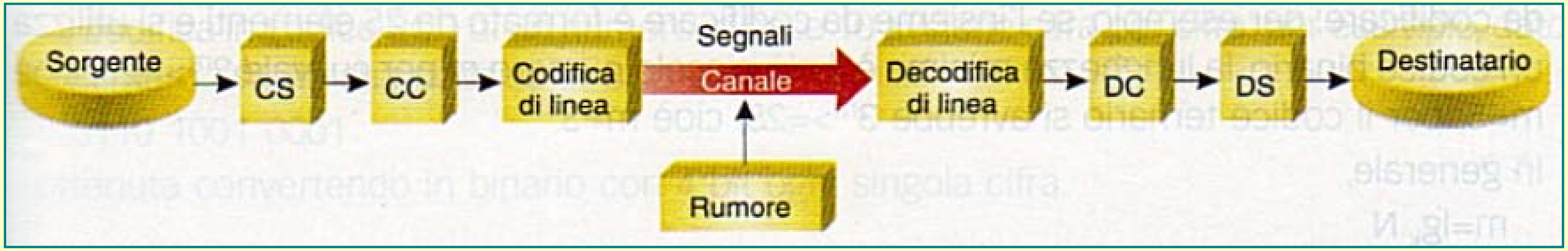
I codici rilevatori e correttori di errori

Modello di un processo di comunicazione



In una trasmissione dati la stazione **mittente** aggiunge ai dati da trasmettere un **codice di controllo** per riconoscere se si sono verificati degli **errori durante la trasmissione**.

La stazione **ricevente** usa questo codice per **individuare ed eventualmente correggere** l'errore.

I codici rilevatori permettono soltanto di capire che i dati non sono corretti.

I codici correttori permettono non solo di capire che si è verificato un errore, ma anche di individuare la posizione dell'errore e quindi di correggerlo.

I **codici correttori** richiedono **molti più bit** dei **codici rilevatori** e quindi sprecano ampiezza di banda. Per questo di solito si usano codici rilevatori.

• Codici ridondanti

Per consentire la **rilevazione** e la **correzione di errori**

si ricorre frequentemente a **codici ridondanti**

ovvero codici che **utilizzano un numero maggiore di bit rispetto al numero strettamente necessario** per codificare le informazioni dell'insieme sorgente.

Ad esempio:

m bit di dati (**l'informazione da trasmettere**)

r bit di controllo (**bit ridondanti**)

→ ciascuna parola del codice utilizza

n = m + r bit

101001010110 Il codice **non è efficiente**: il 33% dei bit sono ridondanti

Efficienza e ridondanza

Un codice: si dice **efficiente** quando utilizza il numero di simboli ***strettamente necessario*** per codificare l'informazione.

si dice **ridondante** quando usa un numero di simboli ***superiore a quello necessario*** anche se utili per qualche scopo.

Se le parole di un codice $\mathbf{C_1}$ hanno ciascuna una lunghezza di **m bit** (considerate *informazione pura*)

e le associamo

alle parole di un altro codice $\mathbf{C_2}$ che ha parole di lunghezza **n bit**

dove **m < n**

per il codice $\mathbf{C_2}$ si definiscono le seguenti quantità:

Efficienza	: $E = m / n$
Ridondanza	: $R = 1 - E$

Si può ossevare che nell'ipotesi **m<n** è sempre **E<1.**
Inoltre, ***più n è piccolo*** **minore** è la ridondanza e **maggiore** è l'efficienza.

Codici ridondanti vengono usati per esempio:

- nei **sistemi di comunicazione**
- nelle **operazioni di memorizzazione dei dati**.

- Nei **sistemi di comunicazione** ciò permette di verificare se le informazioni sono state trasmesse correttamente.
  Se viene usato un codice rilevatore, in caso di errore bisogna chiedere al mittente di ritrasmettere le informazioni.

- In una **operazione di memorizzazione** l'errore si può verificare al momento della **scrittura** o della successiva **lettura** dei dati.
  Se viene rilevato un errore si può ritentare la lettura, ma se l'errore si è verificato in scrittura il problema si può risolvere solo se è stato usato un codice correttore.

La cifra di controllo nell'ISBN-13

Il relativo algoritmo numerico è il seguente:

- Partendo da sinistra, **si moltiplica ogni cifra per un peso** in base alla posizione della cifra stessa: la prima cifra per **1 ,** la seconda per **3** la terza per **1** e così via…..
- **si sommano i risultati**
- **si divide la somma per 10** e si prende il **resto della divisione**
- **si sottrae il risultato da 10**: **questa cifra è la cifra di controllo**

Esempio: dato un codice **978-88-430-2534 ?**

$$9\cdot1+7\cdot3+8\cdot1+8\cdot3+8\cdot1+4\cdot3+3\cdot1+0\cdot3+2\cdot1+5\cdot3+3\cdot1+4\cdot3=117$$

$$\frac{117}{10}=11$$ col resto di 7

$$10-7=3$$ quindi **3** è la 13-esima cifra da aggiungere

ISBN 978-88-430-2534 3

La **prova della correttezza** di un codice ISBN-13 si ottiene **rieseguendo il calcolo iniziale**, stavolta con tutte e 13 le cifre. Il risultato dell'addizione dovrà essere divisibile per 10.

$$9\cdot1+7\cdot3+8\cdot1+8\cdot3+8\cdot1+4\cdot3+3\cdot1+0\cdot3+2\cdot1+5\cdot3+3\cdot1+4\cdot3+3\cdot1=120$$

• Distanza di Hamming

Il numero di bit diversi tra due parole di codice viene detto **distanza di Hamming.**

Si può calcolare il numero di bit diversi facendo l'**or *esclusivo*** delle due stringhe e contando il numero di bit 1 del risultato.

Se due parole hanno distanza ***d*** significa che servono ***d*** **errori** per trasformare una nell'altra.

es.: le stringhe: **1011011** **1010010** hanno **distanza 2**
perché hanno solo 2 bit diversi

Calcolo della distanza di Hamming di un codice

Un codice è costituito da un insieme di parole "legali".

- **Per ogni coppia di parole** del codice **si calcola la distanza di Hamming**
- La **<u>distanza minima</u>** **D** tra le parole legali del codice è

la **distanza di Hamming del codice.**

Esempio: codice con N=4 parole legali.
Il numero delle possibili coppie è dato dalle **combinazioni semplici** di N elementi presi a K a K (nel nostro caso K=2)

$$\binom{N}{K} = \frac{N!}{K!(N-K)!}$$

11010101
10110101 2
01001010 6
01010001 2

8
4
4

La distanza di Hamming del codice è D=2

La distanza di Hamming indica quanti errori si possono **rilevare** e quanti se ne possono **correggere**.

Teorema 1: condizione necessaria e sufficiente affinché un codice sia in grado di **rilevare** ***k*** **errori** è che la distanza di Hamming del codice sia:

$$D = k+1;$$

Teorema 2: condizione necessaria e sufficiente affinché un codice sia in grado di **correggere** ***k*** **errori** è che la distanza di Hamming del codice sia:

$$D = 2k+1;$$

Codici rilevatori

- **Bit di parità semplice** (pari o dispari)
- **Codice CRC** (a ridondanza ciclica)

Codici correttori

- **Bit di parità incrociata** (trasversale e longitudinale)
- **Codice di Hamming**