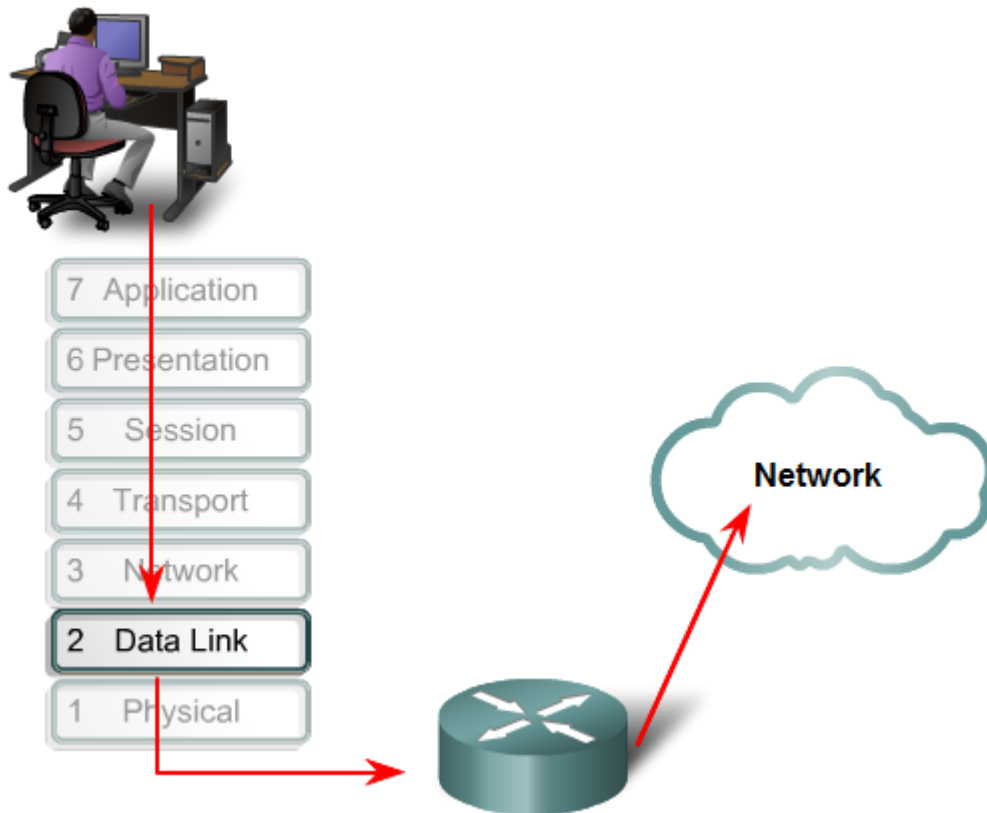


LIVELLO 2: DATA-LINK

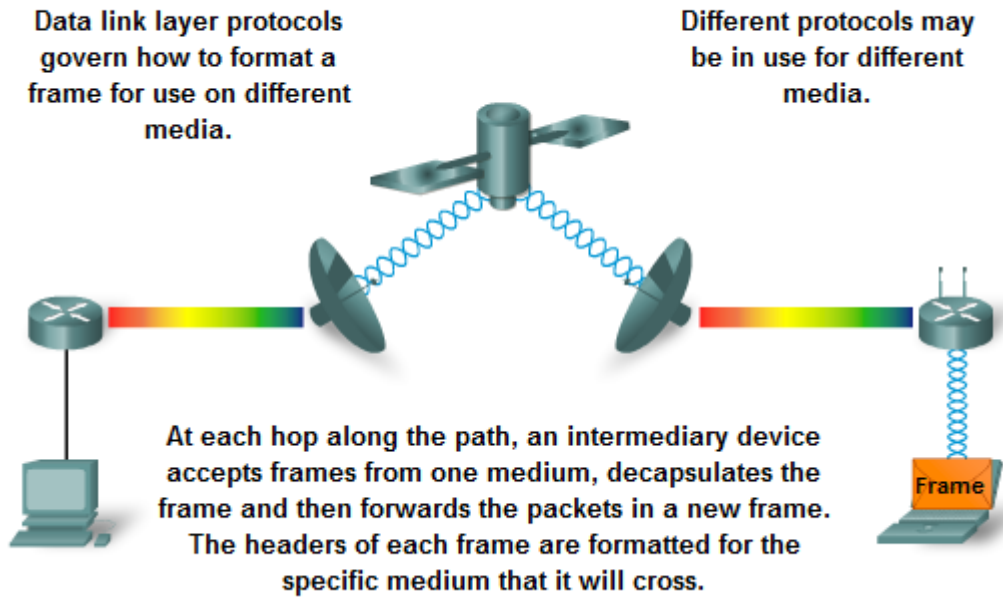


Obiettivo: permettere il trasferimento dei pacchetti su reti di diverso tipo; gestire la comunicazione tra nodi direttamente connessi; garantire il controllo di flusso(ovvero la necessaria sincronizzazione tra sorgente e destinazione); stabilire delle regole per l'accesso al canale di comunicazione nel caso in cui questo risulti condiviso; rilevare eventuali errori di trasmissione e gestire eventualmente la perdita o la duplicazione dei frames(a seconda che il protocollo preveda un riscontro oppure no) per permettere un trasferimento affidabile di dati attraverso il livello fisico. Tutto ciò consente di far apparire, al livello superiore, il mezzo fisico come una linea di trasmissione esente da errori di trasmissione e di rendere invisibile ai livelli superiori le differenze legate alle tecnologie utilizzate

La funzione principale del data-link è quella di permettere il trasferimento dei pacchetti provenienti dal livello di rete su reti di tipo diverso, che utilizzano differenti mezzi di comunicazione(cavi coassiali, doppini di rame, fibre ottiche, microonde) e differenti protocolli.

In ultima analisi tale livello rende invisibile al livello di rete le differenti tecnologie utilizzate nelle reti attraversate.

The Data Link Layer



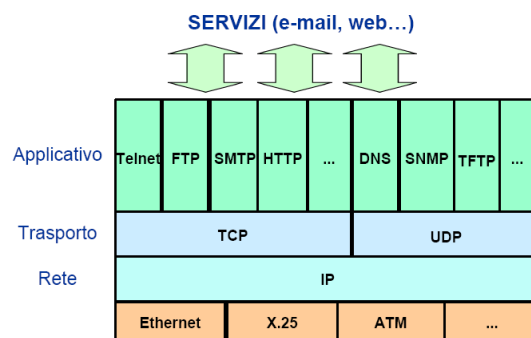
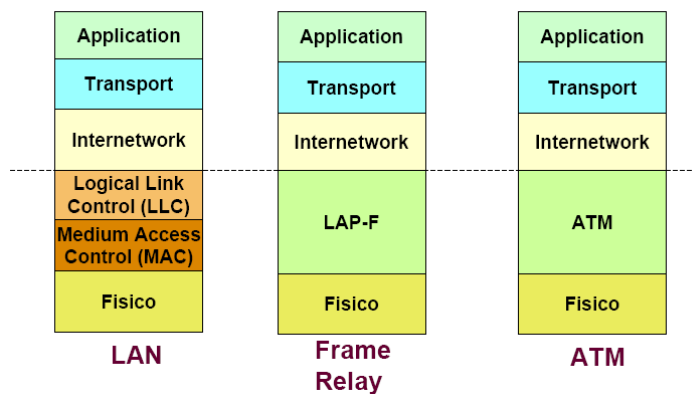
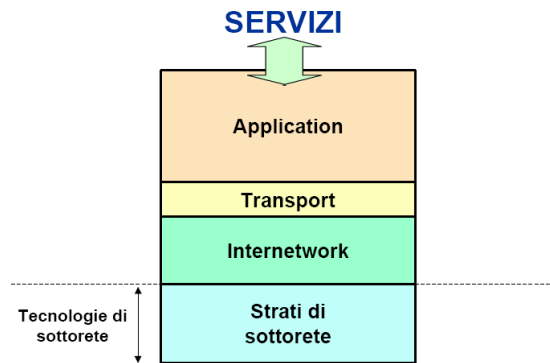
Di seguito verranno brevemente descritti i principali servizi offerti da questo livello al livello immediatamente superiore nel modello di riferimento ISO/OSI.

Sebbene nel livello OSI esista una rigida separazione tra il livello di Data-Link e il sottostante livello fisico in realtà capita che queste funzioni non siano sempre così ben separate.

Vale la pena infatti di ricordare che nell'architettura correntemente utilizzata su Internet il TCP/IP (modello DoD/ARPA) questi due livelli sono riuniti in un singolo livello denominato "interfaccia di rete".

Internet Protocol Suite (TCP/IP)

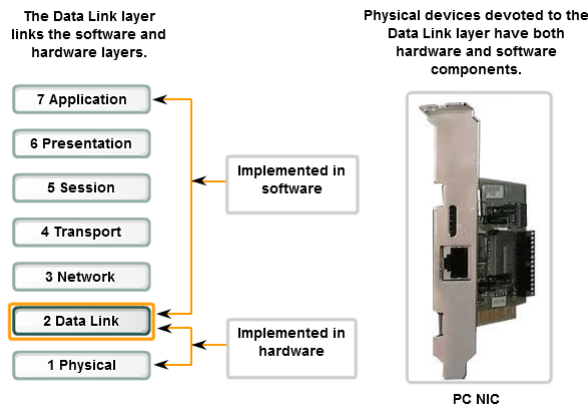
7	applicazione	applicazione (telnet, FTP, SMTP, DNS, HTTP, ecc.)
6	presentazione	
5	sessione	trasporto (TCP e UDP)
4	trasporto	
3	network	network (IP, ARP, ecc.)
2	data link	host - rete (non specificato)
1	fisico	
	OSI	TCP/IP



Hardware o software?

Il livello di Data-Link si presenta come uno strato di connessione tra il software dei processi che operano ai livelli superiori e l'hardware del livello fisico.

Le funzioni assolte dal livello di Data-Link sono spesso svolte da un singolo componente hardware (che svolge anche le funzioni del livello fisico come già anticipato parlando dell'architettura TCP/IP) come, ad esempio, la scheda di rete nel caso delle reti Ethernet o *Network Interface Card*(NIC), all'hardware si affianca anche il software installato che serve a predisporre i pacchetti per la trasmissione sul mezzo fisico.



Protocolli sincroni e asincroni

I protocolli del livello di data-link sono classificati in base al tipo di trasmissione in:

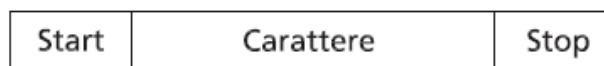
- Asincroni
- Sincroni

L'elemento base che costituisce l'informazione può essere il carattere (a 7 oppure 8 bit) o il bit.

I protocolli asincroni sono orientati al carattere, mentre quelli sincroni possono essere orientati sia al carattere sia al bit.

Un protocollo si dice **asincrono** perché l'intervallo di tempo che intercorre tra l'invio di un carattere ed il successivo è indefinito.

Ogni carattere, codificato generalmente secondo il codice ASCII a 7 o a 8 bit, è preceduto da un bit di start e seguito da uno o due bit di stop. Per tale motivo il protocollo è detto anche start-stop.



I caratteri sono indipendenti e viaggiano separatamente. L'overhead associato alla presenza dei bit di Start and Stop per ogni carattere trasmesso comporta però uno spreco considerevole della banda pertanto questo tipo di protocolli viene utilizzato per la trasmissione di dati a bassa velocità tra computer host.

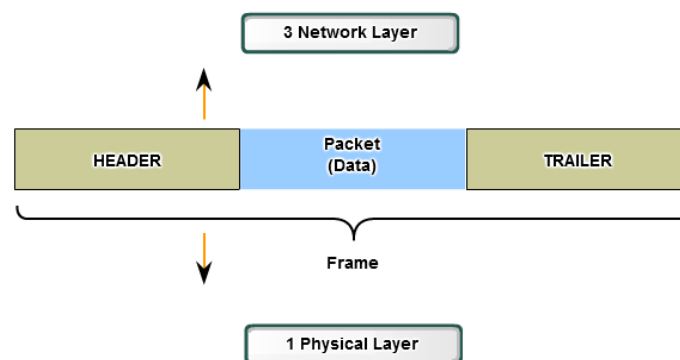
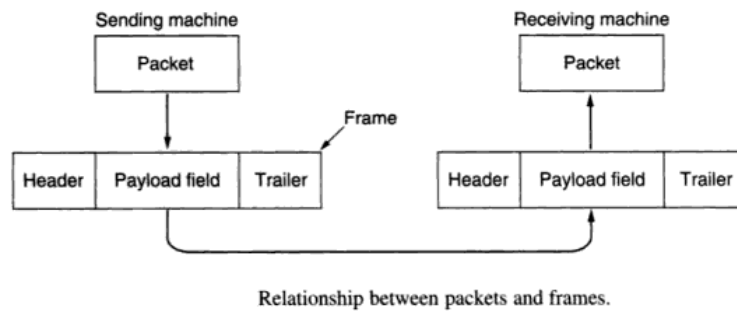
Nella trasmissione sincrona al contrario i caratteri da inviare vengono raggruppati in frame e fatti precedere da caratteri di sincronizzazione che servono a far sì che la stazione in ricezione si sincronizzi sulla velocità della stazione trasmittente.

La trasmissione sincrona è più veloce perché i tempi morti vengono ridotti, ma un errore anche di un bit può danneggiare l'intero frame.

Un'operazione fondamentale nel caso di protocolli sincroni è il *framing* ovvero la suddivisione del messaggio in frame.

Framing

Questo livello si occupa, in trasmissione, di formattare i dati da inviare attraverso il livello fisico, incapsulando il pacchetto proveniente dal livello di rete, in una trama o *frame* provvisto di *header*(intestazione) e *trailer* (coda) , mentre in ricezione compie l'operazione inversa ovvero estrae il pacchetto dal frame per passarlo al livello superiore.



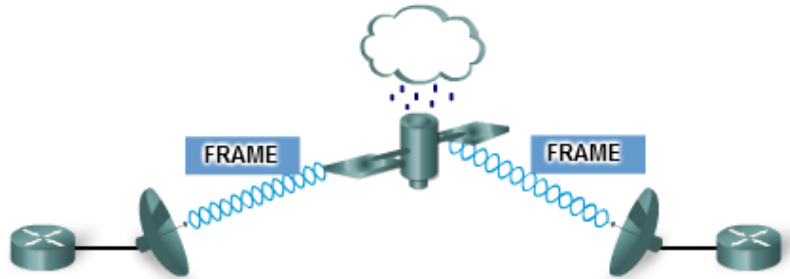
I campi che compongono l' *header* e il *trailer* variano a seconda del protocollo utilizzato a questo livello(quello più diffuso per le LAN è Ethernet, mentre per le WAN vi sono vari protocolli come PPP) e servono a tenendo conto delle specifiche tecnologie utilizzate e delle caratteristiche del canale di comunicazione.

Ad esempio in caso di trasmissioni via satellite(o via radio) la probabilità di errore, di collisione, etc... è sicuramente molto più elevata rispetto alle trasmissioni che avvengono via cavo. Ciò richiede una serie di controlli aggiuntivi e protocolli più sofisticati che si traducono in genere in dimensioni maggiori dell'header e del trailer e dunque dell'overhead legato a tali campi che ha come effetto finale quello di diminuire la velocità di comunicazione.

Data Link Layer Protocols - The Frame

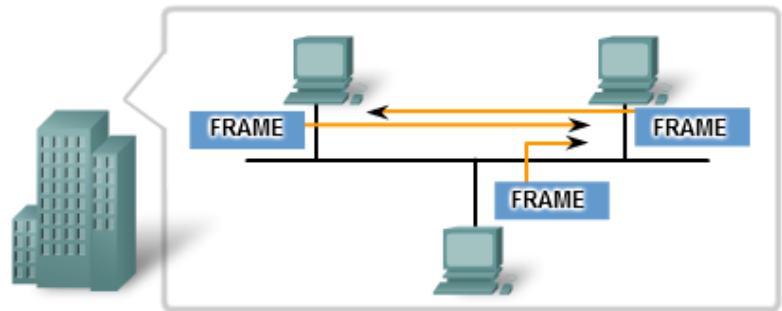
In a fragile environment, more controls are needed to ensure delivery. The header and trailer fields are larger as more control information is needed.

Greater effort needed to ensure delivery = higher overhead = slower transmission rates

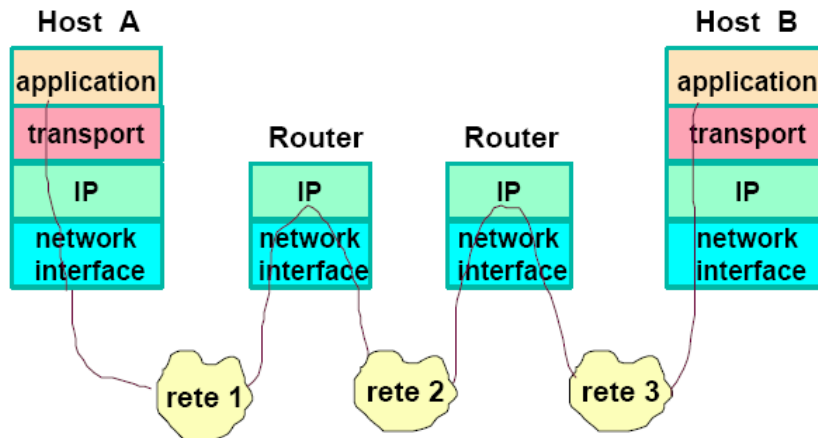


In a protected environment, we can count on the frame arriving at its destination. Fewer controls are needed, resulting in smaller fields and smaller frames.

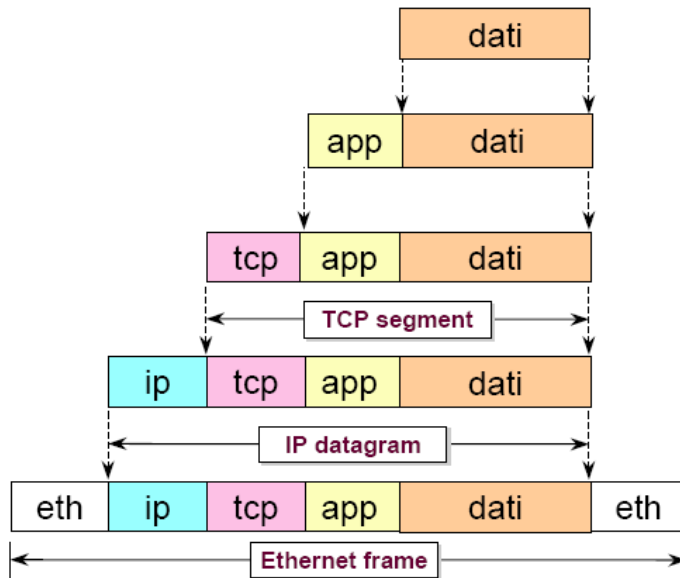
Less effort needed to ensure delivery = lower overhead = faster transmission rates



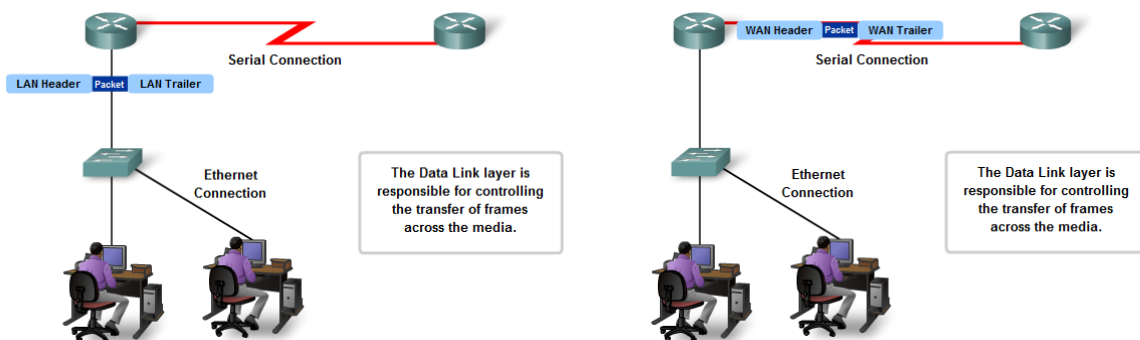
Ad ogni router intermedio il pacchetto viene estratto dalla “busta” del livello data-link(Ethernet,PPP,...) ovvero vengono elaborate e rimossi header e trailer e il pacchetto viene imbustato nella busta di data-link della nuova rete da attraversare.



La figura seguente mostra un esempio di imbustamento o incapsulamento in trasmissione attraverso i vari livelli dell'architettura TCP/IP.

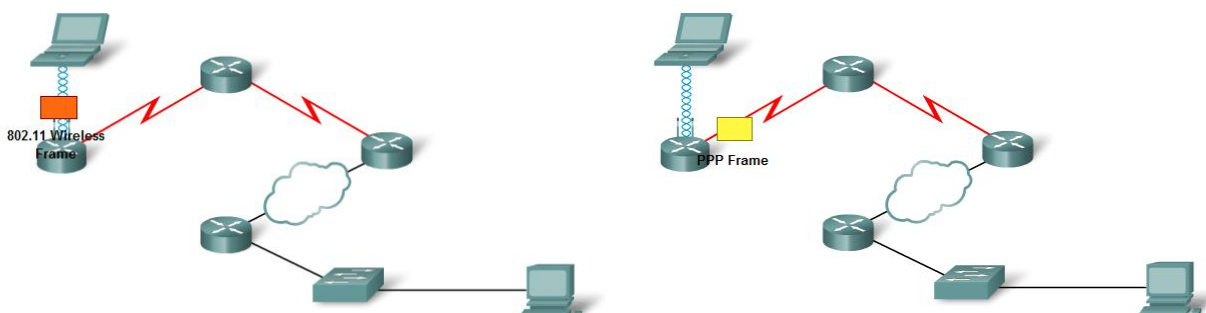


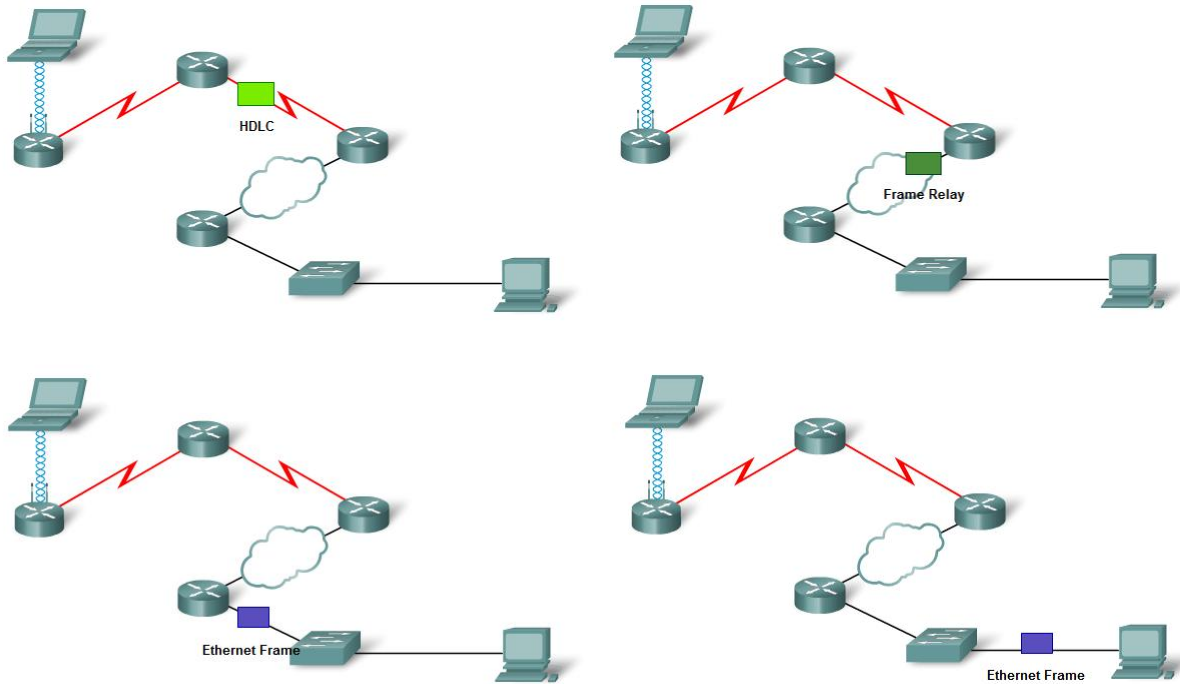
Di seguito è mostrato il cambio di header e trailer tra una LAN e una WAN. Questo cambio avviene ad opera del router che effettua la traduzione da un protocollo di data-link ad un altro. Ovviamente cambiano anche le modalità di trasmissione al livello fisico.



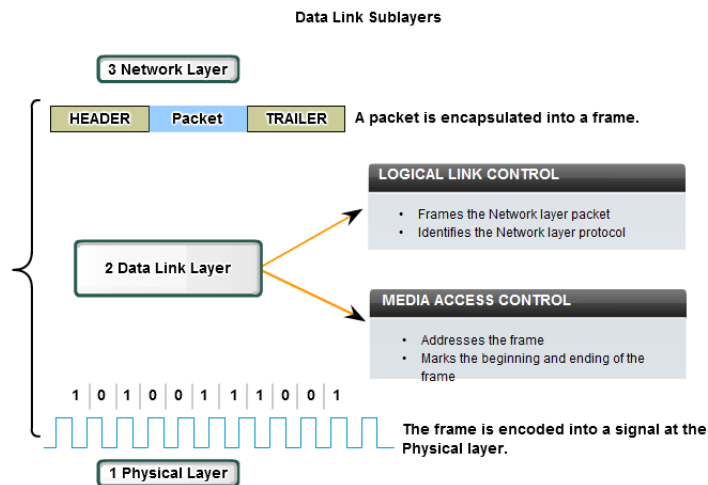
Il pacchetto dunque avrà un'intestazione diversa sul collegamento seriale Punto-A-Punto che ha caratteristiche del tutto diverse da quello locale di tipo Ethernet.

Le seguenti figure mostrano quante volte può cambiare il protocollo durante una tipica trasmissione su Internet.





Sottolivelli del livello Data-Link



Spesso il livello di data-link è suddiviso in due sottolivelli(sublayers). Questo permette di utilizzare lo stesso tipo di frame per mezzi differenti variando solo il sottostrato MAC ad esempio.

Questo accade per esempio per Ethernet(anche se nelle ultime versioni questa suddivisione è stata abolita).

La suddivisione più comune è quella costituita dai due sottolivelli :

- LLC(Logical Link Control)
- MAC(Medium Access Control)

Il primo si occupa di inserire nell'intestazione del frame le informazioni che riguardano il protocollo di rete utilizzato.

Ciò consente di utilizzare più protocolli (IP,IPX) con la stessa interfaccia.

Il sottolivello MAC si occupa del tipo di protocollo da utilizzare al livello Data-Link, dell'indirizzamento, della delimitazione dei frame a seconda della tipologia del canale di comunicazione ovvero del mezzo fisico sul quale vengono trasmessi i segnali.

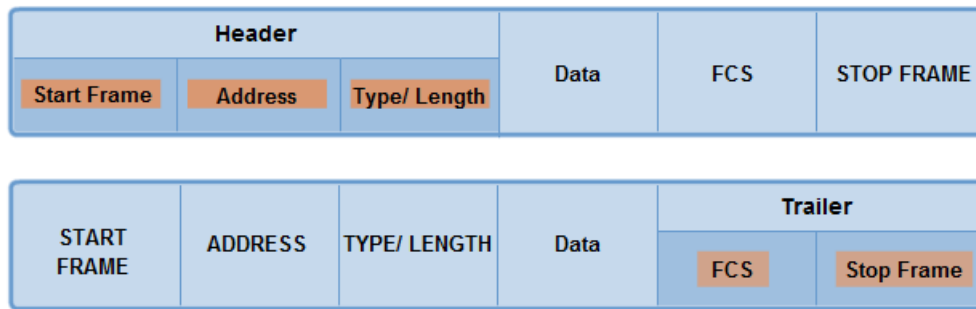
A livello di Data-Link esistono una moltitudine di standard e di variazioni per adeguarsi a nuovi mezzi di trasmissione.

Nella seguente figura sono elencati i principali standard e le organizzazioni che li hanno definiti e contribuiscono al loro aggiornamento.

Standards for the Data Link Layer

ISO:	HDLC (High Level Data Link Control)
IEEE:	802.2 (LLC), 802.3 (Ethernet) 802.5 (Token Ring) 802.11(Wireless LAN)
ITU:	Q.922 (Frame Relay Standard) Q.921 (ISDN Data Link Standard) HDLC (High Level Data Link Control)
ANSI:	3T9.5 ADCCP (Advanced Data Communications Control Protocol)

Intestazione e Trailer



Sebbene, come già accennato, i campi dell'intestazione e della coda di un frame possano variare esistono alcuni campi "tipici" come quelli che ci apprestiamo a descrivere.

Campi dell'intestazione:

- *Inizio frame* : serve a delimitare l'inizio del frame
- *Indirizzi del mittente e del destinatario*: individuano il mittente e il destinatario su una certa sottorete(nel caso della rete Ethernet ad esempio si hanno gli indirizzi MAC che saranno descritti più avanti)
- *Priorità/Qualità del servizio* : indica il livello di priorità o il tipo di servizio che deve essere fornito dal protocollo; infatti, a seconda del tipo di dati trasportati(video/audio, dati,...) o del servizio richiesto, è necessario garantire una minore/maggiore velocità di trasmissione, una minore/maggiore affidabilità, etc...
- *Tipo*: indica il tipo di dati, il tipo di protocollo utilizzato al livello superiore oppure la lunghezza del campo dati
- *LLC Field (Logical Connection Control Field)* : utilizzato per stabilire una connessione logica tra i due nodi
- *Physical link control field*: utilizzato per indicare il tipo di mezzo trasmissivo
- *Flow control field* : utilizzato per effettuare il controllo di flusso tra mittente e destinatario
- *Congestion control field* : utilizzato per effettuare il controllo di congestione

Nel trailer invece campi "tipici" sono:

- *FCS(Frame Check Sequence)* : campo per l'individuazione/correzione degli errori
- *Fine Frame*: delimita la fine del frame

Di seguito viene mostrata la struttura semplificata dei protocolli Ethernet e PPP.

Ethernet Protocol

A Common Data Link Layer Protocol for LANs

Frame

Field name	Preamble	Destination	Source	Type	Data	Frame Check Sequence
Size	8 bytes	6 bytes	6 bytes	2 bytes	46 - 1500 bytes	4 bytes

Preamble - used for synchronization; also contains a delimiter to mark the end of the timing information.

Destination Address - 48 bit MAC address for the destination node.

Source Address - 48 bit MAC address for the source node.

Type - value to indicate which upper layer protocol will receive the data after the Ethernet process is complete.

Data or payload - this is the PDU, typically an IPv4 packet, that is to be transported over the media.

Frame Check Sequence (FCS) - A value used to check for damaged frames.

Point-to-Point Protocol

A Common Data Link Protocol for WANs

Frame

Field name	Flag	Address	Control	Protocol	Data	FCS
Size (bytes)	1 byte	1 byte	1 byte	2 bytes	variable	2 or 4 bytes

Flag - A single byte that indicates the beginning or end of a frame. The flag field consists of the binary sequence 01111110.

Address - A single byte that contains the standard PPP broadcast address. PPP does not assign individual station addresses.

Control - A single byte that contains the binary sequence 00000011, which calls for transmission of user data in an unsequenced frame.

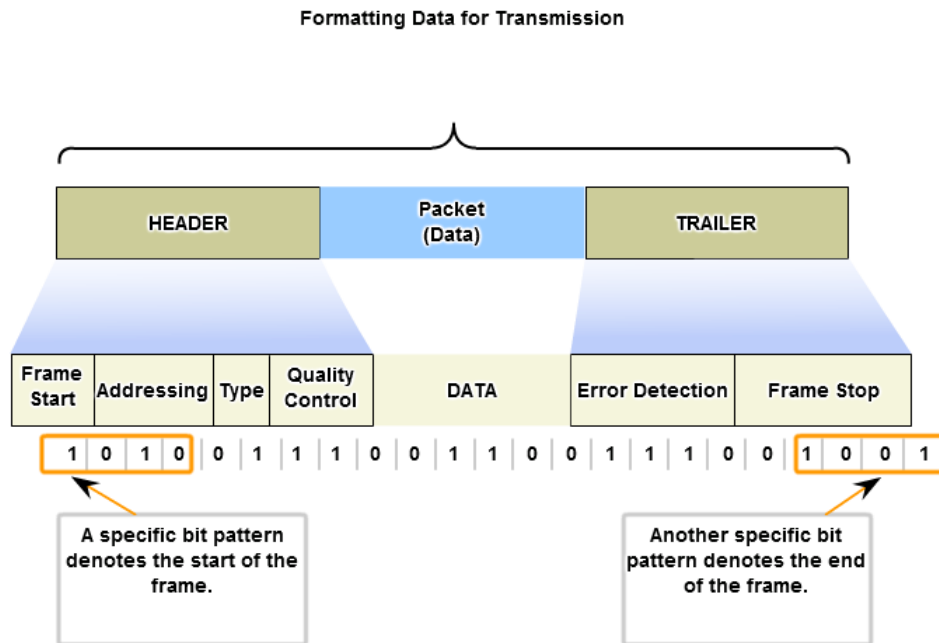
Protocol - Two bytes that identify the protocol encapsulated in the data field of the frame. The most up-to-date values of the protocol field are specified in the most recent Assigned Numbers Request For Comments (RFC).

Data - Zero or more bytes that contain the datagram for the protocol specified in the protocol field.

Frame Check Sequence (FCS) - Normally 16 bits (2 bytes). By prior agreement, consenting PPP implementations can use a 32-bit (4-byte) FCS for improved error detection.

Delimitazione frame

Per poter individuare l'inizio e la fine di un frame occorre in qualche modo delimitarli, perché sarebbe troppo rischioso utilizzare lo spazio che intercorre tra un frame ed un altro per tale delimitazione.

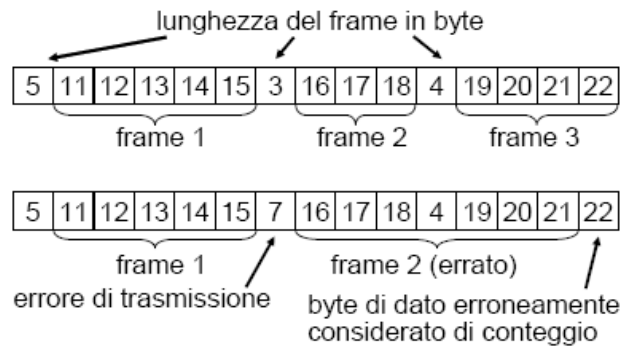


Vengono utilizzati vari metodi per la delimitazione dei frames:

- Conteggio dei caratteri.
- Caratteri di inizio e fine.
- Indicatori (flag) di inizio e fine.

Il metodo del **conteggio di caratteri** (specificando in un campo dell'intestazione il numero di caratteri del frame) è raramente utilizzabile perché, se il campo che contiene il conteggio si rovina durante la trasmissione, non si può più individuare dove comincia il frame successivo pertanto vengono utilizzate le tecniche più affidabili .

Delimitazione della trama con conteggio dei byte

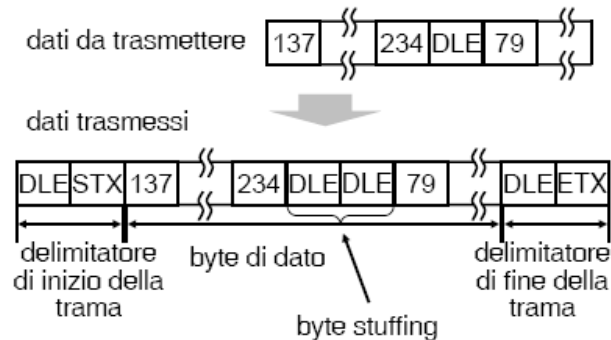


Nella **trasmissione orientata al byte** (il frame mantiene la suddivisione in byte) il frame viene preceduto dalla sequenza di caratteri ASCII **DLE STX** (Data Link Escape Start of TeXt) e finisce con la sequenza **DLE ETX** (Data Link Escape End of TeXt).

Se un frame si rovina e la destinazione perde la sincronizzazione basta trovare il successivo DLE STX o DLE ETX.

Il carattere DLE però può comparire casualmente dentro al frame; perché questi caratteri non interferiscano viene aggiunto un ulteriore DLE (che viene rimosso a destinazione prima di passare al frame al livello di rete) in modo che solo i DLE singoli vengano interpretati come delimitatori; questa tecnica si chiama *character stuffing*.

Delimitazione della trama con caratteri speciali

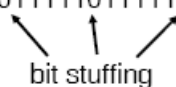


Nella **trasmissione orientata al bit** (il frame può contenere un numero qualsiasi di bite) **ogni frame inizia e finisce con la sequenza 01111110** chiamata flag: questa sequenza può comparire casualmente nei dati, perciò in trasmissione **dopo cinque 1 consecutivi viene sempre inserito uno 0** nel flusso di bit, indipendentemente dal fatto che il bit successivo sia 1 o 0, mentre in ricezione bisogna provvedere ad eliminare i bit inseriti, rimuovendo sempre uno 0 dopo cinque; questa tecnica è chiamata *bit stuffing*.

Delimitazione della trama con riempimento di bit ("bit stuffing")

dati da trasmettere 0110111111111111111111110010

dati trasmessi 0110111110111111011111010010



Rilevamento e correzione degli errori di trasmissione

Per ogni frame da inviare viene calcolato un codice di controllo detto *checksum* (o somma di controllo) legato alla sequenza di bit da trasmettere e il valore ottenuto è trasmesso insieme al frame.

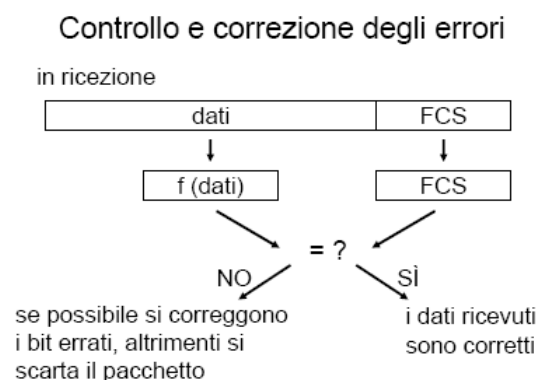
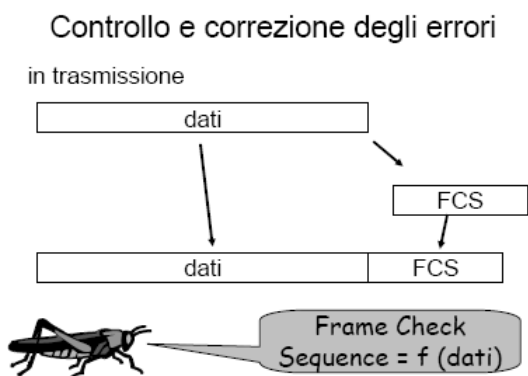
Quando un *frame* arriva a destinazione, il *checksum* viene ricalcolato. Se il nuovo risultato è diverso da quello contenuto nel pacchetto, il livello data link riconosce che deve essersi verificato un errore di trasmissione e provvede ad esempio a scartare il pacchetto ed eventualmente a spedire in risposta un messaggio di errore.

A seconda dell'algoritmo utilizzato per il calcolo del *checksum* è possibile anche un'altra alternativa: correggere l'errore.

Questo richiede ovviamente un algoritmo di calcolo di checksum che permetta non solo la rilevazione dell'errore, ma che fornisca anche la possibilità di risalire alla configurazione originaria.

Un codice per che permetta la correzione degli errori richiede però più spazio nel frame, tanto più quanto maggiore è il numero di errori che riesce a rilevare.

La scelta tra codici rilevatori e correttori dipende in genere dalla velocità delle linee (per linee a bassa velocità aspettare la ritrasmissione potrebbe richiedere troppo tempo) e dalla loro affidabilità (se il tasso di errore sulla linea è molto basso non vale la pena sprecare molta per un codice correttore).



Protocolli orientati alla connessione e non

Il livello di data-link può offrire un servizio:

Non orientato alla connessione, **senza** riscontro

- si mandano frame indipendenti
- i frame non vengono confermati
- non si stabilisce una connessione
- i frame persi non si recuperano (in questo livello)
- appropriato per canali con tasso d'errore molto basso(LAN Ethernet)

Non orientato alla connessione, **con** riscontro

- I frame vengono confermati
- Se la conferma non arriva, il mittente può rispeditore il frame
- appropriato per canali non affidabili (sistemi wireless)

Orientato alla connessione, **con** riscontro

- Tre fasi: apertura connessione, invio dati, chiusura connessione
- Garantisce che ogni frame sia ricevuto esattamente una volta e nell'ordine giusto
- Fornisce al livello network un flusso di bit affidabile

Nel caso in cui il protocollo utilizzato a questo livello prevede il riscontro, per ogni *frame* ricevuto, il destinatario invia al mittente un *frame ACK* di *acknowledgement* ovvero di conferma dell'avvenuta ricezione del *frame* inviato.

Il mittente ripete l'invio dei pacchetti alterati da errori di trasmissione(in questo caso il frame di riscontro è indicato con NACK che sta per *Not - Acknowledged*) e di quelli di cui non ha ricevuto un ACK entro un certo tempo massimo.

Per ottimizzare l'invio degli ACK, si usa una tecnica detta **Piggybacking**, che consiste nell'accodare ai messaggi in uscita gli ACK relativi ad una connessione in entrata, per ottimizzare l'uso del livello fisico i pacchetti ACK possono anche essere raggruppati e mandati in blocchi.

Controllo di flusso

Un altro importante problema di progettazione che si ritrova nel livello di data link è quello gestire una linea condivisa quando più nodi vogliono inviare messaggi nello stesso tempo; inoltre deve decidere cosa fare di un mittente che sistematicamente tende a trasmettere pacchetti più velocemente di quanto il ricevente li accetti. Questa situazione può facilmente essere riscontrata quando il mittente è dislocato su una macchina veloce e il ricevente su una macchina lenta. Il mittente continua a spedire pacchetti ad alta velocità, fino a quando il ricevente non è completamente sopraffatto. Anche se la trasmissione è esente da errori, a un certo punto il ricevente non sarà in grado di gestire i pacchetti in arrivo e inizierà a perderli. La tipica soluzione è quella di introdurre un controllo di flusso per obbligare il mittente a rispettare

la velocità del ricevente nello spedire i pacchetti. Questa imposizione solitamente richiede un certo tipo di meccanismo di riscontro in modo che il mittente possa essere avvisato se il ricevente è in grado di ricevere o meno.

Esistono due strategie principali per il controllo di flusso:

- Stop and wait(*Stop and Go*)
- Sliding windows

Nel primo caso il mittente prima di inviare un nuovo frame attende l'ACK del precedente, se tale riscontro non arriva entro un tempo limite(*timeout*) il mittente provvederà a rispedito il frame.

Per evitare che vi siano duplicazioni causate dalla perdita di frame di riscontro i frame devono essere numerati.

In realtà è sufficiente utilizzare un solo bit che assume successivamente i valori 0 e 1.

Il ricevente si aspetta di ricevere valori con 0 e 1 alternativamente. Se riceve due frame con lo stesso valore (due frame con il valore 0 o due frame con il valore 1) rifiuta l'ultimo frame riconoscendolo come un duplicato del precedente, ma in ogni caso rispedisce il riscontro al mittente in modo che questi riceva prima o poi una conferma dell'avvenuta ricezione.

Se il tempo di trasmissione non è trascurabile attendere l'ACK di ogni frame inviato rallenta eccessivamente la comunicazione. La tecnica Sliding Windows(Finestre scorrevoli) al contrario consente l'invio di più frame prima di fermarsi ad attendere un riscontro.

Nella seguente figura viene illustrata la differenza tra i due protocolli, le frecce indicano lo scorrere del tempo.

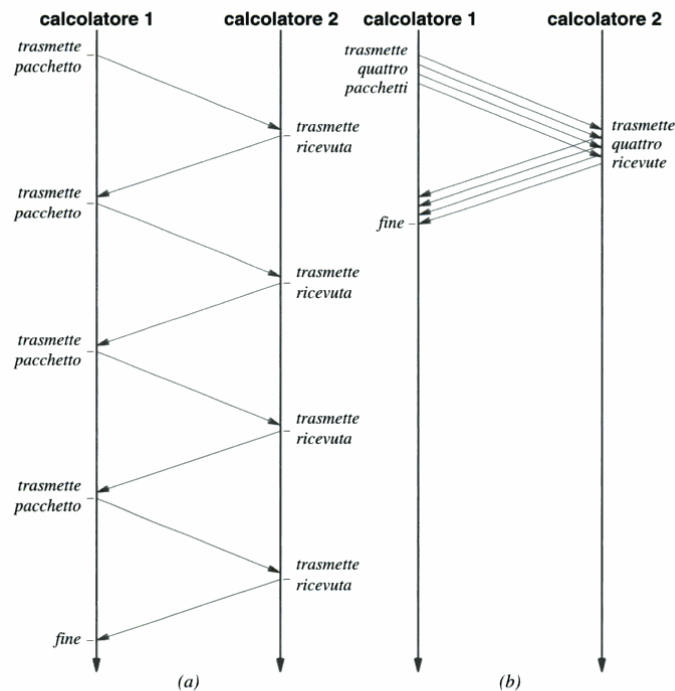


Figura 16.7 Trasmissione di quattro pacchetti (a) in base al protocollo stop-and-go e (b) in base a un protocollo a finestra scorrevole da 4 pacchetti. Il tempo scorre dall'alto verso il basso e ogni freccia rappresenta un messaggio trasmesso.

Dal grafico si può facilmente osservare il tempo richiesto per effettuare la trasmissione nei due casi. Questo tempo è tanto maggiore quanto maggiore è il tempo di risposta.

Nei protocolli sliding windows il mittente posiziona un immaginaria finestra sul flusso dati e trasmette tutti i frame (che corrispondono ad altrettanti pacchetti) contenuti nella finestra. Quando riceve l'ACK di un frame fa scorrere la finestra e trasmette i pacchetti successivi.

I frame sono numerati progressivamente da 0 ad un numero massimo che dipende dal numero di bit utilizzati per la numerazione (con n bit il max è $2^n - 1$). Quando si arriva al numero massimo si riparte da 0.

Nel caso di 3 bit i frame (o i pacchetti corrispondenti) sono numerati da 0 a 7.

Come sarà spiegato in seguito in realtà i pacchetti che possono essere trasmessi contemporaneamente sono solo 7 e non 8 ovvero la dimensione massima della finestra è $2^n - 1$.



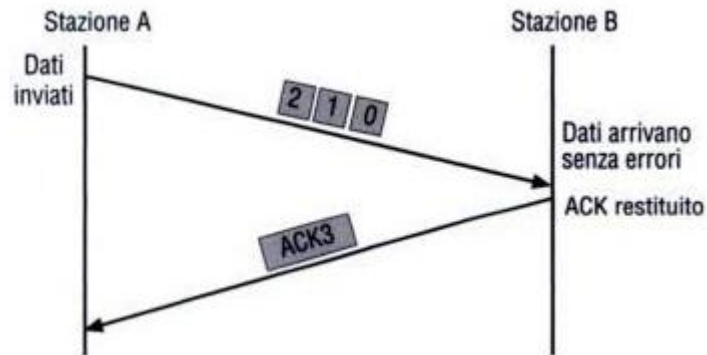
Figura 16.6 Finestra da 4 pacchetti che scorre lungo i dati da trasmettere. La finestra è rappresentata (a) all'inizio della comunicazione, (b) dopo che è stata ottenuta la ricevuta per i primi due pacchetti e (c) dopo che è stata ottenuta la ricevuta per i primi otto pacchetti. Il mittente può trasmettere tutti i pacchetti contenuti nella finestra.

In realtà non è necessario inviare una conferma per ogni frame ricevuto. Essendo numerati è sufficiente che il ricevente confermi l'ultimo frame ricevuto nella corretta sequenza senza confermare quelli precedenti.

In realtà il messaggio di conferma contiene il numero del prossimo frame atteso.

Ad esempio se, come mostrato nella seguente figura, il mittente invia 3 pacchetti numerati 0,1,2 e il ricevente desidera confermarli tutti, invierà un ACK contenente il valore 3 poiché questo è il numero del prossimo frame che si aspetta di ricevere.

Ciò riduce ulteriormente l'overhead necessario per l'invio dei messaggi di riscontro e rende più efficiente l'utilizzo della banda disponibile.



Adesso è possibile capire perché nell'esempio di prima non possono essere spediti 8 pacchetti, ma al massimo 7.

Infatti il primo e l'ottavo avrebbero lo stesso numero di ACK e pertanto il mittente non saprebbe mai se il destinatario ha ricevuto tutti gli 8 pacchetti o non ne ha ricevuto nessuno!

Anche il ricevente utilizza una finestra che fa scorrere sui frame ricevuti. Mentre la finestra del mittente indica i frame che può spedire senza aver ricevuto un riscontro, quella del ricevente contiene i frame che il destinatario può ricevere prima di inviare un messaggio di conferma.

Queste finestre corrispondono in pratica a due buffer (memorie) che consentono di memorizzare un certo numero di frame: la finestra di trasmissione contiene i frame spediti di cui non è pervenuto ancora un *acknowledgment*, mentre quella di ricezione contiene i frame che possono essere accettati senza che vadano persi.

Il motivo della finestra di ricezione è che il ricevente è in grado di memorizzare i frame anche se ne alcuni mancano o sono stati alterati nel corso della trasmissione.

Quando arrivano i frame mancanti il ricevente può riordinarli, consegnare il loro contenuto al livello di rete e inviare al mittente il riscontro dell'ultimo numero progressivo ricevuto correttamente.

Ogni volta che viene spedito un frame viene avviato un timer, se il timer scade prima che sia arrivato l'ACK del frame allora questo verrà rispedito.

La dimensione delle finestre può essere utilizzata per il controllo della congestione (tra l'altro questa è la tecnica utilizzata anche da un protocollo a livello di trasporto come il TCP).

Quando il traffico è scarso la dimensione della finestra di trasmissione vengono aumentate, in questo modo è possibile inviare una maggiore quantità di dati prima che arrivino i messaggi di riscontro.

Nel caso in cui vi sia congestione al contrario le dimensioni della finestra di trasmissione vengono ridotte in maniera che il mittente debba attendere il riscontro dei dati inviati prima di spedirne altri.

La dimensione della finestra di trasmissione dipendono dalla dimensione di quella di ricezione, ovvero da quanti frame il ricevente è in grado di memorizzare prima che vadano persi.

La dimensione della finestra di trasmissione è dunque continuamente negoziata tra mittente e ricevente nel corso della trasmissione.

Esistono due tecniche utilizzate per gestire gli errori di trasmissione nel caso di protocolli sliding windows:

- Go back N
- Selective repeat

Questi due approcci risultano da un compromesso tra larghezza di banda e spazio disponibile per i buffer a livello data-link.

La prima prevede semplicemente di scartare tutti i frame successivi a quello danneggiato senza inviare un messaggio di riscontro per i frame scartati. In altre parole il livello di data-link una volta rilevato l'errore si rifiuta di accettare i frame successivi a quello danneggiato finché questo non viene ritrasmesso (dopo che è scaduto il timeout dal lato sorgente).

Questo approccio riduce moltissimo la dimensione del buffer di ricezione, ma funziona bene solo se la probabilità di errore è bassa, in quanto tutti i frame scartati dovranno essere rispediti consumando parte della banda disponibile.

L'altra strategia quella della "ripetizione selettiva" prevede invece di memorizzare i frame successivi in attesa che il frame danneggiato venga ritrasmesso. Per sollecitarne l'invio prima dello scadere del timeout si invia un frame di NACK.

Questa tecnica evita la ritrasmissione dei frame ricevuti quindi utilizza in modo più efficiente la banda, ma prevede un buffer di ricezione di dimensione tanto maggiore quanto più grande è la finestra di trasmissione.

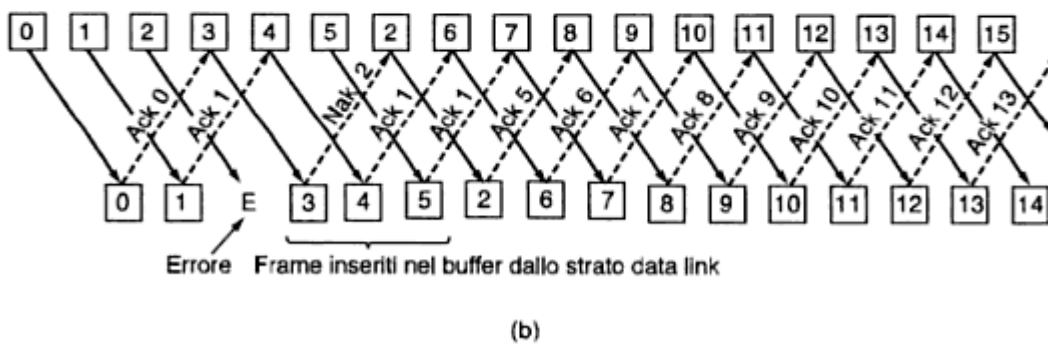
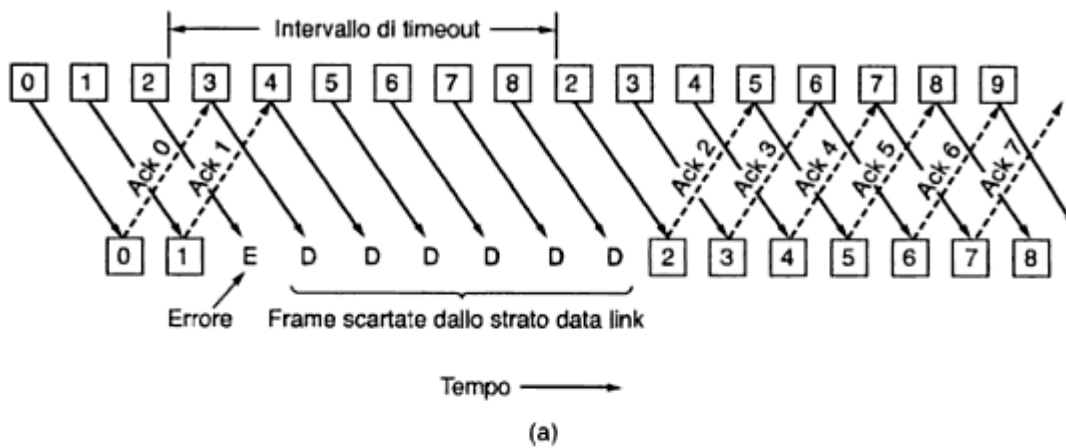


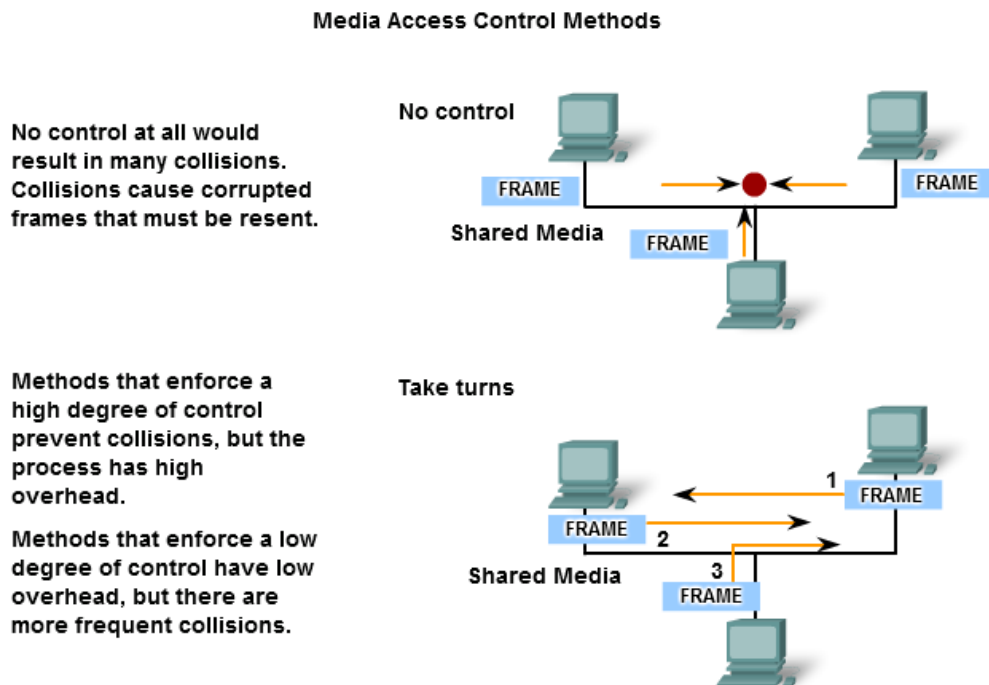
Figura 3.16. Pipelining e ripristino degli errori. Effetto di un errore quando (a) la finestra della destinazione ha dimensione 1 e (b) la finestra della destinazione è grande.

Controllo di accesso al mezzo

Il livello di data-link annovera tra le proprie funzioni quella di decidere le modalità di trasmissione e di condivisione del mezzo.

In particolare stabilire delle regole per l'accesso al mezzo significa specificare quale nodo ha il diritto a trasmettere dati, quando può farlo, quali operazioni preliminari deve effettuare prima della trasmissione, come devono essere gestite le collisioni, etc...

Per quanto concerne il compito di dirimere i problemi connessi alla contesa del mezzo(qualora come spesso avviene questo sia condiviso) è necessario adottare in genere una soluzione di compromesso tra l'esigenza di evitare collisioni(ovvero la sovrapposizione di più trasmissioni) e quella di ridurre l'overhead legato all'applicazione di sofisticati protocolli che evitano le collisioni ma rallentano il trasferimento.



In genere le tecniche applicate differiscono a seconda dei seguenti fattori:

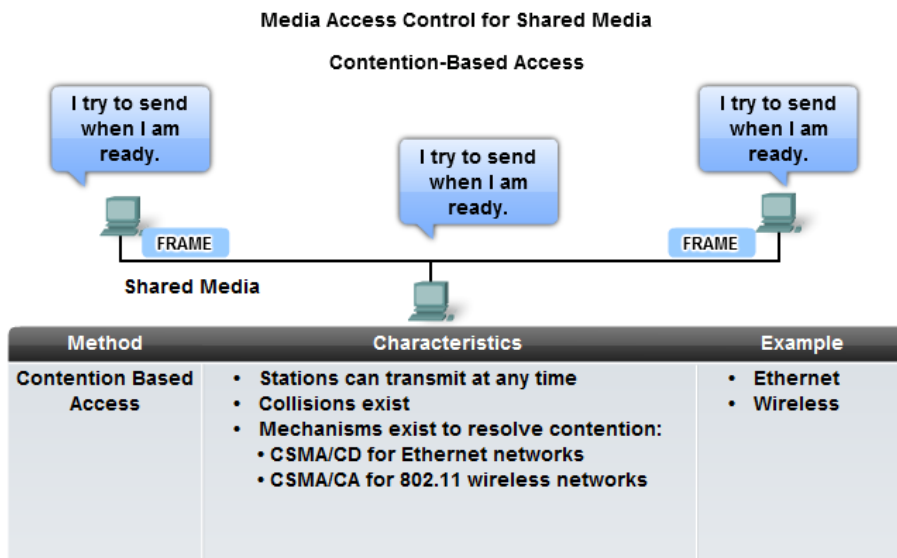
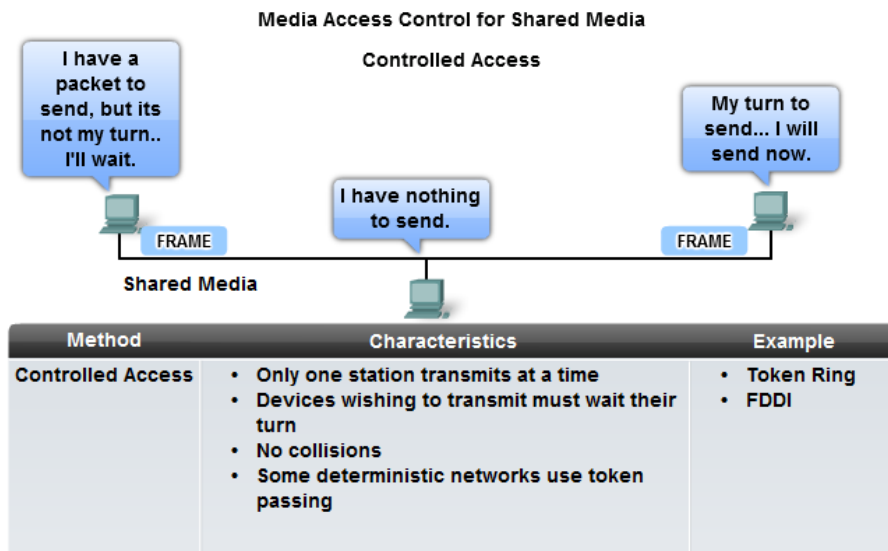
- mezzo condiviso/non condiviso
- tipo di trasmissione: full-duplex/half-duplex
- topologia della rete

Esistono due strategie principali:

- A prenotazione
- A contesa

La differenza è che nel primo caso le collisioni sono del tutto evitate prenotandosi o fissando dei turni di trasmissione, mentre nel secondo caso invece i nodi della rete competono per l'accesso al mezzo e vi è la possibilità concreta che vi siano collisioni ovvero che più di un nodo provi a trasmettere in contemporanea.

Nella seguente figura sono mostrati entrambi i metodi, le implementazioni più diffuse e le caratteristiche principali.



La comunicazione tra due nodi della rete può essere full-duplex o half-duplex. Nel primo caso la comunicazione è bidirezionale nel secondo unidirezionale alternata(nel senso che posso trasmettere in entrambi i versi, ma non contemporaneamente).

