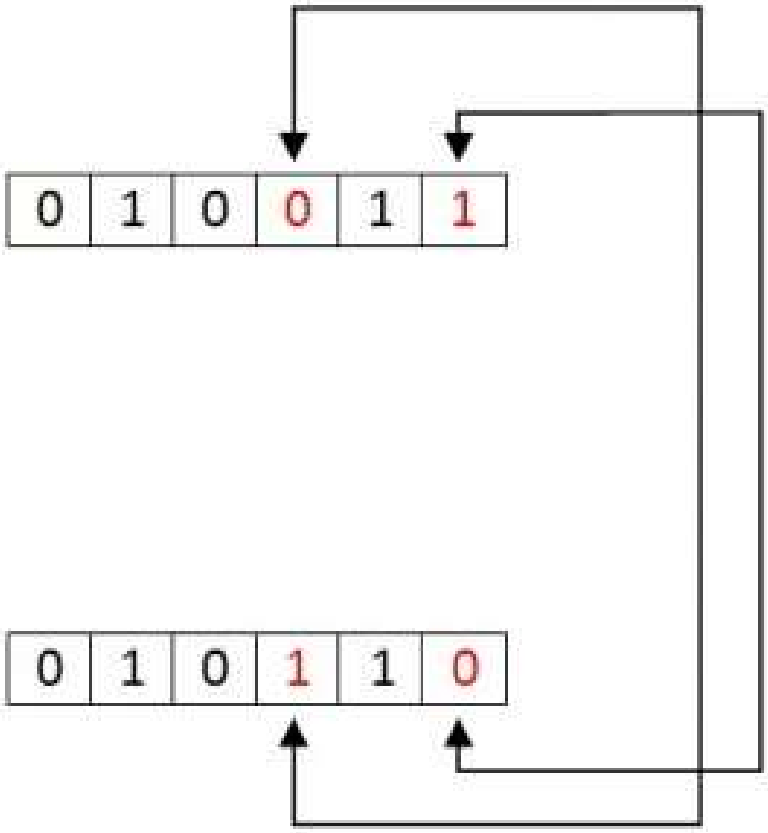
Chapter 10

Error Detection And Correction



Objective

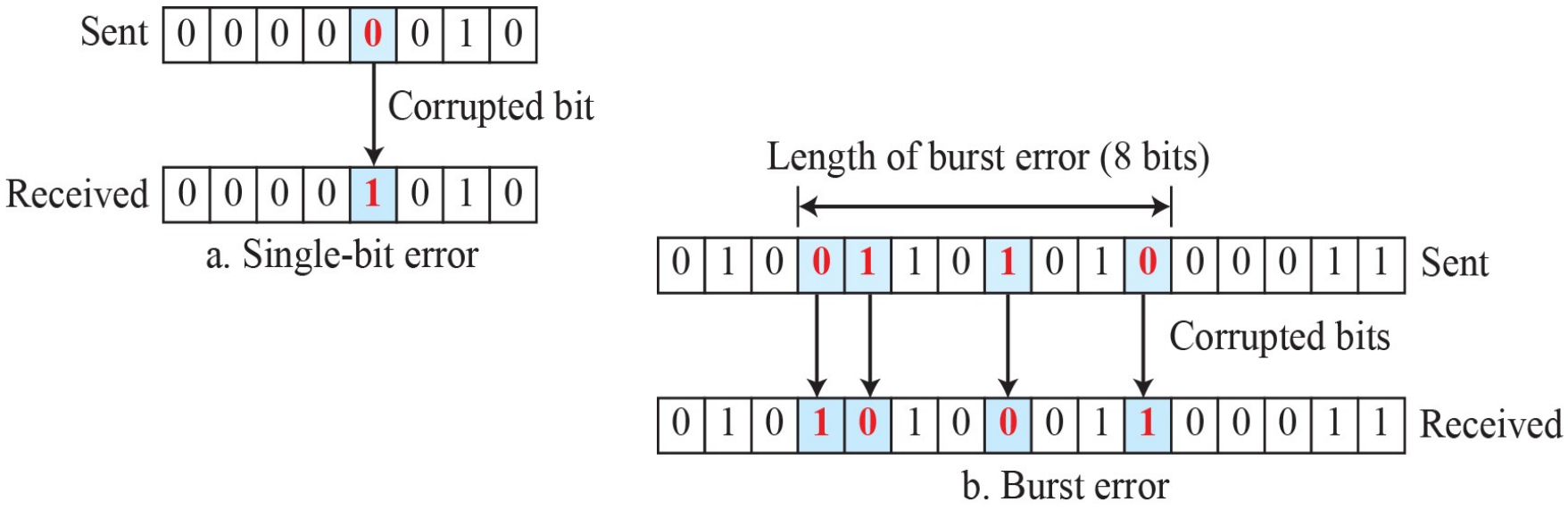
- Types of errors, the concept of redundancy, error detection and correction.
- Block coding, cyclic codes and CRC.
- Checksums
- Forward error correction - interleaving chunks and compounding high and low resolutions packets

Types of Errors

- Whenever bits flow from one point to another, they are subject to unpredictable changes because of interference.
- This interference can change the shape of the signal.
- The term single-bit error means that only one bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 1.

Types of Errors

- The term burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.



Single-bit and burst error

Redundancy

- The central concept in detecting or correcting errors is redundancy. To be able to detect or correct errors, we need to send some extra bits with our data.
- These redundant bits are added by the sender and removed by the receiver.
- Their presence allows the receiver to detect or correct corrupted bits.

Detection versus Correction

- The correction of errors is more difficult than the detection.
- In error detection, we are only looking to see if any error has occurred. We are not even interested in the number of corrupted bits. A single-bit error is the same for us as a burst error.
- In error correction, we need to know the exact number of bits that are corrupted and, more importantly, their location in the message.

Coding

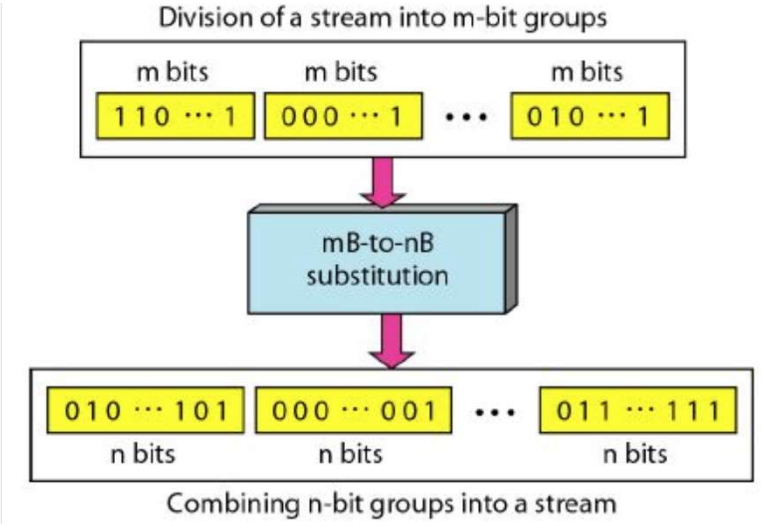
- Redundancy is achieved through various coding schemes. The sender adds redundant bits through a process that creates a relationship between the redundant bits and the actual data bits.

- The receiver checks the relationships between the two sets of bits to detect errors. The ratio of redundant bits to data bits and the robustness of the process are important factors in any coding scheme.

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**.



Hamming Distance

- Hamming distance is a metric for comparing two binary data strings. While comparing two binary strings of equal length, Hamming distance is the number of bit positions in which the two bits are different.

- Suppose there are two strings 11011001 and 10011101. Simply, 11011001 $\oplus$ 10011101 = 01000100. Since, this contains two 1s, the Hamming distance is 2.

Minimum Hamming Distance

- In a set of strings of equal lengths, the minimum Hamming distance is the smallest Hamming distance between all possible pairs of strings in that set.

- Suppose there are four strings 010, 011, 101 and 111.

  $010 \oplus 011 = 001$ (1), $010 \oplus 101 = 111$ (3),

  $010 \oplus 111 = 101$ (2), $011 \oplus 101 = 110$ (2),

  $011 \oplus 111 = 100$ (1), $101 \oplus 111 = 010$ (1),

- Hence, the Minimum Hamming Distance is 1.

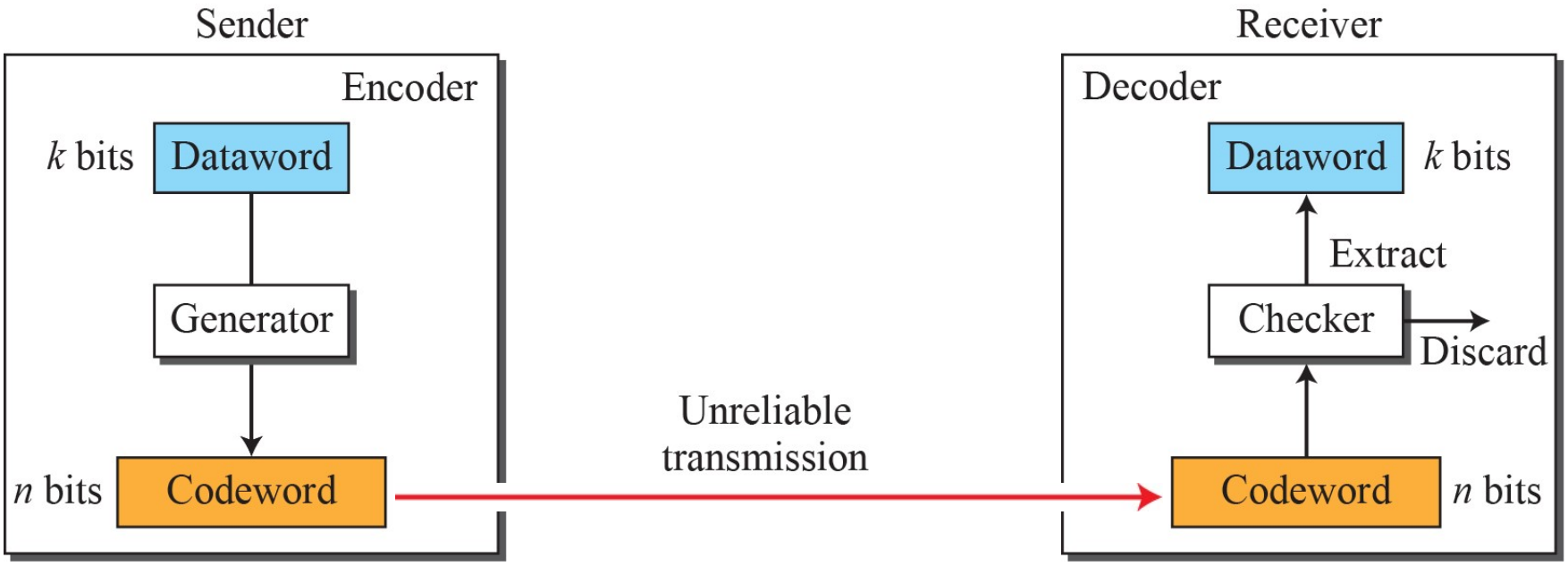
Minimum Hamming Distance

- To design a code that can **detect** *d* single bit errors, the minimum Hamming distance for the set of codewords must be *d* + 1 (or more).

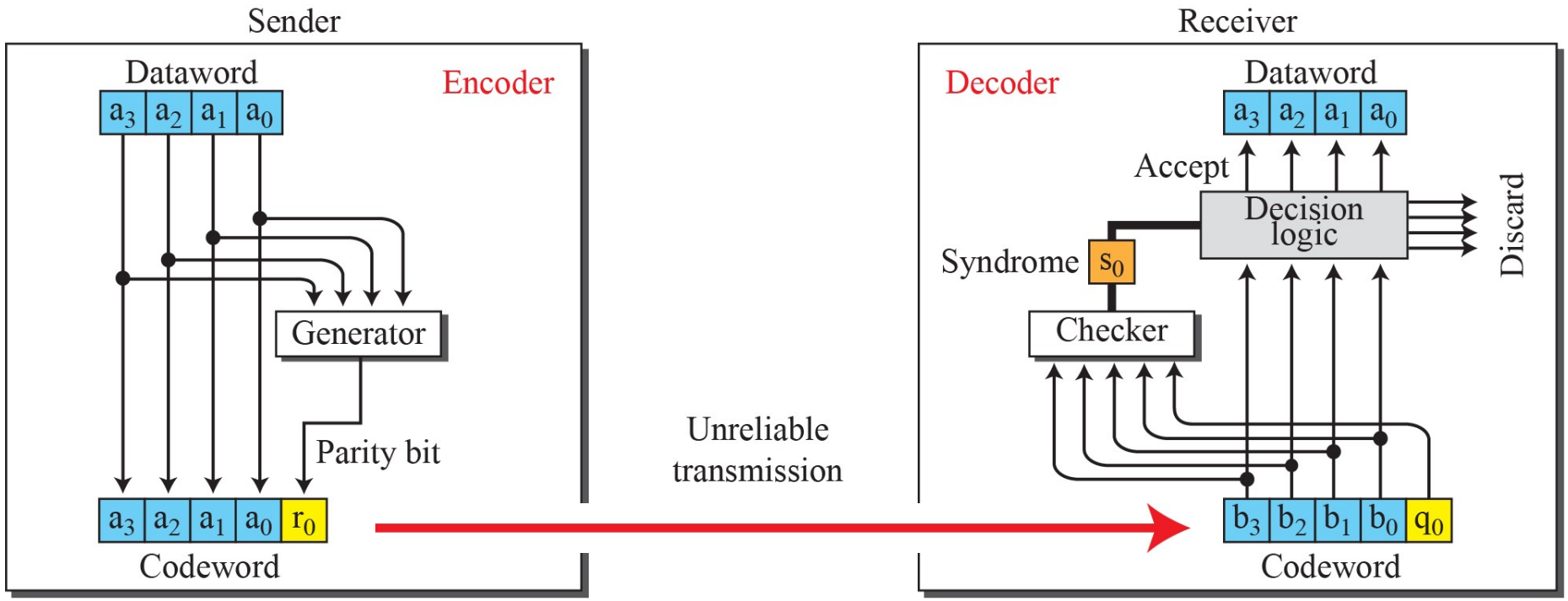
- To design a code that can **correct** *d* single bit errors, a minimum distance of 2*d* + 1 is required.

Dataword	*Codeword*	*Dataword*	*Codeword*
00	000	10	101
01	011	11	110

Error Detection



Process of error detection in block coding



Encoder and decoder for simple parity-check code

CYCLIC CODES

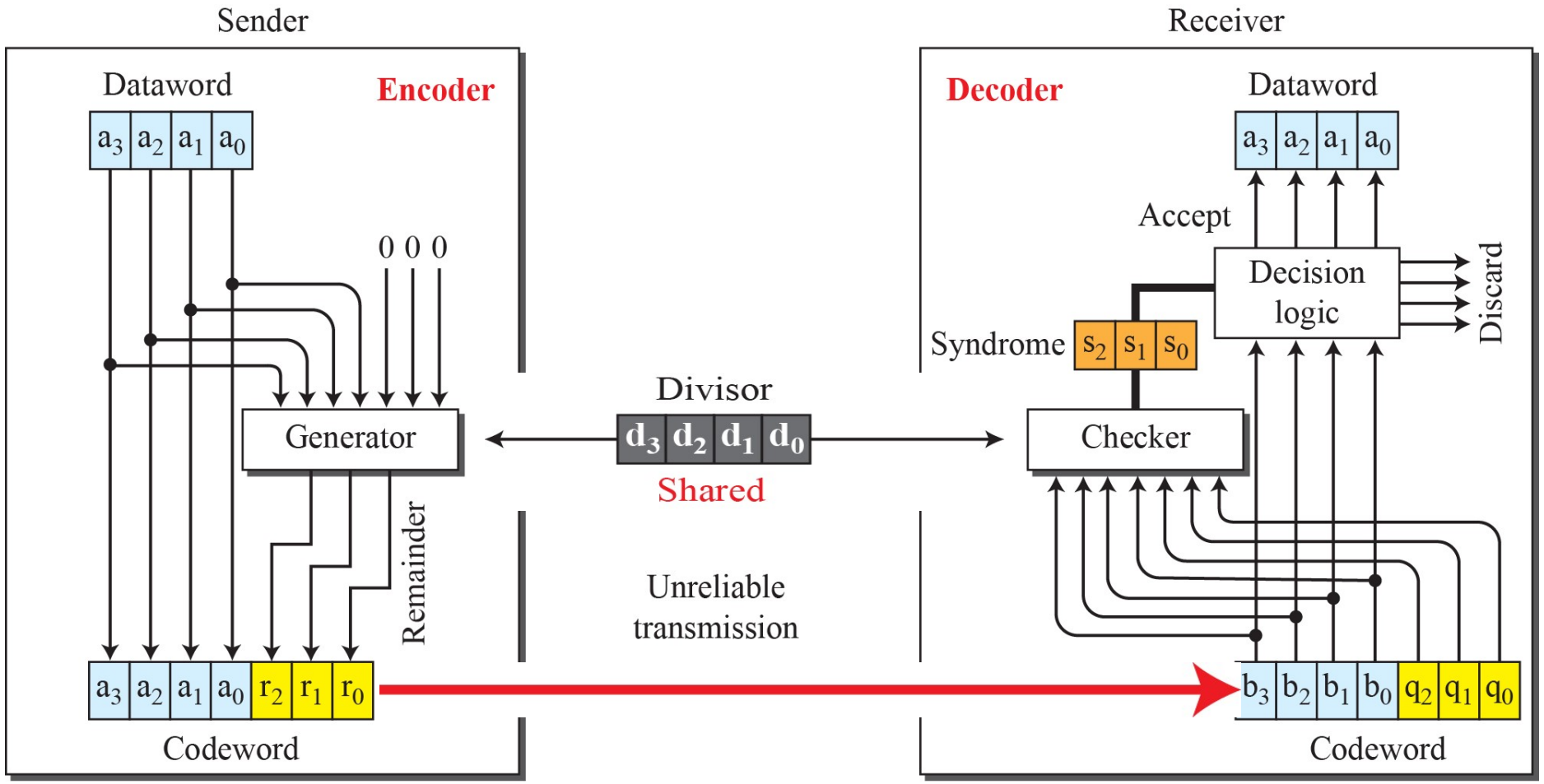
- Cyclic codes are special linear block codes with one extra property. In a cyclic code, if a codeword is cyclically shifted (rotated), the result is another codeword.

- For example, if 1011000 is a codeword and we cyclically left-shift, then 0110001 is also a codeword.

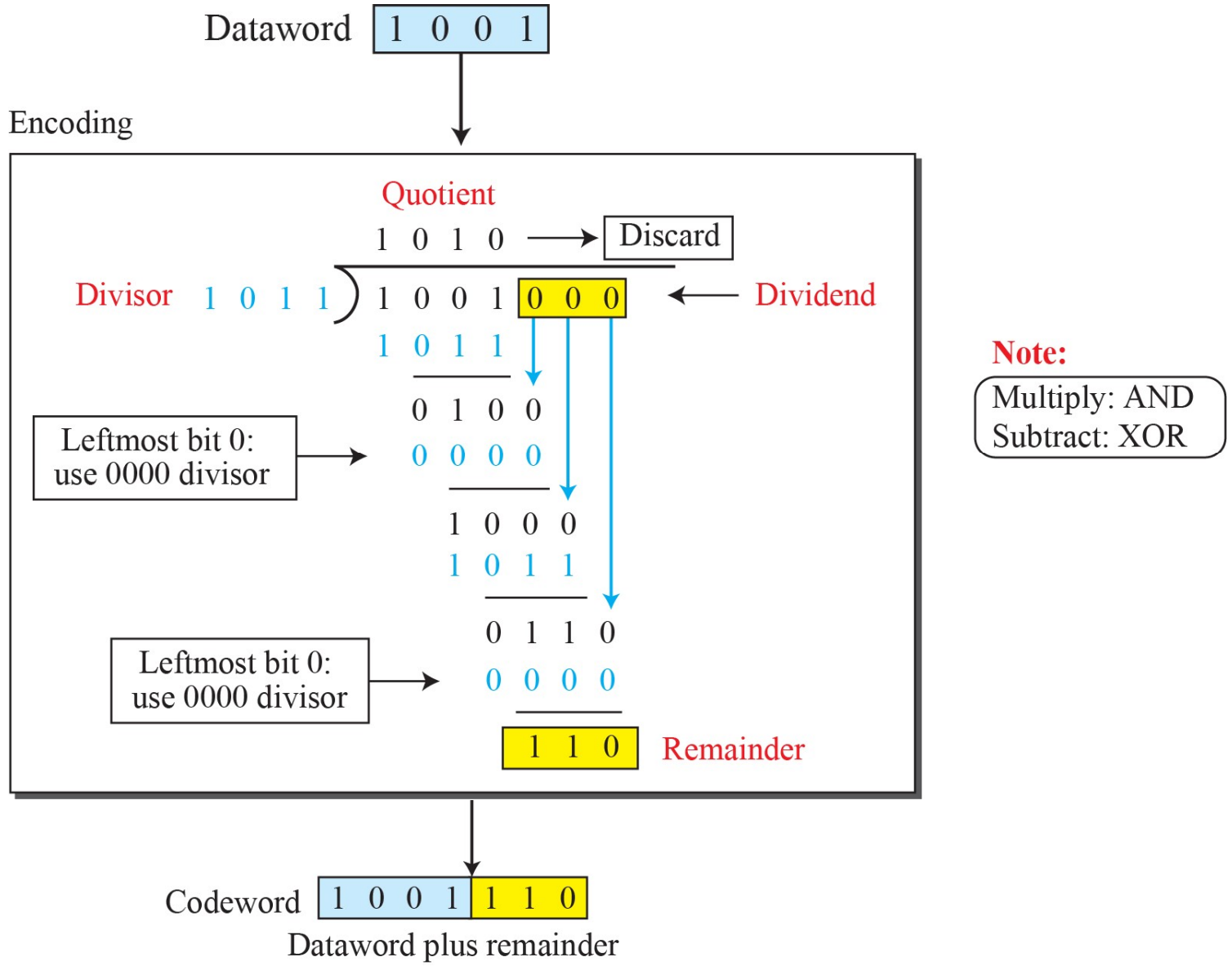
Cyclic Redundancy Check

- We can create cyclic codes to correct errors. However, the theoretical background required is beyond the scope of this book.

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



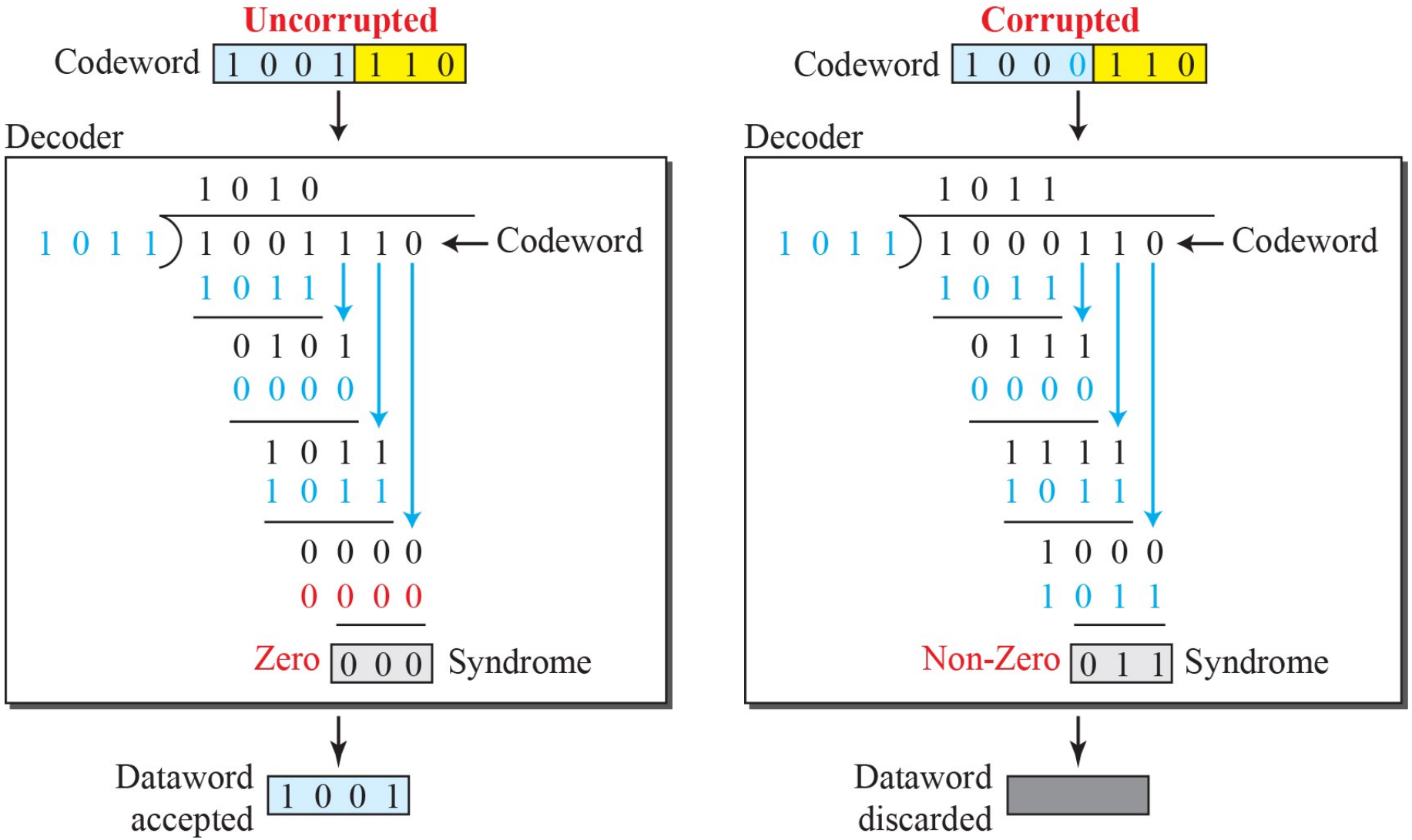
CRC encoder and decoder



Note:

Multiply: AND
Subtract: XOR

Division in CRC encoder



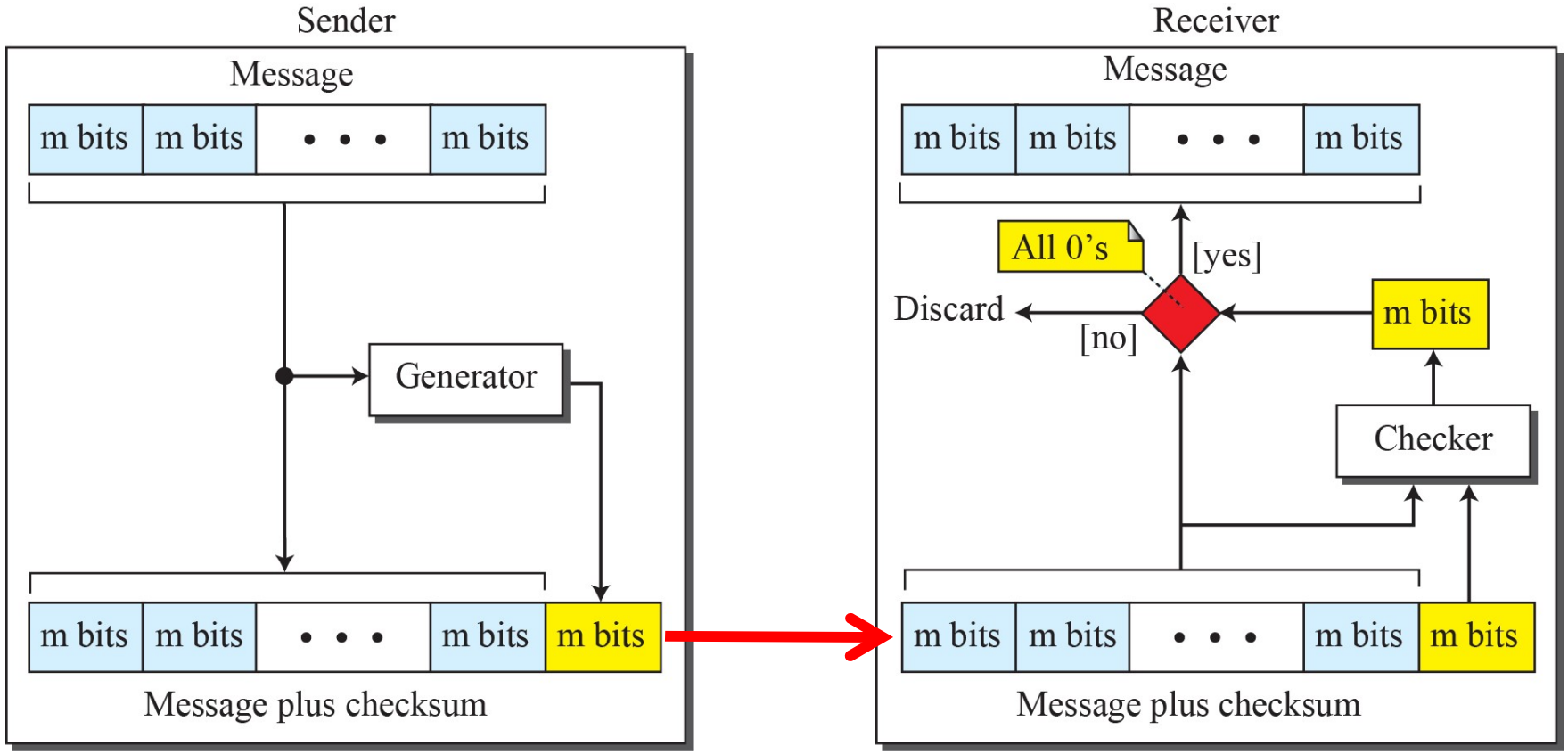
Division in the CRC decoder for two cases

Advantages of Cyclic Codes

- We have seen that cyclic codes have a very good performance in detecting single-bit errors, double errors, an odd number of errors, and burst errors.
- They can easily be implemented in hardware and software. They are especially fast when implemented in hardware.
- This has made cyclic codes a good candidate for many networks.

CHECKSUM

- Checksum is an error-detecting technique that can be applied to a message of any length.
- In the Internet, the checksum technique is mostly used at the network and transport layer rather than the data-link layer.
- However, to make our discussion of error detecting techniques complete, we discuss the checksum.


Sender
Message
m bits
m bits
m bits
Generator
m bits
m bits
m bits
m bits
Message plus checksum
Receiver
Message
m bits
m bits
m bits
All 0's
[yes]
Discard
[no]
m bits
Checker
m bits
m bits
m bits
m bits
m bits
Message plus checksum

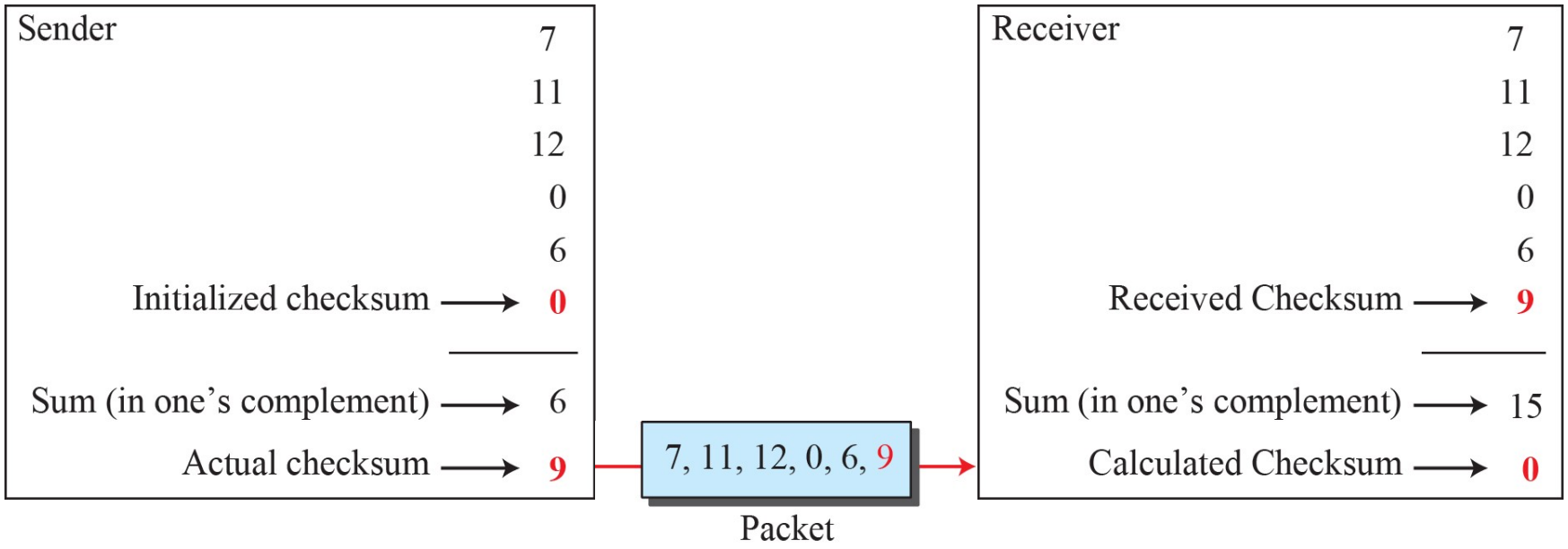
Checksum

Procedure to calculate the traditional checksum

Sender	*Receiver*
1. The message is divided into 16-bit words.	1. The message and the checksum is received.
2. The value of the checksum word is initially set to zero.	2. The message is divided into 16-bit words.
3. All words including the checksum are added using one's complement addition.	3. All words are added using one's complement addition.
4. The sum is complemented and becomes the checksum.	4. The sum is complemented and becomes the new checksum.
5. The checksum is sent with the data.	5. If the value of the checksum is 0, the message is accepted; otherwise, it is rejected.

Concept

- The idea of the traditional checksum is simple. We show this using a simple example.

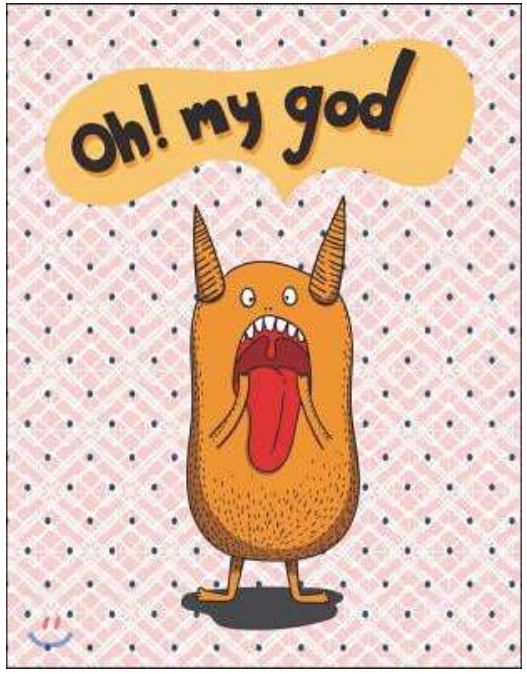


Example

Other Approaches

- Fletcher checksum and Adler checksum

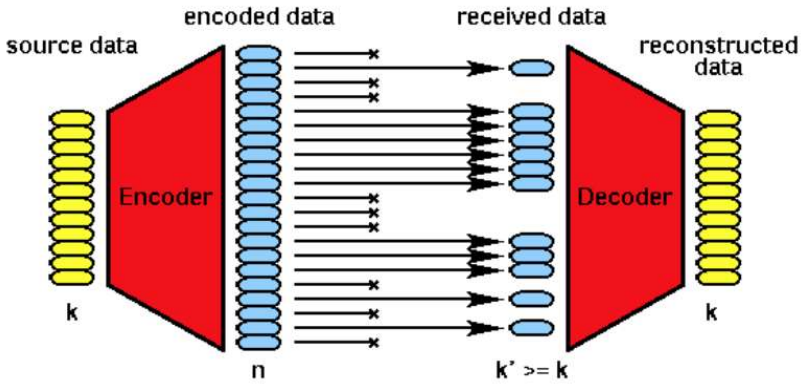
  …… bla~~ bla~~ bla~~ bla~~.



FORWARD ERROR CORRECTION

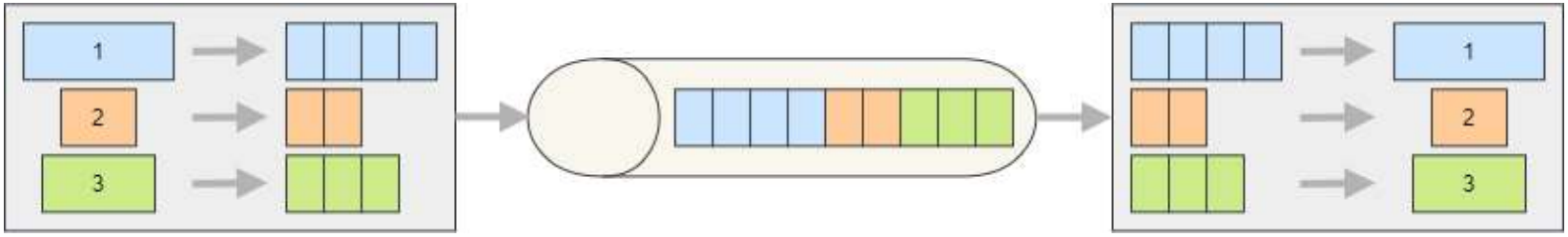
- We discussed error detection and retransmission in the previous sections. However, retransmission of corrupted and lost packets is not useful for real-time multimedia transmission.

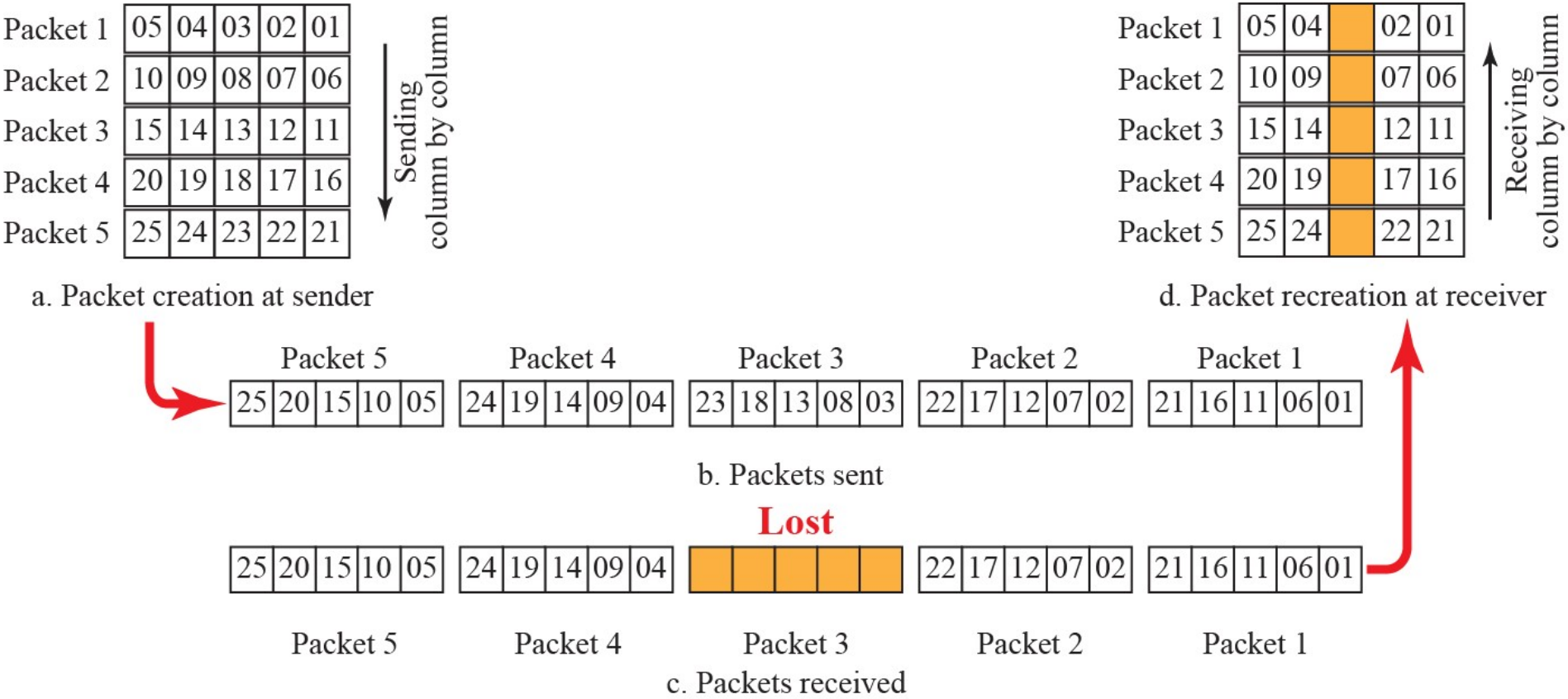
- We need to correct the error or reproduce the packet immediately.



Chunk Interleaving

- FEC in multimedia is to allow some small chunks to be missing at the receiver.
- We cannot afford to let all the chunks belonging to the same packet be missing; however, we can afford to let one chunk be missing in each packet.



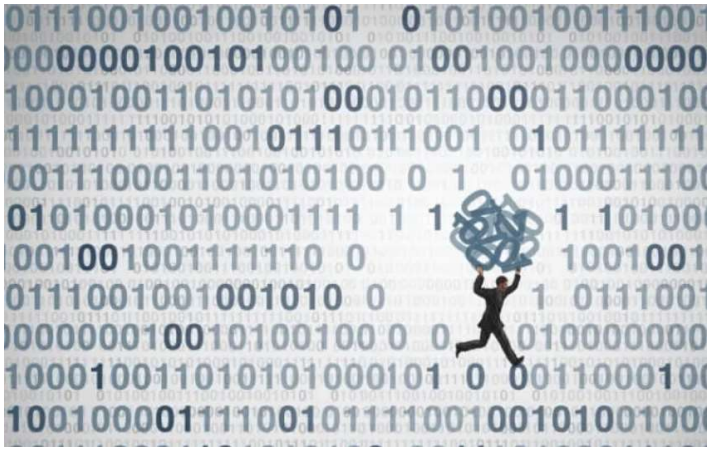


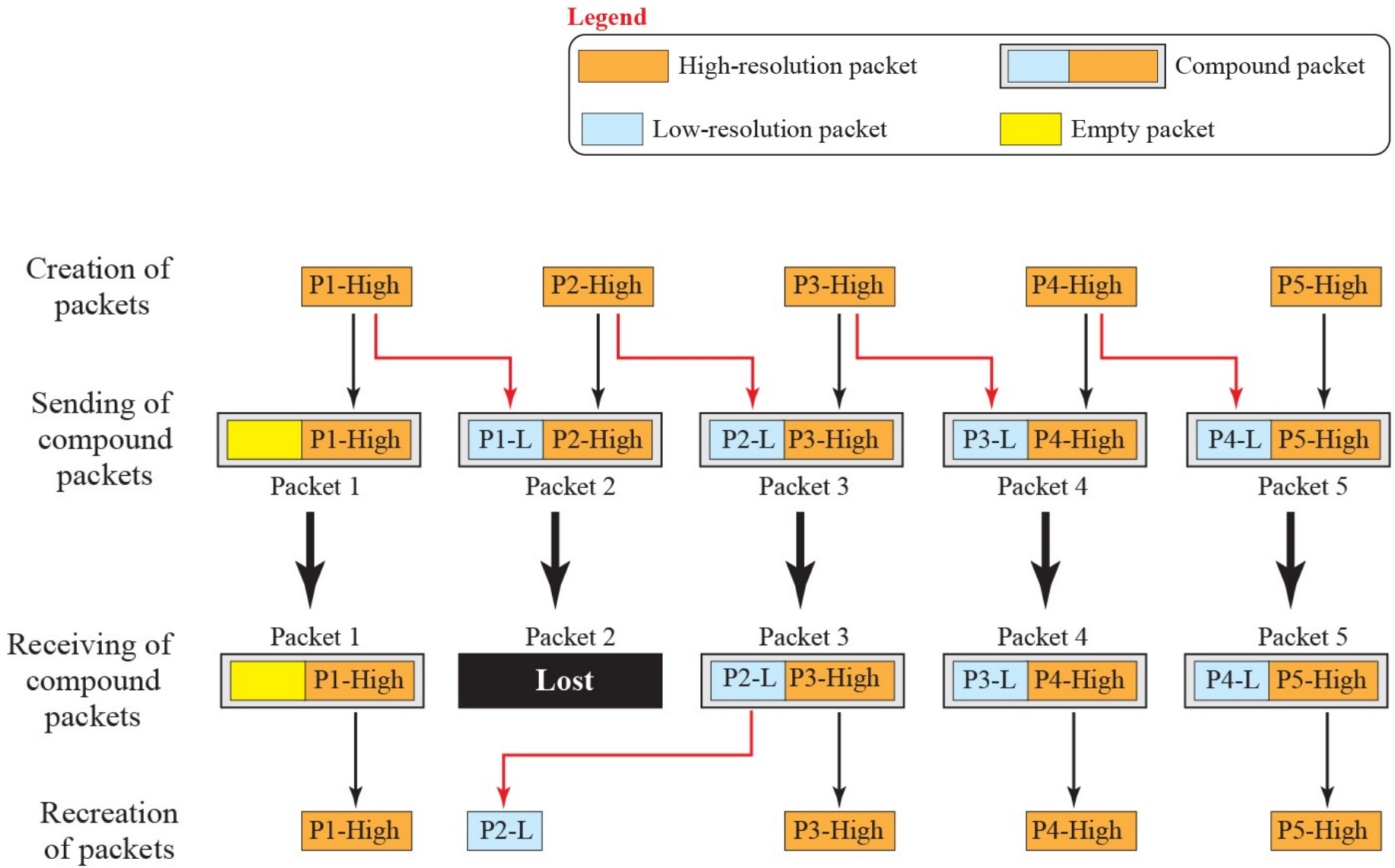
Interleaving

Compounding

- Still another solution is to create a duplicate of each packet with a low-resolution redundancy and combine the redundant version with the next packet.

- For example, we can create four low-resolution packets out of five high-resolution packets and send them.





Compounding high-and-low resolution packets