**Data Link Layer**

* Il Data Link Layer (livello di collegamento dati: livello 2) ha la funzione principale di fornire allo strato di rete servizi per il recapito di dati al nodo direttamente adiacente sulla rete.
* Il compito del data link layer e’ 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 riceve i dati dallo strato di rete (pacchetti), li organizza in trame (frame) eventualmente spezzando in piu’ frame il blocco di dati ricevuto dal livello 3, 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 riceve i dati dallo strato fisico, effettua i controlli necessari, elimina header e trailer, ricombina i frame e passa i dati ricevuti allo strato di rete.

Per poter svolgere le sue funzioni il data link layer dovra’ curare i seguenti aspetti:

* la organizzazione del flusso di bit in frame, con controllo per la sincronizzazione, inserimento e rimozione di header e trailer, riordinamento dei frame in ricezione.
* organizzare il trasferimento dei dati in modo da gestire eventuali errori di trasmissione, utilizzando codici di correzione degli errori o codici di identificazione degli errori e gestendo la ritrasmissione dei frame errati.
* realizzare il controllo di flusso, per utilizzare in modo efficiente il canale trasmissivo impedendo al contempo ad un trasmettitore veloce di sovraccaricare un ricevitore lento.

**Controllo di flusso**

* Puo’ capitare che una sorgente sia in grado di trasmettere ad un tasso piu’ alto della capacita’ 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).
* Il protocollo prevedera’ dei frame di controllo con cui il ricevente puo’ inibire e riabilitare la trasmissione di frame, cioe’ il protocollo stabilisce quando il trasmittente puo’ inviare frame.

**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, inviera’ ad A un frame di controllo, cioe’ un frame privo di dati, allo scopo di avvisare A che puo’ trasmettere un nuovo frame
* Il frame di riscontro di indica generalmente con il termine ACK (ACKnowledge) o RR (Receiver Ready)
* Va osservato che il traffico dati e’ simplex, ma i frame devono viaggiare nelle due direzioni, quindi il canale fisico deve essere almeno half-duplex

**Protocollo con timeout**

* Il numero di sequenza (seq) impedisce che il ricevente accetti un pacchetto duplicato (inviato in caso di perdita dell’ack)
* Nel caso di ricezione con errore si aspetta lo scadere del timeout
* Protocollo PAR (Positive Acnowledgement with Retransmission) o anche ARQ (Automatic Repeat reQuest)
* Il tempo di timeout non deve essere troppo corto per evitare continue ritrasmissioni
* Il protocollo può fallire se un ack ritarda ad arrivare e nel frattempo scade il timeout e si ritrasmette (si perde il sincronismo dell’ack). Dipende dal fatto che l’ack è anonimo (senza seq)

**Protocolli a finestra scorrevole**

* Questo tipo di protocolli necessita’ di maggiori risorse di buffer:
  + in trasmissione devono essere memorizzati i frame inviati in attesa di riscontro, per poterli ritrasmettere in caso di necessita’
  + ad ogni riscontro ricevuto, vengono liberati i buffer relativi ai frame riscontrati, per occuparli con i nuovi frame trasmessi
  + a seconda del protocollo anche in ricezione di deve disporre di buffer, ad esempio per memorizzare frame fuori sequenza;
  + ad ogni riscontro inviato, i frame riscontrati vengono passati allo strato di rete ed i relativi buffer vengono liberati per poter accogliere nuovi frame in arrivo ed una maggiore complessita’ di calcolo
* La dimensione della finestra (W) puo’ essere fissata a priori dal protocollo, ma esistono protocolli che permettono di modificarne il valore dinamicamente tramite informazioni di controllo del protocollo
* Il mittente mantiene in un buffer di dimensione w tutti i frame nella finestra nel caso debbano essere ritrasmessi
* Quando il buffer è pieno (non si sono ricevuti ack) il livello data link del mittente non accetta più pacchetti dal livello di rete
* Se w=1 si ha un protocollo stop-and-wait (si aspetta l’ack prima di spedire un nuovo frame)
* Il protocollo stop-and-wait spreca banda per le attese in caso di mezzo fisico con ritardo non trascurabile

**Protocollo Go-back-n**

Questo protocollo segue la logica che in ricezione vengano rifiutati tutti i frame successivi ad un frame danneggiato o mancante, esso infatti spedisce più di un frame ed aspetta un riscontro da parte del destinatario. Mantiene una copia dei frame spediti fino all’arrivo del riscontro positivo.

**Funzionamento:**

Esistono due possibilita’:  
• **frame errato**: in questo caso B scarta il frame:

* se A non invia frame successivi, non accade nulla fino allo scadere del timer di A, quindi A ricomincia ad inviare frame a partire dal primo non riscontrato
* se A invia frame successivi, B risponde con un REJ dei frame ricevuti, in modo da notificare ad A che il frame indicato nel REJ e’ andato perso; al primo REJ ricevuto, A ricomincia dal primo frame non riscontrato

**• ACK errato**: in questo caso B ha accettato il frame:

1. se A non invia frame successivi, allo scadere del timer:

* A invia nuovamente il frame; B lo rifiuta (duplicato) ma invia nuovamente l’ACK
* alternativamente, al timeout A puo’ inviare un frame di controllo per chiedere conferma dell’ultimo frame ricevuto correttamente, a cui B risponde con l’ACK relativo

1. se A invia frame successivi, B risponde con l’ACK del frame successivo; poiche’ gli ACK sono cumulativi, l’ACK del frame successivo riscontra anche quello di cui A non ha ricevuto l’ACK, quindi il trasferimento dati continua senza interruzioni

**Protocollo Selective Reject**

Il protocollo selective reject prevede che in ricezione possano essere accettati frame fuori sequenza, utilizzando un meccanismo di ritrasmissione selettiva dei frame errati

In questo modo si riduce ulteriormente il numero di frame ritrasmessi, mantenendo la caratteristica di recapitare allo strato di rete i dati nell’ordine corretto

In ricezione i frame fuori ordine (ma dentro la finestra di ricezione) vengono mantenuti nei buffer fino a che non siano stati ricevuti tutti i frame intermedi

**Funzionamento:**

* Quando si ha un frame perduto, B ricevera’ il frame successivo fuori sequenza, al quale rispondera’ con un ACK relativo al frame perduto
* A non ritrasmette tutti i frame successivi a quello, ma solo quello perduto, quindi proseguira’ con la normale sequenza
* B ha memorizzato i frame successivi, ed alla ricezione del frame ritrasmesso liberera’ tutti i buffer inviando un ACK relativo all’ultimo frame ricevuto correttamente
* In caso di perdita dell’ACK, sara’ il timeout di A a generare un frame di sollecito di ACK per B, che rispondera’ di conseguenza

**Codifica Polinomiale (CRC)**

La tecnica consiste nel considerare i dati da inviare come un polinomio di grado m-1

* Trasmettitore e ricevitore si accordano sull’utilizzo di un polinomio generatore G(x) di grado r
* Il trasmettitore aggiunge in coda al messaggio una sequenza di bit di controllo (CRC) in modo che il polinomio associato ai bit del frame trasmesso, costituito dall’insieme di dati e CRC, sia divisibile per G(x)
* In ricezione si divide il polinomio associato ai dati ricevuti per G(X)
  + se la divisione ha resto nullo, si assume che la trasmissione sia avvenuta senza errori
  + se la divisione ha resto non nullo, sono certamente avvenuti errori