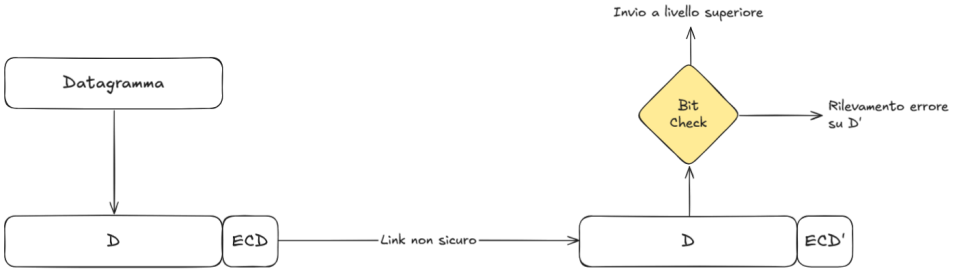
Rilevamento e correzione degli errori

Nel livello di collegamento, viene utilizzato un meccanismo di controllo chiamato **EDC (Error Detection Code)** per verificare l'integrità dei dati trasmessi. I dati trasmessi sono indicati con **D**, mentre il codice di rilevamento degli errori è **EDC**. Dopo la trasmissione, il ricevente ottiene **D'** e **EDC'**, che potrebbero contenere errori.



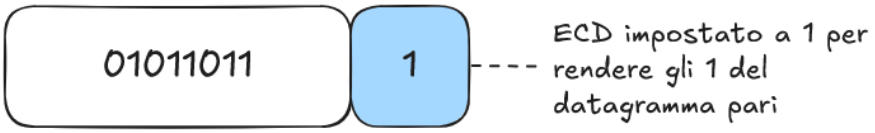
Il destinatario ricalcola **EDC'** sui dati ricevuti **D'** e lo confronta con il valore ricevuto. Se i due valori non corrispondono, viene rilevato un errore e la gestione dell'errore dipende dal protocollo utilizzato. Più è grande ECD più è funzionale l'error detection, ne vediamo due.

Controllo di parità

Il **bit di parità** è un metodo semplice per rilevare errori:

- Se il numero di bit 1 in **D** è **pari**, il bit di parità viene impostato a **0**.
- Se è **dispari**, viene impostato a **1,** così da farlo diventare pari.

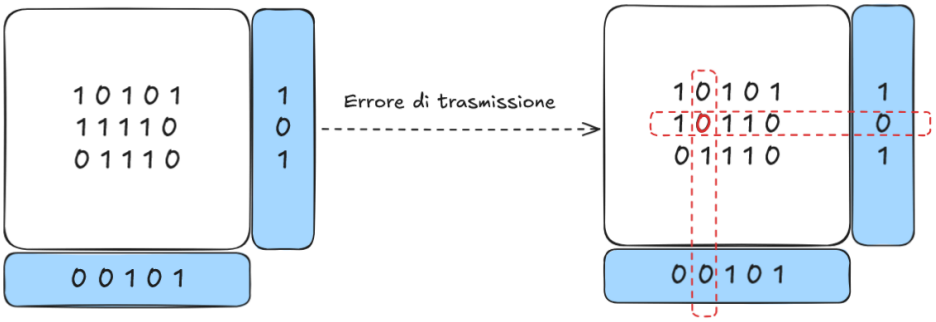
Questo metodo rileva solo errori con un singolo bit modificato, ma non identifica dove si trova.



Parità bidimensionale

Per individuare **quale bit è stato alterato**, si usa una **struttura a matrice**:

- Si calcolano i bit di parità **per riga e per colonna**.
- Se un bit viene alterato, la sua riga e colonna avranno un valore errato di parità.
- Incrociando i due valori errati, si può **identificare e correggere** il bit sbagliato.



Checksum Internet

Il **checksum** viene usato nei protocolli **TCP/UDP** per rilevare errori nei pacchetti. Il calcolo avviene nel **campo header** del pacchetto, si utilizza nei livelli più alti perchè pesante.

Lato mittente

1. I dati del segmento vengono trattati come una sequenza di interi a 16 bit.
2. Si calcola la somma in complemento a uno di tutti i blocchi.
3. Il risultato viene inserito nel campo **checksum** dell'header.

Lato ricevente

1. Ricalcola il checksum con lo stesso metodo del mittente.
2. Se il valore calcolato **non** corrisponde a quello ricevuto → **Errore rilevato**.
3. Se i valori coincidono → **Il pacchetto è considerato corretto** (anche se non è garantito che sia privo di errori).

Il checksum rileva solo alcuni errori (bit invertiti), ma non è infallibile per errori complessi.

Cyclic Redundancy Check (CRC)

Il **CRC** è un metodo più avanzato di EDC, affidabile e utilizzato in protocolli come **Ethernet e Wi-Fi**, capace di rilevare errori multipli.

Componenti principali

- **D**: Dati da trasmettere (visti come un numero binario).
- **G**: Generatore di CRC, un pattern binario noto sia dal mittente che dal destinatario.
- **R**: Resto della divisione binaria, usato per verificare la presenza di errori.
- **r:** grado del generatore ovvero $|G| - 1$.

Lato mittente

1. **Aggiunta di zeri**: Si concatenano **r zeri** ai dati **D**.
2. **Divisione binaria** tra D e G ovvero lo XOR
3. Il resto della divisione è **R**, che viene aggiunto a **D** per formare **<D, R>**.
4. Il frame finale viene trasmesso.

Lato ricevente

1. **Riceve il frame** <D, R>.

2. **Esegue la divisione binaria** con il generatore **G, di nuovo lo XOR**.
3. Se il resto è **zero**, i dati sono corretti.
4. Se il resto è **diverso da zero**, viene rilevato un errore.

Il CRC può rilevare fino a $|G|$ **bit cambiati** nella trasmissione ma **dispari**.