Distanza di Hamming

Con due CODEWORD è possibile definire una distanza formale in termini di Differenze tra singoli BIT. (ex: Se le due Codeword sono a distanza due dobbiamo CAMBIARE due bit per passare dall'una all'altra).

Date due codeword è possibile definire una distanza formale in termini di differenze tra i singoli bit. Se le due codeword sono a distanza 2 è necessario cambiare due bit per passare dall'una all'altra.

Il canale di trasmissione fisica "rovina il messaggio", ovvero introduce errori di trasmissione nella sequenza inviata. Normalmente in un byte di dati è più probabile che sia un solo bit ad essere modificato rispetto a 2 o più bit.
Il controllo di parità rileva gli errori in numero dispari , ovvero se c'è un errore in un solo bit lo rileva ma non su due, ed è poco probabile che ci siano 2 o più errori, per tanto funziona abbastanza bene.

Keyword 3 parità pari sul bit 1

000	001
011	010
101	100
110	111

distanza del vocabolario = 2

$$P(e=1) \gg P(e>1)$$

	0	3	5	6
0	0	2	2	2
3	2	0	2	2
5	2	2	0	2
6	2	2	2	0

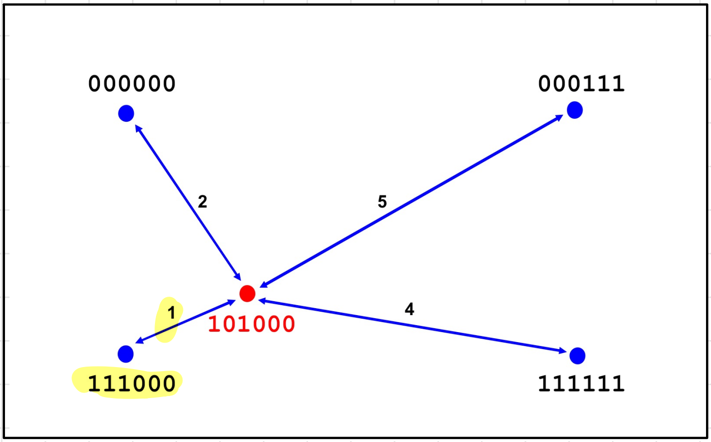
=> minimo = 2

Il trasmettitore invia di certo l'informazione corretta (combinazioni di sinistra, compreso bit di parità in verde), tuttavia il ricevitore potrebbe ricevere informazioni errate (combinazioni in rosso) per via di errori nel canale di trasmissione. Il destinatario conosce le combinazioni errate e le scarta in caso di ricezione.

Nella ricezione della risposta si traccia la possibile sorgente dell'errore valutando, tra quelle possibili, quale di queste ha distanza minore da quella ricevuta, in quanto il canale è probabile che abbia cambiato meno bit possibile.
Il mittente stesso corregge l'errore sulla base della sequenza più vicina.

Il destinatario sa quali sono le combinazioni errate e le scarta;

- Nella ricezione della RISPOSTA si traccia la possibile sorgente dell'errore, valutando quale è a distanza MINORE. (SARÀ IL CANALE CHE AVRA CAMBIATO MENO BIT)



- il mittente corregerà l'errore

Il vocabolario con controllo di parità ha sempre distanza = 2.
La regola generale dice che :

- **Per rilevare e errori è necessario un vocabolario con distanza e + 1.**
- **Per correggere e errori è necessario un vocabolario a distanza 2e + 1**

Rilevare C errori → VOCABOLARIO CON DISTANZA C+1

Correggere C errori → VOCABOLARIO CON DISTANZA 2C+1

Ridondanza

La ridondanza è funzione dei dati già noti e non presenta informazioni aggiuntive rispetto a quelle già note.
Avendo **m** bit di dati, quanti bit **r** di ridondanza devo aggiungere per **rilevare** gli errori singoli ? Serve aggiungere un solo bit di ridondanza, quello di parità.
Bisogna poter calcolare quanti sono gli **r** bit di ridondanza da introdurre per **correggere** gli errori singoli e la formula per il calcolo di **r** è funzione di **m**. Perchè possa correggere errori singoli server un dizionario a distanza d=3.

Consideriamo un vocabolario con m bit dati e r bit di controllo

Poniamo $m+r = n$

Le combinazioni possibili (con n bit) sono 2^n , di cui solo 2^m sono valide

Consideriamo una codeword valida: 10001

Cambiando un solo bit per volta, possiamo scrivere altre **n** codeword, tutte non valide:

11001

Combinazioni errate ottenibili cambiando n bit della sequenza uno alla volta + Codeword corretta

Se $d=3$, allora $(n+1)\,2^m \leq 2^n$

Semplificando:

$$(m+r+1)\,2^m \leq 2^{m+r}$$

Ricorda

$$(m+r+1) \leq 2^r$$

$$m+1 \leq 2^r - r$$

Nota: L'espressione $m+1 \leq 2^r - r$ è valida **solo** per $d=3$.

Esempi:

$m=8 \Rightarrow r=4 \qquad (8+4+1) \leq 2^4 \qquad 13 < 16$

$m=11 \Rightarrow r=4 \qquad (11+4+1) \leq 2^4 \qquad 16 = 16$

Codici di hamming

Quando passo da un valore di n al successivo introduco un bit di parità (controllo). Tra i valori di n mancano le potenze del due e in corrispondenza di queste inserisco il bit di controllo ottenuto dallo XOR tra i bit dei dati, in particolare si fa lo XOR delle sole posizioni per le quali il bit di parità figura nella espansione binaria dei bit dei dati.

Esempio:

Dati originali 10010001100

xx1x001x0001100

1 2 4 8

Mancano le potenze del 2 e le ottengo facendo lo XOR con gli altri DATI.

$b_1 = 3 \otimes 5 \otimes 7 \otimes 9 \otimes 11 \otimes 13 \otimes 15$

$b_2 = 3 \otimes 6 \otimes 7 \otimes 10 \otimes 11 \otimes 14 \otimes 15$

$b_4 = 5 \otimes 6 \otimes 7 \otimes 12 \otimes 13 \otimes 14 \otimes 15$

$b_8 = 9 \otimes 10 \otimes 11 \otimes 12 \otimes 13 \otimes 14 \otimes 15$

$3 = 1+2$

$5 = 1 \quad +4$

$6 =$ non vi è 1 $\; 2+4$

$7 = 1+2+4$

$9 = 1 \quad\quad +8$

$10 = \quad 2 \quad +8$

$11 = 1+2 \quad +8$

$12 = \quad\quad 4+8$

$13 = 1 \quad +4+8$

$14 = \quad 2+4+8$

$15 = 1+2+4+8$

Al verificarsi di un errore nella trasmissione della sequenza vengono a cambiare i valori ottenuti dagli XOR nei bit di controllo e quindi si riesce facilmente ad individuare la posizione esatta del bit in cui si è verificato l'errore.
Se la posizione modificata è una potenza del due allora si ha un errore nella trasmissione del bit di controllo e si può facilmente implementare la correzione di un bit di controllo attraverso un contatore.