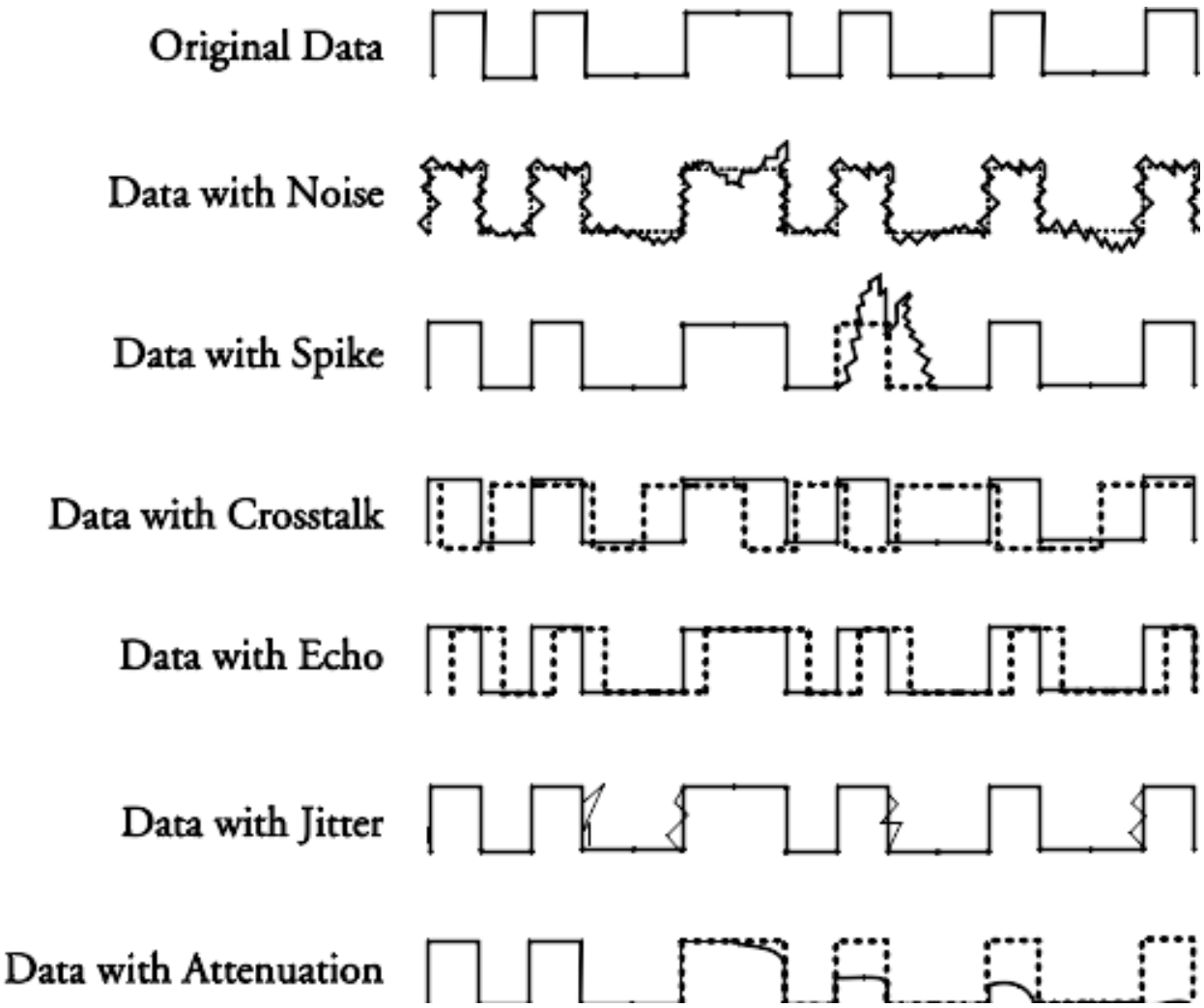
Error Control



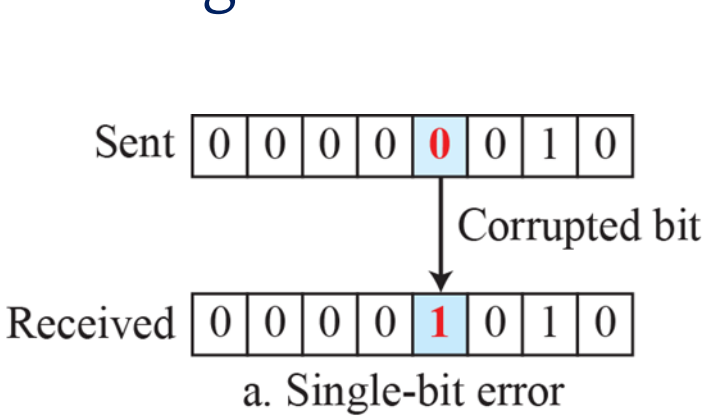
Common forms of data errors at the data-link level

Type of Errors

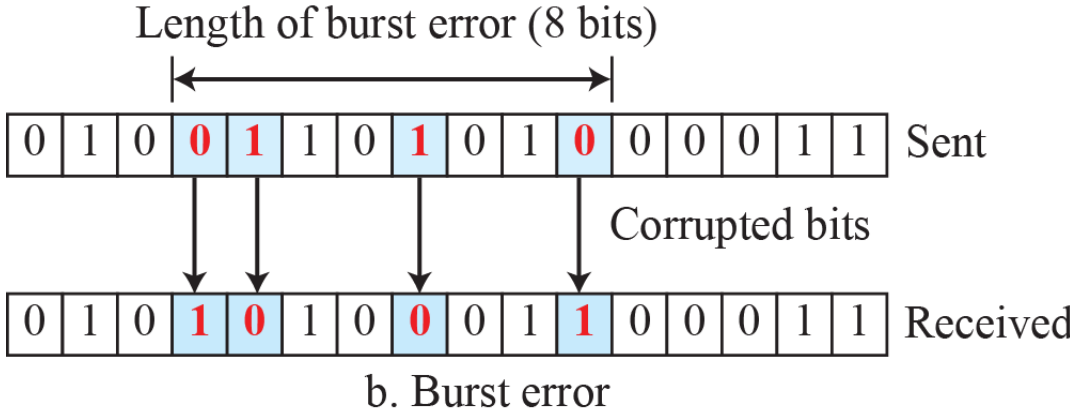
Interference/noise can change the shape of the signal and result in errors.

Single-bit error means that only 1 bit of a given data unit (such as a byte, character, or packet) is changed from 1 to 0 or from 0 to 10.

Burst error means that 2 or more bits in the data unit have changed from 1 to 0 or from 0 to 1.



a. Single-bit error



b. Burst error

Error Control

- Digital transmission systems introduce errors
  - BER: Wired electrical communication ~ 10^{-6}
  - BER: Optical communication ~ 10^{-9}
  - BER: Wireless ~10^{-3}
- Applications require certain reliability level
  - Data applications require error-free transfer
  - Voice & video applications tolerate some errors
- Error control used when transmission system does *not* meet application requirement

Error Control

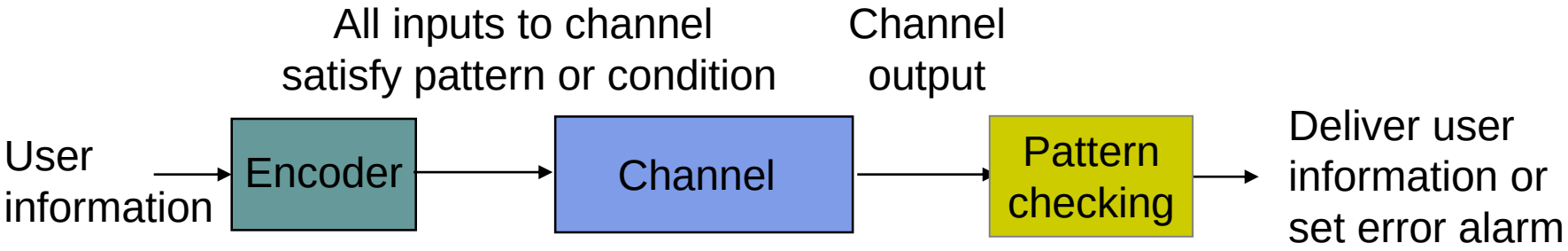
- Error control ensures a data stream is transmitted to a certain level of accuracy despite errors
- Two basic approaches:
  - Error ***detection*** & retransmission (ARQ)
  - Forward error ***correction*** (FEC)

Error Control

- Two basic approaches:
  - Error detection and Automatic Repeat Request (ARQ)
    - need for return/feedback channel
    - retransmissions wastes bandwidth
  - Forward error control (FEC)
    - Suitable when return channel not available
    - Or retransmission for correction of a few errors not very efficient
    - Eg: satellite and deep space commn., storage devices, etc
    - Additional redundancy, complex processing,…

Key Idea

- All transmitted data blocks ("codewords") satisfy a pattern
- If received block doesn't satisfy pattern, it is in error
- Redundancy: Only a subset of all possible blocks can be codewords
  - map k bits into n bits, n>k
  - $2^n - 2^k$ unused words
- Blindspot: when channel transforms a codeword into another codeword



Redundancy for Error Detection/Correction

- *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.*
- *Presence of redundant bits allows the receiver to detect or correct corrupted bits.*

Error Control: Example

- *Single Parity Code*: append an overall parity check to k information bits

  Info Bits: $b_1, b_2, b_3, \ldots, b_k$

  Check Bit: $b_{k+1} = [b_1 + b_2 + b_3 + \ldots + b_k]$ modulo 2

  Codeword: $(b_1, b_2, b_3, \ldots, b_k, b_{k+1})$

- All codewords have even # of 1s
- Receiver checks to see if # of 1s is even
  - All error patterns that change an odd # of bits are detectable
  - All even-numbered patterns are undetectable
- Parity bit used in ASCII code

Examples of Single Parity Code

Information (7 bits): (0, 1, 0, 1, 1, 0, 0)
Parity Bit: $b_8 = 0 + 1 + 0 + 1 + 1 + 0 = 1$
Codeword (8 bits): (0, 1, 0, 1, 1, 0, 0, 1)

- If single error in bit 3 : (0, 1, 1, 1, 1, 0, 0, 1)
  - # of 1's =5, odd
  - Error detected
- If errors in bits 3 and 5: (0, 1, 1, 1, 0, 0, 0, 1)
  - # of 1's =4, even
  - Error not detected

Hamming Distance

Hamming distance, **d(x,y)**, is a metric for comparing two binary data strings/words ***x*** and ***y***; it is the number of bit positions in which the two strings differ

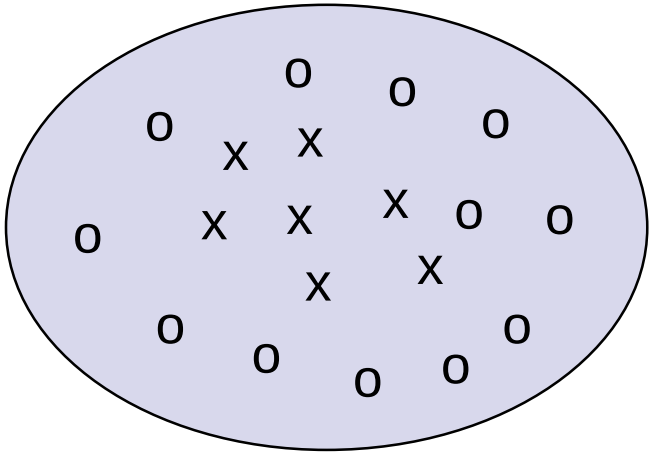
It can be found by XORing the words and counting the number of 1's

Example: Hamming distance between two pairs of words.

1. The Hamming distance $d(000, 011)$ is 2 because $(000 \oplus 011)$ is 011 (two 1s).
2. The Hamming distance $d(10101, 11110)$ is 3 because $(10101 \oplus 11110)$ is 01011 (three 1s).

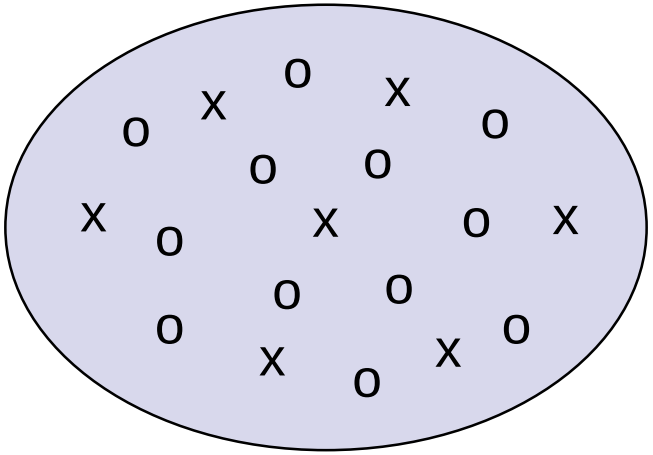
What is a good code?

- Many channels have preference for error patterns that have fewer # of errors
- These error patterns map transmitted codeword to nearby n-tuple
- If codewords are close to each other, then detection failures will occur
- Good codes should maximize separation between codewords



Poor distance properties

x = codewords
o = **noncodewords**



Good distance properties

How effective are error detection codes?

- Effectiveness measured by the probability that it fails to detect an error
  - to compute it, prob. of occurrence of errors are required

- Three models of errors:
  - Random error vector model
  - Random bit-error model
  - Burst error model

How effective are error detection codes?

- Code words with n bits
- Define *error vector*: $\underline{e} = [e_1, e_2, \cdots, e_n]$
  - $e_i = 1$ if error in the i^{th} transmitted bit

Random error vector model: all 2^n error vectors are equally likely

- Eg: erro vectors [1,0,…,0] and [1,1,…,1] have the same error prob.
- single parity code fails when error vector has even no. of 1s
- hence, prob. of error detection failure: ½
- Is it possible to detect more errors if we add more check bits?

What if bit errors are random?

- Random bit-error model: many transmission channels introduce bit errors at random, independently of each other, and with probability p
- Some error patterns are more probable than others:

$$P[10000000] = p(1-p)^7 = (1-p)^8 \left(\frac{p}{1-p}\right) \text{ and}$$

$$P[11000000] = p^2(1-p)^6 = (1-p)^8\left(\frac{p}{1-p}\right)^2$$

- In any worthwhile channel $p < 0.5$, and so $(p/(1-p) < 1$
- It follows that patterns with 1 error are more likely than patterns with 2 errors and so forth
- What is the probability that an undetectable error pattern occurs?

Single parity check code with random bit errors

- Undetectable error pattern if even # of bit errors:

$$P[\text{error detection failure}] = P[\text{undetectable error pattern}]$$
$$= P[\text{error patterns with even number of 1s}]$$

$$= \binom{n}{2} p^2(1-p)^{n-2} + \binom{n}{4} p^4(1-p)^{n-4} + \ldots$$

- Example: Evaluate above for $n = 32$, $p = 10^{-3}$

$$P[\text{undetectable error}] = \binom{32}{2} (10^{-3})^2 (1 - 10^{-3})^{30} + \binom{32}{4} (10^{-3})^4 (1 - 10^{-3})^{28}$$

$$\approx 496\,(10^{-6}) + 35960\,(10^{-12}) \approx 4.96\,(10^{-4})$$

- For this example, roughly 1 in 2000 error patterns is undetectable

Coding For Error Detection

- 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 the 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.

How about data rate?

Error Detection

How can errors be detected by using block coding? If the following two conditions are met, the receiver can detect a change in the original codeword.

1. The receiver has (or can find) a list of valid codewords.

2. The original codeword has changed to an invalid one.

Example

Let us assume that k = 2 and n = 3. Table shows the list of datawords and codewords.

A code for error detection

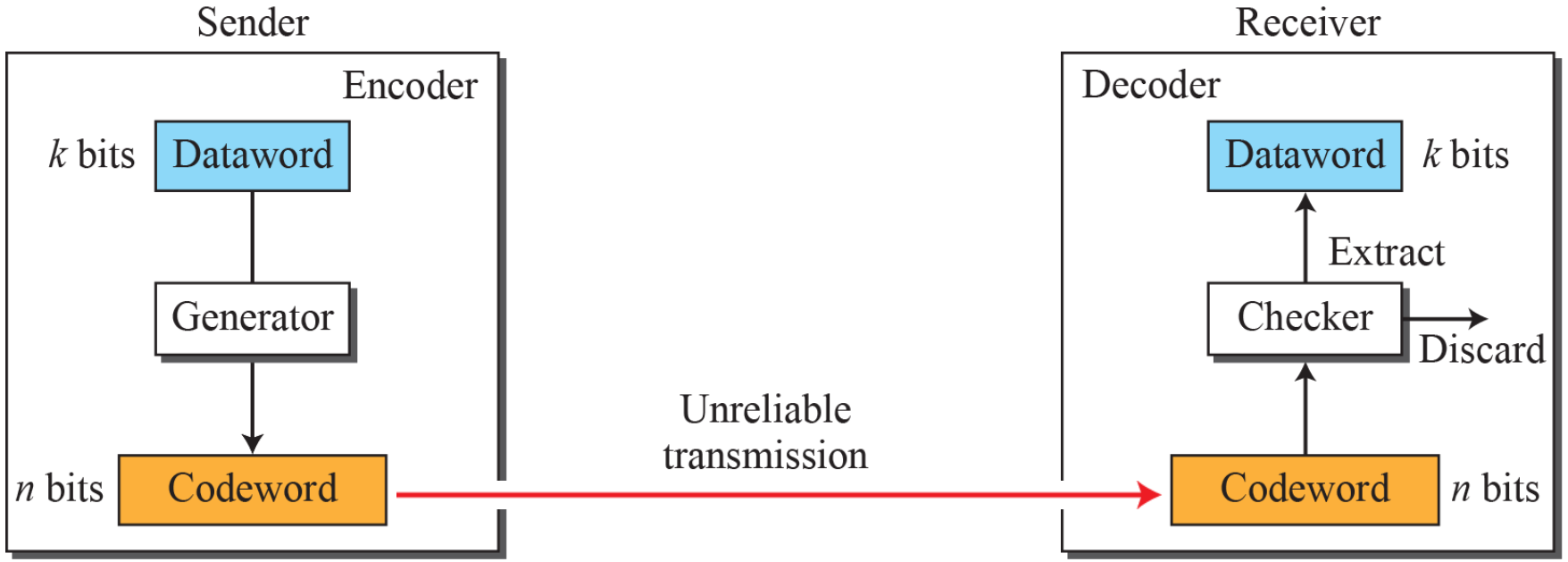
Datawords	*Codewords*	*Datawords*	*Codewords*
00	000	10	101
01	011	11	110

Unused words: 001, 010,100,111

Assume the sender encodes the dataword 01 as 011 and sends it to the receiver. Consider the following cases:

1. The receiver receives 011. It is a valid codeword. The receiver extracts the dataword 01 from it.
2. The codeword is corrupted during transmission, and 111 is received (the leftmost bit is corrupted). This is not a valid codeword and is discarded.
3. The codeword is corrupted during transmission, and 000 is received (the right two bits are corrupted). This is a valid codeword. The receiver incorrectly extracts the dataword 00. Two corrupted bits have made the error undetectable.

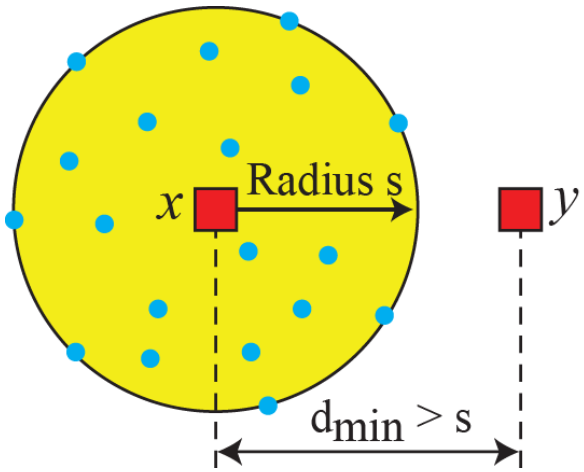
Process of error detection in block coding

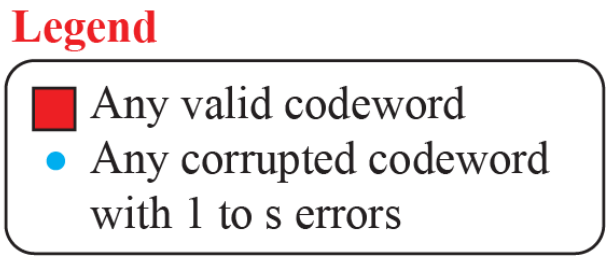


Geometric concept of d_{min} in error detection

Minimum Hamming distance d_{min} : the minimum Hamming distance between all possible pairs of codewords

To guarantee the detection of up to s errors in all cases, the minimum Hamming distance in a block code must be $d_{min}=s+1$





Example

The minimum Hamming distance in the example scheme is 2.
This code guarantees detection of only a single error. For example, if the third codeword (101) is sent and one error occurs, the received codeword does not match any valid codeword. If two errors occur, however, the received codeword may match a valid codeword and the errors are not detected.

A code for error detection

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

Unused words: 001, 010,100,111

Example

A code scheme has a Hamming distance $d_{min} = 4$. This code guarantees the detection of up to three errors ($d = s + 1$ or $s = 3$).