UNIVERSITA' DEGLI STUDI DI CATANIA

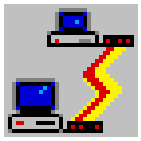
FACOLTA' di INGEGNERIA
CORSO di LAUREA In INGEGNERIA INFORMATICA

Data Link Layer: parte 1

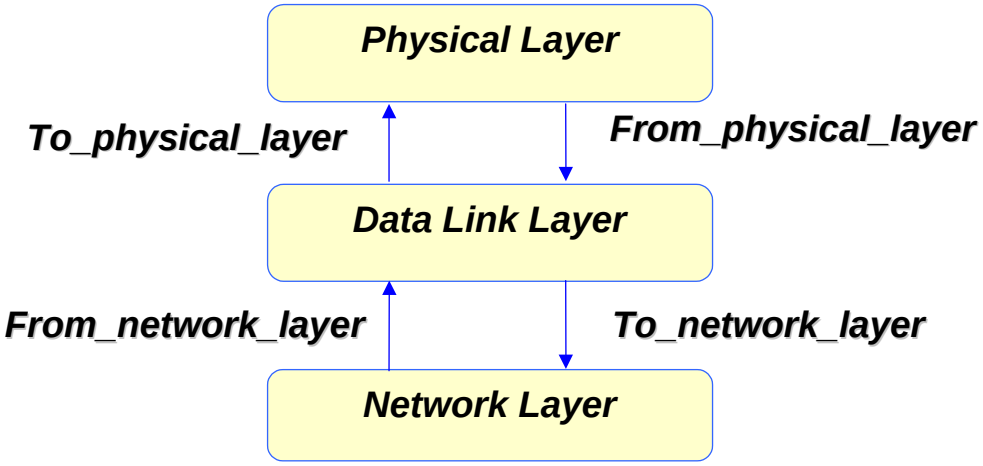
Corso di RETI di CALCOLATORI

PROF. OrazioMIRABELLA

Data Link Layer



- Secondo livello dell'architettura OSI
- Fornisce i mezzi funzionali e procedurali per il trasferimento trasparente, affidabile ed efficiente di unità dati tra Network Layer Entities
- Trasforma un formato grezzo di bit in pacchetti di bit ordinati (frame) e privi di errori
- Nasconde i dettagli su come le risorse di comunicazione fornite dal Physical Layer sono usate per fornire il servizio
- Il Physical Layer, il Data Link Layer ed il Network Layer possono essere considerati come dei processi che comunicano mediante lo scambio di messaggi (vedi lucido seguente).

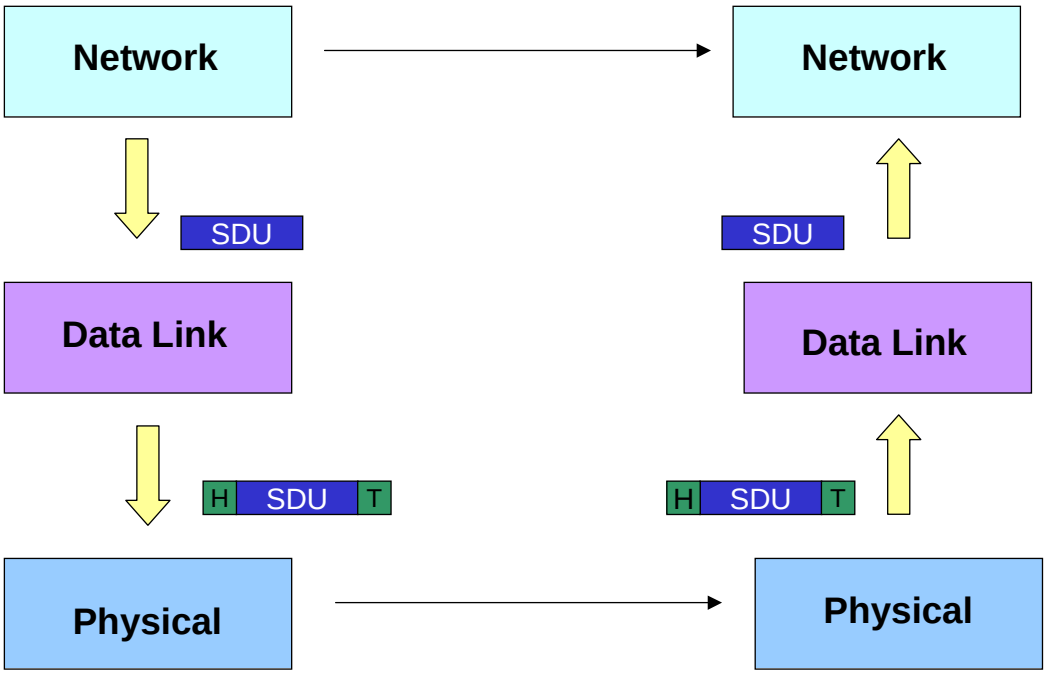


From_physical_layer : ***utilizzata per prelevare pacchetti di dati, (in formato grezzo), dal livello fisico.***

To_physical_layer : ***utilizzata per dare al livello fisico le "frame" arrivate dal N.L.***

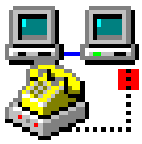
From_network_layer : ***utilizzata per prelevare le frame spedite dall' host remoto.***

To_network_layer : ***utilizzata per spedire frame all'host remoto.***



Attività realizzate nel Data Link Layer

- Impacchettamento o Framing
  - Controllo degli errori
    - Controllo di flusso
      - Rilevazione degli errori
        - Protocolli



FRAMING

- Organizzazione del flusso di dati entrante, in frame.
- Le frame hanno un opportuno e ben definito formato
- Per formato si intende non solo la dimensione di ogni campo della frame ma anche il tipo di codifica usata
- I pacchetti di dati forniti al D.L.L. dal Physical Layer possono essere affetti da errore

Formato delle FRAMES

Synchronization Field	Address Field	Control Field	Data Field	Error check Field	Synchronization Field

Synchronization Field Consente di distinguere l'inizio e la fine di una frame, mediante l'uso di opportuni pattern di bit.

Address Field Contiene l'indirizzo del destinatario/sorgente.

Control Field Contiene una serie di informazioni di controllo utilizzate per individuare il tipo di frame.

Data Field Contiene un pacchetto del Network Layer.

Error Check Field Contiene una sequenza di byte utilizzati per la rilevazione di errori presenti nella frame.

Tipi di delimitatori

- Conteggio dei caratteri
- Caratteri di inizio e fine, con riempimento di caratteri
- Indicatori (flag) di inizio e fine, con riempimento di bit
- Violazione della codifica del livello fisico

Conteggio dei Caratteri

- Il mittente inserisce in testa alla frame da spedire un numero che indica il numero di bit di cui è composto il pacchetto
- Il destinatario riesce a distinguere una frame dalla successiva

VANTAGGI:

Facilità nel distinguere una frame dalla successiva

PROBLEMI:

Disturbi sulla linea di trasmissione falsano il valore del numero di bit della frame ⇨ perdita di pacchetti

Caratteri di Inizio e Fine

- Si aggiungono due insiemi di caratteri ASCII all'inizio e alla fine della frame.

DLE STX cioè **D**ata **L**ink **E**scape **S**tart of **Te**x**t**

DLE ETX cioè **D**ata **L**ink **E**scape **E**nd of **Te**x**t**

VANTAGGI:

Se viene rilevato un pacchetto errato viene automaticamente innescato un meccanismo di risincronizzazione

PROBLEMI:

Possibilità che si presenti una sequenza uguale a quella dei delimitatori.
Una soluzione è l'uso della tecnica "Character Stuffing"

Esempio di Character Stuffing

Data la frame:

DLE	ETX	C	A	D	DLE	S	DLE	ETX

Il mittente per evitare problemi di interpretazione inserirà un ulteriore campo di valore DLE nella frame, cioè:

DLE	ETX	C	A	D	DLE	DLE	S	DLE	ETX

Il destinatario rimuoverà il valore DLE ed otterrà la frame corretta.

Flag di Apertura e Chiusura

- Ogni frame è delimitata da una sequenza di bit detta *FLAG* che è identificata da 01111110.

VANTAGGI:

I pacchetti possono avere un numero arbitrario di caratteri con un numero arbitrario di bit per carattere ed inoltre la risincronizzazione è sempre possibile

PROBLEMI:

Se si presenta una sequenza di bit uguale a quella del flag si utilizza la tecnica “Bit Stuffiing”

Esempio di Bit Stuffing

Data la sequenza di bit:

01111110011101100101111101000110111110

viene modificata dal mittente nella sequenza:

011111100111011001011111010001101111110

Il destinatario, alla ricezione della frame, dopo cinque bit 1 consecutivi, eliminerà lo stuffed bit 0, ottenendo la giusta sequenza.

Una modalità alternativa può essere la sostituzione del quinto bit 1 consecutivo con uno 0 in trasmissione(stuffing) ed il ripristino del valore originale in ricezione (de-stuffing)

Violazione della Codifica

In molte reti (sopratutto LAN) si codificano i bit al livello fisico con una certa ridondanza.

Ad esempio:

- il valore 1 di un bit di dati è codificato con la coppia high/low di bit fisici;
- il valore 0 di un bit di dati è codificato con la coppia low/high di bit fisici.

Codifiche come questa (Manchester encoding, usata in IEEE 802.3) hanno lo scopo di consentire una facile determinazione dei confini di un bit dati, poiché esso ha sempre una trasmissione nel mezzo.

Le coppie high/high e low/low possono quindi essere usate per delimitare i frame.

Controllo degli Errori

Il campo controllo permette di distinguere fra i vari tipi di frame e di realizzare alcuni meccanismi utili per una corretta gestione della comunicazione (ack/controllo di flusso)

- Il controllo di errore viene effettuato in quanto a causa di disturbi elettrici il segnale rappresentante lo stream di bit trasmesso potrebbe essere alterato

- I problemi affrontati dal controllo di errore sono relativi alla sicurezza nella ricezione delle frame da parte del livello di rete remoto e nel controllo relativo all'ordine con cui le frame vengono ricevute

Controllo degli Errori

Possiamo distinguere due casi a seconda del tipo di servizio:

- ***Connection Oriented:*** Il mittente dopo aver spedito una frame attenderà una conferma (*ack)* spedita dal destinatario

Si possono verificare due tipi di problemi:

- *Perdita della frame:* Comporterebbe l'attesa infinita di un ack al mittente.

Soluzione: Uso di time-out per la ritrasmissione.

- *Perdita dell'ack:* Comporta la duplicazione inconsapevole da parte del mittente di frame.

Soluzione: Uso di numeri di sequenza per ricevere tutte le frame in modo ordinato e senza duplicati.

- ***Connectionless:*** Non è possibile alcun riscontro da parte del mittente sull'avvenuta ricezione.

Controllo di Flusso

- Il controllo di flusso viene utilizzato quando il mittente spedisce dati con una velocità maggiore di quella con cui il destinatario vuole, o può, accettarli.
- E' utilizzato per garantire che il destinatario abbia sempre buffer a disposizione in cui memorizzare le frame ricevute in modo da evitare perdite di pacchetti.
- La soluzione a tale problema varia in corrispondenza dei diversi protocolli di comunicazione utilizzati nel DLL. Tutti i protocolli devono comunque vietare al mittente di inviare frame fino a quando non riceverà il permesso dal destinatario.

Codici di rilevazione degli Errori

- Codice polinomiale o CRC code.
- Un codice polinomiale considera una stringa di "k" bit come un polinomio in "k" termini .
- La stringa di bit : 1 1 0 1 si traduce in

  1*x^3 + 1*x^2 + 0*x^1 + 1*x^0
- Le operazioni su tali polinomi, addizione e sottrazione, vengono implementate come "OR ESCLUSIVO".

CRC code o Codice Polinomiale

I passi da eseguire per implementare un CRC code sono i seguenti:

- ✓ Mittente e ricevente devono accordarsi sulla scelta di un Generatore Polinomiale G(x).
- ✓ Porre il bit più alto e quello più basso della stringa da trasmettere a “1”.
- ✓ Calcolare la checksum relativa al pacchetto da trasmettere e accodarlo alla fine del pacchetto.
- ✓ In tal modo siamo certi che il polinomio risulta divisibile per G(x).
- ✓ Se il ricevente, quando ottiene la frame e dividendola per G(x) ottiene un resto diverso da zero siamo sicuri che c’è stato un errore durante la trasmissione.

Premesse sui protocolli

- Il Physical Layer, il Data Link Layer ed il Network Layer sono indipendenti e comunicano attraverso scambio di messaggi;
- Due host A e B per comunicare utilizzano un servizio affidabile orientato alla connessione;
- L’host A che spedisce i dati ha una risorsa infinita, cioè non esiste il tempo di attesa per le loro produzioni.
- Il Data Link Layer quando riceve un pacchetto di dati introduce in esso due ulteriori campi di controllo Header e Trailer.

HD	DATI	TD

- L’hardware di trasmissione di ogni host della rete si occuperà di calcolare la checksum del pacchetto e di inserire il campo CRC ad essa riferito alla fine del pacchetto.

Premesse sui Protocolli

Definiamo delle procedure di libreria standard per la comunicazione utilizzate in molti protocolli.

- ***To_physical_layer***
- ***From_physical_layer***

➔ Utilizzate per inviare e ricevere un pacchetto.

- ***Wait_for_event*(&event)**

➔ Rimane attiva fino a quando non si verifica l'evento relativo alla ricezione di un pacchetto.

- ***To_network_layer***
- ***From_network_layer***

➔ Utilizzate per ricevere ed inviare pacchetti dal D.L.L al livello di rete

Premesse sui Protocolli

- ***Start_timer***
- ***Stop_timer***

➔ Utilizzate per attivare e disattivare il timer.

- ***Enable_network_layer***
- ***Disable_network_layer***

➔Utilizzate in protocolli in cui il livello di rete non ha sempre pacchetti da spedire.

Protocolli SIMPLEX

- Simplex non limitato
- Simplex Stop And Wait
- Simplex per canale disturbato

Simplex non limitato

- Protocollo più semplice da implementare, ma anche il meno realistico.
- Tempo di elaborazione trascurabile;
- Buffer di dimensione infinita;
- Canale di comunicazione privo di errori;
- Dati trasmessi in una sola direzione;
- Procedure *sender* e *receiver* del Simplex non limitato

Simplex non limitato

Procedura sender *:* esegue in un ciclo infinito tre sole operazioni:

1. rileva il pacchetto dal livello di rete;
2. costruisce una frame illimitata in cui memorizza il pacchetto;
3. spedisce la frame.

```
Void sender1(void)
{
frame s;
packet buffer;
while (true) {
        from_network_layer(&buffer);
        s.info=buffer;
        to_physical_layer(&s);

}                }
```

Simplex non limitato

Procedura Receiver *:* esegue in un ciclo infinito tre sole operazioni:

1. attende che *event* assuma il valore *frame_arrival*;
2. preleva la frame dal livello di rete e la memorizza;
3. dispone la frame sul livello di rete.

```
Void receiver1(void)
{
frame r;
event_type event;
while (true) {
        wait_for_event(&event);
        from_ physical _layer(&r);
        to_network_layer(&r.info);

}                }
```

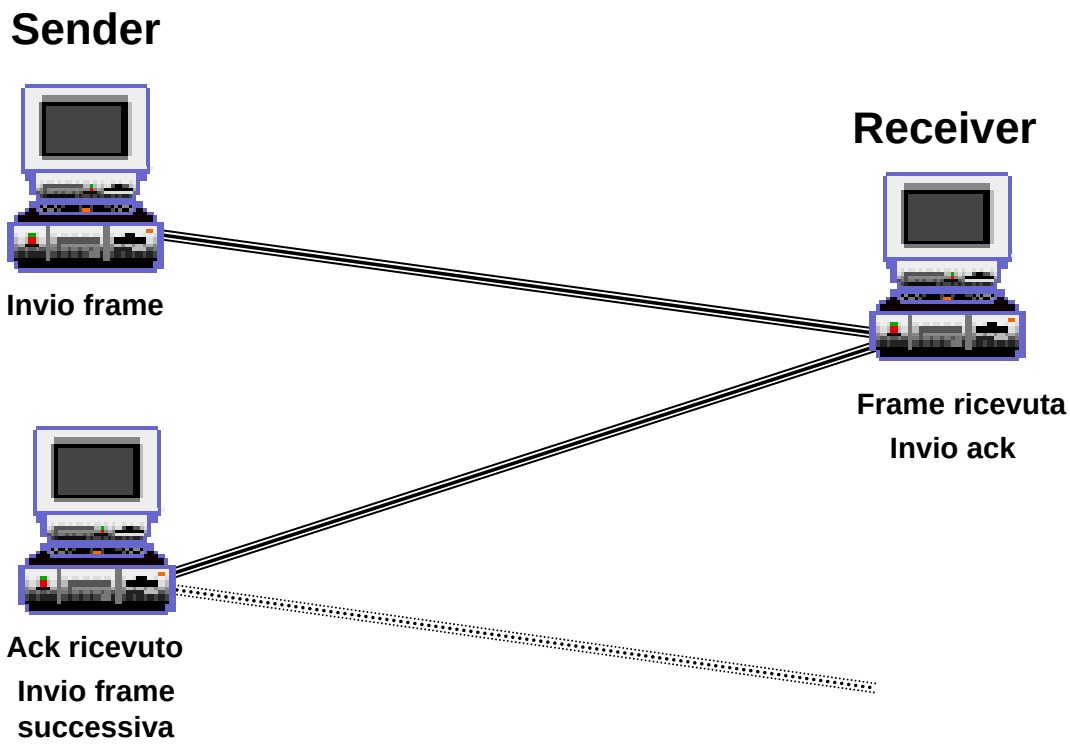
Simplex STOP and WAIT

- Buffer di dimensione infinita;
- Canale di comunicazione privo di errori;
- Dati trasmessi in una sola direzione;

PROBLEMI: Dimensione finita del buffer;
Velocità di elaborazione delle frame da parte del destinatario.

SOLUZIONI:
- ✸ Inserire, nel protocollo del Simplex non limitato, un ritardo nella trasmissione del mittente.
  - ⇨ Uso della larghezza di banda sotto l'ottimo
- ✸ spedire una informazione di "riscontro" al mittente (Ack).
  - ⇨ Il canale fisico è Half Duplex

Simplex STOP and WAIT



Simplex STOP and WAIT

```
Void sender2(void)
{
frame s;
packet buffer;
event_type event;

while (true) {

from_network_layer(&buffer);
      s.info=buffer;
      to_physical_layer(&s);
      wait_for_event(&event);

}              }
```

```
Void receiver2(void)
{

frame r,s;
event_type event;

while (true) {

      wait_for_event(&event);
      from_ physical _layer(&r);
      to_network_layer(&r.info);
      to_physical_layer(&s);

}              }
```

Simplex per Canale Disturbato

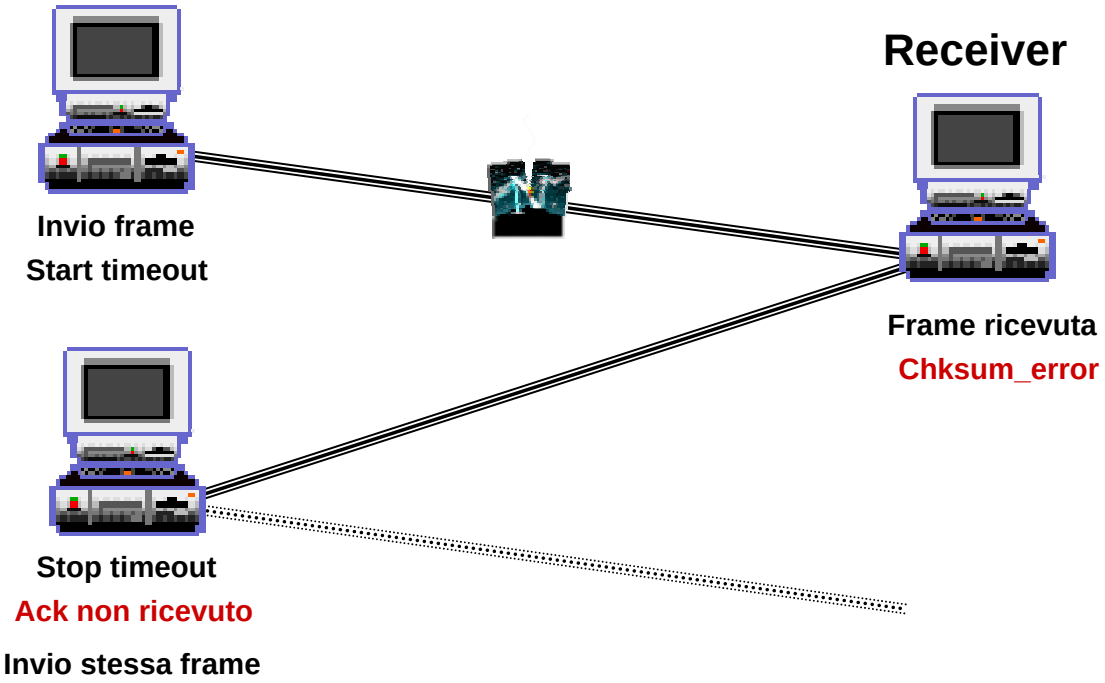
- *Buffer di dimensione infinita;*
- *Canale di comunicazione non esente da errori;*
- *Canale fisico half duplex.*

PROBLEMI:

- *Frame perse*
- *Frame danneggiate*
- *Ack perso (duplicati)*

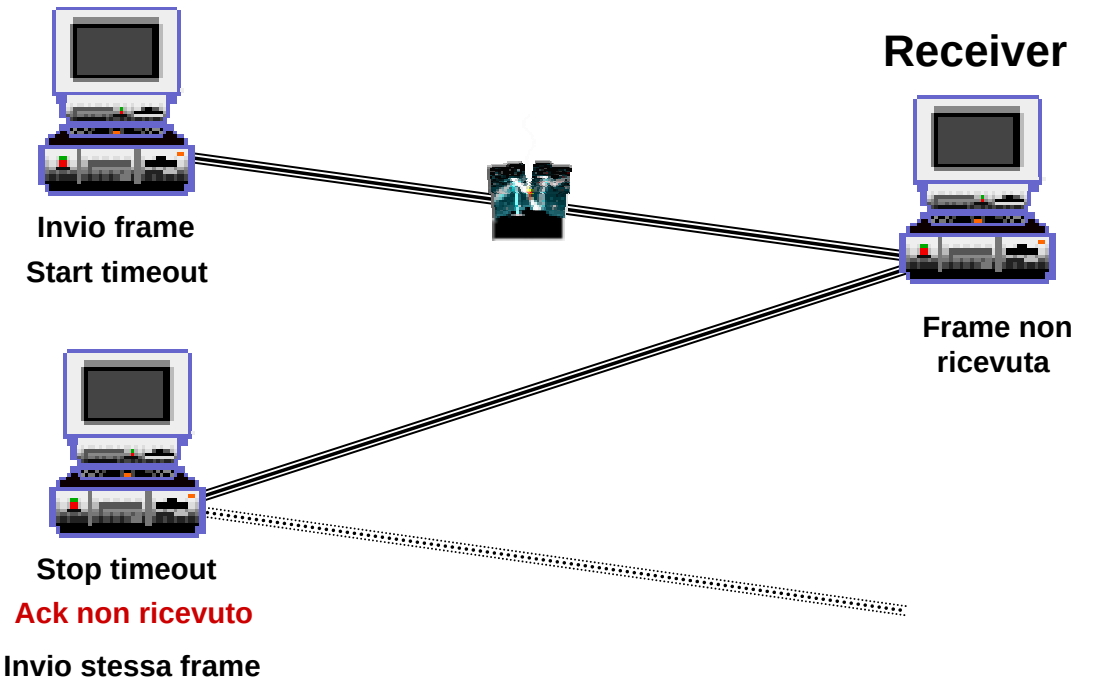
Simplex per Canale Disturbato

La frame ricevuta è danneggiata quindi il ricevente non spedisce l'ack, e il destinatario rispedisce la stessa frame.



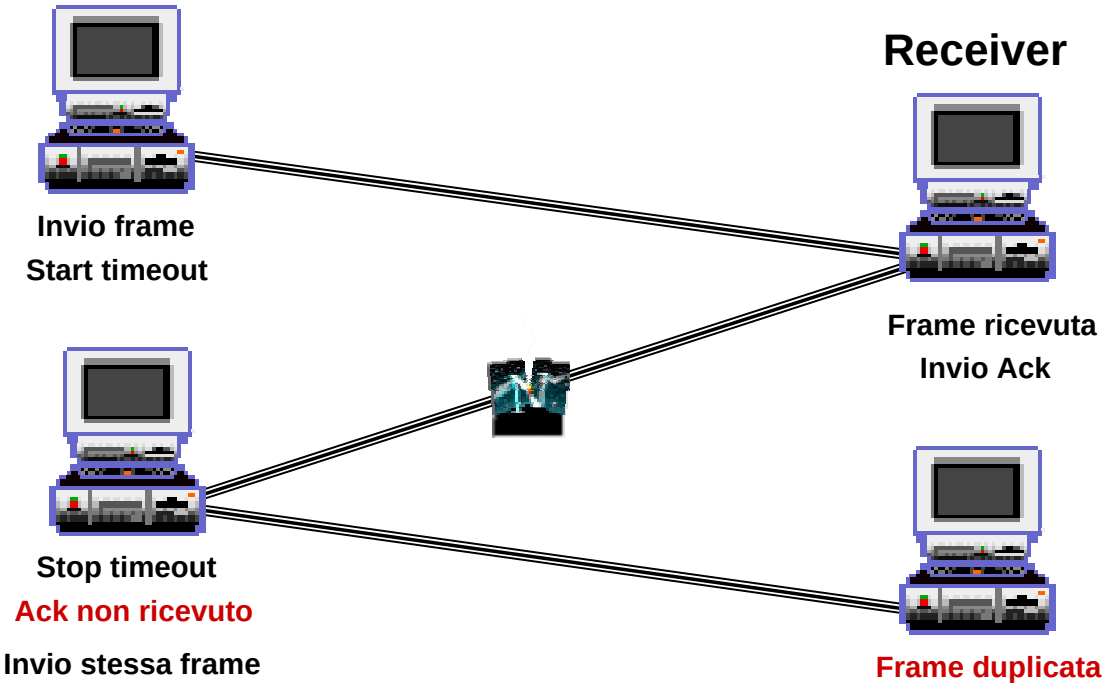
Simplex per Canale Disturbato

La frame non è stata ricevuta quindi il ricevente non spedisce l'ack, e il destinatario rispedisce la stessa frame.



Simplex per Canale Disturbato

La frame è stata ricevuta corretta; il ricevente spedisce l'ack, ma questo si perde, il destinatario rispedisce la stessa frame creando un duplicato.



Simplex per Canale Disturbato

```
Void sender3(void)
{
seq_nr  next_frame_to_send;
frame s;
packet buffer;
event_type event;

next_frame_to_send=0;
from_network_layer(&buffer);

while (true) {
        s.info=buffer;
        s.seq= next_frame_to_send;
        to_physical_layer(&s);
        start_timer(s.seq);
        wait_for_event(&event);
        if (event==frame_arrival) {
             from_network_layer(&buffer);
             inc(next_frame_to_send);
                }                    }
}
```

```
Void receveir3(void)
{
seq_nr_seq_expected
frame r,s;
event_type event;

frame.expected=0;

while (true) {
      wait_for_event(&event);
       if (event==frame_arrival) {
            from_physical_layer(&r);
            if (r.seq==frame_expected)
                 to_network_layer(&r.info);
                 inc(frame_expected);
                                  }
            to_physical_layer(&s);
            }
}
```

Protocolli a Finestra Scorrevole

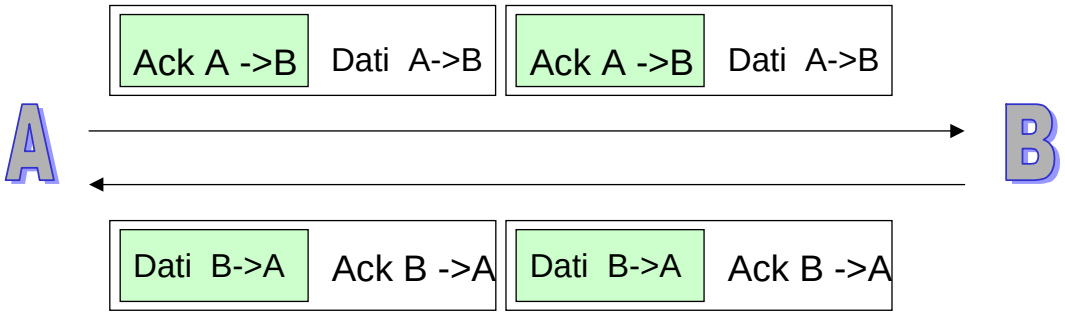
- Sliding window
  - Sliding window di un solo bit
    - Go Back N
      - Selective Repeat

Protocolli a Finestra Scorrevole

- Canale di comunicazione “duplex”.
- Tecnica di utilizzo del canale di comunicazione di tipo ***Piggy Backing.***
- Protocolli **Sliding Window**
- Ogni frame spedita contiene un numero progressivo che varia da zero ad un valore massimo pari a “2^n - 1”.

PIGGYBACKING

Utilizza un solo canale di comunicazione per dati ed ack, con la variante che gli ack non viaggiano da soli ma vengono accodati alle frame che il riceiver deve rispedire al sender.



Vantaggi :

Ottimizzazione dell'uso della larghezza di banda.

Svantaggi :

L'utilità di agganciare un ack alla frame che viene spedita in senso opposto è limitata dal tempo che il DLL può attendere per l'ack.

Se il riceiver non ha frame pronte, l'ack verrà spedito da solo

SLIDING WINDOW

- I protocolli SLIDING WINDOW stabiliscono che sia il mittente che il ricevente mantengano una "finestra" di trasmissione e una di ricezione dove sono memorizzati i numeri progressivi corrispondenti alle frames da ricevere o da spedire.

- Le Finestre associate al mittente e al destinatario non devono avere obbligatoriamente la stessa dimensione e neanche gli stessi limiti inferiori e superiori.

SLIDING WINDOW

- Il meccanismo di spedizione di frame e di ricezione di ack è gestito dal **MITTENTE** tramite l'uso dei numeri progressivi della finestra; questi vengono incrementati di un valore all'estremità superiore della finestra se arriva un nuovo pacchetto dalla rete, oppure vengono incrementati di un valore all'estremità inferiore della stessa se arriva un ack.

- Il mittente deve mantenere in memoria tutte le frame spedite di cui non ha ancora ricevuto l'ack.
  Per tale motivo se la dimensione massima della finestra è "n" il mittente ha bisogno di un buffer di dimensione "n" per la memorizzazione delle frame.

SLIDING WINDOW

- Il meccanismo di spedizione di frame e di ricezione di ack è gestito dal **RICEVENTE** in maniera simile a quella del mittente; infatti il ricevente aumenta di un valore l'estremità superiore della finestra, se arriva un pacchetto con numero progressivo pari all'estremità superiore prima di essere incrementata, oppure aumenta di un valore l'estremità inferiore della stessa quando invia un ack.

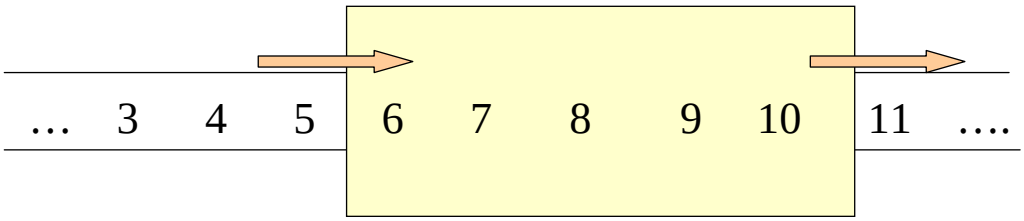
SLIDING WINDOW AD 1 SOLO BIT

In questo protocollo la dimensione massima della finestra è di un bit e la sua implementazione si basa sul protocollo STOP AND WAIT visto precedentemente.

- Vi è un accordo a priori tra i due D.L.L. presenti nelle due macchine su chi deve iniziare per primo la trasmissione.

Una volta stabilita la precedenza inizia la trasmissione; il mittente inizia ad inviare le frame, il ricevente controlla, di volta in volta, se il pacchetto ricevuto è un duplicato o meno utilizzando il protocollo 3.

- Se è il pacchetto atteso viene passato al livello di rete, la finestra viene fatta slittare in avanti e viene inviato un ack con il numero della frame ricevuta.
  Se il mittente riceve un ack con lo stesso numero della frame che cercava di spedire, carica un nuovo pacchetto e lo manda al livello di rete; altrimenti continua a rispedire la stessa frame.



Finestra scorrevole sugli indici dei frame

Limiti:

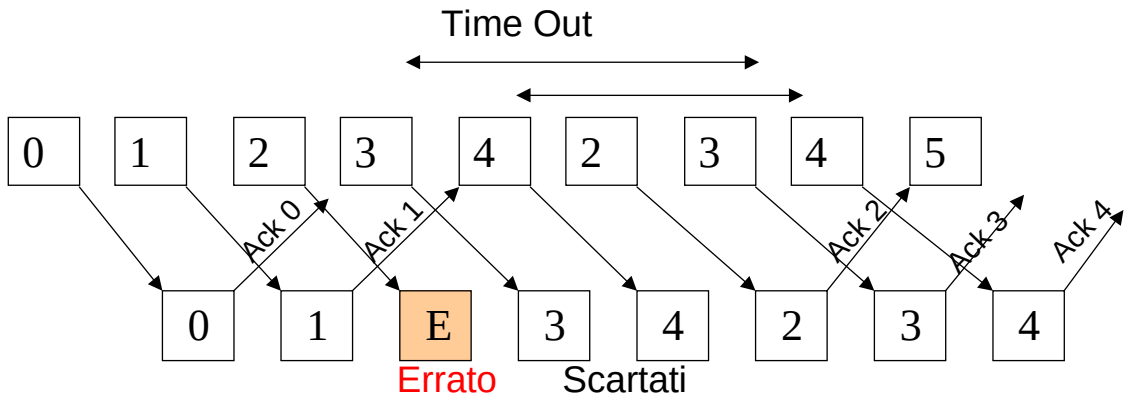
Se il tempo di andata e ritorno del segnale (***round-trip time***) è alto, con una finestra ad 1 bit c'è un elevata inefficienza,perché si sta quasi sempre fermi aspettando l'ack.

GO BACK N

Per migliorare le cose, si può consentire l'invio di un certo numero di frame anche senza aver ricevuto l'ack del primo. Questa tecnica va sotto il nome di ***pipelining***.

• se arriva un frame danneggiato o con un numero di sequenza non progressivo, il destinatario ignora tale frame e tutti i successivi, non inviando i relativi ack. Ciò corrisponde ad una finestra di dimensione uno nel ricevitore, che quindi accetta i frame solo nell'ordine giusto;

• il mittente ad un certo punto va in time-out sul frame sbagliato, e poi su tutti quelli successivi (scartati dal destinatario), e quindi provvede a ritrasmettere la sequenza di frame che inizia con quello per il quale si è verificato il time-out.



Funzionamento del protocollo go-back-n

Limiti:

il mittente deve mantenere in un apposito buffer tutti i frame non confermati per poterli eventualmente ritrasmettere. Se il buffer si riempie, il mittente deve bloccare il livello network fino a che non si ricrea dello spazio. Inoltre, vi è spreco di banda se il tasso d'errore è alto e/o il time-out è lungo.

SELECTIVE REPEAT

Le frame successive ad una persa o danneggiata, non vengono scartate dal destinatario.

- La finestra del ricevente ha dimensione MAX_SEQ costante invece quella del mittente varia da 0 a MAX_SEQ.

Il ricevente è provvisto di un buffer per ogni numero progressivo nella finestra, ad ognuno di questi è associato un bit detto *arrived* che indica se il buffer è pieno o vuoto.

- Per ogni frame si controlla che il suo numero progressivo appartenga alla finestra e non sia un duplicato.
  In tal caso viene memorizzata, altrimenti viene scartata.

La frame ricevuta verrà mantenuta nel D.L.L. fino all'arrivo di tutte le frame con numeri progressivi più bassi in modo da consegnarle al livello di rete nell'ordine corretto.