

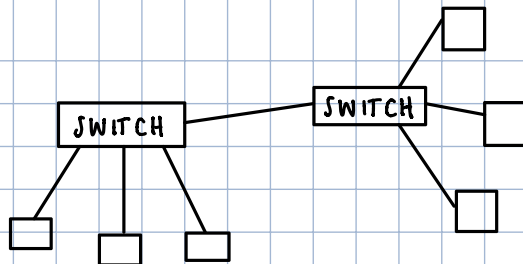
Mancano i primi 5 min



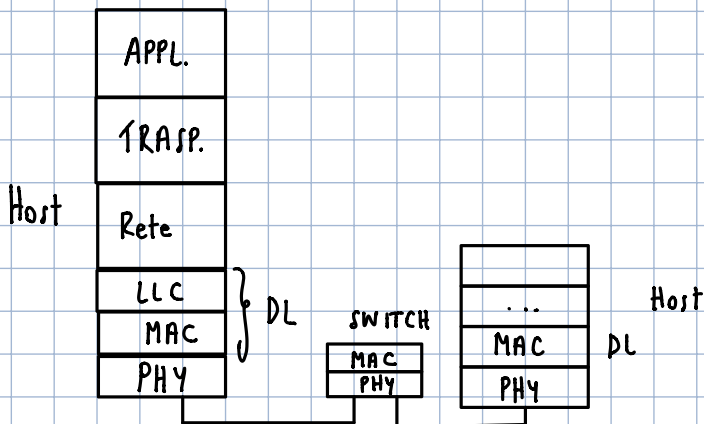
Il livello fisico di Ethernet ha subito diversi sviluppi in quanto sono stati introdotti diversi tipi di portanti fino ad arrivare alla fibra ottica.

L'ethernet più comune sul mercato è 1000BASE-SX, ove 1000 è la velocità utilizzata nel livello fisico mentre SX ci indica il portante (doppini di rame, fibra ottica ecc..)

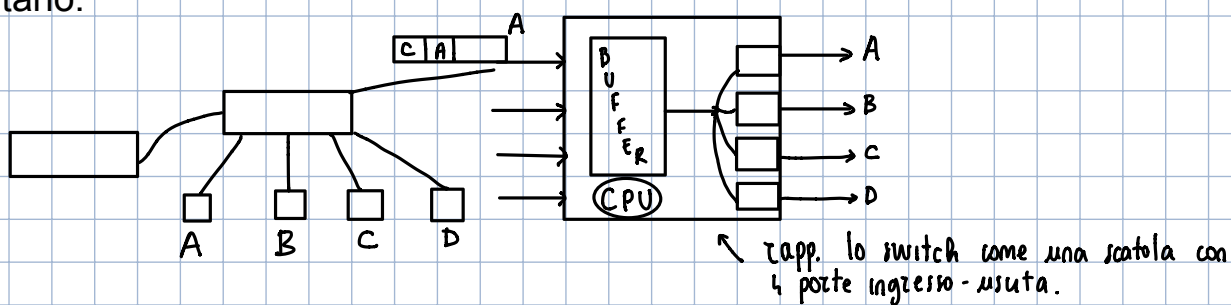
Un rete si realizza: mettendo degli switch collegato a degli host e da ognuna di queste porte può aggiungere anche altri switch.



Architettura protocollare con lo switch :



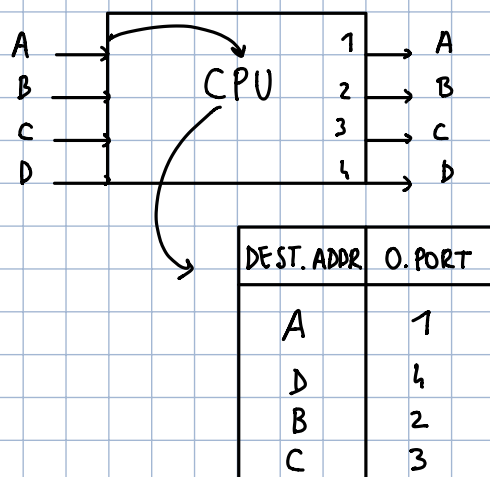
Lo switch commuta, in quanto legge la MAC PDU, vede a chi è destinata e la lancia solo a al destinatario.



Da A entra una trama che finisce nel buffer e viene letto il destinatario, successivamente viene rilanciato ma se ci sono già trame si poggia nel buffer e viene mandata subito dopo. Per sapere dove sta esattamente il destinatario, lo switch ha una tabella (forwarding table).

Vantaggi dello switch:

- Non ci sono mai collisioni.
- Il vantaggio dello switch è che se A scarica un file verso C un hub ripeterebbe questo pacchetto a tutto il resto della rete quindi occuperebbero capacità dove non serve inoltre c'è un problema di privacy. Se mettiamo uno switch questo problema non si pone.



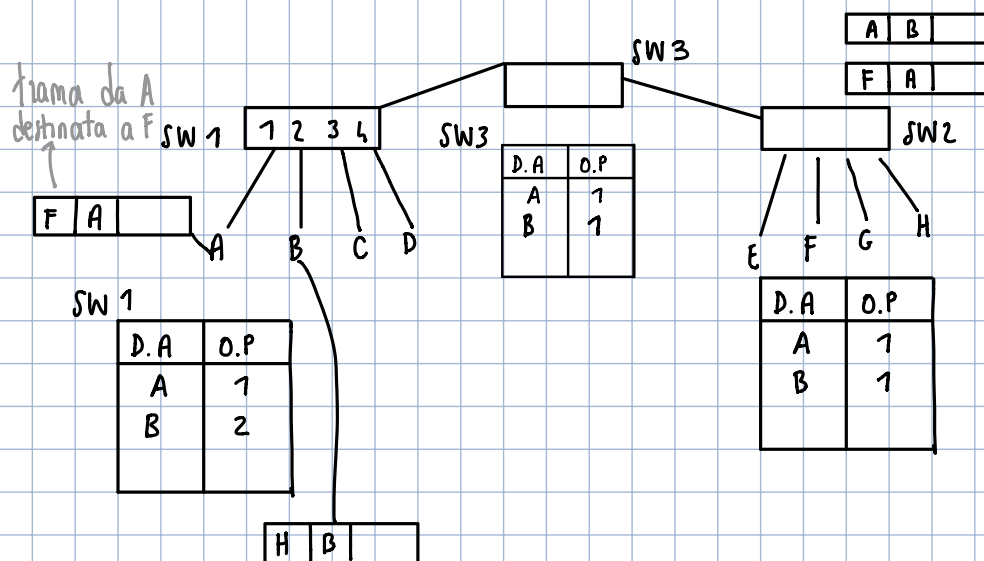
chi passa la tabella allo switch?

ALGORITMO DI ADDRESS LEARNING ADDRESS SELF LEARNING

Inizialmente la tabella è vuota, quindi lo switch si riduce ad hub momentaneamente e manda a tutti (si perde leggermente in efficienza) però si salva poi l'indirizzo di destinazione. Mano mano che riceve trame lo switch apprende.

È importante che questa tabella abbia un soft-state, ossia che le informazioni in essa non siano permanenti. Ci deve essere un time to live: questo tempo se è troppo piccolo non va bene perché lo switch si comporterebbe sempre da hub, se è troppo grande si continuano a mandare pacchetti ad un'uscita che è stato tolto.

Cosa succede se ad uno switch sono collegati altri switch?



Affinchè l'algoritmo di self learning funzioni è necessario che il grafo dei switch è un albero, ossia che ci sia un percorso aperto tra gli switch senno potrebbero esserci più percorsi per una trama.

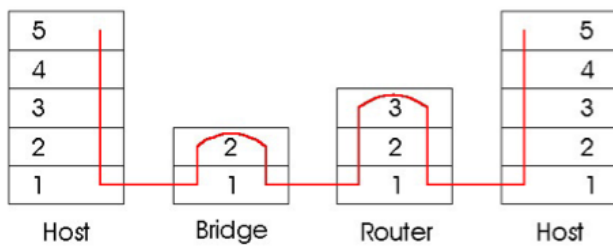
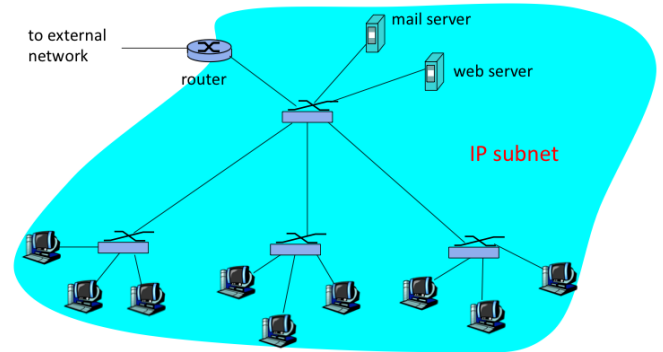
Di solito si tende però a creare grafi, quindi si accoppia al self learning l'algoritmo di spanning tree. Ossia viene cercato un albero che connette tutti i nodi all'interno del grafo.

In passato lo switch si chiamava bridge.

Se volessi collegare una rete Ethernet a una rete Wi-Fi questo non sarebbe possibile a livello MAC, non hanno un linguaggio comune, quest'ultimo viene trovato da un livello superiore ossia quello di rete.

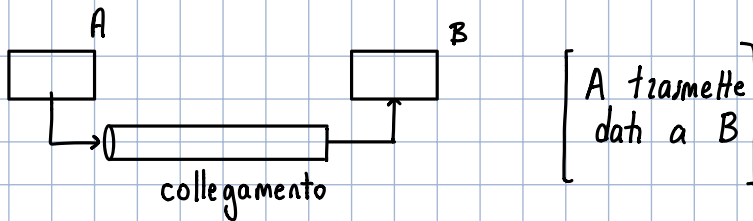
Il router arriva quindi al livello di rete ed è un collante che consente a tutte le reti di comunicare tra di loro.

Il linguaggio comune è IP.



Data link (DL)

I protocolli del DL si occupano del trasferimento affidabile dei dati in un collegamento punto punto. Immaginiamo collegamenti che vanno da un terminale ad un altro e vediamo i protocolli relativi ad essi. La visibilità del DL è sul singolo collegamento.

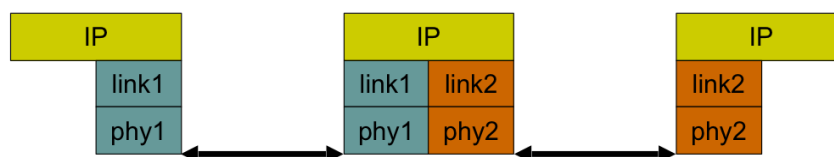


I protocolli di stato di collegamento si possono progettare:

- con connessione: più complessi ma consentono di recuperare errori, controllare il flusso
- Senza connessione

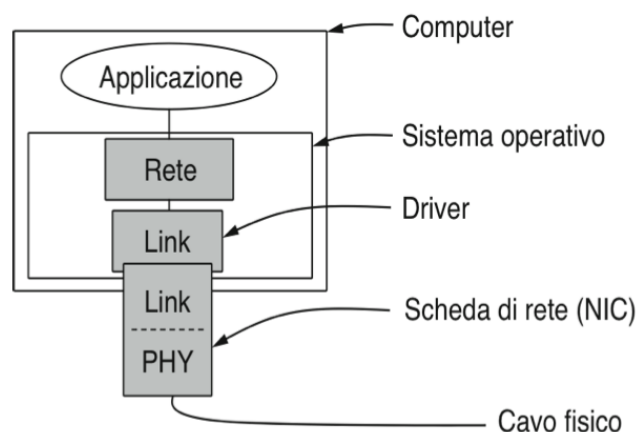
DL: architettura

Consideriamo 3 collegamenti, collegamenti diversi possono avere collegamenti fisici diversi e protocolli di collegamento diversi



Il protocollo viene aperto e chiuso a cavallo di ogni link.

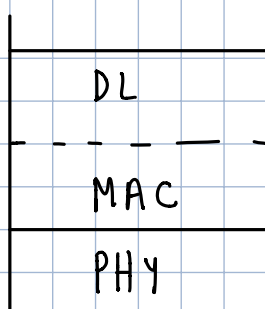
Il DL che si occupa dei protocolli MAC sta nell'interfaccia, la restante sta nel driver ossia librerie software che implementano il data link.



Quali funzioni assolve un protocollo di DL?

Non tutte fanno tutto ma nell'insieme possono :

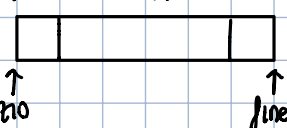
- **Delimitazione** : si occupa di delimitare i pacchetti tra i bit che arrivano dal livello fisico.



← il livello fisico si occupa del trasferimento dei bit.

viene delimitato il pacchetto

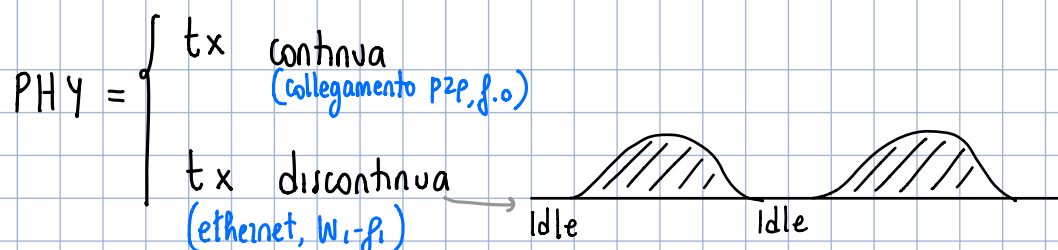
PDU : DL - PDU



- **rivelazione degli errori**
- **recupero degli errori**, avviene attraverso ritrasmissione (funzione principale)
- **Controllo di flusso**, in ogni secondo viene riferito quanti dati possono essere mandati.
Serve a proteggere il ricevitore dal sovraccarico.
- **Gestione**
- Accesso multiplo
- Indirizzamento

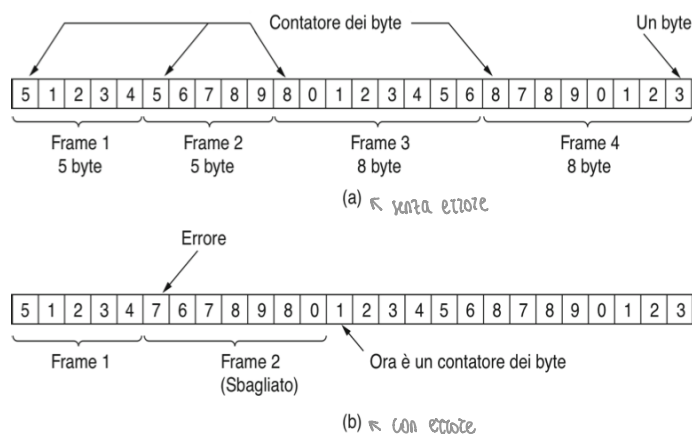
Delimitazione delle trame

Il livello fisico può avere una trasmissione continua e una discontinua



La delimitazione delle trame riguarda solamente quando la trasmissione è continua in quanto quando è discontinua la delimitazione la fa automaticamente il livello fisico.

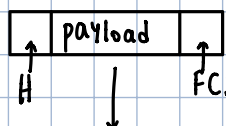
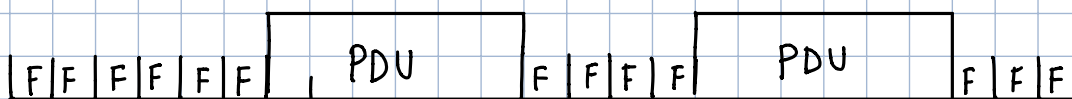
Un modo di delimitare è che all'inizio di ogni PDU scrivo quanto è lunga. Ad esempio scrivo 5 alla prima, 5 alla seconda e 8 alla terza. Scrivo quanti byte è lunga quella trama. Se leggo male un singolo byte sbaglio tutto quello che viene dopo (errore moltiplicativo)



Questo meccanismo è allora troppo delicato perché se sbaglio succede un macello.

Allora una volta creata una PDU ci metto all'inizio e alla fine un byte speciale che si chiama FLAG, dopo di che quando ricevo una sequenza di byte ininterrotta la delimito dicendo che i flag rappresentano l'inizio e la fine di una trama.

Il trasmettitore quando non ha dati trasmette solo flag, poi quando ha la PDU la manda e successivamente altri flag.

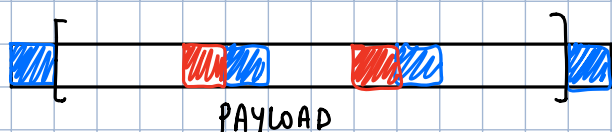


chi mi dice che non ci posso trovare un flag nella PDU?

interromperei prima la lettura del pacchetto.

allora si fa **byte stuffing**

Inserisco un marcatore, un byte di escape che indica che c'è un byte di flag e a sua volta si fa per i byte di ESC.



In questo modo occupo più spazio, raddoppio la lunghezza perché aggiungo un ESC per ogni flag.

I byte di flag senza esc davanti funzionano effettivamente da flag, quelli che hanno davanti un esc fanno sì che si eliminano quei byte.

Lo stuffing viene fatto a livello di DL prima della trasmissione.

$L = \# \text{ byte trama prima del byte stuffing}$

[I byte sono equiprobabili

$$Pr(B=x) = \frac{1}{256} \quad x = b_1, \dots, b_{256}$$

$$E[\# \text{ byte stuffing}] = \underbrace{\frac{L}{256}}_{= \text{n}^\circ \text{ medio byte FLAG}} + \underbrace{\frac{L}{256}}_{= \text{n}^\circ \text{ medio byte ESC}} = \frac{L}{128}$$

Recupero degli errori