Unit#5
Topics

Part A. **Data Link Layer – Selected topics**

Part B. **Consolidation of our learning in last 4 units**

Life of a web page request in internet involving protocols of all layers

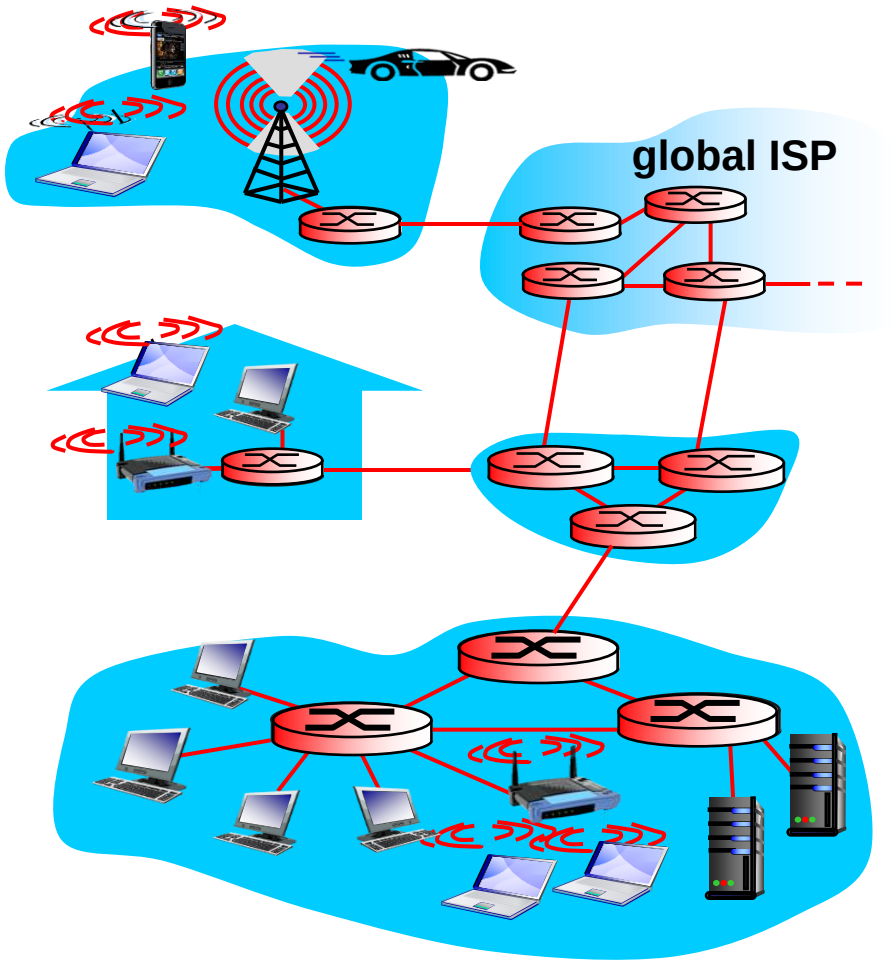
Part A

Data Link layer

Link layer: introduction

terminology:

- hosts and routers: nodes
- communication channels that connect adjacent nodes along communication path: links
  - wired links
  - wireless links
  - LANs
- layer-2 packet: frame, encapsulates datagram



data-link layer has responsibility of transferring datagram from one node to physically adjacent node over a link

Link layer services

- *framing, link access:*
  - encapsulate datagram into frame, adding header, trailer
  - channel access if shared medium
  - "MAC" addresses used in frame headers to identify source, dest
    - different from IP address!
- *reliable delivery between adjacent nodes*

Link layer services (more)

- *flow control:*
  - pacing between adjacent sending and receiving nodes
- *error detection*:
  - errors caused by signal attenuation, noise.
  - receiver detects presence of errors:
    - signals sender for retransmission or drops frame
- *error correction:*
  - receiver identifies *and corrects* bit error(s) without resorting to retransmission

Let us focus only on the functions which we have not covered in other layers :

- error detection
- sharing a broadcast channel: Multiple Access Control (MAC)
- local area networks
    - LAN switches
    - VLANs

Note

Data can be corrupted during transmission.

Some applications require that errors be detected and corrected.

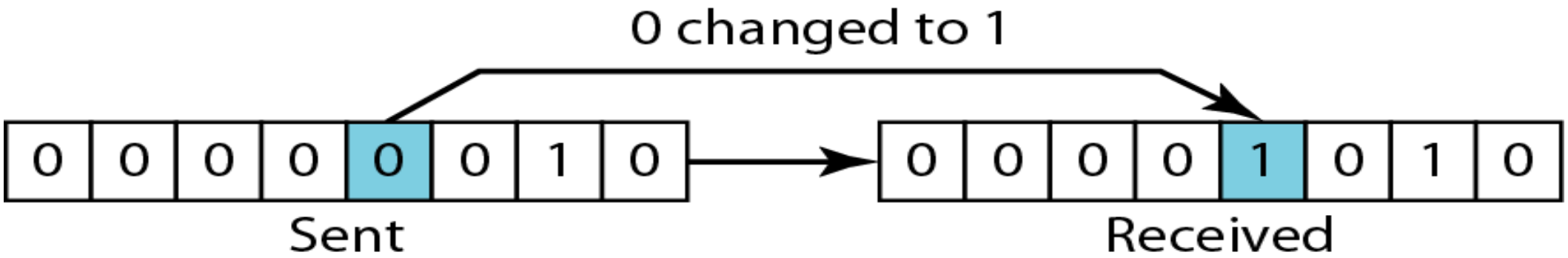
Types of Errors

Note

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

Types of Errors

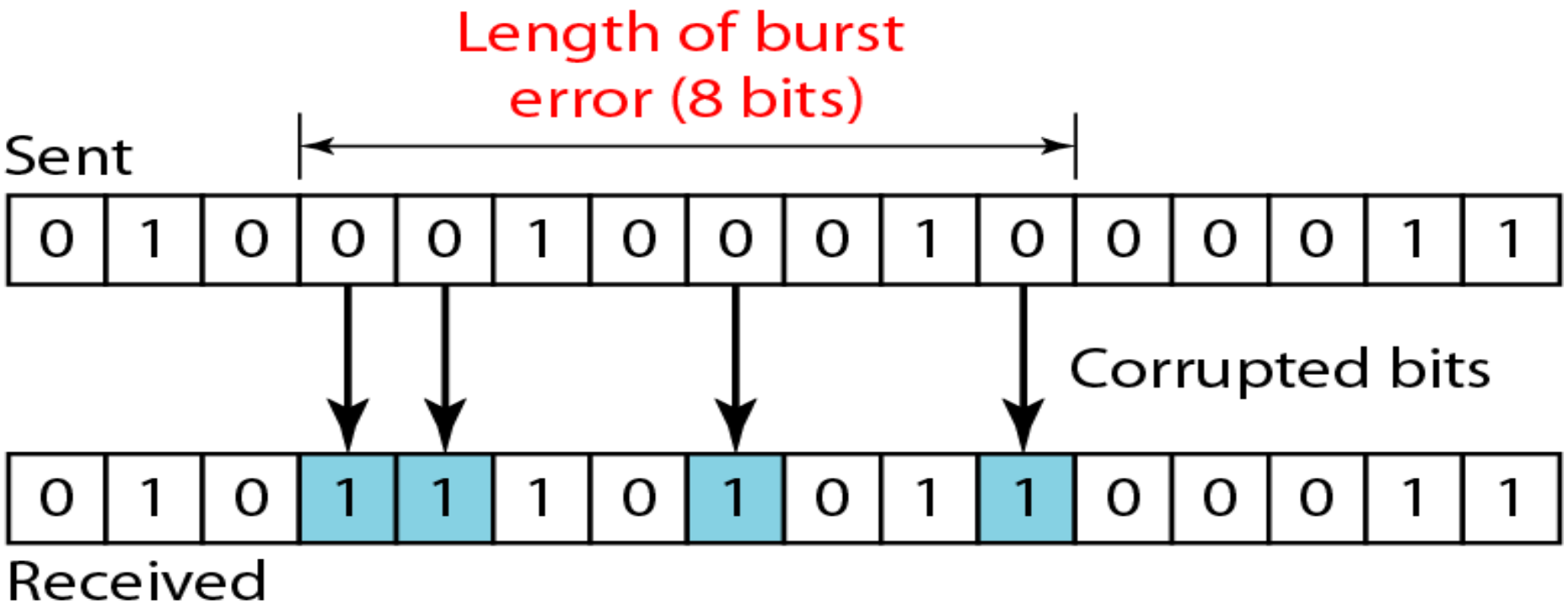
Figure 10.1 ***Single-bit error***



Types of Errors

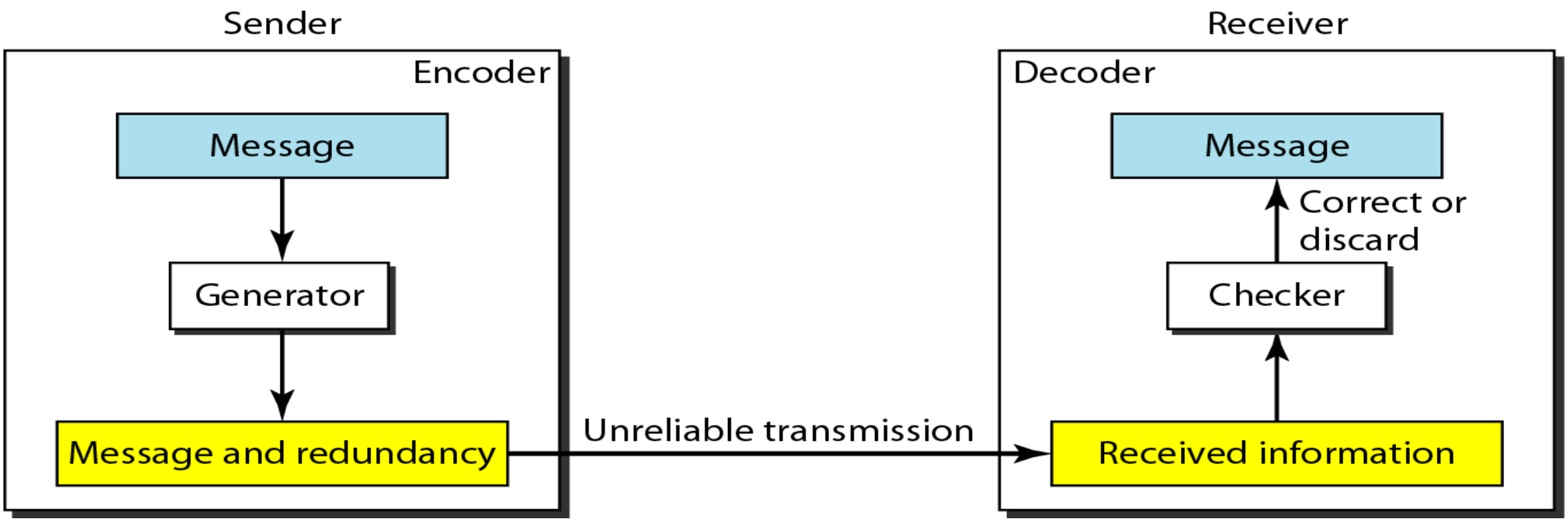
A burst error means that 2 or more bits in the data unit have changed.

Figure 10.2 *Burst error of length 8*



Note

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


Sender
Encoder
Message
Generator
Message and redundancy
Unreliable transmission
Receiver
Decoder
Message
Correct or
discard
Checker
Received information

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

CHECKSUM

The checksum is used in the Internet by several protocols although not at the data link layer.

CHECKSUM–Principle

Sender site:

1. **The message is divided into 16-bit words.**
2. **The value of the checksum word is set to 0.**
3. **All words including the checksum are added using one's complement addition.**
4. **The sum is complemented and becomes the checksum.**
5. **The checksum is sent with the data.**

CHECKSUM–Principle

Receiver site:

1. **The message (including checksum) is divided into 16-bit words.**
2. **All words are added using one's complement addition.**
3. **The sum is complemented and becomes the new checksum.**
4. **If the value of checksum is 0, the message is accepted; otherwise, it is rejected.**

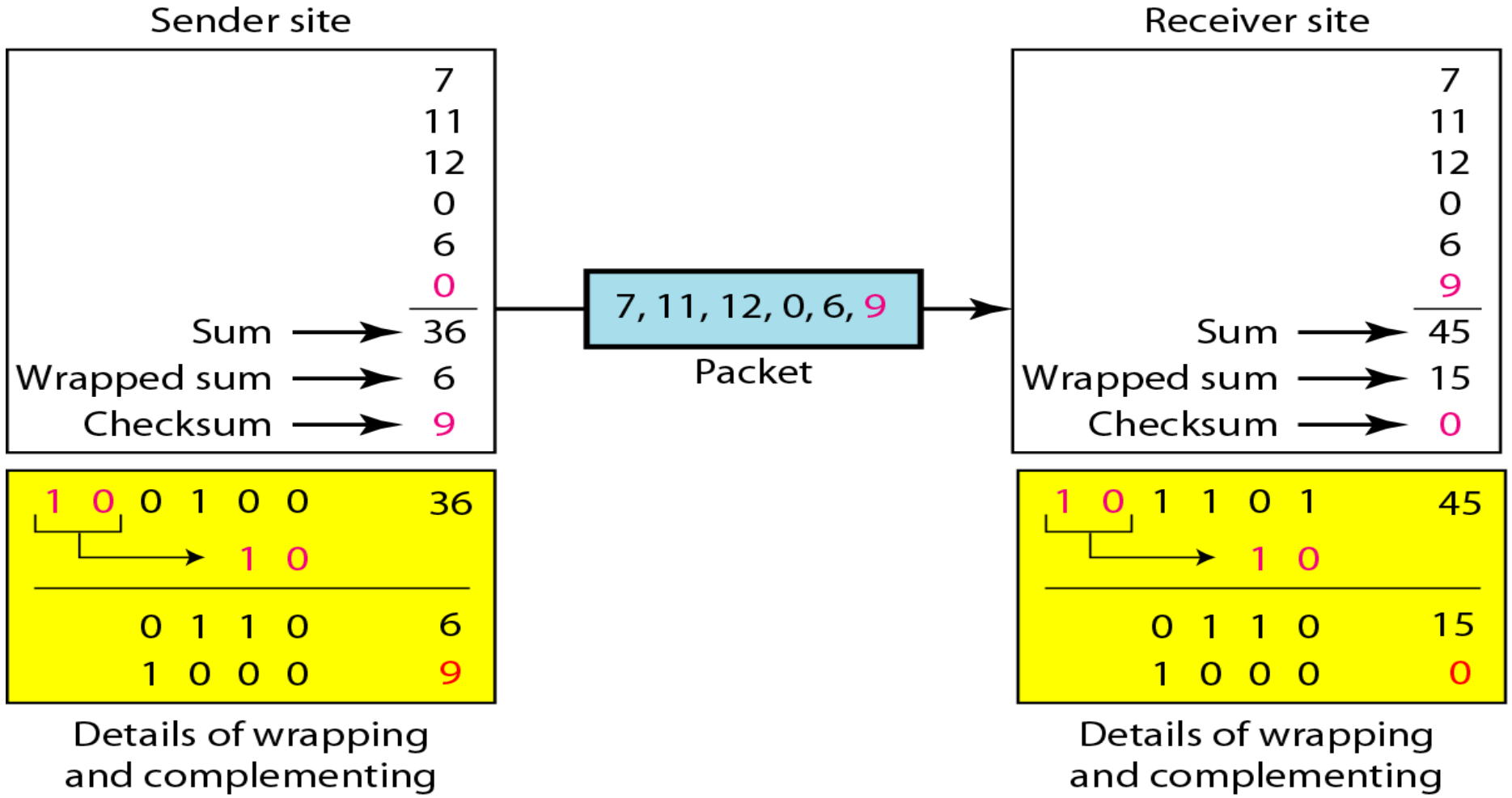
Example 10.18

Suppose our data is a list of five 4-bit numbers that we want to send to a destination. In addition to sending these numbers, we send the sum of the numbers. For example, if the set of numbers is (7, 11, 12, 0, 6), we send (7, 11, 12, 0, 6, 36), where 36 is the sum of the original numbers. The receiver adds the five numbers and compares the result with the sum. If the two are the same, the receiver assumes no error, accepts the five numbers, and discards the sum. Otherwise, there is an error somewhere and the data are not accepted.

Example 10.19

We can make the job of the receiver easier if we send the negative (complement) of the sum, called the checksum. In this case, we send (7, 11, 12, 0, 6, −36). The receiver can add all the numbers received (including the checksum). If the result is 0, it assumes no error; otherwise, there is an error.

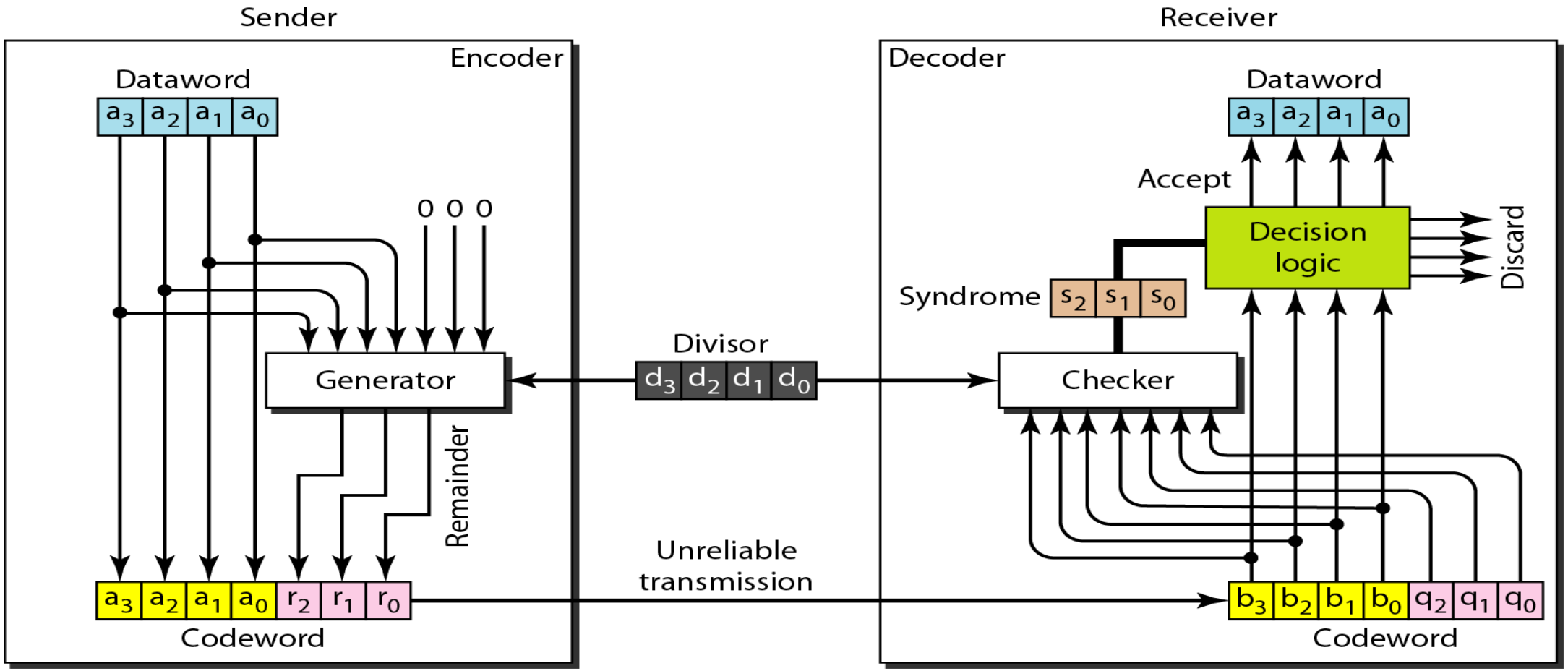
Figure 10.24 *Example 10.22*



Cyclic Redundancy Code (Check)

CRC

CRC encoder and decoder



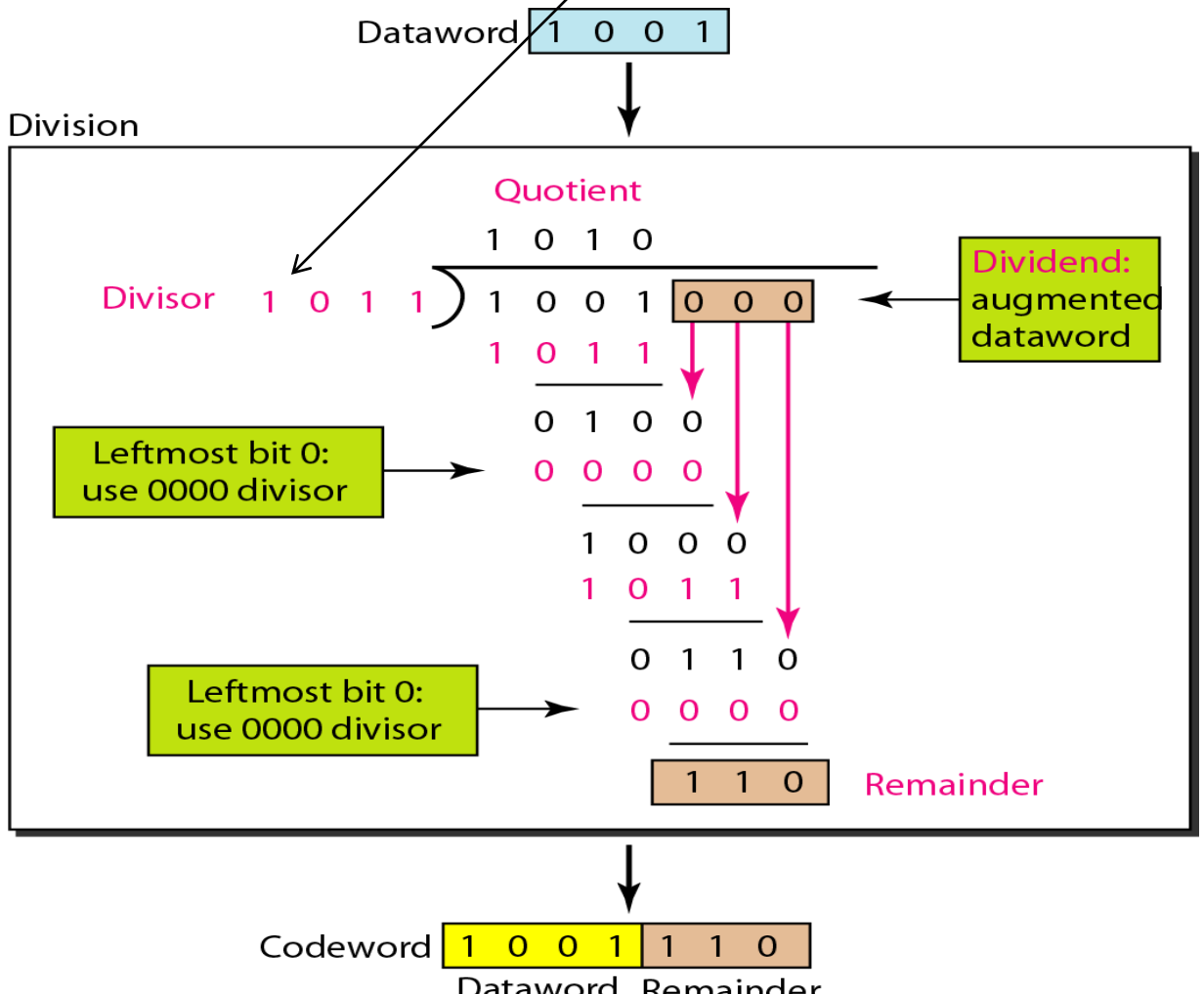
Example

CRC – at Sender

Data word to be sent from the sender : 1001

CRC bits appended , after the computation , using the divisor 1011, shown here : 110

What is sent after appending (called as codeword): : 1001110



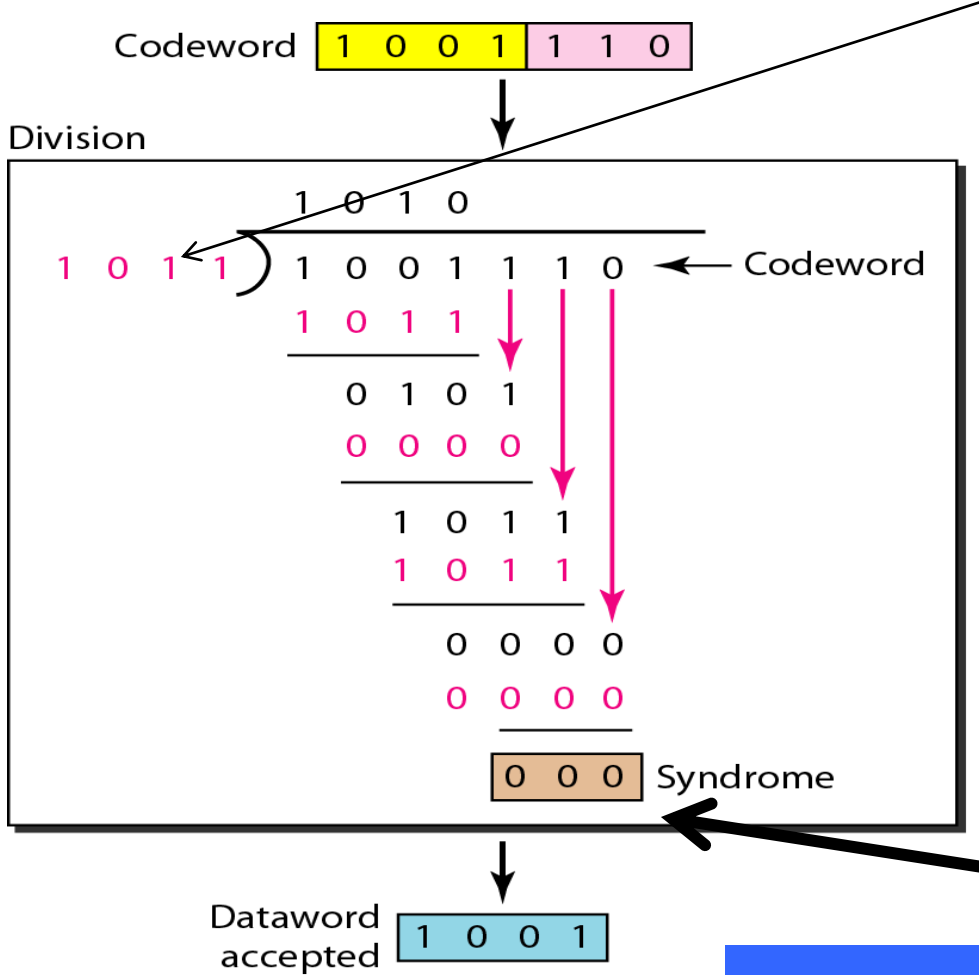
Example

CRC – at Receiver [when there is no error]

Codeword received : 1001110

Reminder , after the computation shown here, using the divisor 1011 : 000

So, the dataword accepted, 1001

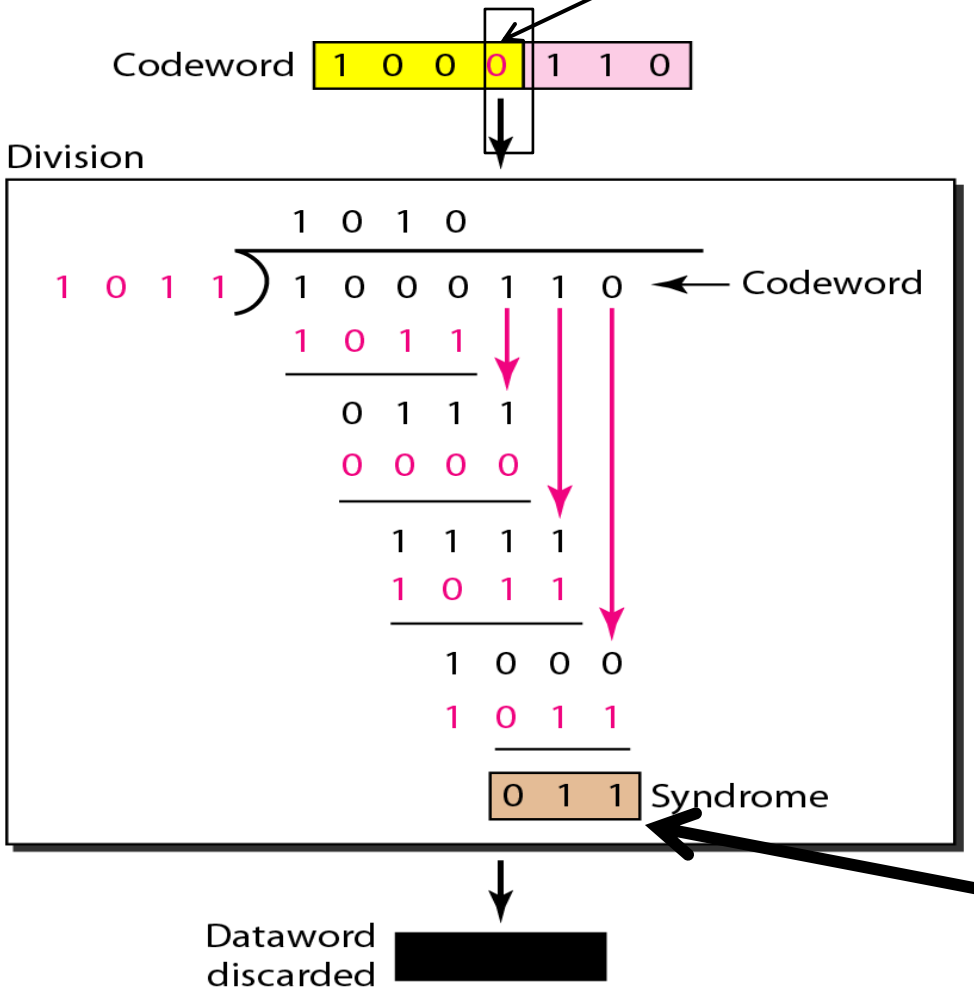


If the reminder is 0, there is no error

Example

CRC – at Receiver [when there is an error]

Here is the error – 1 bit is corrupted



If the reminder is not 0, there is an error

CRC – Summary of this example
How error is detected at the receiver ?