

Ergo...

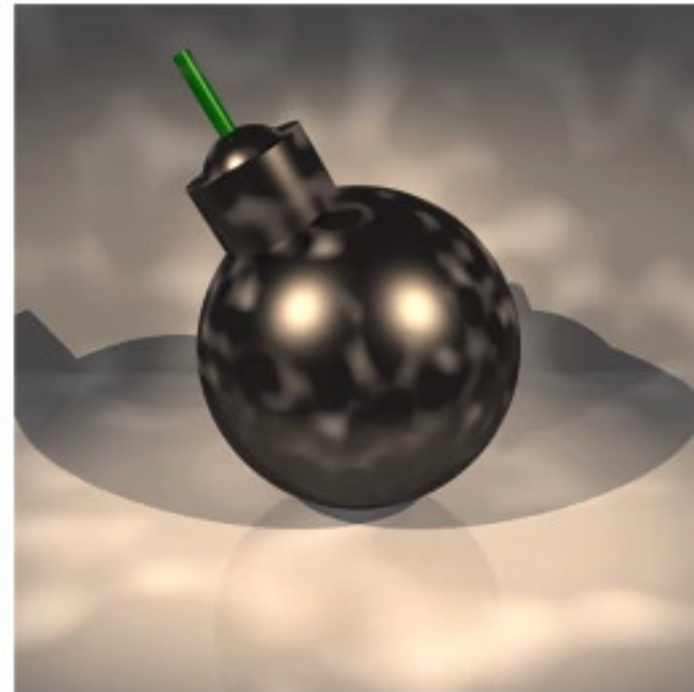
- ◆ Error detection è utile quando il canale ha ***pochi errori*** (è molto affidabile), oppure quando un errore non è ***critico***
- ◆ Ma spesso serve anche fare **error correction**, cioè cercare anche di correggere gli errori

Error correction...

- ◆ Anche perché, nelle reti, quando invece il canale non è molto affidabile, ritrasmettere troppi pacchetti diventerebbe **oneroso**: facendo direttamente **correzione degli errori** sui pacchetti, si evita la ritrasmissione

Error control: distanze

- ◆ Prima di procedere, ci occorre una ***misura del "danno"*** che un errore può fare
- ◆ Gli errori ***meno gravi*** sono quelli che danneggiano ***un*** solo bit del messaggio...
- ◆ ... poi seguono quelli che ne danneggiano ***due***, etc etc



La distanza

- ◆ Diciamo allora che due messaggi (di lunghezza uguale ovviamente) sono ***distanti 1*** se sono diversi solo per 1 bit, ...
- ◆ ***Distanti 2*** se sono diversi per 2 bits, etc etc
- ◆ Questa misura di distanza si chiama **distanza di Hamming**



Le tecniche di error control...

- ◆ Sono quindi ***più o meno potenti*** a seconda di quanta distanza di Hamming riescono a sopportare nei messaggi (cioè, di quant'è la gravità massima dell'errore che riescono a sopportare)



Error detection

- ◆ Vediamo allora cosa si può fare nel caso dell'error detection
- ◆ Ovviamente, dovremo avere ***dell'informazione extra*** rispetto ai dati (come nel caso del framing) che ci permetta stavolta di trovare gli errori



Error detection

- ◆ Avremo quindi un **encoding** (in cui aggiungiamo la protezione dagli errori), e poi una fase di **decoding** dove ci liberiamo dalla protezione dagli errori e teniamo il dato originale



Error detection

- ◆ L'error detection sarà quindi nella fase di decoding, dove il ricevente controllerà se ci sono stati errori
- ◆ Modo semplice? Basta ad esempio verificare se il messaggio arrivato non può essere stato creato con l'encoding (messaggio "illegale")



Ma...

- ◆ Questa è una condizione ***sufficiente***, non ***necessaria***
- ◆ Potrebbero esserci talmente tanti errori da trasformare un messaggio “legale” in un altro messaggio “legale”
- ◆ Quindi, quand’è che siamo sicuri di trovare l’errore?
- ◆ Dobbiamo limitare la potenza dell’errore!



Riformulando...

- ◆ Dato un certo modo di proteggere dagli errori (encoding/decoding) vorremmo sapere: qual'è il massimo errore che riusciamo a trovare?



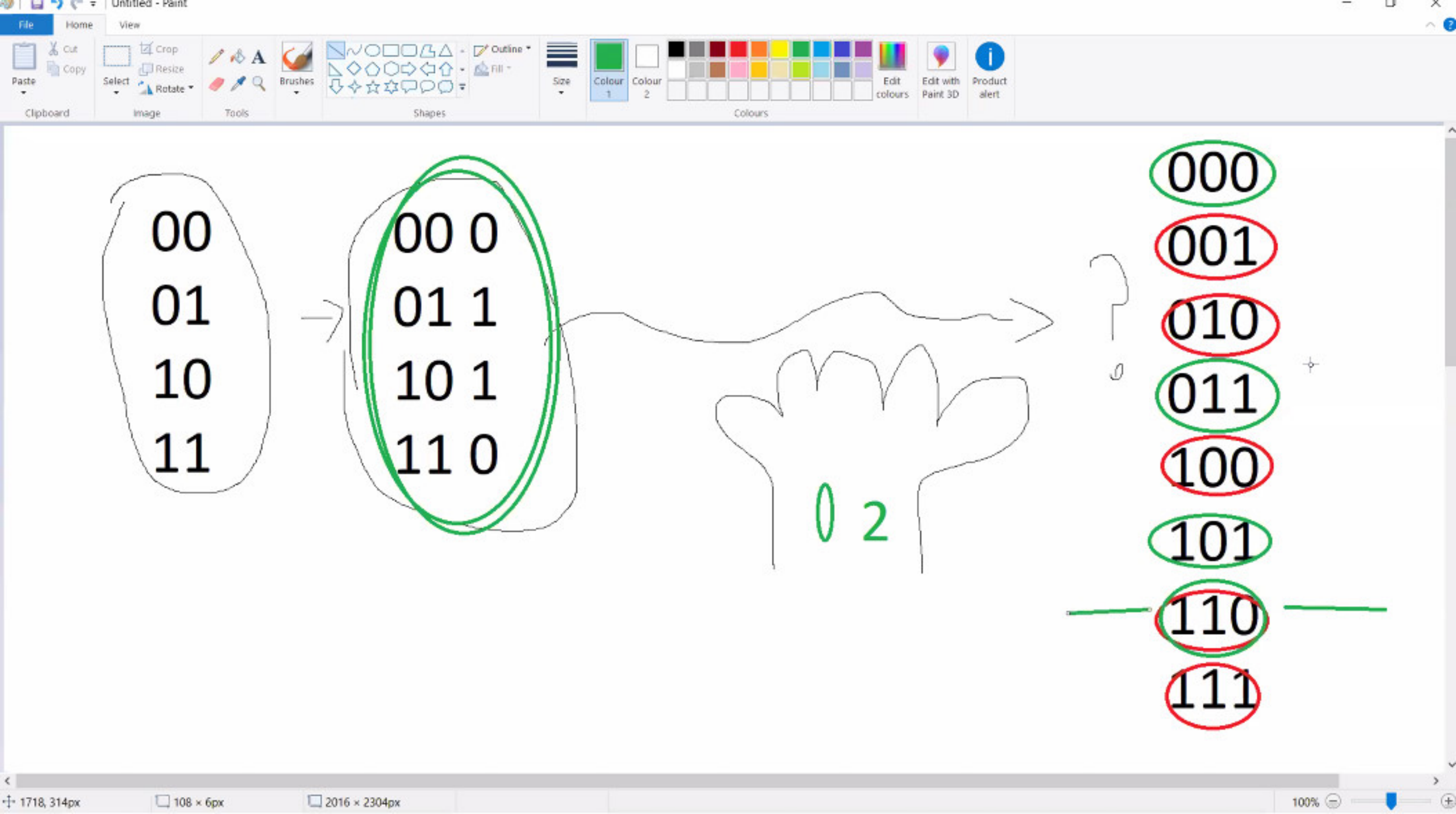


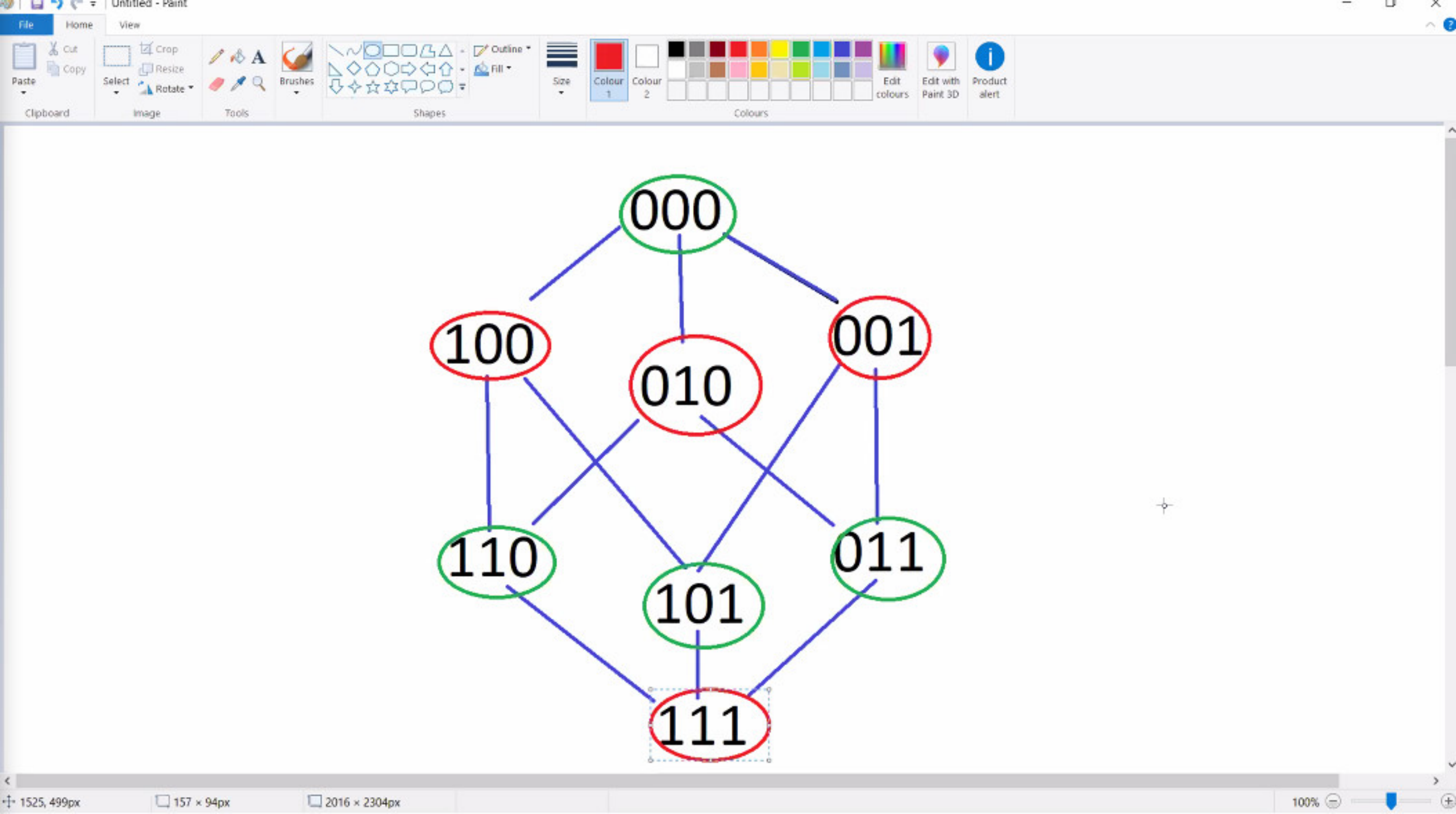
Esempio

- ◆ Una semplice tecnica è quella del bit di parità (**parity bit**):
- ◆ Ogni tot bits (diciamo, **m**), ***inseriamo*** uno 0 o un 1 a seconda che la somma degli m bits precedenti sia ***pari o dispari***
- ◆ $m=2$: 01 \rightarrow 011 10 \rightarrow 101

Disegnamolo...







Quanto è potente il parity bit "m"?



- ◆ Facile vedere che, **qualunque sia l'm:**
- ◆ Presi due messaggi diversi, ***i loro messaggi codificati sono a minimo a distanza 2***
- ◆ → anche se c'è un errore (1 bit), un messaggio codificato non può mai diventare un altro messaggio!