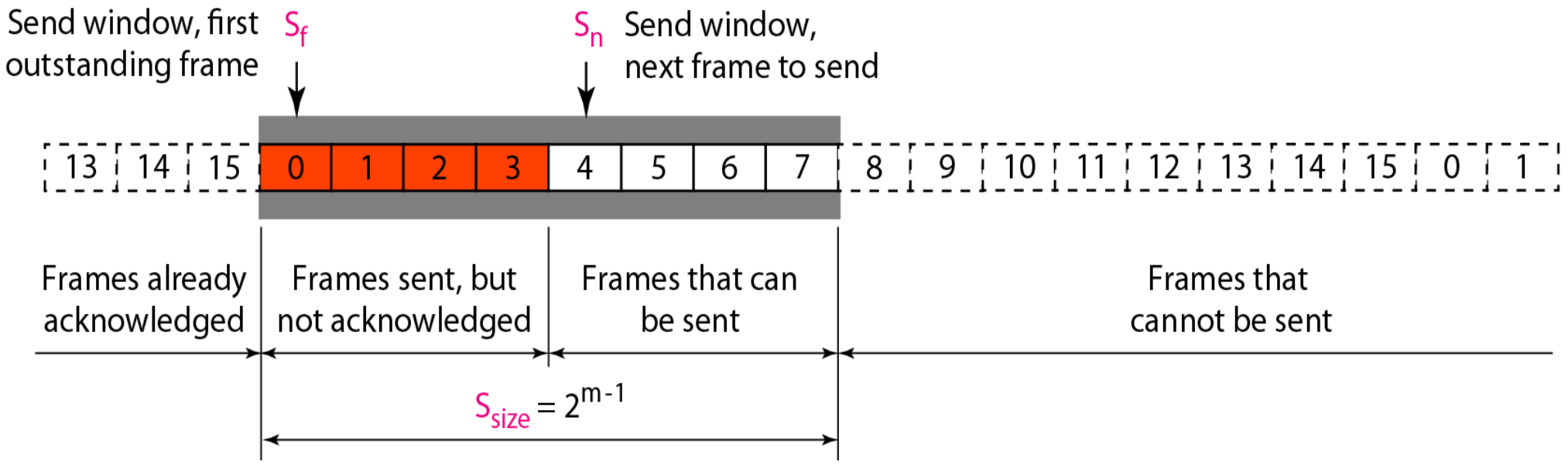
- In Go-Back-N ARQ, the size of the send window must be less than 2^m;
  - the size of the receiver window
    - is always 1.

Flow diagram forGo-Back-N

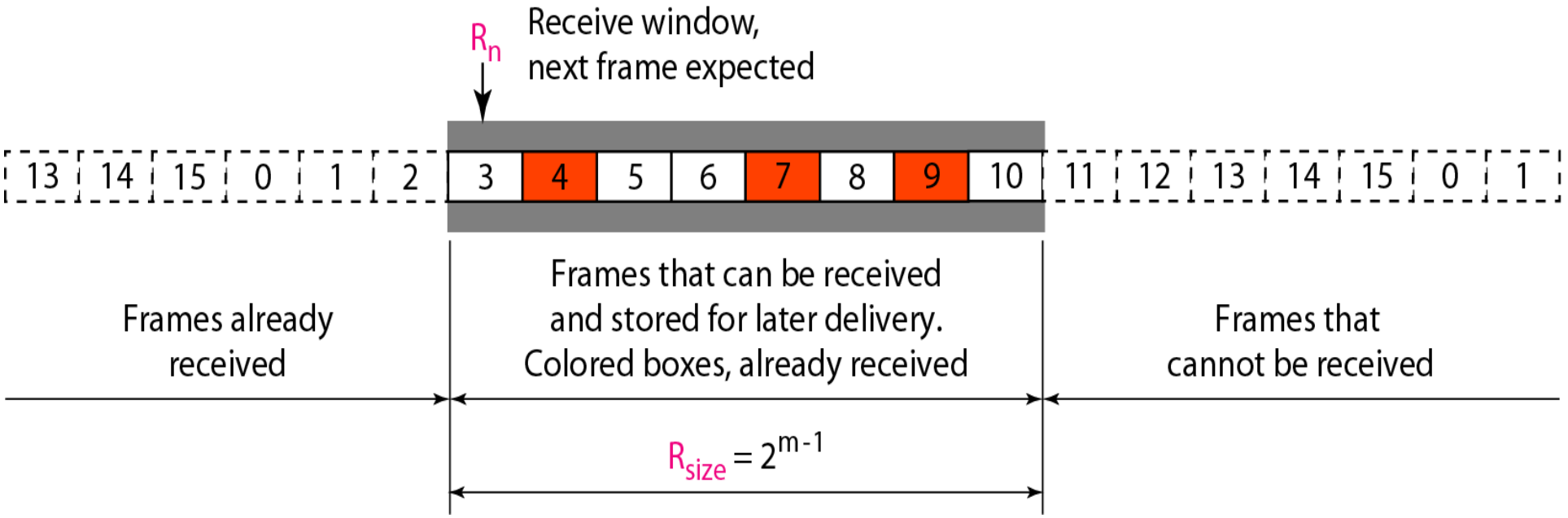


3. Selective Repeat Automatic Repeat Request

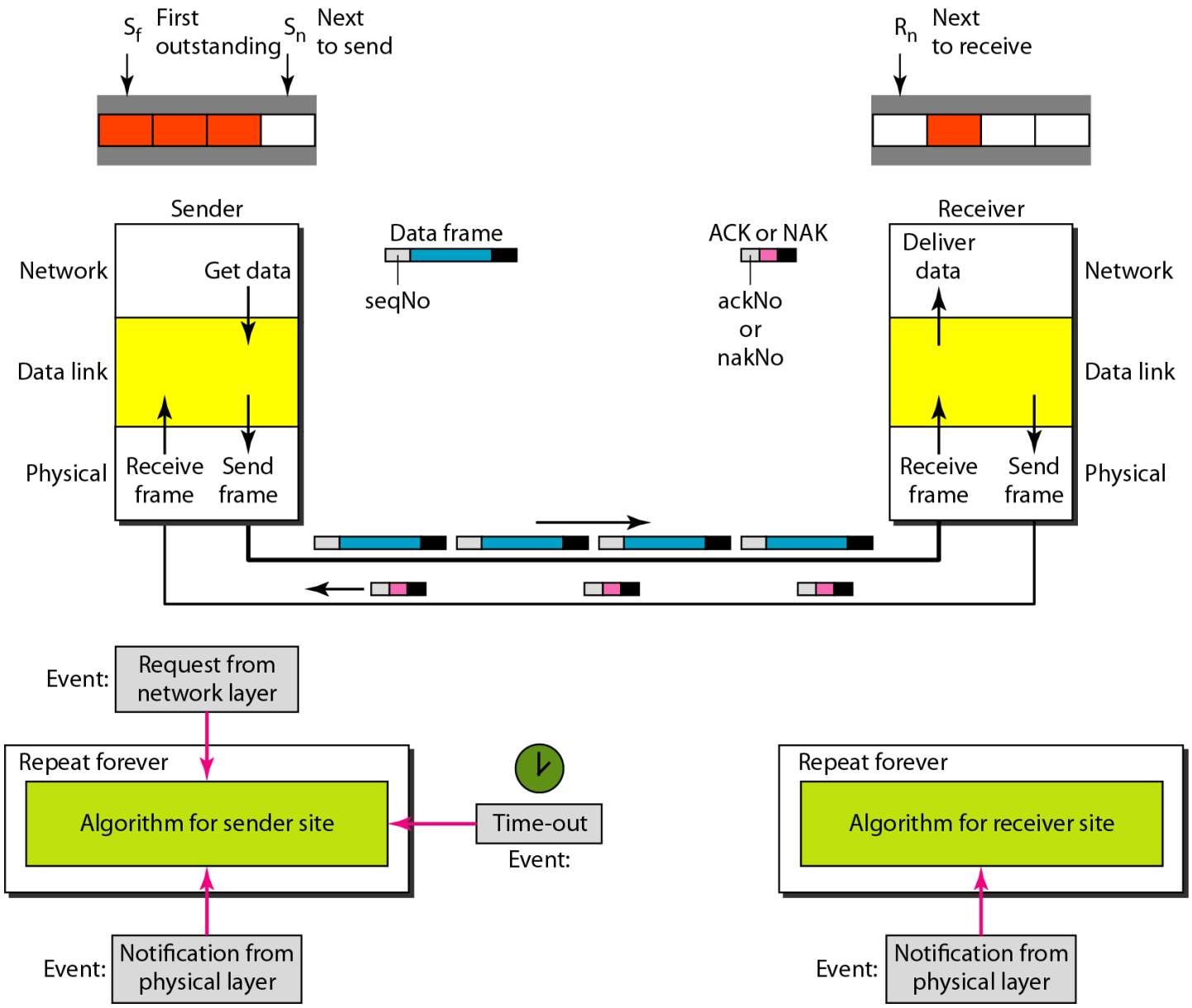
- ***Send window for Selective Repeat ARQ***



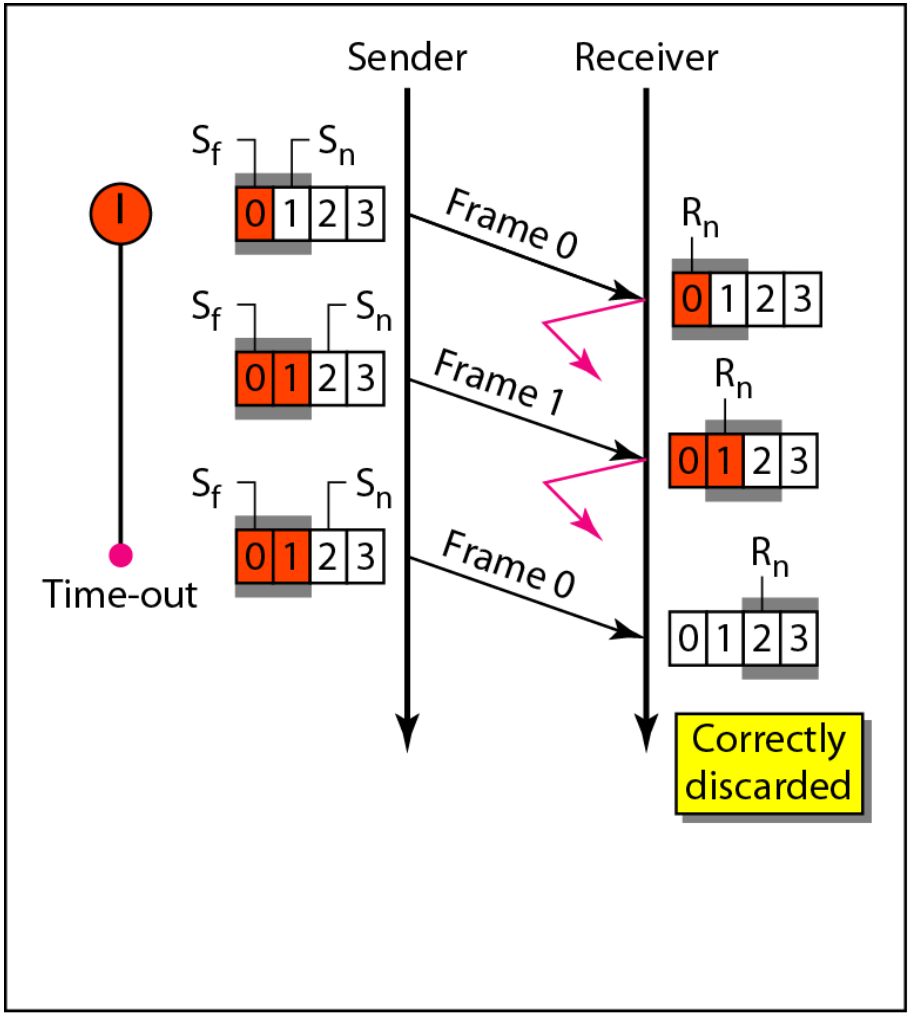
- ***Receive window for Selective Repeat ARQ***



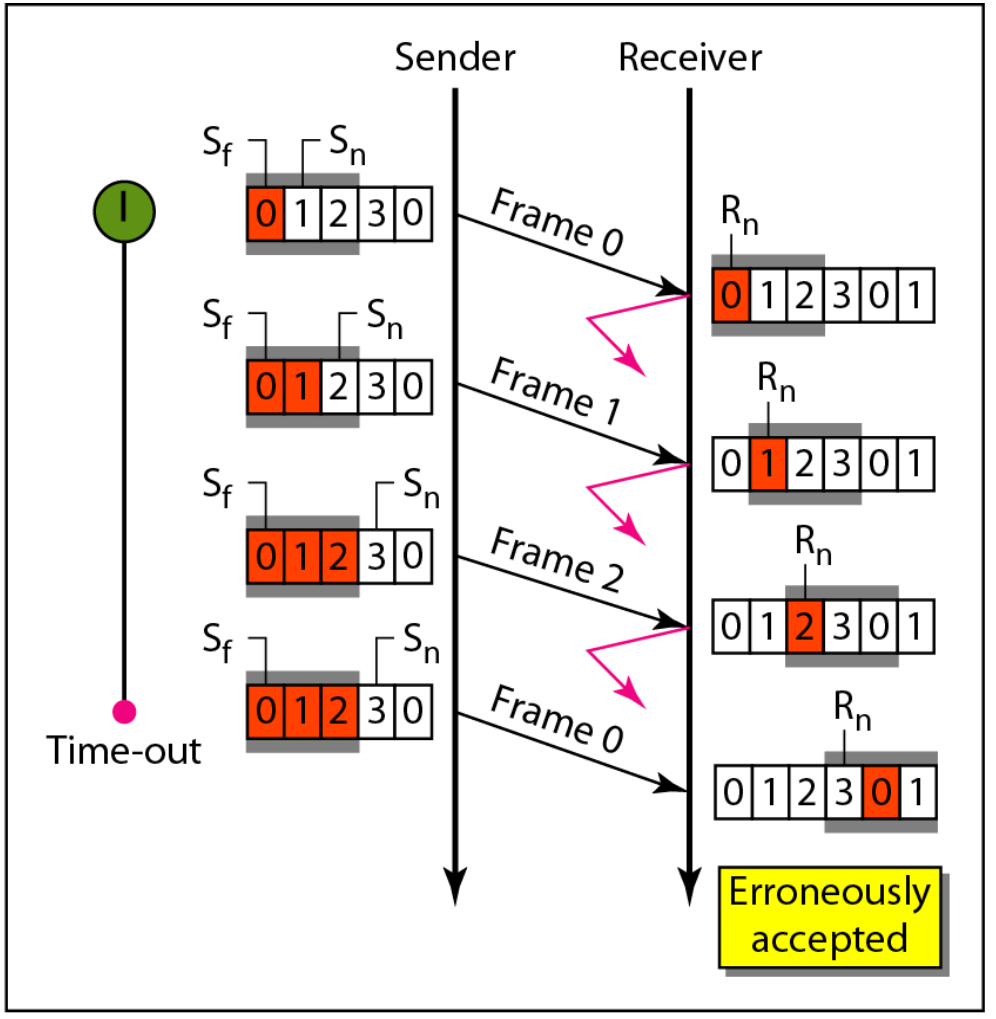
Design of Selective Repeat ARQ



Selective Repeat ARQ, window size



a. Window size = 2^{m-1}

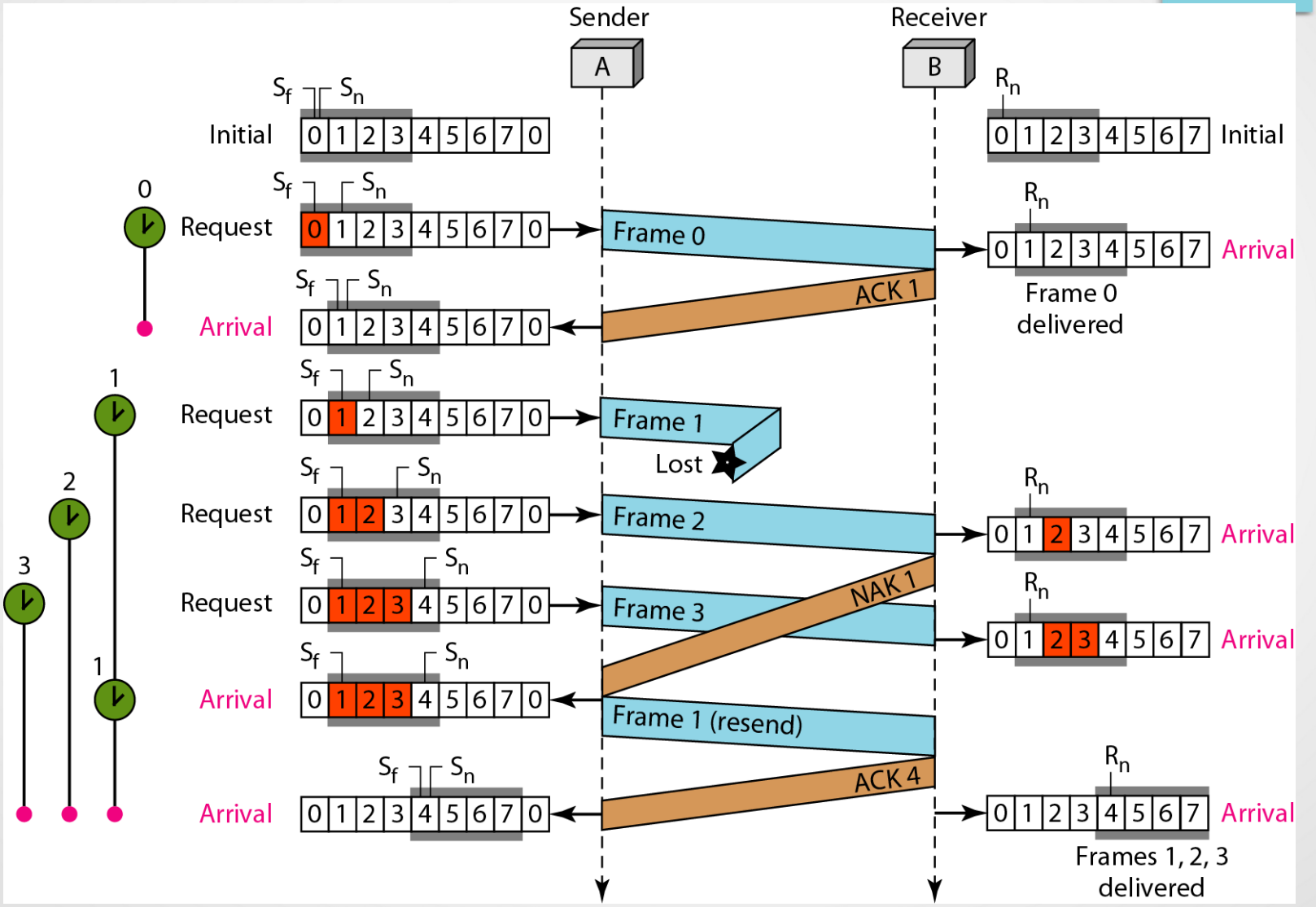


b. Window size > 2^{m-1}

Note

In Selective Repeat ARQ, the size of the sender and receiver window must be at most one-half of 2^m.

Flow diagram for Selective Repeat



HDLC

High-level Data Link Control (HDLC) is a bit-oriented protocol for communication over point-to-point and multipoint links. It implements the ARQ mechanisms we discussed in this chapter.

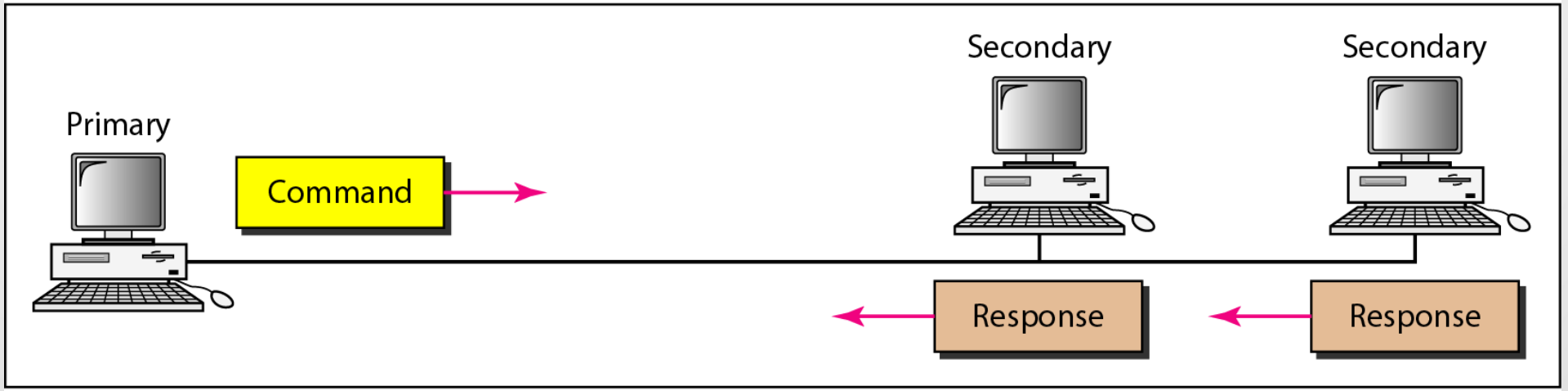
Configurations and Transfer Modes:

HDLC provides two common transfer modes that can be used in different configurations:

1. Normal response mode (NRM)
2. Asynchronous balanced mode (ABM).

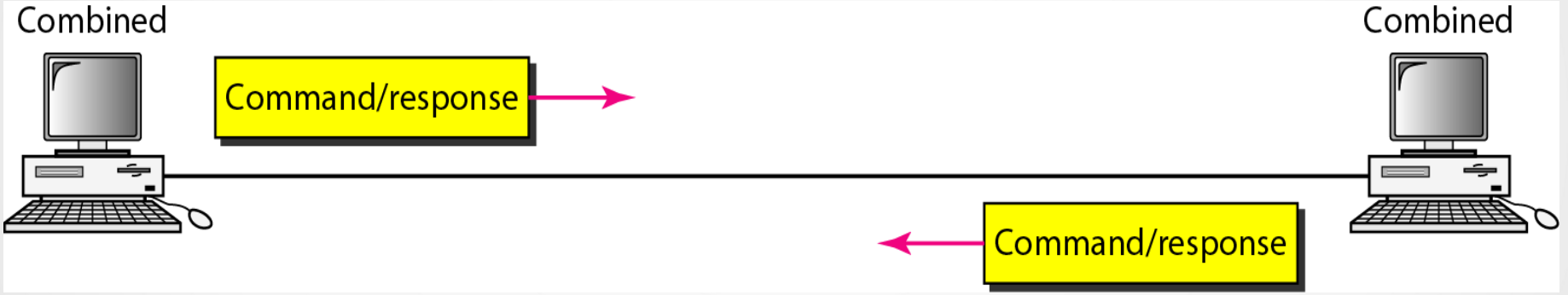
Figure : *Normal response mode*

a. Point-to-point



b. Multipoint

Figure : *Asynchronous balanced mode*



Frames

HDLC defines three types of frames:

1. Information frames (I-frames),

2. Supervisory frames (S-frames), and

3. Unnumbered frames (U-frames).

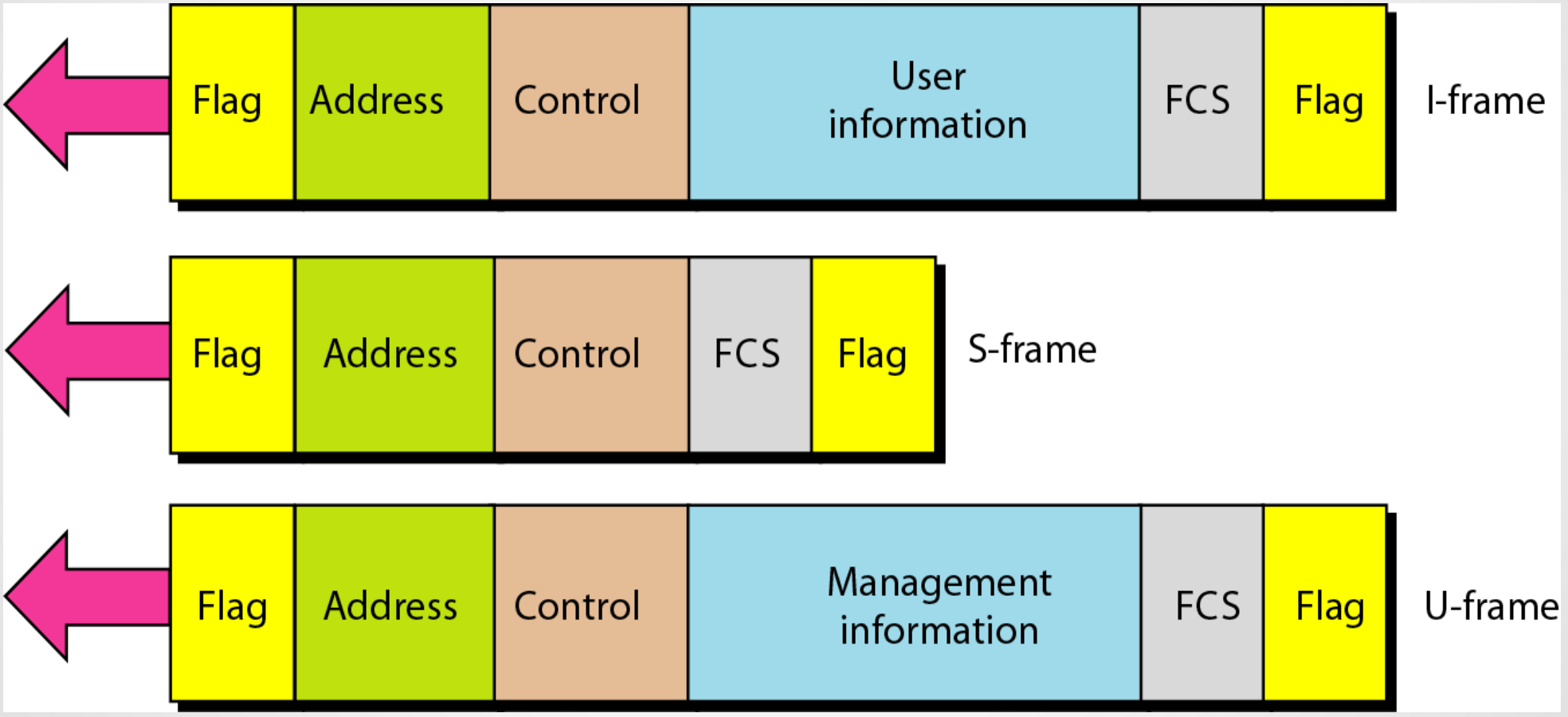
Each type of frame serves as an envelope for the transmission of a different type of message.

I-frames are used to transport user data and control information relating to user data (piggybacking).

S-frames are used only to transport control information.

U-frames are reserved for system management. Information carried by U-frames is intended for managing the link itself.

Figure : *HDLC frames*



Flag field: 8-bit sequence with the bit pattern 01111110.

Address field: Address of the secondary station. If a primary station created the frame, it contains a ***to*** address. If a secondary creates the frame, it contains a ***from*** address.

Control field: 1 or 2-byte segment of the frame used for flow and error control.

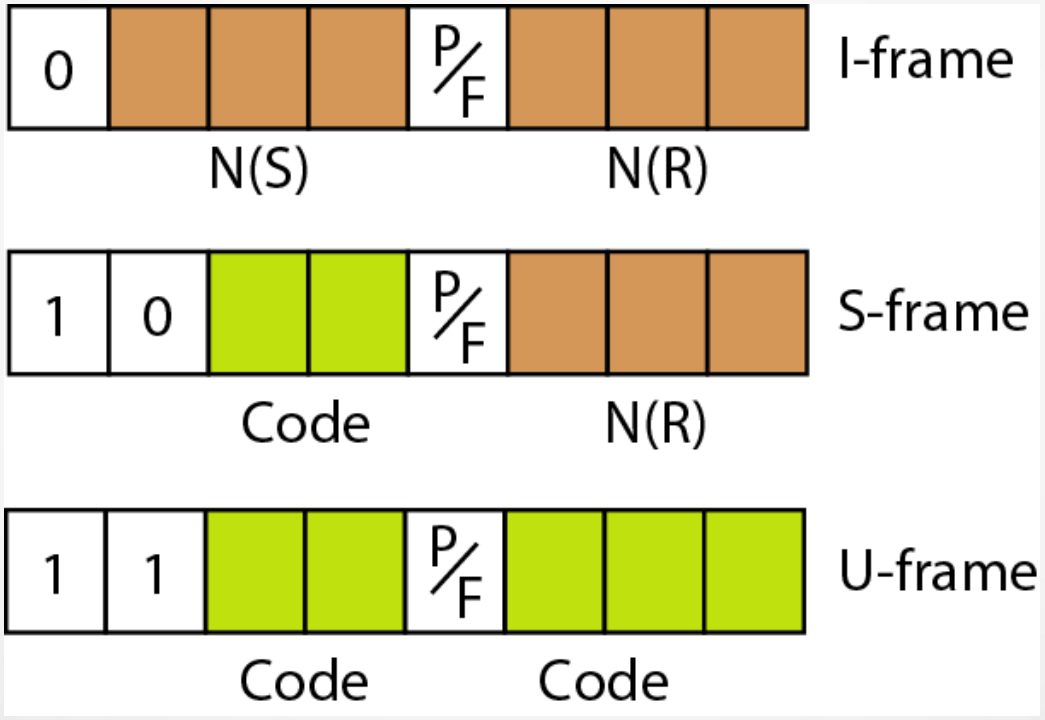
Information field: It contains the user's data from the network layer or management information. Its length can vary from one network to another.

FCS field: The frame check sequence (FCS) is the HDLC error detection field. It can contain either a 2- or 4-byte CRC.

Figure : *Control field format for the different frame types*

Control Field:

The control field determines the type of frame and defines its functionality.



Control Field for I-Frames

First bit: defines the type i.e. if control field is 0, this means the frame is an I-frame.

Next 3 bits: called N(S),define the sequence number of the frame. (0-7)

Last 3 bits: called N(R), correspond to the acknowledgment number when piggybacking is used.

P/F bit: when it is set (bit = 1), means **Poll or Final**.

It means **poll** when the frame is sent by a primary station to a secondary (when the address field contains the address of the receiver).

It means **final** when the frame is sent by a secondary to a primary (when the address field contains the address of the sender).

Control Field for S-Frames

Supervisory frames are used for flow and error control whenever piggybacking is either impossible or inappropriate.

First 2 bits: of the control field is 10, this means the frame is an S-frame.

Last 3 bits: called N(R), corresponds to the acknowledgment number (ACK) or negative acknowledgment number (NAK) depending on the type of S-frame.

2 bits : called **code** is used to define the type of S-frame itself.

With 2 bits, we can have four types of S-frames, as described below:

1. Receive Ready (RR): If code = 00**,** it is an RR S-frame.

This kind of frame acknowledges the receipt of a safe and sound frame or group of frames. In this case, the value N(R) field defines the acknowledgment number.

2. Receive not ready (RNR): If code= 10, it is an RNR S-frame.

It acknowledges the receipt of a frame or group of frames, and it announces that the receiver is busy and cannot receive more frames, i.e. congestion control mechanism by asking the sender to slow down.

3. Reject (REJ): If code= 01, it is a REJ S-frame.

This is a NAK frame,that can be used in Go-Back-N ARQ to improve the efficiency of the process by informing the sender, before the sender time expires, that the last frame is lost or damaged. N(R) = NAK.

3. Selective reject (SREJ): If code = 11, it is an SREJ S-frame.

This is a NAK frame used in Selective Repeat ARQ. Note that the HDLC Protocol uses the term selective reject instead of selective repeat. The value of N(R) = NAK

Table *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