**Data-Link Lv2**

Il ***Data Link Layer*** (anche ***livello di collegamento dati***, o più semplicemente: ***livello 2***) ha la funzione principale di fornire allo strato di rete servizi per il recapito di dati al nodo direttamente adiacente sulla rete e potrebbe fornire i mezzi per rilevare e possibilmente ***correggere errori*** che possono verificarsi nel livello fisico.

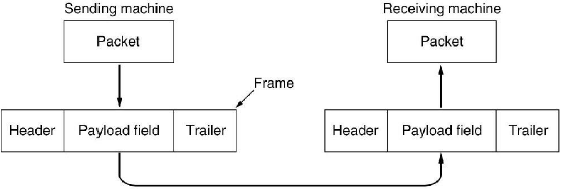
Il compito del data link layer è quindi quello di ***organizzare*** il trasferimento dei dati tra ***due apparati adiacenti***, e di fornire una interfaccia definita per consentire allo strato di rete di ***accedere*** ai ***servizi*** offerti.

Per realizzare le sue funzioni il data link layer:

* Ricevei dati dallo strato di rete (***pacchetti***);
* Li organizza in trame(***frame***) eventualmente ***spezzando*** in più frame il blocco di dati ricevuto dal livello3;
* Aggiunge ad ogni ***frame*** una intestazione ed una coda (***header*** e ***trailer***), e passa il tutto allo ***strato fisico*** per la trasmissione.

In ricezione il data link layer:

* Ricevei dati dallo strato fisico;
* Effettua i controlli necessari, elimina ***header*** e ***trailer***, ***ricombina i*** ***frame*** e passa i dati ricevuti allo ***strato di rete***.



***Servizi del livello di link:***

**Framing**

Il termine ***framing*** fa riferimento alle seguenti operazioni:

* ***Incapsulamento*** dei dati con una intestazione (***header***) e una eventuale coda (***trailer***).
* ***Interpretazione*** dei bit presenti nelle intestazioni (ed eventualmente nelle code).

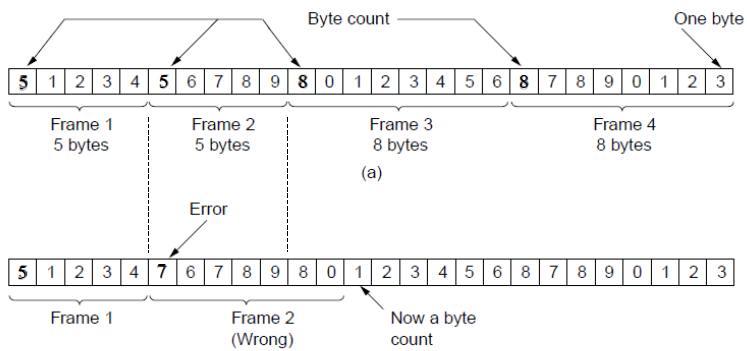
Al fine di fornire servizi al ***livello di rete***, il ***livello data link*** deve usufruire dei servizi forniti dal ***livello fisico***.

Lo strato fisico non può ***garantire*** il trasferimento ***privo di errori***, che dovranno essere gestiti dal ***DLL*** (librerie caricate dinamicamente in fase di esecuzione), che organizza i bit in ***frame***, ed effettua i controlli ***per ogni frame***.

L'approccio del ***livello data link*** è quello di dividere il flusso dei bit in ***frame*** (o ***pacchetti***), esistono 3 metodi:

1. ***Conteggio dei caratteri:***

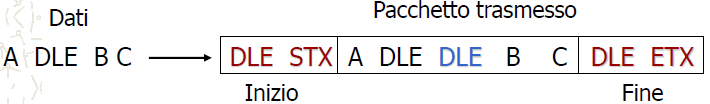
*Un campo dell’intestazione indica il numero di caratteri nel pacchetto*. Se si perde il sincronismo non si riesce a trovare l’inizio di un pacchetto successivo.



1. ***Carattere di inizio e fine:***

I pacchetti sono iniziati dai caratteri ASCII ***DLE*** (***Data*** ***Link*** ***Escape***) e ***STX*** (***Start*** ***of*** ***TeXt***) e terminati da DLE ***ETX*** (***End*** ***of*** ***TeXt***). Ci si può sincronizzare nuovamente cercando la sequenza ***DLE STX.***

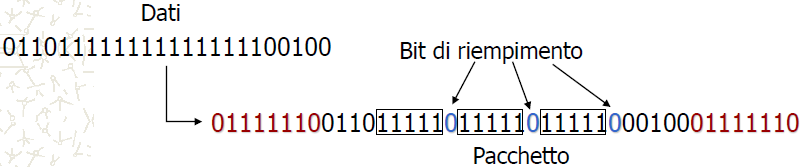
I dati nel pacchetto non possono contenere queste due sequenze. In trasmissione si duplica ogni ***DLE*** nei dati che poi si elimina in ricezione. Un ***STX*** o ***ETX*** preceduto da ***due*** ***DLE*** è un dato del pacchetto.



1. ***Indicatore (flag) di inizio e fine:***

I pacchetti sono iniziati e terminati con una sequenza speciale di bit. ***Flag byte*** = **01111110**

Per evitare che il ***flag*** ***byte*** possa trovarsi all’interno dei dati del pacchetto, viene inserito un bit 0 dopo ogni gruppo di 5 bit a 1. Il bit inserito viene eliminato in ricezione (***riempimento di bit***).



**Rilevazione degli errori**

Gli errori sono causati dall’attenuazione del segnale e da rumore elettromagnetico.

Il controllo dell’errore si basa su ***codici di ridondanza***, che aggiungono bit alla parola dati per verificarne la correttezza. Tali codici si suddividono in:

***codici rilevatori***: in grado unicamente di rilevare la presenza o meno di errori nel ***frame***, ma non la loro posizione. In questo caso il ricevente può chiedere la ritrasmissione del messaggio.

***codici correttori***: in grado di rilevare una o più posizioni errate nel ***frame*** e quindi di correggerle per semplice inversione del bit. Vi sono 3 principali codici a rilevazione d'errore , diversi tra loro:

1. ***Checksum:***

Per la rilevazione di tali errori, nell`header di ogni trama il Lv 2 inserisce un campo denominato ***checksum.*** Questo campo è il risultato di un ***calcolo*** fatto utilizzando i ***bit della trama*** (somma in algebra modulo 2, XOR, dei codici delle parole). La destinazione ripete il calcolo e confronta il risultato col ***checksum*** nell`header, se coincidono allora la ***trama*** è corretta.



La ***Checksum*** è una delle tecniche di rilevazione errori maggiormente utilizzata per trasmissione a breve distanza.

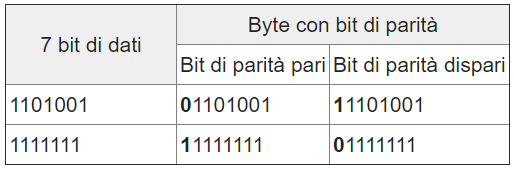
L'obiettivo è quello di ottenere la più alta possibilità di rilevare errori con la minor ridondanza introdotta.

1. ***Controllo di parità:***

Tale sistema prevede l'aggiunta di un ***bit*** ***ridondante*** ai dati, calcolato a seconda che il numero di bit che valgono 1 sia pari o dispari. Ci sono due varianti del bit di parità: **bit di parità pari** e **bit di parità dispari**.

Quando si usa un ***bit di parità pari***, si pone tale bit uguale a 1 se il numero di "1" in un certo insieme di bit è dispari (facendo diventare il numero totale di "1", incluso il bit di parità, pari).

Quando invece si usa un ***bit di parità dispari***, si pone tale bit uguale a 1 se il numero di "1" in un certo insieme di bit è pari (facendo diventare il numero totale di "1", incluso il bit di parità, dispari).



1. ***Codici di ridondanza ciclica:***

Gli ***n bit*** del blocco da ***trasmettere*** vengono considerati come ***coefficienti*** ***di un*** ***polinomio*** ***di grado n-1*** nella variabile x. Tale polinomio, che chiameremo ***M(x),*** viene poi diviso per un altro polinomio fissato dalle convenzioni internazionali, chiamato **polinomio generatore** **G(x)**, le cui caratteristiche sono:

* è sempre di grado ***inferiore al polinomio M(x)*** da tra smettere;
* ha sempre il coefficiente del ***termine x^0 uguale a 1***.

In trasmissione, insieme al blocco di bit che costituisce ***M(x),*** viene anche mandato il blocco di controllo ***R(x),*** ottenuto ***dividendo M(x) per G(x)*** con le regole di divisione modulo 2 (effettuando lo XOR delle due stringhe di bit).

Le cifre di controllo calcolate vengono dette FCS (Frame Check Sequence) o ***CRC*** (***Cyclic Redundancy Check***). Per rilevare la presenza di un errore il ricevitore divide il messaggio ricevuto ***per*** ***G(x)*** e verifica che il ***resto*** sia ***nullo***. Se non lo è il ricevitore deve chiedere la ripetizione del messaggio.

**Controllo di flusso**

Può capitare che una ***sorgente*** sia in grado di trasmettere ad un tasso ***più alto*** della capacità di ***ricevere*** a destinazione. Senza controllo, questo implica che la destinazione inizierebbe a ***scartare frame trasmessi*** correttamente per ***mancanza di risorse***(tempo di processamento, ***buffer***).

In ricezione, il data link layer verrà ***svegliato*** per ***prelevare*** ***dati*** allo ***strato*** ***fisico***, processarli, e passarli allo ***strato*** ***di*** ***rete.***

Il protocollo deve poter gestire questa situazione e prevedere ***meccanismi*** per ***rallentare*** la trasmissione.

Tipicamente il protocollo prevederà dei frame di controllo con cui il ricevente può ***inibire e riabilitare*** la trasmissione di ***frame***, cioè il protocollo stabilisce ***quando*** il trasmittente può inviare frame.

Un semplice meccanismo può essere quello di ***valutare*** i ***tempi*** ***di*** ***risposta*** del ricevente, ed inserire dei ***ritardi*** nel processo di trasmissione per adattarlo alla capacità di ricezione.

**Il frame data-link**

Prevede un’intestazione (header) e una coda (trailer) aggiunti al pacchetto passato dal livello di rete



Le informazioni di framing e di checksum sono gestite in hardware. La presenza di campi di controllo dipende dal protocollo di comunicazione utilizzato nel livello data link

* + Tipo del pacchetto (***type***) (es. data, ack, nack )
  + Numero di sequenza del pacchetto (***seq***)
  + Numero di riscontro (***ack***)

**RDT – Trasferimento affidabile**

Tecnica per controllare se i dati arrivano a buon fine o meno. Per rilevare gli errori si usa:

* ***notifica*** ***positiva*** (***ACK***): il ricevente comunica espressamente al mittente che il pacchetto ricevuto è corretto
* ***notifica negativa*** (***NAK***): il ricevente comunica espressamente al mittente che il pacchetto contiene errori
* il mittente ***ritrasmette*** il pacchetto se riceve un ***NAK***

**Protocollo stop-and-wait:**

Il ***protocollo stop-and-wait*** prevede che A, dopo aver inviato il frame, si ***fermi*** per attendere un ***riscontro.***

B, una volta ***ricevuto il frame***, invierà ad A un ***frame di controllo***, cioè un frame privo di dati, allo scopo di avvisare A che ***può trasmettere*** un nuovo frame.

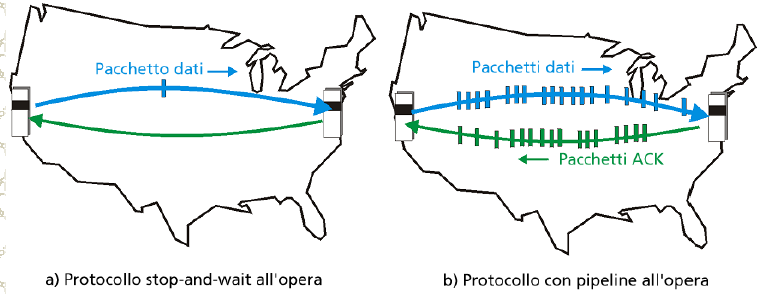
Il frame di riscontro si indica generalmente con il termine ***ACK*** (***ACKnowledge***) o ***RR*** (***Receiver*** ***Ready***). Se i pacchetti contenenti ACK/NAK si danneggiano, bisogna fare il checksum a ACK/NAK.

Il ***Piggy backing*** è la pratica di mandare ***l'ACK*** di un frame ricevuto insieme al prossimo frame da inviare, e non in un frame contenente solo il flag. Si risparmia banda, ma è utile solo per i canali ***full duplex*** e solo se il nuovo pacchetto da inviare non si fa attendere troppo.

**Protocolli con Pipeline:**

Il mittente ammette più pacchetti in transito, ancora da notificare

* l’intervallo dei numeri di sequenza deve essere incrementato
* buffering dei pacchetti presso il mittente e/o ricevente



Esistono 2 forme generiche di ***protocolli*** con ***pipeline***:

1. **Go Back N**

Per migliorare l’efficienza della trasmissione, e quindi sfruttare al meglio l’intero canale, occorre spedire più frame prima di fermarsi ed aspettare un riscontro.

Questo protocollo spedisce ***più di un frame*** prima di ricevere un ***riscontro cumulativo***. Mantiene una copia del frame spediti fino all’arrivo di un riscontro. I frame vengono numerati per essere identificati dal destinatario. Il funzionamento di questo protocollo si basa su concetto di ***finestra scorrevole***. Questo protocollo necessità di maggiori risorse di ***buffer***.

Ad ogni riscontro ricevuto vengono liberati i ***buffer*** relativi ai ***frame*** riscontrati per occuparli con nuovi frame trasmessi.

1. **SELECTIVE REJECT (Ripetizione selettiva)**

Il ricevente invia riscontri specifici per tutti i pacchetti ricevuti correttamente