

Dati link layer si interfaccia con il livello fisico, in particolare Riceve i segnali del livello fisico, Impacchetta i bit in dei frame di livello DLL. Fornisce un recapito affidabile se richiesto. Gestisce gli errori (non volontari) <sup>PREVENZIONE DI ERRORI - si può fare abbastanza bene</sup> <sup>Correzione degli errori - Non n° sicurezza</sup>

**Compiti del DLL** poiché sono il Liv. Fisico a ricevere i messaggi. Abbiamo due operazioni: **Correzione degli errori** - Non n° sicurezza.

**Data link layer si interfaccia con il livello fisico**, in particolare recepisce i segnali del livello fisico ed impacchetta i bit in dei frame di livello DLL. Fornisce un **recapito affidabile** se richiesto. **Gestisce gli errori** (non volontari) dovuti al canale di trasmissione dovuti al fatto che il livello fisico "rovina" il messaggio. Si possono rilevare e correggere gli errori, solo che mentre la rilevazione si può fare abbastanza bene mentre per la correzione si può al massimo garantire una qualità elevata del servizio ma **non sempre è possibile**.

Consiste nell'incapsulare i dati provenienti dal layer superiore, aggiungendo informazioni necessarie per garantire la corretta trasmissione sulla rete. Considerando che i dati inviati sono **Codificati** e ricevuti correttamente dal destinatario.

**Framing dei dati** **Codifica dei livelli**: Segnali 1 e 0 (1 e 0 comunicazione 0 non c'è) // **Codifica tre livelli e cinque**: Portano più informazioni. <sup>Cinque livelli: Se n° è zero solo  $\frac{1}{5}$  si perde</sup> <sup>Nb: tre livelli:  $\frac{1}{3}$  si perde</sup>

Il "framing dei dati" consiste nell'incapsulare i dati provenienti dal layer superiore in una struttura di frame, aggiungendo informazioni necessarie per garantire la corretta trasmissione dei dati sulla rete.

Il framing dei dati è importante perché permette ai dispositivi di comunicare su una rete di distinguere i dati inviati da un dispositivo da quelli inviati da altri dispositivi, garantendo che i dati inviati siano **codificati** e ricevuti correttamente dal destinatario.

Codifica di due livelli: un segnale ad 1 implica che ci sia una comunicazione mentre a zero significa o stato logico basso oppure assenza di comunicazione. Come si distingue tra 0 e assenza di comunicazione? Si introducono più livelli.

Codifica di tre livelli o cinque livelli: portano più informazioni, infatti avrei più bit per ogni posizione.

**Avere più livelli permette di perdere meno informazioni durante il passaggio da uno stato logico ad un altro in quanto se si verifica un errore su cinque livelli si perderà solo 1/5 dell'informazione, mentre con un errore in un livello nel caso di tre soli livelli si perderebbe 1/3 dell'informazione.**

**Codifica 4b-5B** <sup>Utilizzato per aumentare l'affidabilità della trasmissione e semplificare la sincronizzazione tra mittente e destinatario. (Le tabelle sono concepite da mittente e destinatario)</sup> <sup>Danno Problemi di Sincronizzazione</sup>  
Nella codifica 4B5B, ogni gruppo di 4 bit viene mappato in un gruppo di 5 bit, utilizzando una tabella di codifica predefinita per **ANUNCIARE** l'affidabilità della trasmissione. <sup>Ogni Combinazione di 4 bit viene mappata in un codice da 5 bit. Ci sono 16 Combinazioni di 4 bit, e 32 Codici di 5 bit. Garantisce Corretta Ricezione dei dati (evitando gli 1111)</sup>

La tecnica di codifica 4B5B viene utilizzata a livello fisico di una rete per aumentare l'affidabilità della trasmissione e semplificare la sincronizzazione tra mittente e destinatario. La tabella di codifica 4B5B è conosciuta sia dal mittente che dal destinatario.

**Nella codifica 4B5B, ogni gruppo di 4 bit viene mappato in un gruppo di 5 bit, utilizzando una tabella di codifica predefinita, così da poter aumentare l'affidabilità della trasmissione.** In particolare, ogni possibile combinazione di 4 bit viene mappata in un codice di 5 bit univoco. Questo significa che ci sono un totale di 16 combinazioni di 4 bit e 32 codici di 5 bit. La decodifica 4B5B garantisce una corretta ricezione dei dati, evitando le sequenze 1111 che potrebbero causare problemi di sincronizzazione. **Scartare queste combinazioni tuttavia induce lo spreco di un bit ogni 5.**

<sup>Utilizzato per la trasmissione di dati nelle reti a larga banda come HDMI. La codifica di 8 bit in un simbolo a 10 bit. 2 Byte viene convertito in un simbolo prima della trasmissione.</sup>

**Codifica 8B-10B** <sup>Sviluppata per garantire sulle lunghe sequenze ugual numero di 0 e di 1 in modo da **PERDERE** lo scambio energetico nella trasmissione.</sup>

La codifica 8B10B è una tecnica di codifica utilizzata per la trasmissione di dati su reti a larga banda come HDMI. La tecnica prevede la codifica di ogni gruppo di 8 bit di dati in un simbolo a 10 bit. Ciò significa che ogni byte di dati viene convertito in un simbolo a 10 bit prima della trasmissione. La codifica 8B10B è stata sviluppata per garantire, sulle lunghe sequenze uguale numero di zeri e uno nella codifica.

**La codifica 8B10B utilizza una tabella di codifica predefinita per codificare ogni gruppo di 8 bit di dati in un simbolo a 10 bit.** La tabella di codifica è progettata in modo da garantire che il numero di zeri e uno nella codifica sia bilanciato, in modo da **azzerare lo scambio energetico** nella trasmissione. La codifica con uguale numero di zeri e uno nella sequenza permette un annullamento dello scambio energetico.

**Quantità di informazione** <sup>Potremmo Aumentare la **Quantità** di informazioni grazie all' aumentare del n° di simboli che può avere il segnale. (es: Alto, Alto, Alto) ecc....</sup>

<sup>Si determinano le Assenze di segnale come tutti i simboli allo stato logico medio.</sup>

Usando tre livelli logici (alto, medio, basso) nella trasmissione si possono trasmettere più informazioni grazie alle 2<sup>3</sup> combinazioni possibili. Aumentando il numero di simboli (forme ottenibili con i livelli) si può **aumentare la quantità di informazioni** portate dal segnale, misurata in bit. Ad esempio con 3 simboli si hanno alto-alto-alto, alto-alto-medio, ecc... **Si discrimina assenza di segnale come tutti i simboli allo stato logico medio.**

## Rilevazione e Correzione degli errori

**Rilevazione:** Utilizzo per individuare l'errore e correggerlo direttamente.  
**Correzione:** Consente sia di rilevare che correggere.  
**Entambi:** metodi sono basati sulla presenza di ridondanza nella comunicazione.

La **rilevazione** degli errori consente di individuare la presenza di un errore di trasmissione in una frame, ma non di correggerlo direttamente. Invece, la **correzione** degli errori consente sia di rilevare che di correggere gli errori di trasmissione.

Entrambi i metodi sono basati sulla presenza di **ridondanza** nella comunicazione, ovvero dati che non aggiungono informazioni che non si hanno già.

Queste informazioni ridondanti permettono al destinatario di confrontare i dati ricevuti con la somma di controllo o valore di parità e individuare eventuali errori di trasmissione.  
La **correzione** è più complessa.

La rilevazione degli errori utilizza tecniche come la checksum o il controllo di parità per aggiungere informazioni ridondanti ai dati trasmessi. Queste informazioni ridondanti permettono al destinatario di confrontare i dati ricevuti con la somma di controllo o il valore di parità e individuare eventuali errori di trasmissione.

La correzione degli errori è più complessa e richiede più risorse di elaborazione rispetto alla rilevazione degli errori.

- **Controllo di parità** Se i due bit non corrispondono allora si farà mandare nuovamente il segnale o utilizzare la politica di correzione. (SARABICE MA PUO' CORREGGERE ERRORI DI UN SOLO BIT)  
Viene aggiunto un bit di parità ogni byte trasmesso. Durante la ricezione, il destinatario calcola il bit di parità dei dati ricevuti e lo confronta con quello che gli è stato spedito.

Il controllo di parità è una tecnica di rilevazione degli errori utilizzata nelle trasmissioni seriali di dati. In questa tecnica, viene aggiunto un bit di parità alla fine di ogni byte di dati trasmesso. Il bit di parità viene calcolato in modo da garantire che il numero totale di bit di valore 1 nel byte di dati, compreso il bit di parità, sia pari o dispari.

Durante la ricezione, il destinatario calcola il bit di parità del byte di dati ricevuto e lo confronta con il bit di parità trasmesso. Se i due bit di parità non corrispondono, significa che si è verificato un errore di trasmissione nel byte di dati. In questo caso, il destinatario può richiedere una ritrasmissione del byte di dati o adottare altre tecniche di correzione degli errori.

Il controllo di parità è una tecnica relativamente semplice e **facile da implementare**, ma ha alcuni limiti. In particolare, il controllo di parità può individuare solo gli errori a singolo bit. Se ci sono più di due bit errati nel byte di dati trasmesso, il controllo di parità non sarà in grado di individuare l'errore.

- **Parità a due dimensioni** Utilizzato per verificare la presenza di errori all'interno della matrice (dove vengono disposti i nostri bit ricevuti). Con questo metodo si può ritrovare la posizione esatta dell'errore poiché controllando il bit di parità all'interno di righe e colonne, si verifica la corretta corrispondenza, se risulta incostante nell'intersezione, lì c'è l'errore.  
La correzione degli errori si basa sulla probabilità che l'errore sia uno.

Il controllo di parità può essere utilizzato anche per controllare sequenze di bit in matrici. In questo caso, vengono utilizzati un controllo di parità in riga e uno in colonna per individuare e correggere eventuali errori nella matrice.

Durante la ricezione, il destinatario calcola sia il bit di parità della riga che quello della colonna del byte di dati ricevuto e li confronta con i bit di parità trasmessi. Se uno dei due bit di parità non corrisponde, significa che si è verificato un errore nella riga o nella colonna del byte di dati. Il destinatario può quindi individuare la posizione esatta dell'errore, che si trova all'intersezione della riga e della colonna errate.

Tuttavia la correzione degli errori si basa sulla **probabilità che l'errore sia solo uno**, pertanto si necessita di massimizzare la probabilità che ci sia un solo errore.

- **CRC (codice di ridondanza ciclica)**

La tecnica CRC prevede la generazione di un **polinomio di grado  $n$** , dove  **$n$**  è il numero di bit della sequenza di dati.

Si prende il polinomio  **$M(x)$**  ottenuto dalla sequenza  **$m$**  da trasmettere che presenta una ridondanza di  **$r$**  bit. Si prende il resto della divisione tra il polinomio  **$M(x)$**  ed un generatore  **$G(x)$** , noto a mittente e destinatario, e lo si moltiplica per un polinomio di grado  **$r$** .

Durante la trasmissione, il valore di controllo CRC viene trasmesso insieme ai dati. Il destinatario riceve i dati e il valore di controllo CRC e utilizza lo stesso polinomio generato dal mittente per calcolare il valore di controllo CRC sui dati ricevuti. Se il valore di controllo CRC calcolato dal destinatario corrisponde al valore trasmesso dal mittente, allora i dati sono stati trasmessi correttamente. In caso contrario, significa che si è verificato un errore di trasmissione. Il generatore  **$G(x)$**  deve avere alcune caratteristiche, ovvero deve avere il primo e l'ultimo bit posti ad 1 e non deve essere ottenibile come prodotto di altri polinomi.

Il CRC è un controllo semplice da implementare ed efficace dal punto di vista della rilevazione.

## Somma

1 0 0 1 1 0 0 1 0

$$x^8 + x^5 + \cancel{x^4} + x$$

+

1 0 1 1 1 0 1

$$x^6 + \cancel{x^4} + x^3 + x^2 + 1$$

⇓

$$x^8 + x^6 + x^5 + x^3 + x^2 + x + 1$$

## Prodotto

1 0 0 1 1 0 0 1 0

$$x^8 + x^5 + x^4 + x$$

•

1 0 0 1 1 0 1 1

$$x^7 + x^4 + x^3 + x + 1$$

$$\Rightarrow x^{15} + \cancel{x^{12}} + \cancel{x^{11}} + x^9 + \cancel{x^8} + \cancel{x^{12}} + x^9 + \cancel{x^8} + x^6 + x^5 + \cancel{x^{11}} + \cancel{x^8} + x^7 + \cancel{x^5} + x^4 + \cancel{x^8} + \cancel{x^5} + x^4 + x^2 + x$$

$$\Rightarrow x^{15} + x^7 + x^6 + x^5 + x^2 + x$$

# Calcolare il CRC

- Inizia per 1 e finisce per 1 //
  - Non deve essere ottenibile attraverso il prodotto di altri Polinomi.
- generatore  $\Rightarrow 11101$

## RIDONDA N2A

1 1 0 0 0 0 1 1 1 0 1 0 1 1 0 1 0 0 1 0 0 0 1 1 1 1 0 0 0 0

1 1 1 0 1

0 0 1 0 1 0 1

1 1 1 0 1

0 1 0 0 0 1

1 1 1 0 1

0 1 1 0 0 0 1

1 1 1 0 1

1 1 1 0 1

1 1 1 0 1

0 0 0 0 0  $\Rightarrow$  CRC ✓

$x \neq$  da 0 0 0 0 0  $\Rightarrow$  CRC X