

PER ALTRI APPUNTI CONSULTARE IL SITO:
https://luigi-v.github.io/Appunti_Universita/

Una panoramica sulle reti

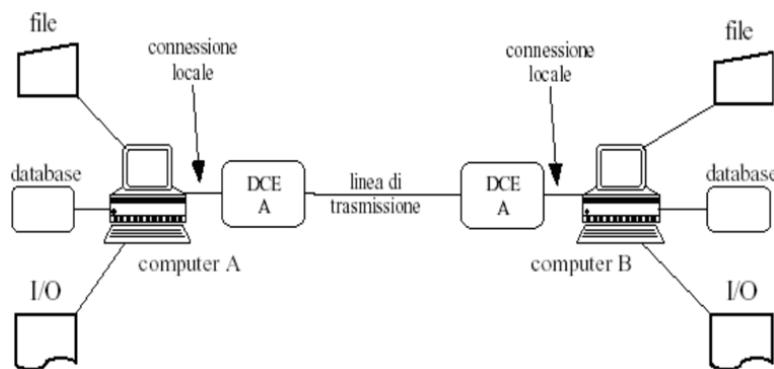
Una rete è un insieme di dispositivi informatici connessi tra loro che permette la trasmissione di dati ed informazioni da un capo all'altro, è composta da componenti **hardware**, quali sono gli **apparati per la trasmissione**, e da componenti **software**, quali sono **protocolli** e **drivers**, in modo da fornire una comunicazione affidabile, efficiente e scalabile.

Una rete deve essere capace di suddividere le informazioni in **pacchetti**, deve poter rilevare e correggere dati corrotti e persi e deve poter trovare cammini ottimali in modo da far arrivare i pacchetti ad una specifica destinazione nel miglior modo possibile.

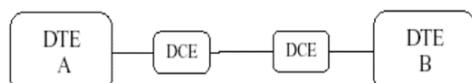
Una rete prevede la presenza di:

- **Hosts**, dispositivi connessi alla rete;
- Nodi di commutazione (**router**), dispositivi che riconoscono l'apertura di una connessione e che permettono ai dati di arrivare a destinazione;
- **Links**, "ponti" che connettono i nodi di commutazione tra loro

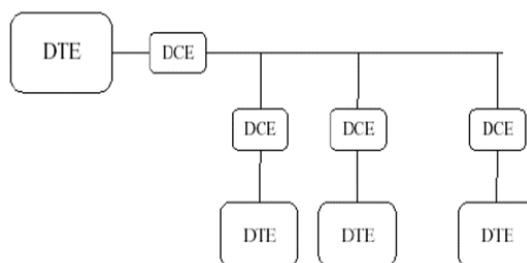
L'Host assume anche il nome di **DTE (Data Terminal Equipment)**, il che può essere un supercalcolatore, un semplice PC o un qualsiasi altro oggetto connesso in rete come utente finale. Ogni DTE è collegato ad una linea di trasmissione tramite un apposito dispositivo, il **DCE (Data Circuit Terminating Equipment)**: se la linea di trasmissione è una linea telefonica, il DCE è un normale modem; in ambito ethernet il DCE è uno switch o un router.



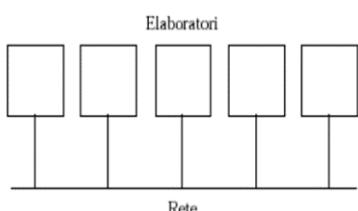
Un circuito fisico è detto **punto-a-punto** se collega solamente **due DTE** tra loro ed è spesso utilizzato nella connessione tra due soli computer oppure nella connessione tra un computer ed un terminale.



Un circuito **multipunto**, invece, consiste nel mettere **più di due DTE** sulla stessa linea. Tale tipo di rete può creare problemi di contesa.



Le reti **broadcast** sono dotate di un unico canale di comunicazione condiviso da **tutti i DTE**, dove i **pacchetti** inviati da un elaboratore vengono **ricevuti da tutti gli altri**: il destinatario viene specificato all'interno del **pacchetto**. Nel caso i destinatari siano tutti gli altri elaboratori si utilizza un particolare indirizzo, in modo tale che tutti gli altri elaboratori prendano in considerazione il pacchetto: in tal caso si parla di **broadcasting**.



Una variante del **broadcasting** è il **multicasting**: in quest'ultimo caso il pacchetto viene considerato da **un sottoinsieme di elaboratori** ed ignorato dai restanti. In ciascun pacchetto è presente un bit che specifica se la connessione sia multicasting o meno. I rimanenti N-1 bit specificano l'indirizzo del destinatario.

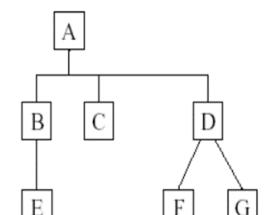
La comunicazione può avvenire in molteplici modi: con la **trasmissione simplex** i dati viaggiano in un'unica direzione; con la trasmissione **half-duplex** i dati viaggiano in entrambe le direzioni ma **non contemporaneamente**; con la trasmissione **full-duplex** i dati viaggiano in **entrambe le direzioni contemporaneamente** ed è particolarmente indicata per le reti multipunto.

La **commutazione di pacchetto (multiplazione statistica)** è l'operazione che suddivide il messaggio in molteplici pacchetti, ciascuno corredata da diverse informazioni di controllo, come l'identificativo di mittente e destinatario, numero d'ordine del pacchetto rispetto all'intero messaggio, eccetera (tali informazioni di controllo costituiscono **l'header** del pacchetto). Ogni **pacchetto viene instradato** indipendentemente e probabilmente su **percorsi differenti** rispetto agli altri pacchetti dello stesso messaggio. È importante sapere che la rete non ne garantisce il corretto arrivo.

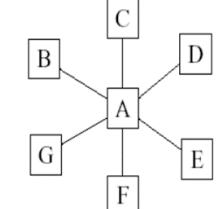
La **commutazione di circuito (multiplazione deterministica)** simula **un unico canale** tra le due stazioni che necessitano di comunicare, dedicando l'intera capacità del canale trasmissivo a quella specifica comunicazione.

Riguardo i vari tipi di **topologie delle reti**, la scelta avviene in base a diversi criteri, quali sono l'affidabilità, la scalabilità, il rendimento e i costi.

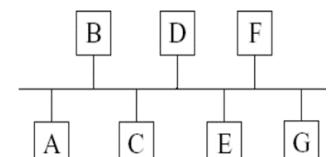
La **rete gerarchica o ad albero** è la rete più comune: il traffico di dati va dai nodi più bassi verso i nodi più alti, fino ad arrivare al nodo più in alto nell'intera struttura, il quale è il più potente siccome deve gestire le richieste di tutti gli utenti inferiori. Il nodo principale, però, può essere un problema per l'intera rete: se è sovraccarico di lavoro, può diventare un **collo di bottiglia** per l'intera rete, comportando un notevole **rallentamento**. Peggio ancora se tale nodo non funzionasse più: in tal caso la rete cadrebbe. Da ciò si possono prendere provvedimenti permettendo ad altri nodi lo svolgimento delle stesse operazioni del nodo principale nel caso in cui quest'ultimo dovesse venire a mancare.



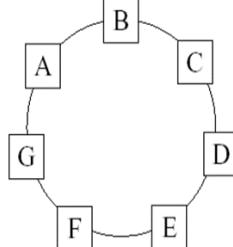
La **rete a stella** ha un nodo centrale in grado di gestire le richieste di tutti gli altri nodi. Ha un funzionamento molto simile alla rete gerarchica, comportando addirittura gli stessi e identici rischi, con la differenza che non vengono utilizzati i livelli.



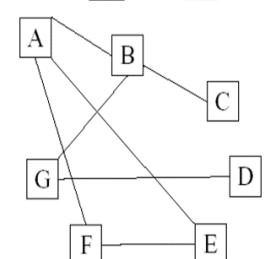
La **rete dorsale o a bus condiviso** viene adottata nelle reti locali di tipo *ethernet* ed è composta da un unico cavo che connette tutti i nodi. La trasmissione di un nodo viene ricevuta da tutti gli altri, quindi può trasmettere un solo elaboratore alla volta mentre tutti gli altri dovranno astenersi. È, quindi, necessario l'utilizzo di un sistema di **arbitraggio** che risolva i conflitti quando due o più nodi vogliono trasmettere contemporaneamente. Anche tale rete offre diversi inconvenienti, i quali sono legati al portante: il portante è unico nella rete, quindi nel caso soffra di problemi prestazionali ne risentirebbe l'intera rete; un'eventuale interruzione del portante metterebbe fuori uso l'intera rete.



La **rete ad anello** fornisce una comunicazione **unidirezionale**, permettendo la comunicazione con qualsiasi stazione siccome si utilizza un circuito chiuso su sé stesso. Anche in questa topologia di rete è necessario un meccanismo di **arbitraggio**, di solito basato sul possesso di un **token** che abilita alla trasmissione.



La **rete a maglia** consiste nel collegare le varie stazioni con diversi circuiti, permettendo ottime prestazioni siccome il traffico verrebbe ripartito su molteplici percorsi. Inoltre, l'esistenza di molteplici percorsi implica alta affidabilità nell'intera struttura. Allo stesso tempo, però, i costi dei collegamenti possono anche essere elevati e la gestione della struttura è altamente complicata.



Un **protocollo** è una serie di norme e **convenzioni** inerenti allo **scambio di dati**, comandi ed informazioni di controllo.

Riguardo le tipologie di reti, veniamo a conoscenza delle seguenti:

- **PAN (Personal Area Network)**, riguarda la connessione tra due dispositivi in un raggio di azione di pochi metri. Di solito tale connessione viene effettuata tramite cavi USB o tramite una soluzione wireless come il bluetooth;

- **LAN (Local Area Network)**, sono possedute da organizzazioni private e hanno una portata che arriva fino a qualche chilometro. Si distendono, in genere, nell'ambito di un singolo edificio o campus e vengono in genere utilizzate per connettere PC e altri dispositivi tra loro. Il mezzo trasmissivo utilizzato è il classico doppino di rame;
- **WAN (Wide Area Network)**, sono reti geografiche che si estendono a livello nazionale, continentale o addirittura planetario. Hanno una portata vastissima, pari a migliaia di chilometri, e un numero di terminali connessi notevolmente elevano (anche migliaia). Tali reti utilizzano la struttura a maglia, in modo da avere molteplici percorsi in cui trasmettere i dati. Una tipica WAN connette molteplici LAN tra loro;
- **MAN (Metropolitan Area Network)**, sono reti metropolitane che hanno un'estensione tipicamente urbana e sono generalmente pubbliche (generalmente si possono utilizzare previo pagamento di un'opportuna tariffa).

In base alla tipologia di rete, cambia il modo con cui viene effettuata la comunicazione tra due **DTE** nella rete, quali sono il **Connection Oriented Mode** (orientato alla connessione) e il **Connectionless Mode** (non orientato alla connessione). La commutazione di circuito utilizza la **Connection Oriented Mode**, mentre la commutazione di pacchetto utilizza la **Connectionless Mode**.

Nel **Connection Oriented Mode**:

1. I due DTE si assicurano della reciproca **presenza in linea, prima** di effettuare lo scambio dei dati;
2. Fatta tale verifica, viene instaurata la connessione* (o **sessione**), la quale durerà tutto il tempo richiesto dallo scambio dei dati;
3. Finito lo scambio, viene **terminata la connessione** (o sessione).

La connessione è gestita dal software dei due DTE, il quale svolge diverse funzioni, quali sono la gestione della velocità di scambio, il controllo delle regole di scambio, la capacità di interruzione della controparte e il controllo degli errori con eventuale correzione. Tali controlli sono fondamentali nelle WAN, data la bassa affidabilità delle linee.

Nel **Connectionless Mode**, invece, un DTE può inviare dati all'altro DTE anche senza aver instaurato una connessione*. Il vantaggio principale rispetto al **Connection Oriented Mode** è che non sono necessari servizi di controllo o di supporto, vantaggioso per le LAN ma non opportuno per le WAN, data la scarsa affidabilità. Il problema fondamentale del **Connectionless Mode** riguarda il controllo degli errori che possono verificarsi: non essendoci controlli immediati durante la trasmissione, il DTE sorgente non può sapere com'è andata la trasmissione. La soluzione è quella di affidare il controllo degli errori alle applicazioni.

Una **intranetwork** è una rete formata da molteplici reti diverse collegate tra loro. La definizione può sembrare simile a quella delle reti WAN, ma strutturalmente sono diverse: la **intranetwork** utilizza apparecchiature speciali dette **gateway**, le quali rendono possibili i trasferimenti tra reti diverse.

La seguente tabella illustra in maniera completa le differenze tra Connection Oriented e Connectionless. Con circuito virtuale si intende il path nella sottorete che i pacchetti seguiranno.

Caratteristica	Connection Oriented	Connectionless
Creazione del circuito	Richiesto	Non richiesto
Indirizzamento	Ogni pacchetto contiene un piccolo numero, il virtual circuit (VC)	Ogni pacchetto contiene gli indirizzi sorgente e destinazione completi
Informazioni di stato	Ogni circuito virtuale richiede spazio di tabella nella sottorete	La sottorete non conserva informazioni di stato
Instradamento	Percorso scelto alla creazione del circuito virtuale: tutti i pacchetti seguiranno tale percorso	Ogni pacchetto viene instradato singolarmente
Effetti dei guasti nei router	Tutti i circuiti che passano attraverso il router guasto vengono terminati	Nessuno, al massimo viene perso qualche pacchetto durante il guasto
Controllo di congestione	Semplice	Complesso

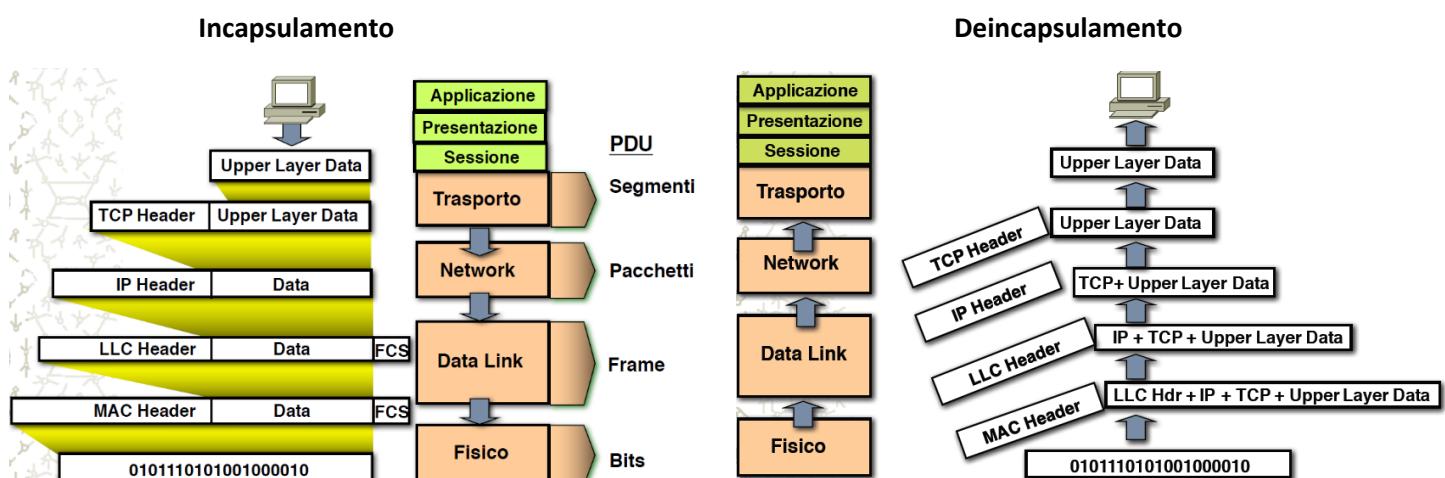
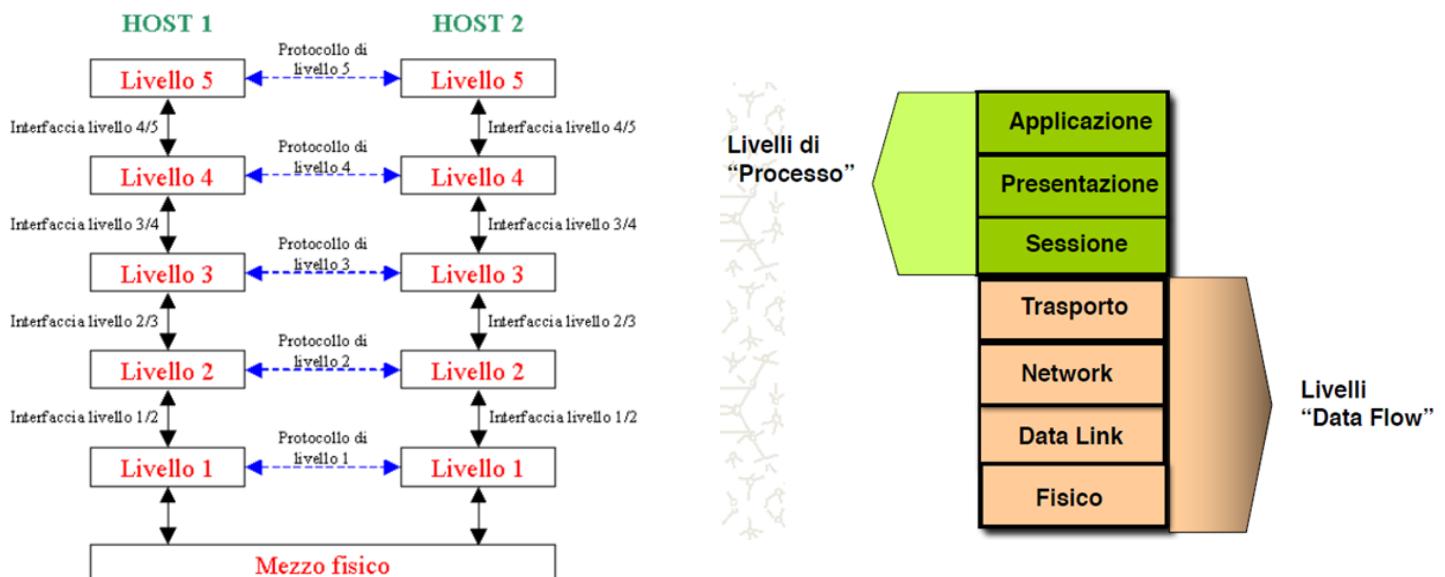
[*con "instaurare una connessione" si intende la scelta di un path verso cui mandare i pacchetti. Sostanzialmente, con il Connection Oriented viene stabilito un path; con il Connectionless non viene stabilito alcun path, quindi ogni pacchetto viene spedito indipendentemente dagli altri.]

Modello ISO-OSI

Il **modello ISO-OSI** è costituito da una pila (o **stack**) di protocolli i quali regolano l'implementazione di un sistema di comunicazione. Ogni livello fornisce un diverso livello di astrazione e individua un protocollo di comunicazione del livello medesimo. Concetto fondamentale di tale modello è che ***ogni livello incapsula i messaggi del livello n in messaggi del livello n-1***: sostanzialmente i messaggi dei livelli più bassi incapsulano i messaggi dei livelli più alti. In questo modo viene realizzata una **comunicazione multilivello**, conferendo modularità al sistema con maggiore semplicità di progettazione e gestione della rete.

I livelli del modello ISO-OSI sono i seguenti (dal più basso al più alto):

- **Livello fisico:** si occupa della trasmissione di ogni singolo bit attraverso un canale di trasmissione;
 - **Livello data link:** si occupa del controllo degli errori di comunicazione, in modo da far apparire la trasmissione senza errori ai livelli superiori;
 - **Livello di rete:** gestisce l'instradamento dei pacchetti di dati verso la stazione destinataria nel migliore dei modi possibili;
 - **Livello di trasporto:** in ricezione si occupa di memorizzare e riordinare ciò che è in arrivo dai livelli inferiori per poi passare il tutto al livello superiore, altrimenti si occupa di suddividere i messaggi provenienti dallo strato superiore e di controllare eventuali errori;
 - **Livello di sessione:** si occupa di gestire le modalità di dialogo (simplex, half-duplex, full-duplex) tra gli elaboratori;
 - **Livello di presentazione:** gestisce il formato della codifica dei dati e la sua eventuale conversione. Le entità di questo livello si occupano anche della compressione dei dati e delle tecniche di crittografia;
 - **Livello di applicazione:** contiene tutti i servizi e protocolli per l'utilizzo della rete.



Livello Fisico

I **segnali** sono variazioni di grandezze fisiche capaci di trasportare informazioni e possono essere di due tipi:

- **Analogici**, sono quei segnali che possono assumere un qualsiasi valore compreso tra un certo valore massimo ed un certo valore minimo consentiti dal canale di comunicazione;
- **Digitali**, o *numerici*, sono quei segnali che possono assumere solo due o un numero discreto di valori.

Un segnale di **periodo T** può essere sviluppato in **serie di Fourier** in una **somma di infinite sinusoidi di ampiezza base** alla banda del canale di comunicazione, quale è un range di frequenze che il canale di comunicazione fa passare.

Un **segnale analogico** è formato da **molteplici onde sinusoidali**. Ogni ciclo consiste in un arco che va al di sopra della linea del tempo e in un arco che va al di sotto della linea del tempo. Un'onda sinusoidale è caratterizzata da tre parametri, quali sono:

- **Aampiezza**, indica l'altezza del picco dell'onda sinusoidale ed è proporzionale all'energia trasportata;
- **Frequenza**, è il numero di onde sinusoidali in un secondo, quindi il numero di periodi T (un periodo è un'onda sinusoidale completa, quindi un intero ciclo di segnale) e si indica con $f = 1/T$;
- **Fase**, descrive la posizione dell'onda rispetto al tempo 0. Indica la posizione del primo ciclo ed è misurata in gradi o radianti.

Livello Fisico: campionamento

Il **campionamento** non è altro che la conversione **da segnale analogico a segnale digitale**. Il teorema del **campionamento di Nyquist** afferma che: dato un segnale con banda limitata B, si può ricostruire il segnale se la frequenza di campionamento è maggiore o uguale a $2B$. In generale, la frequenza di campionamento dovrà essere leggermente maggiore a $2B$, in modo da stabilire un piccolo range extra (**banda di guardia**) in modo da evitare che i filtri taglino parti utili del segnale.

Livello Fisico: capacità del mezzo

Shannon e Nyquist hanno enunciato teoremi che esprimono la massima velocità di trasmissione per ogni tipo di canale. Il **teorema di Nyquist** permette di stabilire la **massima quantità di informazione trasmessa (bit rate)** in un canale non rumoroso. Il **teorema di Shannon** permette di stabilire la **massima quantità di informazione trasmessa (bit rate)** in un canale rumoroso.

L'aumento dei livelli trasmittivi comporta l'aumento della quantità di informazioni che vengono trasmesse nello stesso tempo. L'aumento dei livelli comporta che il singolo livello diventi sempre più piccolo, rendendolo addirittura indistinguibile a causa del **rumore**, il quale è sempre presente.



Livello Fisico: rumori

Il rumore è una forma di energia indesiderata che influisce sul segnale utile, degradandone l'informazione. È considerato, quindi, un disturbo. I principali rumori sono i seguenti:

- Rumore bianco, energia distribuita equamente in tutte le frequenze;
- Rumore di intermodulazione, causato dalla non linearità dei dispositivi elettronici e consiste nella presenza di armoniche indesiderate nel segnale di uscita, originariamente non presenti nel segnale di ingresso;
- Rumore di modo comune, rumore presente in ingresso da uno strumento di misura, unito al normale segnale da misurare;
- Rumore di quantizzazione, perdita di informazione che ha luogo durante la trasformazione di un segnale analogico in digitale;
- Rumore termico, dovuto all'agitazione termica degli elettroni presenti in una resistenza. Tale rumore è considerabile anche un rumore bianco.

Livello Fisico: trasmissione dei segnali

Parlando della trasmissione dei segnali, è detta **analogica**, se il segnale viene trasmesso senza curarsi del suo significato: in tal caso la trasmissione si limita a recapitare il segnale, magari amplificandone l'intensità quando necessario. La **trasmissione digitale**, invece, tiene conto del contenuto dei dati nel caso si debba amplificare l'intensità del segnale. Siccome l'amplificazione del segnale danneggierebbe il contenuto informativo, nel caso della trasmissione digitale quest'ultimo viene estratto e si rigenera il segnale tramite opportuni ripetitori.

Una volta generato il segnale da trasmettere, questo può essere **immesso direttamente sul canale**: in tal caso si parla di **trasmissione in banda base**. Nel caso si abbia la necessità di trasmettere il segnale su frequenze diverse, si adopera la **modulazione**.

Riguardo la codifica dei dati, i dati numerici vengono rappresentati dai segnali tramite **sequenze di impulsi discreti**. Il dato binario è quindi codificato in modo da far corrispondere il valore di un bit ad un certo livello del segnale. Il ricevitore, cioè il dispositivo che riceverà i segnali, dovrà sapere quando inizia e quando finisce il bit, leggere il valore del segnale al momento giusto e determinare il valore del bit in base alla codifica utilizzata. La migliore valutazione si ottiene effettuando il campionamento del segnale al tempo pari a metà del bit.

La codifica adottata per determinare il valore dei bit viene scelta in base ad alcuni criteri, quali sono:

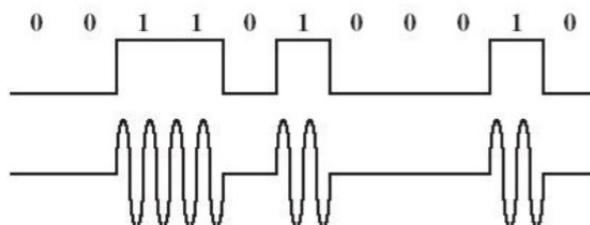
- **Spettro del segnale:**
 - Le alte frequenze richiedono bande maggiori;
 - L'assenza di componente continua è preferibile.
- **Sincronizzazione temporale:**
 - Il ricevitore deve essere sincronizzato con il trasmettitore per identificare i bit.
- **Rilevazione di errori** (caratteristica dei livelli superiori, ma può essere utile anche nel livello fisico).

Livello Fisico: modulazione

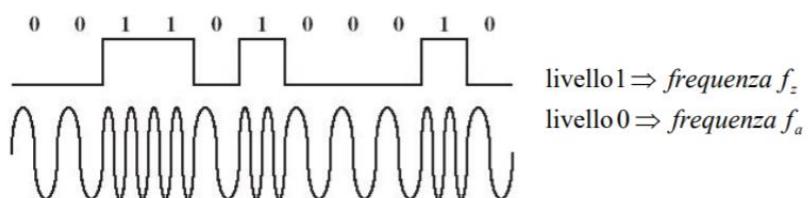
La **modulazione** è un processo che permette di associare un segnale generalmente contenente informazioni (segnales **modulante**) ad un altro segnale (segnales **portante**), sviluppato ad alta frequenza (frequenza portante > frequenza modulante). Il risultato di tale processo è la **conversione** del **segnales modulante** dalla banda base alla banda traslata, generando quindi un nuovo segnale (segnales **modulato**) la cui banda sarà la **traslata**. Utilizzando una **portante** ad alta frequenza, quindi, si può spostare la banda necessaria alla trasmissione in un range più opportuno.

Le **tecniche** di **modulazione** principali che vedremo riguardano **l'ampiezza**, la **frequenza** e la **fase**.

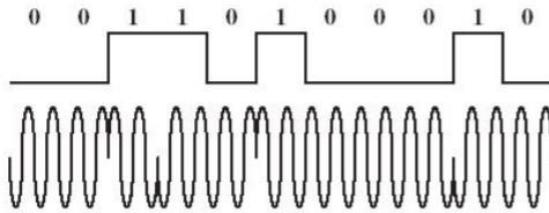
- **Modulazione ASK (Amplitude Shift Keying)**: partendo da un segnale numerico (segnales **modulante**), è possibile modulare in **ampiezza** un segnale **portante** moltiplicando la sua ampiezza per il segnale numerico.



- **Modulazione FSK (Frequency Shift Keying)**: partendo da un segnale numerico (segnales **modulante**), è possibile modulare un segnale **portante** modificando la sua **frequenza** in funzione del segnale **modulante**, così facendo corrispondere le due frequenze a due valori del bit.



- **Modulazione PSK (Phase Shift Keying)**: partendo da un segnale numerico (segnales **modulante**), è possibile modulare in **fase** un segnale **portante** associando un certo valore di fase ad un certo valore di bit.



Altre tecniche di modulazione sono le seguenti:

- Modulazione BPSK (Bifase PSK) consiste nell'associare ai due bit due valori diversi di fase della portante;
- Modulazione 4PSK utilizza ben quattro valori di fase diversi della portante, quali sono 0° , 90° , 180° , 270° , ognuno associato a 2 bit di codifica;
- Modulazione 8PSK utilizza ben otto valori di fase diversi della portante, ognuno associato a 3 bit di codifica;

Livello Fisico: multiplazione

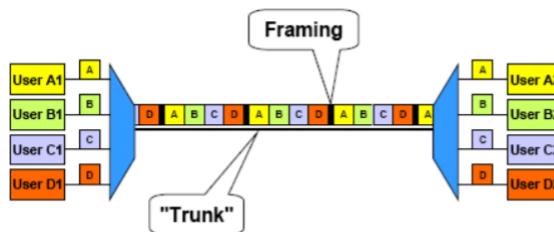
La **multiplazione** è la tecnica di trasmissione che permette di **combinare più segnali analogici o digitali** in un **solo segnale** (detto **multiplato**) trasmesso in uscita su uno stesso collegamento fisico.

Prima di vedere le tecniche di multiplazione, è bene sapere che:

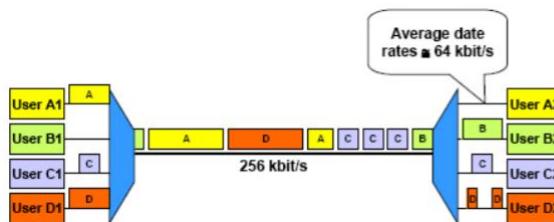
- Ogni **intervallo temporale** si chiama **slot** e può contenere uno o più bit;
- Il **flusso dei dati** è organizzato in **trame**. Una trama è **l'insieme di tutti gli slot** utilizzati ad una specifica frequenza;
- Il **flusso dei dati relativo ad una sola linea del tempo** viene detto **canale**.

Le tecniche di multiplazione che vedremo sono le seguenti:

- **TDM (Time Division Multiplexing) (per segnali digitali)**, tecnica di multiplazione secondo la quale ogni dispositivo ottiene a turno l'uso esclusivo del canale di comunicazione per un breve lasso di tempo, quindi con la possibilità di sfruttare l'intera banda per sé. Esistono due metodi di multiplazione TDM:
 - **Deterministico**: le **trame** vengono **allocate in modo fisso e nell'ordine stabilito**. Un grande svantaggio del TDM deterministico è il caso nel quale non venga utilizzato il mezzo trasmittivo: verrebbero inviate **trame idle**, quindi vuote, in modo da sprecare tempo e canale. I vantaggi è che le trame sono già ordinate e non sono necessari schemi di indirizzamento.



- **Statistico**: le **trame** vengono **allocate in modo dinamico**, in base alla quantità di dati da spedire. Gli svantaggi del TDM statistico riguardano la necessità di uno schema di indirizzamento e il fatto che sia più lento rispetto al TDM deterministico. Il vantaggio riguarda, invece, il fatto che le trame vengono allocate dinamicamente evitando sprechi di tempo e canale, comportando quindi ad un ottimo utilizzo del mezzo trasmittivo.

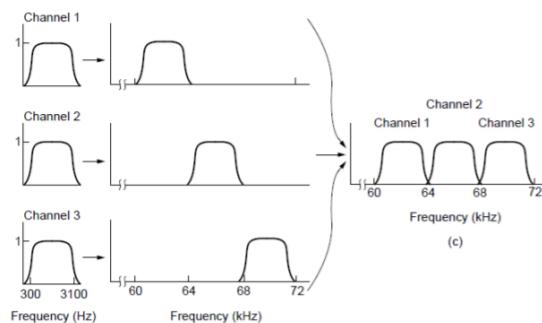


- **SDM (Space Division Multiplexing) (per segnali digitali)**, tecnica di multiplazione secondo la quale ad **ogni dispositivo** viene associato un **proprio canale**. La differenza principale dal TDM è proprio l'utilizzo dei canali:

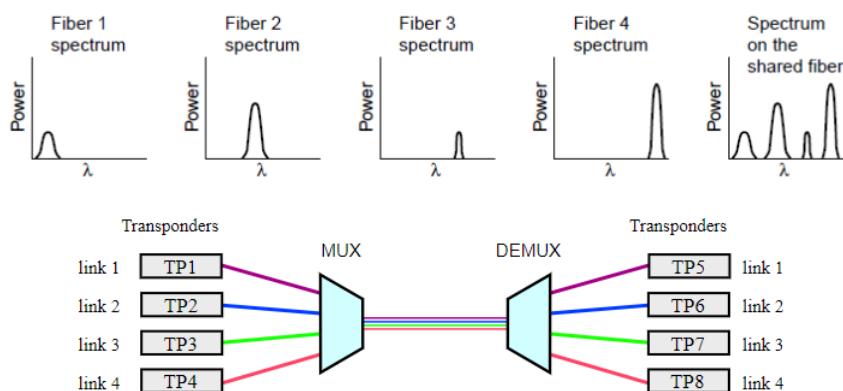
mentre più dispositivi possono usare un unico canale di comunicazione, nel SDM è presente un canale di comunicazione per ogni singolo dispositivo.



- **FDM (Frequency Division Multiplexing)** (*per segnali analogici*), tecnica di multiplazione secondo la quale *l'intero canale trasmissivo è diviso in sottocanali*, ognuno costituito da una banda di frequenza e separato da un altro grazie ad un piccolo *intervallo di guardia*. Ciò rende possibile la condivisione dello stesso canale da parte di diversi dispositivi che utilizzano bande di frequenze diverse, in modo da poter comunicare contemporaneamente senza problemi di interferenza. In ricezione, opportune operazioni di demodulazione permetteranno di separare i diversi traffici.



- **CDM (Code Division Multiplexing)**, tecnica di multiplazione che *moltiplica l'informazione binaria per* una certa parola di codice detta **chip**. La sequenza in uscita dal moltiplicatore sarà successivamente modulata e trasmessa sul canale;
- **WDM (Wave Division Multiplexing)** (*per segnali ottici*), tecnica di multiplazione molto simile alla FDM che *consente di veicolare molteplici lunghezze d'onda* all'interno dello stesso portante fisico.



Livello Fisico: mezzi trasmissivi

I mezzi trasmissivi non sono altro che i mezzi con cui vengono trasportati dati ed informazioni. Essi sono divisi in **wired** e **wireless**: i mezzi trasmissivi **wired** sono i soliti cavetti di rame (doppino, cavo coassiale) e la fibra ottica; i mezzi trasmissivi **wireless** sono i suoni, la luce, i raggi infrarossi, la radiofrequenza e gli infrarossi.

Ogni mezzo trasmissivo è caratterizzato dalla banda, dal **delay**, dal costo e dalla facilità di installazione e manutenzione. I mezzi trasmissivi **wired (elettrici)** sono attualmente i più utilizzati, in particolare nelle reti locali: tali mezzi devono rendere sicura la trasmissione dell'energia da un estremo all'altro con il minimo disturbo. Con le tecnologie attuali è possibile realizzare mezzi trasmissivi **wired** che permettano la trasmissione dei dati ad una velocità fino a 1 Gb/s (1000 Mb/s).

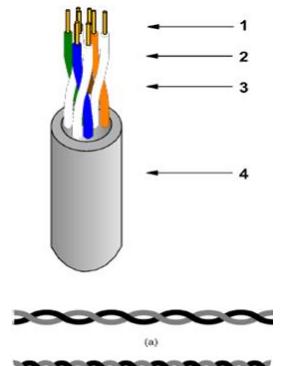
Purtroppo, un mezzo trasmissivo ideale, cioè che non riceva il minimo disturbo o distorsione, non esiste. Piuttosto esistono i mezzi trasmissivi ideali, i quali sono caratterizzati da bassa dissipazione e bassa esposizione ai disturbi: con tali mezzi, quasi tutta la potenza trasferita viene ricevuta dal ricevitore ed il segnale non viene distorto.

Inoltre, tali mezzi sono soggetti anche a **disturbi elettromagnetici**, ai quali ultimamente si sta prestando particolare attenzione. L'utilizzo delle **schermature** ed una **corretta messa a terra** riduce drasticamente i disturbi elettromagnetici, migliorando notevolmente le caratteristiche del cavo. Le schermature più utilizzate nelle LAN sono fogli di alluminio e trecciole di fili di rame che avvolgono il cavo.

Altro tipo di disturbo è la **diafonia**, il quale è un fenomeno che si presenta in caso di accoppiamento elettrico tra mezzi trasmissivi vicini non isolati adeguatamente: il segnale trasmesso su un cavo genera un segnale nel cavo vicino, il quale si sovrapporrà al segnale di quest'ultimo.

Livello Fisico: doppino

Il **doppino** è il classico mezzo trasmissivo della telefonia e consiste in due fili di rame attorcigliati tra loro ricoperti da una guaina isolante. Tale guaina, detta **binatura**, riduce i disturbi elettromagnetici, in particolare quando si utilizzano cavi con più coppie: nel caso di cavi con più coppie, è necessario adottare passi di binatura differenziati da coppia a coppia per ridurre la **diafonia**. Infatti, se i passi di binatura fossero uguali, ogni conduttore di una coppia si troverebbe affiancato con uno dei due conduttori dell'altra coppia, aumentando la diafonia. Nati come mezzo trasmissivo con banda molto ridotta, attualmente i doppini hanno raggiunto elevate prestazioni grazie ai nuovi materiali isolanti utilizzati.



Livello Fisico: cavo coassiale

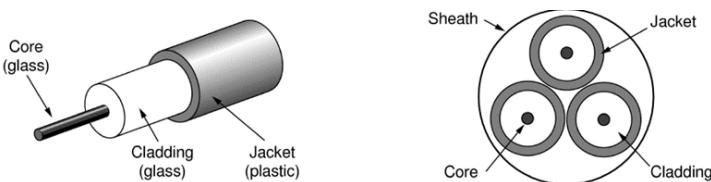
Il cavo coassiale, oggi giorno, è stato sostituito dai doppini e dalla fibra ottica. Viene utilizzato solamente nelle reti geografiche. Il cavo coassiale a banda base consiste di un filo di rame rigido circondato da una garza metallica che funge da schermo. La banda di tale cavo dipende dalla lunghezza del cavo: più è corto, più è possibile aumentare la velocità di trasmissione. Il cavo coassiale a banda larga, invece, usa la trasmissione analogica, quindi molto simile alla trasmissione televisiva.

Livello Fisico: trasmissione Power-Line

La trasmissione Power-Line è una tecnologia di trasmissione dati che utilizza la rete di alimentazione elettrica come mezzo trasmissivo: si sovrappone alla corrente elettrica una frequenza più elevata su cui è stata modulata l'informazione da trasmettere.

Livello Fisico: fibra ottica

La fibra ottica è un cavo composto da un'anima trasparente di silicio puro (**core**) avvolta in un rivestimento di silicio puro con indice di rifrazione diverso (**cladding**). La parte in silicio è ricoperta da una guaina di plastica nera. È importante sapere che core e cladding devono avere indici di rifrazione diversi, in particolare nel cladding dovrà essere minore rispetto all'indice nel core.



Il **core** è il nucleo centrale in cui viaggia la luce, il **cladding** è il suo rivestimento: la luce entra nel core ad un certo angolo e si propaga mediante una serie di riflessioni generate nell'impatto con il **cladding**.

Normalmente, molteplici fibre ottiche sono raggruppate insieme intorno ad un filo di metallo che facilita la posa del cavo.

La fibra presenta molteplici vantaggi, quali sono l'immunità ai disturbi, banda decisamente alta (quindi velocità trasmissiva notevolmente elevata) e costo relativamente basso. Gli svantaggi, invece, riguardano la dispersione del segnale e la difficoltà di interfacciamento.

La trasmissione all'interno del core (propagazione della luce) può avvenire in due modalità diverse:

- **Fibra multimodale:** è una fibra il cui nucleo è abbastanza ampio da poter permettere diversi angoli di rimbalzo della luce trasmessa. Essa è divisa in due tipologie:
 - Fibra multimodale **step-index**: la variazione dell'indice di rifrazione tra core e cladding è talmente elevata da causare molta dispersione modale (ritardo della luce);
 - Fibra multimodale **graded-index**: la variazione dell'indice di rifrazione tra core e cladding rallenta i raggi più centrali;
- **Fibra monomodale:** è una fibra nella quale la luce viaggia in maniera diretta senza riflessioni, quindi non è presente dispersione modale.

Parte della luce che si propaga lungo la fibra viene assorbita dal materiale o si diffondono in esso, costituendo una perdita del segnale trasmesso. Minore è l'attenuazione, maggiore è la distanza di trasmissione.

Riguardo, quindi, i segnali ottici, si utilizza la multiplazione **WDM**, già vista precedentemente. Se le distanze coperte dal segnale multiplato sono notevolmente grandi, può essere necessario rigenerare e risincronizzare il segnale.

Bisogna sapere che quando si trasmette su fibra è necessario effettuare due conversioni: in trasmissione, da elettrico a luminoso; in ricezione, da luminoso ad elettrico.

Livello Fisico: trasmissione wireless

L'aria è un buon mezzo di trasmissione siccome risulta semplice generare le onde radio: possono viaggiare per lunghe distanze e penetrano facilmente negli edifici. Ogni trasmissione via etere (aria) deve poter utilizzare due stazioni, quali sono trasmittente e ricevente. La trasmissione è resa possibile grazie alle antenne.

Livello Fisico: radiodiffusione

La radiodiffusione viene generalmente utilizzata per la trasmissione analogica in broadcast ed utilizza due tecniche trasmissive in base alla regione di frequenze:

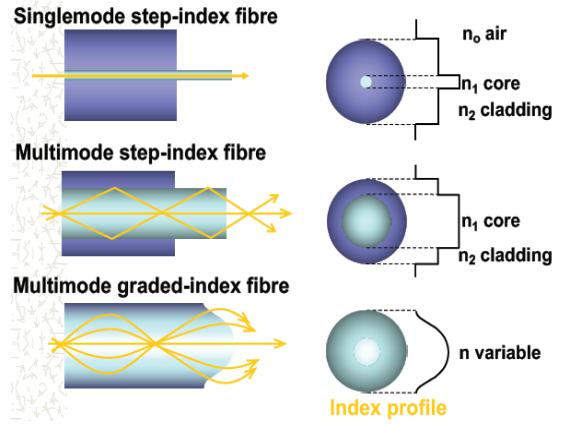
- Nella regione fino al MHz, il segnale segue la curvatura terrestre superando bene gli ostacoli;
- Nella regione dal MHz fino al GHz, il segnale viene assorbito dalla superficie della terra ma viene riflesso molto bene dalla ionosfera.

Livello Fisico: trasmissione via ponte radio

La trasmissione via ponte radio si utilizza per grandi frequenze (1-40 GHz) ed instaura una comunicazione ottica rettilinea punto a punto tra sorgente e destinazione: ciò significa che sorgente e destinazione devono essere allineati e ben visibili tra loro. Utilizzando diverse stazioni ripetitrici, è possibile raggiungere elevate distanze. Di solito, le connessioni a breve distanza utilizzano frequenze più alte, antenne più piccole e sono soggette a minori interferenze.

Livello Fisico: trasmissioni satellitari

Il satellite si comporta come una normale stazione ripetitrice. Il segnale viene mandato dalla stazione terrestre al satellite, il quale lo rimanda alla stazione di destinazione sulla terra. Il satellite opera su molteplici bande di frequenza, utilizzando la tecnologia FDM, gestendo molteplici comunicazioni contemporaneamente. Le bande utilizzate si aggirano tra 1 e 10 GHz.



Livello Data-Link

Il compito del **data-link** è quello di organizzare il trasferimento di dati tra due dispositivi adiacenti e connessi tra loro. Il data-link si occupa di **incapsulare i dati in frame (pacchetti)**, di **rilevare e correggere errori** nei pacchetti, assicurare una consegna affidabile, assicurare il controllo del flusso (il mittente viene obbligato a rispettare una **certa velocità con cui spedire** i pacchetti, evitando che il destinatario non riesca a gestire un carico troppo pesante e che perda i pacchetti).

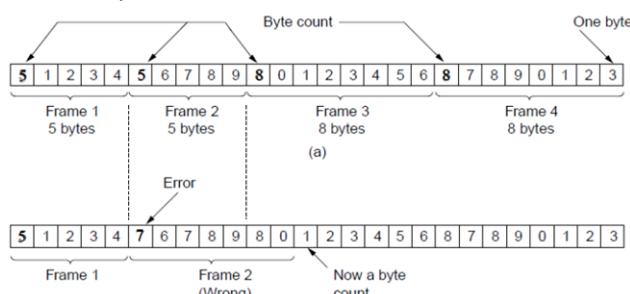
In **trasmissione**, il data-link **raggruppa i dati provenienti dallo strato superiore in frame, aggiunge header e trailer** per **ogni frame**, imposta i **bit** per la **rilevazione degli errori**, controllo di flusso, ecc., per poi mandarli al livello fisico; in **ricezione**, il **data-link riceve i dati** dallo strato fisico, **rimuove header e trailer da ogni frame, controlla e gestisce gli errori** di trasmissione e **passa** i dati al **livello superiore**, il livello di rete. Tali funzioni sono realizzate da un adattatore.

Start flag	type	seq	ack	Pacchetto (livello rete)	Check sum	End flag
------------	------	-----	-----	--------------------------	-----------	----------

Livello Data-Link: suddivisione in frame (Framing)

Lo strato fisico non può effettuare tutti questi controlli, di conseguenza non può garantire un trasferimento senza errori: perciò il **data-link provvede a suddividere i dati in frame** in modo da poterli controllare tutti in ricezione, permettendo l'identificazione di ognuno di essi tramite apposite tecniche:

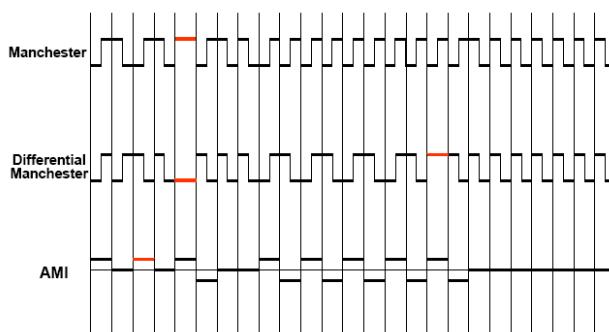
- **Conteggio dei caratteri:** un campo dell'intestazione indica il numero dei caratteri nel pacchetto, in modo da capire quando finisce e quando inizia il prossimo;



- **Indicatori di inizio e fine:** i pacchetti iniziano e terminano con una sequenza speciale di bit, chiamata **byte flag**, dal valore **01111110**. Per evitare che tale byte sia presente nel contenuto del pacchetto, si inserisce uno 0 dopo ogni gruppo di cinque bit pari ad 1 consecutivi. Tale bit viene eliminato in ricezione;



- **Violazione di codifica:** è possibile segnalare la fine e l'inizio di un pacchetto commettendo una violazione riguardo la codifica utilizzata.



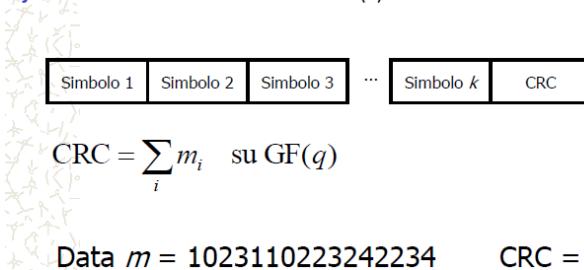
Livello Data-Link: rilevazione di errori

Il livello fisico offre un canale di trasmissione non privo di errori. Per rendere **possibile la rilevazione di errori**, viene inserito **nell'header** un campo denominato **checksum**, il quale è il risultato di un calcolo effettuato con i bit del corpo del pacchetto. La destinazione, una volta ricevuto il pacchetto, effettua **di nuovo il calcolo del checksum** e lo **compara** con il valore posto **nell'header**: se i valori coincidono, il corpo del pacchetto è corretto.



Altra tecnica per la rilevazione degli errori è la **CRC (Cyclic Redundancy Code)**, la quale consiste nel considerare i dati da inviare come un **polinomio**: il trasmettitore accoda ai dati del pacchetto una serie di bit di controllo, i quali saranno inclusi nel frame; trasmettitore e destinatario si accorderanno riguardo il **polinomio generatore** calcolato tramite i **campi di Galois**. In ricezione, si divide il polinomio associato al frame per il polinomio generatore: se il resto sarà nullo non ci saranno errori, altrimenti sono certamente avvenuti errori.

Cyclic Redundance Check-sum su GF(5).



EX: Il campo di Galois è un campo $(+,*)$ con un certo numero di elementi su cui sono definite due operazioni aritmetiche che godono della proprietà commutativa ed associativa. Le operazioni vengono effettuate seguendo l'aritmetica binaria.

Livello Data-Link: controllo del flusso

Può capitare che il mittente sia in grado di **trasmettere più velocemente rispetto alla capacità di ricezione** del destinatario. Nel caso il destinatario faccia da **collo di bottiglia**, si inizierebbero a **perdere frame**. Il protocollo dovrà poter gestire questa situazione.

Un metodo per gestire il flusso è quello di valutare i tempi di risposta del ricevente ed **inserire del ritardo** nel processo di **trasmissione**: ciò è un problema perché il tempo di risposta **non è una costante**, quindi comporterebbe un grosso limite in termini di efficienza.

Tecniche RDT (Real Data Transport)

Vediamo quindi alcune tecniche riguardanti la gestione del flusso:

RDT 1.0: il canale sottostante è **perfettamente affidabile**, quindi esente da errori senza perdite di pacchetto. Il mittente invia i dati nel livello fisico, mentre il destinatario li legge dal livello fisico.

RDT 2.0: il canale sottostante può contenere errori, rilevabili grazie all'**ACK (Acknowledge)** e al **NAK**:

- **L'ACK** è una **notifica positiva** comunicata dal ricevente che afferma di aver ricevuto il pacchetto correttamente;
- Il **NAK** è una **notifica negativa** comunicata dal ricevente che afferma di aver ricevuto il pacchetto contenente errori. Nel caso si riceva un **NAK**, il pacchetto viene **rtrasmesso**.

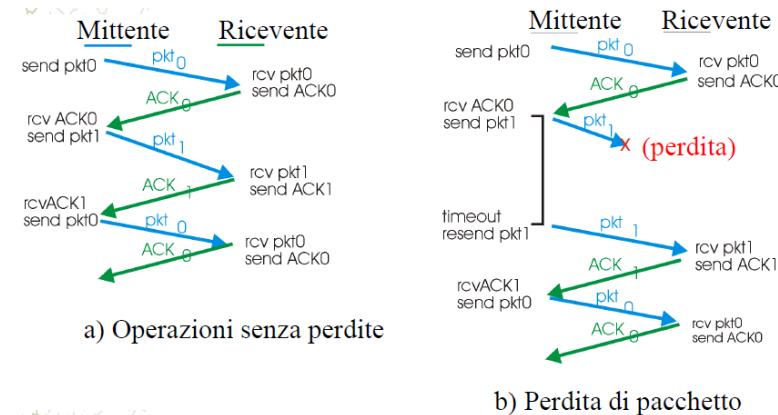
RDT 2.0 introduce, rispetto al RDT 1.0, la **rilevazione degli errori** e il **feedback** da parte del destinatario.

Sostanzialmente, tale procedura viene chiamata protocollo **stop-and-wait**. Tale protocollo prevede che il mittente, dopo aver trasmesso un pacchetto, debba aspettare un riscontro dal destinatario, quale è **L'ACK** o il **NAK**. Si nota che il traffico dati è **simplex**, ma il canale è **half-duplex** o **full-duplex** perché i dati viaggiano comunque in entrambe le direzioni. Può, però, presentarsi un grave problema: il **feedback** da parte del destinatario potrebbe arrivare **corrotto**: in tal caso, il mittente non sa cosa sia accaduto al destinatario. Una soluzione sarebbe quella di aggiungere il **checksum al feedback**, in modo tale che nel caso arrivi alterato si rinvii il pacchetto. Tale metodo produce pacchetti duplicati tra sender e receiver e, inoltre, il receiver non sa se il pacchetto ricevuto sia nuovo o duplicato: si aggiunge, quindi, un nuovo campo al pacchetto, quale è il numero di sequenza (l'introduzione del **numero di sequenza** riguarda RDT 2.1).

RDT 3.0: il canale sottostante può contenere errori e addirittura perdite, quindi non solo c'è il rischio che si **perda il pacchetto**, bensì è possibile **perdere anche il feedback!** Ciò rende insufficiente le tecniche viste poco fa. Una soluzione è quella di far attendere al mittente un **ACK** per un **determinato tempo**: nel caso non riceva alcun ACK ritrasmetterà il

pacchetto. Un altro problema si presenta nel caso il **pacchetto o l'ACK sia in ritardo**: in tal caso la trasmissione verrà **duplicata**, ciò è già gestito dai **numeri di sequenza**, quindi il **receiver** dovrà specificare il **numero di sequenza del pacchetto** a cui il **feedback** fa riferimento.

RDT 3.0 introduce un **timer** per il quale, se alla scadenza non si ha ricevuto il feedback da parte del destinatario, si rispedisce il pacchetto; oltre al timer, introduce nell'ACK un campo contenente il numero di pacchetto a cui il feedback fa riferimento, nel caso siano stati inviati dei duplicati.



Tecniche con pipeline

RDT è poco efficiente siccome utilizza il **protocollo stop-and-wait**. Un protocollo con pipeline migliorerebbe decisamente le prestazioni, siccome il mittente trasmetterebbe anche senza feedback. In parole poche, il mittente trasmette senza attendere l'ACK, mandando in **buffering molteplici pacchetti** presso il ricevente che, una volta elaborato il pacchetto, manderà il feedback. Tale tecnica con pipeline è implementata dai protocolli a **finestra scorrevole**: permettono di **inviare K frame prima di fermarsi ad aspettare un riscontro** (K stabilito e fisso, con $K \leq 2^n - 1$, dove n è il numero di bit che rappresenta il massimo numero di sequenza). Poiché in ricezione arrivano **molteplici frame**, essi **devono essere numerati** in modo da capire quali vengano persi. In trasmissione devono essere memorizzati i frame inviati in attesa del feedback, in modo da poterli rinviare in caso di necessità. Ad **ogni feedback ricevuto**, viene **liberato il buffer** associato al frame a cui il feedback fa riferimento. Anche in ricezione si deve disporre di un buffer, in modo da memorizzare i frame fuori sequenza. I frame di cui è stato mandato il feedback passano al livello di rete e il relativo buffer viene liberato.



<https://www.youtube.com/watch?v=9BuaeEjeQI>

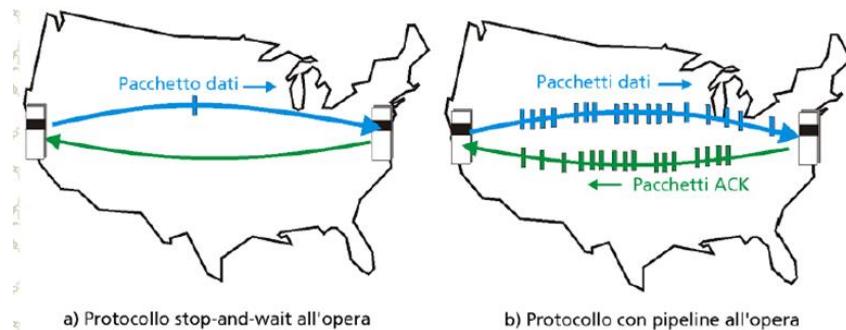
Precisamente, si usa una "**finestra**" per **contenere i frame da inviare**. La dimensione iniziale della finestra è pari a K .

Riguardo il mittente, durante la trasmissione la **finestra scorre in avanti**: ad **ogni frame inviato**, il **limite sinistro** della finestra **cresce di uno**, **fino** al chiudersi. Una volta chiusa, cioè quando sono stati inviati tutti i **K frame**, la trasmissione si ferma; per **ogni frame riscontrato**, invece, il **limite destro** della finestra **aumenta di uno**, permettendo al trasmittente di inviare nuovi frame. La dimensione della finestra varia da 0 a K .

Riguardo il **destinatario**, anch'egli utilizza una **finestra che scorre in avanti**: ad **ogni frame ricevuto**, il **limite sinistro** della finestra **cresce di uno**; **per ogni ACK inviato**, il limite **destro cresce di uno**. Anche la dimensione di questa finestra varia da 0 a K : quando la finestra si azzera significa che si devono inviare tutti i riscontri perché la ricezione è bloccata.

Riguardo il **buffer**, il mittente **conserva tutti i frame** di cui non si è ancora ricevuto l'ACK, nel caso sia necessario rinviarlo. Nel caso il buffer sia pieno, il data-link del mittente non accetterà altri pacchetti dal livello di rete.

Inoltre, l'invio di **ACK** e **NAK** prevede il numero di sequenza del frame a cui si fa riferimento.



Una tecnica utilizzata nei protocolli pipeline per l'invio degli ACK è il **piggybacking**: per motivi di efficienza, l'**ACK non viene spedito in un frame vuoto**, bensì in un frame di dati inserendolo **nell'header**. Quando si trasmette un ACK, quindi, si aspetta il momento in cui si debba trasmettere un frame di dati. Nel caso non si debba inviare nulla, si invia comunque un frame vuoto con un ACK, prima che scada il timer.

Esistono due protocolli che utilizzano la tecnica a **finestra scorrevole**: **Go-Back-N** e **selective reject**.

- **Go-Back-N**: introduce i **riscontri cumulativi**: un feedback N è interpretato come **riscontro cumulativo** perché indica che tutti i pacchetti con numero di sequenza $\leq N$ sono stati correttamente ricevuti dal receiver. Introduce anche un **timer per il primo pacchetto della finestra in transito**: se interviene un timeout vengono rispediti tutti i pacchetti presenti nella finestra. Il receiver, invece, invia un feedback cumulativo riferente al pacchetto N se i pacchetti con numero di sequenza $\leq N$ sono stati ricevuti ed in ordine. In tutti gli altri casi, il receiver rispedisce un **ACK** relativo al pacchetto più recente con l'ordine giusto. Sarebbe inutile conservare i pacchetti non in ordine siccome verrebbe rispedito dopo il timeout.

Il vantaggio principale riguarda il buffering, il quale non viene utilizzato siccome non c'è bisogno di memorizzare alcun pacchetto fuori ordine. L'unica cosa da conservare è il numero di sequenza del prossimo pacchetto da spedire.

Nel caso al mittente arrivi un **NAK** al posto dell'**ACK**, egli rinvierà nuovamente quel frame: il destinatario lo rifiuterà siccome lo avrà già accettato ma invierà comunque l'**ACK**, il quale farà proseguire la finestra del mittente.

- **Selective reject**: il ricevente **invia riscontri specifici per tutti i pacchetti ricevuti correttamente**, quindi il mittente **ritrasmette solamente i pacchetti per il quale non ha ricevuto l'ACK**. Anche qui è presente un timer per i pacchetti non riscontrati. In ricezione i frame fuori ordine, ma nella finestra, vengono mantenuti nei buffer finché non siano stati ricevuti tutti i frame intermedi. In questo modo si riduce drasticamente il numero di pacchetti ritrasmessi.

Sostanzialmente, quindi, quando si ha un frame perduto, il destinatario riceverà il frame successivo fuori sequenza di cui manderà l'ACK correlato. A differenza del Go-Back-N, il protocollo selective reject provvederà con lo **spedire solamente i pacchetti perduti**, proseguendo con la normale sequenza. Una volta ricevuto il pacchetto precedentemente perso, il destinatario libererà tutti i buffer e manderà un ACK riferito al pacchetto ricevuto. In caso di perdita dell'**ACK**, provvederà il timer come già spiegato. Nel caso della selective reject è necessario ridurre la finestra ad una dimensione pari a $K \leq 2^{n-1}$.

Livello Data-Link: collegamenti di rete

Il livello data-link gestisce due tipi di collegamenti: **punto-a-punto** e **broadcast**. Parlando del **broadcast**, quando una **trama** viene inviata sul mezzo condiviso, tutti i dispositivi collegati la ricevono: ogni **scheda di rete (NIC)** confronta il suo **MAC address** con quello di destinazione del pacchetto. Nel caso i **MAC address** coincidano, il NIC corrispondente accetterà il pacchetto anziché ignorarlo.

Nel caso più stazioni tentino di **trasmettere allo stesso tempo sullo stesso canale trasmittivo**, si genera una **collisione**. Sia T il tempo di una trasmissione tra due stazioni, una collisione viene rilevata in tempo $2T$.



I protocolli che regolano la trasmissione in modo da non far presentare **collisioni** sono detti **protocolli ad accesso multiplo**. Tali protocolli **regolano** anche la **velocità** con cui i dispositivi possono comunicare: se un solo nodo deve comunicare, può farlo con una **velocità pari a K bps**; se N nodi devono comunicare, possono farlo con una velocità pari a **K/N bps ciascuno**. Tali protocolli sono suddivisi in tre categorie:

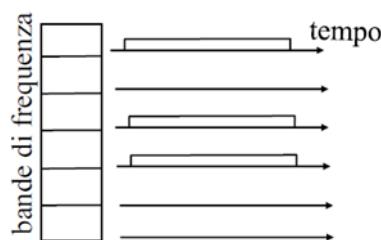
- **Protocolli a suddivisione del canale:** il **canale** viene **suddiviso in parti più piccole**;
- **Protocolli ad accesso casuale:** i **canali non** vengono **divisi** e possono presentarsi collisioni. I nodi coinvolti ritrasmettono ripetutamente i pacchetti;
- **Protocolli a rotazione o collision free:** **ciascun nodo ha il suo turno** per trasmettere, il quale viene **regolato in base alla quantità** di dati da trasmettere.

Protocolli a suddivisione del canale

1. **TDMA (Time Division Multiple Access)**: suddivide il canale in **slot di intervalli di tempo**.



2. **FDMA (Frequency Division Multiple Access)**: suddivide il canale in **bande di frequenza diverse**. Se uno slot o una banda rimane vuota, allora sarà inattiva.

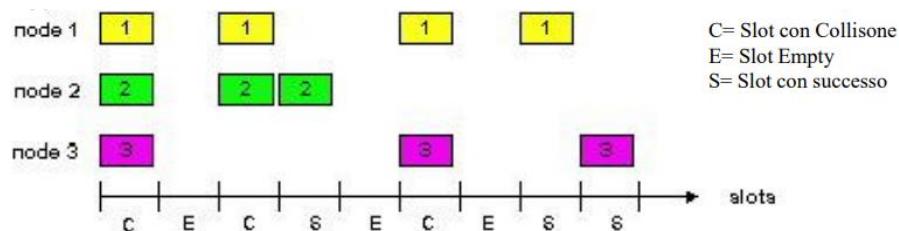


Protocolli ad accesso casuale

Con i protocolli ad accesso casuale, il **nodo mittente trasmette** sempre alla **massima velocità** consentita dal canale, cioè K bps. Non essendoci coordinazione tra i nodi, c'è sempre il rischio di collisione. Varie implementazioni di tale protocollo sono le seguenti:

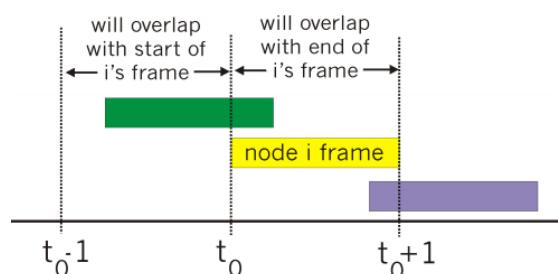
1. Slotted ALOHA

Quando a un nodo arriva un nuovo pacchetto da spedire, il nodo attende fino all'inizio dello slot successivo. **Se non si verifica alcuna collisione**, si può **trasmettere un nuovo pacchetto nel prossimo slot**. Nel caso, invece, si verifichi una **collisione**, il nodo la rileva prima della fine dello slot e **ritrasmette con probabilità P il suo pacchetto negli slot successivi**. Tecnica molto semplice che consente ad ogni nodo di sfruttare la massima velocità del mezzo trasmisivo, ma che presenta anche un grande svantaggio: in caso di collisioni potrebbero esserci di mezzo slot inutilizzati, siccome si decide in modo casuale con probabilità P quando ritrasmettere il pacchetto che ha riscontrato la collisione.



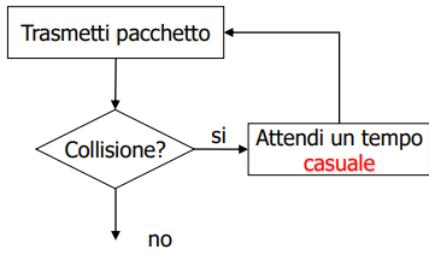
2. ALOHA puro

A differenza dello Slotted ALOHA, **appena arriva un pacchetto viene trasmesso immediatamente**, aumentando i rischi di collisione: il fatto che venga trasmesso subito implica il fatto che la trasmissione non avvenga all'inizio dello slot, quindi c'è il rischio di "invadere" un altro slot. Le probabilità di provocare una collisione sono decisamente più alte.



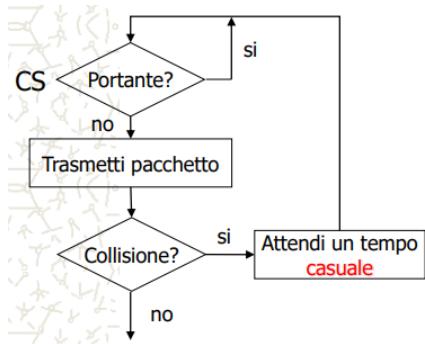
3. Accesso multiplo a rilevazione della portante – CSMA

Con il **CSMA**, il **nodo si pone in ascolto prima di trasmettere**: se il canale è **libero**, allora **trasmette il pacchetto**; se qualcun altro sta **già trasmettendo**, il **nodo aspetta un altro intervallo di tempo**. Può capitare, però, che agli altri nodi sul canale non venga notificata in tempo la trasmissione di un certo nodo, facendo risultare libero il canale (ritardo di propagazione): nel caso si effettuino, quindi, due o più trasmissioni contemporaneamente, si genera una collisione, la quale bloccherà le trasmissioni. Notificata la collisione, i nodi attenderanno un tempo casuale per la prossima trasmissione. Maggiore è il ritardo di propagazione e maggiore è la probabilità di riscontrare una collisione.



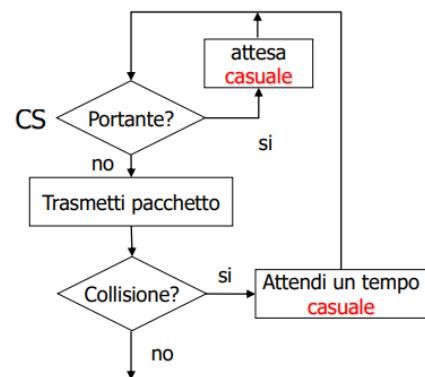
4. CSMA persistente

Il **CSMA persistente ascolta e controlla continuamente il canale, attendendo che si liberi**. In caso di **collisione, attende un tempo casuale e ripete i controlli**. Tale tecnica è problematica in caso di tre nodi: non appena il nodo A finirà di trasmettere, B e C troveranno il canale libero e trasmitteranno contemporaneamente, causando una collisione.



5. CSMA non persistente

Il **CSMA non persistente** si differenzia dal **persistente** dal fatto che quando un **nodo vuole trasmettere ma trova il canale occupato, non resta ad ascoltare in continuazione ma attende un tempo casuale e riprova**. Tale meccanismo riduce sensibilmente le collisioni, siccome ogni stazione attende un tempo randomico. Il problema di tale tecnica è che nel caso ci siano molteplici nodi connessi alla rete, il tempo di attesa può crescere enormemente.

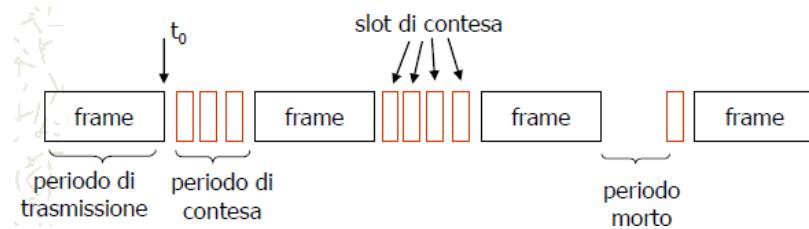


6. CSMA p-persistente

Nel **CSMA p-persistente**, chi vuole trasmettere ascolta continuamente il canale e, nel caso sia libero, **trasmette con probabilità P nello slot attuale**. Nel caso non trasmetta, quindi con probabilità 1-P, attende il prossimo slot e, se libero, riprova a trasmettere con probabilità P.

7. CSMA/CD

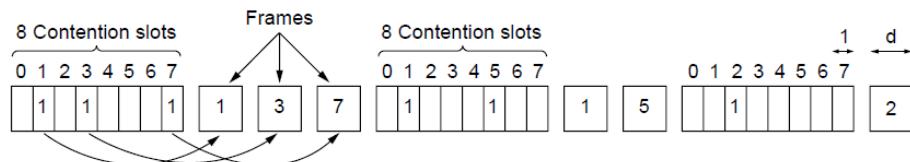
Nel **CSMA/CD**, **ogni nodo ascolta continuamente il canale e trasmette solo se è libero**; se si verifica una collisione, i nodi bloccano la trasmissione. A seguito della collisione, il nodo mittente trasmette una serie di **jamming** (interferenze) per comunicare a tutti i nodi la collisione. Il nodo mittente riproverà poi la trasmissione dopo un tempo casuale.



Protocolli a rotazione o collision free

1. Mappa di bit elementare:

Sulla rete ci sono molteplici nodi, dei quali sono alcuni dovranno trasmettere. La **mappa di bit elementare permette ad ogni nodo di notificare, durante un periodo di contesa**, se deve trasmettere o meno: ogni nodo trasmetterà 1 nel caso debba **trasmettere**, 0 altrimenti (tale bit viene denominato come **notifica**). I nodi che trasmetteranno 1, dopo il periodo di contesa, potranno trasmettere i propri **frame**, uno per volta onde evitare collisioni. Se un nodo comunica la notifica una volta finito il periodo di contesa, quest'ultimo verrà scartato e dovrà attendere il prossimo periodo di contesa, il quale si presenterà una volta finite le trasmissioni dati.

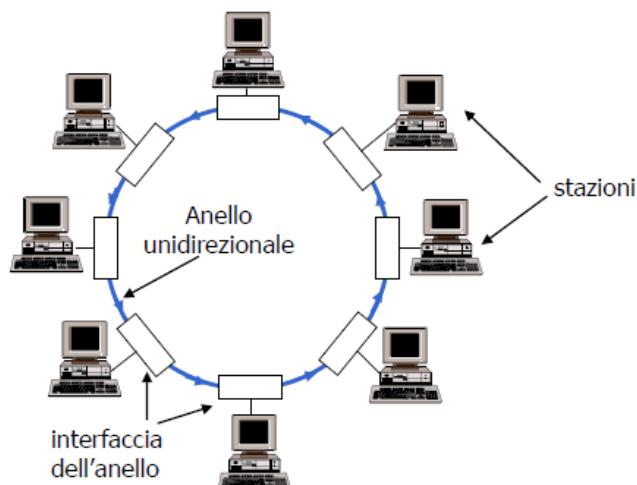


Le notifiche sono controllate da un'apposita stazione. L'efficienza di tale sistema è bassa, siccome per un gran numero di nodi potrebbe sovraccaricarsi la stazione di controllo.

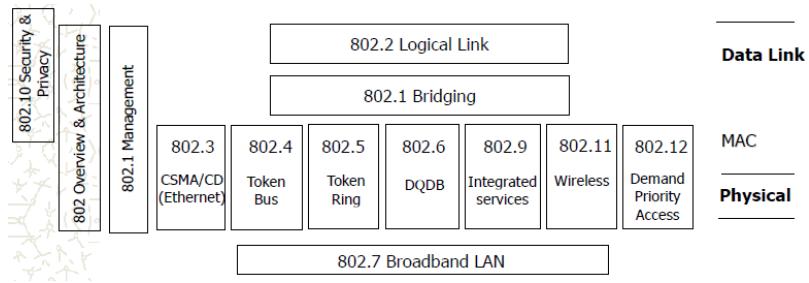
2. Token ring:

Il sistema token ring **non** utilizza un mezzo **broadcast**, bensì un **insieme di nodi punt- a-punto**. Tale protocollo prevede l'utilizzo di una **rete** con topologia **ad anello**. Sull'anello circola un piccolo **frame detto token**, il quale autorizzerà la trasmissione al nodo cui ne terrà il possesso. Il token è una sequenza di bit che circola sull'anello quando i nodi sono tutti in **idle**. Quando un nodo vuole trasmettere, si impossessa del token. Ovviamente i nodi senza il token non sono autorizzati a trasmettere.

Tale protocollo è poco efficiente in caso di basso carico, siccome la stazione che deve trasmettere deve comunque attendere di ricevere il token, anche se il canale è libero. Se il carico è alto (cioè tutti vogliono trasmettere), il token è il sistema perfetto!



Livello Data-Link: standard IEEE 802, MAC ed LLC



Il progetto IEEE 802 si occupa di fornire una serie di standard per le **LAN** e le **MAN** ai livelli data-link e fisico.

Il progetto IEEE 802 suddivide il data-link in due sottolivelli: **LLC** e **MAC**.

Il **sottolivello LLC (Logical Link Control)** è comune a tutte le LAN e costituisce l’interfaccia verso il livello rete. In **trasmissione** si occupa di “preparare” i dati verso il livello fisico **inserendo** un suo **header** con le informazioni riguardanti il numero del frame, il feedback, ecc.. In **ricezione**, invece, si occupa di **rimuovere l’header** per poi passare i dati al livello di rete.

Il **sottolivello MAC (Media Access Control)** è specifico di **ogni LAN** e risolve il problema della condivisione del mezzo trasmittivo, quindi: in **trasmissione** sceglie chi può usare il canale; in **ricezione** determina a quali sistemi è destinato un certo pacchetto. La soluzione riguardo la trasmissione è data da vari algoritmi di **MAC**; la soluzione riguardo la ricezione implica la presenza di indirizzi a livello MAC, in modo da trasformare la trasmissione **broadcast** in:

- **Trasmissione punto a punto**, se l’indirizzo di destinazione è unico;
- **Trasmissione punto a molti**, se l’indirizzo di destinazione indica molteplici sistemi;
- **Trasmissione broadcast**, se l’indirizzo di destinazione indica tutti i sistemi.

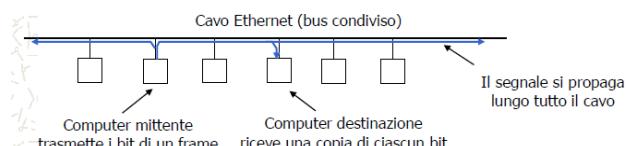
L’indirizzo MAC (chiamato anche **indirizzo LAN**, **indirizzo fisico** o **indirizzo Ethernet**) è paragonabile al codice fiscale di una persona: è univoco e non dipende dalla “posizione” del dispositivo. È importante non fare confusione con l’indirizzo IP, il quale è paragonabile all’indirizzo dell’abitazione (certo, è univoco anche quest’ultimo, ma non fa riferimento al singolo device).

Ogni adattatore (scheda di rete) in una LAN ha un proprio indirizzo MAC, il quale ha sei coppie ognuna formata da due caratteri esadecimali, quindi FF-FF-FF-FF-FF-FF.

È importante sapere che il **MAC** è sempre **connectionless**, quindi non si occupa della correzione degli errori riguardo la trasmissione. In ogni caso, le LAN sono reti sicure ed affidabili, quindi non è necessario controllare e correggere errori. Se ciò fosse richiesto, se ne occuperebbe il **sottolivello LLC**.

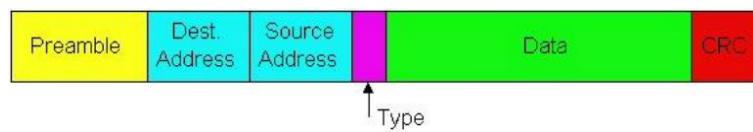
Livello Data-Link: Ethernet

Ethernet è una serie di tecnologie standardizzate per reti locali, che ne definisce le specifiche tecniche a livello fisico e a livello MAC.



I pacchetti Ethernet sono composti dai seguenti campi:

- **Preamble**: serve a “svegliare” l’adattatore del ricevente e a sincronizzare il clock del mittente e del ricevente;
- Indirizzo **MAC destinazione**;
- Indirizzo **MAC sorgente**;
- **Ethertype (campo di tipo)**: indica il tipo di protocollo del livello di rete in uso;
- **Payload (data)**: contiene i veri dati del pacchetto, per un massimo di 1500 byte;
- **CRC**: consente all’adattatore di rilevare eventuali errori nel pacchetto.



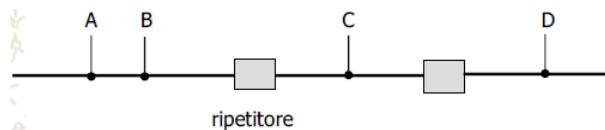
Un frame valido deve essere lungo almeno 64 byte.

EX: L'insieme dei protocolli Ethernet domina tutt'ora il mercato delle LAN. Originariamente la velocità di trasmissione era 10Mbit/s su cavo coassiale, invece oggi si usano mezzi trasmissivi migliori in modo da comportare prestazioni migliori, fino a 10Gbit/s (Gigabit Ethernet). Sul mezzo condiviso l'assenza di trasmissione è identificata dall'assenza di segnale, quindi non sono possibili codifiche che utilizzino il segnale a 0 volt per identificare un bit. Lo standard Ethernet utilizza la codifica Manchester con segnali a +0.85V e -0.85V.

Livello Data-Link: dispositivi

Prima di andare avanti, è necessario introdurre i **transceiver** (*ricetrasmettitore*), i quali sono dispositivi **composti da un trasmittitore e da un ricevitore**, i quali condividono alloggiamento e circuiti. Vengono utilizzati su connessioni **half-duplex** e **full-duplex**.

Per costruire reti più ampie, può essere necessario l'utilizzo dei **ripetitori**. Un ripetitore opera al livello fisico e non fa altro che **amplificare e ritrasmettere il segnale**. Fra due transceiver (la cui distanza massima può essere fino a 2.5 Km) possono esserci massimo 4 ripetitori.



Introduciamo ora i seguenti dispositivi:

Il **bridge** è un dispositivo hardware o software in grado di collegare tra loro due o più reti diverse (solitamente delle LAN) in modo che possano comunicare tra loro. Sostanzialmente anche lo **switch** opera nella stessa maniera, ma offrendo una differenza sostanziale: il bridge “**unisce**” le reti che collega, formando una LAN più grande, quindi con un unico dominio di collisione; lo **switch**, invece, permette la comunicazione tra le diverse reti ma tenendole costantemente separate, con domini di collisione separati e diversi, confinando i malfunzionamenti dovuti a stazioni difettose, aumentando la sicurezza della rete. Sia **bridge** che **switch** sono composti da un certo **numero di porte**. Solitamente il numero di porte dello switch è maggiore rispetto al numero di porte del bridge.

Sia il **bridge** che lo **switch**, riguardo la **ricezione**, usano una **tavella di indirizzi MAC** per capire chi è il reale **destinatario**. Tale tabella permette di accoppiare ad ogni porta un indirizzo MAC. Al primo boot la tabella sarà vuota, quindi si sentirà la necessità di riempirla: basterà mandare pacchetti da un nodo all'altro e il gioco è fatto! Il problema è che, avendo la tabella vuota, non si conosce la porta per il nodo di destinazione. Quindi si prosegue mandando in **broadcasting** il pacchetto: si salveranno porta e MAC del dispositivo che risponderà. Sostanzialmente, quindi, bastano pochi pacchetti per poter riempire la tabella.

Ogni **switch** è composto da:

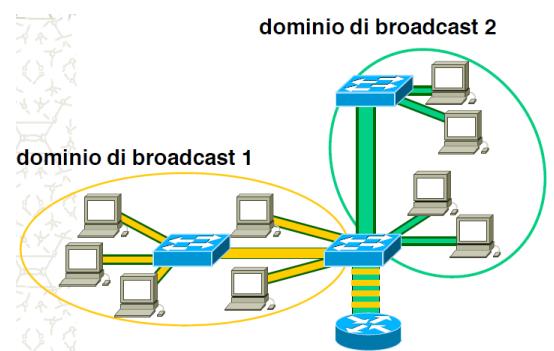
- **Memoria condivisa**, nella quale vengono memorizzati i pacchetti, i quali verranno poi mandati alla porta di destinazione;
- **Fabric**, dispositivo che recapita le trame in input verso una porta di output;
- **Architettura bus**, i dati passano tramite un bus interno ad alta velocità. La comunicazione interna usa TDMA (Time Division Multiple Access).

Livello Data-Link: Virtual LAN (VLAN)

Sappiamo che, tramite l'utilizzo dello switch, è possibile dividere una LAN in più reti locali i cui domini di broadcast sono separati ma che condividono comunque la stessa infrastruttura fisica. Tali reti vengono dette **VLAN** (*Virtual Local Area Network*) e **possono comunicare tra loro solo a livello di rete**.

Le porte **Trunk** sono delle porte che permettono di trasferire **frame** appartenenti a **VLAN differenti**.

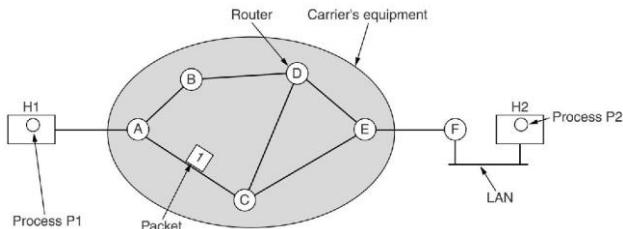
Ciascuna VLAN è identificata da un numero, detto **VID (VLAN ID)**. Per trasmettere dati, è necessario che si possa identificare a quale VLAN appartiene ogni pacchetto. Per fare questo, il frame 802.1Q è caratterizzato dal VID, il quale verrà interpretato dallo switch.



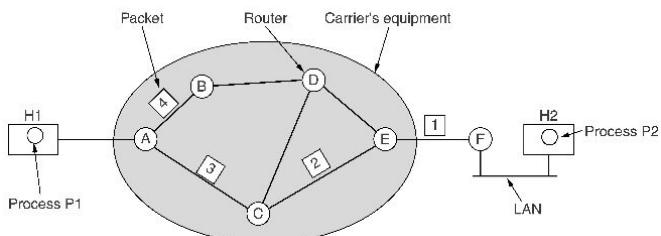
Livello Rete / Network

Sappiamo che **due host** sono **separati da un certo numero di nodi intermedi**, separati a loro volta (ma non per forza) da reti funzionanti con tecnologie diverse, e che tra tali nodi è possibile intraprendere molteplici percorsi. Lo **strato di rete permette di determinare quale tragitto (path) dovranno seguire i dati (instradamento)**, deve evitare di sovraccaricare le linee quando sono disponibili percorsi alternativi (**congestione**) e deve risolvere i problemi connessi al transito attraverso reti differenti (**internetworking**).

Inizialmente si prevedeva solamente l'utilizzo del servizio **Connection Oriented**, mentre in seguito si è sentita la necessità di introdurre nello standard anche il servizio **Connectionless**. Il servizio senza connessione (**Connectionless**) richiede che i **pacchetti** vengano **instradati indipendentemente** uno dall'altro.



Sostanzialmente, l'idea di base è quella di stabilire un **path**, il quale è **costituito da** una serie di **router che il pacchetto dovrà attraversare**. Ovviamente i pacchetti con la stessa connessione seguiranno lo stesso path; di conseguenza, i pacchetti seguiranno un determinato path non in base alla destinazione, bensì in base alla connessione instaurata, la quale è identificata tramite un **id** nell'**header** del **pacchetto**.



La funzione principale dello **strato di rete** è **l'instradamento (routing)**: tale processo permette al router di scegliere, tramite un certo algoritmo, la linea di uscita verso cui instradare i dati. Tale operazione, ovviamente, verrà ripetuta per ogni pacchetto nel caso la connessione sia **Connectionless**; nel caso della **Connection Oriented** viene effettuata una sola volta. Precisamente, possiamo distinguere due operazioni:

- **Inoltro (forwarding)**: definisce le regole con le quali un pacchetto viene inoltrato a livello fisico verso l'uscita (normalmente sulla base della lettura di una tabella di instradamento);
- **Instradamento**: definisce le regole con le quali viene scelto un percorso in rete tra sorgente e destinazione (sulla base delle quali vengono scritte le tabelle di instradamento).

È importante sapere che un algoritmo di routing deve essere corretto, meno soggetto ad errori, stabile e ottimizzato.

Livello Rete / Network: protocollo, indirizzi e pacchetti IP (Internet Protocol)

Il **protocollo IP** ha la funzione di **recapitare dati** dalla **sorgente** alla **destinazione** tramite reti interconnesse tra loro. Il recapito viene effettuato in maniera **diretta** se **sender** e **receiver** fanno parte della **stessa rete**, altrimenti in maniera **indiretta** nel caso viaggi tramite dei **router**. Se possibile il pacchetto viaggia per intero, altrimenti viene spezzettato in più parti trasportate individualmente: in tal caso il pacchetto viene riassemblato a destinazione. Tale protocollo fornisce un servizio **Connectionless** inaffidabile.

Ogni **interfaccia di rete** (cioè ogni connessione ad una rete) deve **avere un indirizzo IP**: i computer generalmente ne hanno affiliato uno solo, mentre i **server** ne **hanno molteplici**.

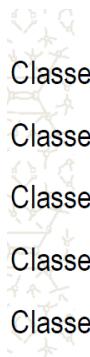
L'**indirizzo IP** è formato da **32 bit** rappresentati da **4 numeri decimali** separati da un punto che possono assumere un valore nel range [0 to 255]. Ogni indirizzo IP contiene una parte che **specifica la rete (prefisso)** ed una parte che **identifica l'host** all'interno della rete (**suffisso**). Prefisso e suffisso dipendono dalla classe dell'indirizzo IP, la quale vedremo tra poco.

Esempio di indirizzo IP: XX.XX.XX.XX

È importante sapere una cosa: mentre ***l'indirizzo MAC identifica univocamente il dispositivo, l'indirizzo IP identifica univocamente la connessione dispositivo-rete***: da ciò consegue che se un computer ha molteplici connessioni di rete, allora avrà assegnato un indirizzo IP per ogni connessione.

Gli indirizzi IP sono raggruppati in diverse categorie, dette ***classi***:

- Gli indirizzi di ***classe A*** hanno il primo campo che assume un valore nel range [0 to 127]: il ***primo*** campo è dedicato al prefisso, i restanti al suffisso; la sequenza di bit comincia con 0;
- Gli indirizzi di ***classe B*** hanno il primo campo che assume un valore nel range [128 to 191]: i primi ***due*** campi sono dedicati al prefisso, i restanti al suffisso; la sequenza di bit comincia con 10;
- Gli indirizzi di ***classe C*** hanno il primo campo che assume un valore nel range [192 to 223]: i primi ***tre*** campi sono dedicati al prefisso, l'ultimo al suffisso; la sequenza di bit comincia con 110;
- Gli indirizzi di ***classe D*** hanno il primo campo che assume un valore nel range [224 to 239]: sono indirizzi dedicati al ***multicasting***; la sequenza di bit comincia con 1110;
- Gli indirizzi di ***classe E*** hanno il primo campo che assume un valore nel range [240 to 255]: sono indirizzi dedicati ad utilizzi sperimentali; la sequenza di bit comincia con 1111.



	8 bits	8 bits	8 bits	8 bits	Address	Class	Network	Host
Classe A:	Network	Host	Host	Host	10.2.1.1	A	10.0.0.0	0.2.1.1
Classe B:	Network	Network	Host	Host	128.63.2.100	B	128.63.0.0	0.0.2.100
Classe C:	Network	Network	Network	Host	201.222.5.64	C	201.222.5.0	0.0.0.64
Classe D:	Multicast	I bit iniziali determinano la classe, che a sua volta determina il confine tra prefisso e suffisso.			192.6.141.2	C	192.6.141.0	0.0.0.2
Classe E:	Research				130.113.64.16	B	130.113.0.0	0.0.64.16
					256.241.20.10	Nonexistent		

- Nel caso un indirizzo IP contenga tutti i ***bit pari a 0*** nel campo di host (***suffisso***), allora si sta indicando la rete.
- L'indirizzo IP contenente tutti i ***bit pari a 0*** nel campo di rete (***prefisso***) indica ***"questa rete"***.
- L'indirizzo IP contenente tutti i ***bit pari a 0*** sia nel campo di rete che nel campo di host indica ***"questo host di questa rete"***.
- L'indirizzo IP contenente tutti i ***bit pari ad 1*** sia nel campo di ***rete*** che nel campo di ***host*** indica l'indirizzo ***broadcast della rete locale***, quindi viene utilizzato per mandare un pacchetto IP ad ogni host sulla propria rete.
- L'indirizzo contenente tutti i ***bit pari ad 1*** nel campo ***host*** indica il ***broadcast nella rete specificata nel campo rete***, quindi viene utilizzato per mandare un pacchetto IP ad ogni host appartenente ad una certa rete remota.

L'indirizzo **127.0.0.0** indica l'interfaccia di ***loopback***, la quale identifica la macchina locale detta ***localhost***.

Affinché tutto funzioni correttamente, gli indirizzi IP devono essere assegnati da una ***autorità centrale*** che garantisca l'unicità delle assegnazioni (siccome ogni indirizzo IP deve essere unico in tutta la rete). Per internet gli indirizzi sono assegnati dalla ***ICANN***, la quale ha poi delegato organizzazioni regionali assegnando loro gruppi di indirizzi da riassegnare al loro interno.

Lo spazio di assegnamento equivale a circa due miliardi di indirizzi, ma con il passare degli anni ci si è accorti di una certa ***carenza di indirizzi***. Per risolvere tale problema, è stata sviluppata una tecnica detta ***subnetting***, la quale permette di suddividere un campo di indirizzi in gruppi più piccoli, trattando un sottogruppo come se fosse una rete a sé stante.

Esempio: un campus ha rete associata 100.0.0.0 (***classe A***) e si vuole suddividere il suo campo di indirizzi: è possibile associare 100.1.0.0 ad un dipartimento, 100.2.0.0 ad un altro dipartimento e così via, in modo da considerare le reti dei dipartimenti come reti a sé stanti e di classe più piccola, il tale ***classe B***. Qualsiasi rete può essere subnettata.

Altro esempio: sia 172.16.0.0 una rete, eventuali subnet possono essere 172.16.1.0, 172.16.2.0, 172.16.3.0 e così via.

Il processo di ***subnetting*** è la ***divisione di una singola rete in sottoreti*** i cui dispositivi avranno l'indirizzo di rete identico. Per svolgere le proprie funzioni, il ***subnetting*** fa uso della ***network mask*** (o ***subnet mask***), la quale permette di suddividere la parte prefisso dalla parte suffisso.

La network mask raffigurerà con i bit 1 il prefisso, con i bit 0 il suffisso. Per capire a quale rete appartiene un indirizzo IP, si mette in AND bit a bit l'indirizzo IP con la network mask.

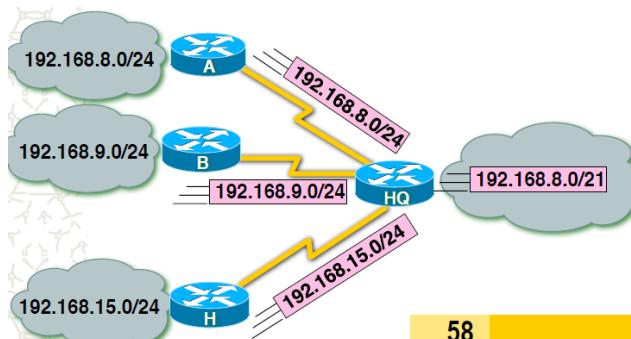
Network		Host	
172.16.2.160	10101100	00010000	00000010 10100000
255.255.0.0	11111111	11111111	00000000 00000000
	10101100	00010000	00000000 00000000
Network Number	172	16	0 0

- Subnets non in uso — schema di default

Esempio: 193.206.144.64/26 -> sono necessari 26 bit per rappresentare l'indirizzo di rete.

Livello Rete / Network: Classless InterDomain Routing

Già diversi anni fa Internet cresceva più velocemente di quanto si potesse pensare. Era, dunque, necessario adottare una soluzione: abbandonare le classi di rete. Si decide quindi di adottare un sistema che consente una migliore gestione degli indirizzi di rete evitando sprechi: il **CIDR**. Tale sistema permette la **suddivisione dell'indirizzo IP in prefisso e suffisso senza la suddivisione in classi**. Secondo questo standard ogni record della **tavella di routing** specifica la destinazione con la sua maschera, causando però un grave problema: l'aumento delle reti indirizzabili può far esplodere le dimensioni della tabella di routing. Per ovviare a questo problema, gli indirizzi vengono assegnati alle varie organizzazioni regionali e locali che risultano verso l'esterno solo come una rete.



Livello Rete / Network: IPv6 (IP version 6)

Agli inizi degli anni 90 si iniziò la ricerca di un successore di IPv4 siccome si sentiva la necessità di ampliare lo spazio degli indirizzi. Fu così sviluppato un protocollo progettato sul modello dell'IPv4, ampliando e migliorando le sue caratteristiche: **IPv6**. Tale protocollo offre **indirizzamento illimitato, riduce i tempi di elaborazione** del router e **supporta pacchetti di grosse dimensioni**.

L'IPv6 prevede indirizzi a 16 byte (128 bit), con 8 campi composti da 4 cifre esadecimale e separati tra loro da “:”.

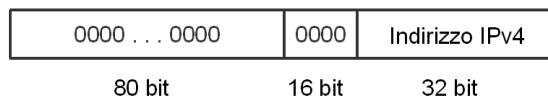
Esempio: 8000:0000:0000:0000:0123:4567:89AB:CDEF

Gli **indirizzi IPv6** sono soggetti ad ottimizzazioni riguardo la loro rappresentazione: si possono omettere gli zeri all'inizio di un gruppo e si possono omettere gruppi di zeri consecutivi, rappresentandoli con la seguente sequenza “::”.

Esempio: 8000::123:4567:89AB:CDEF

Come l'IPv4, anche l'IPv6 definisce **campi** appartenenti alla **rete** e **campi** appartenenti all'**host**. Il formato dell'intestazione (**header**) dell'IPv6 è stato notevolmente semplificato rimuovendo e modificando molteplici campi; anzi, alle volte è possibile attribuire anche molteplici header.

È importante sapere che partendo da un indirizzo IPv6 è possibile ricavarsi l'IPv4. Gli IPv4 sono rappresentati sottoforma di IPv6 da 6 gruppi di zeri e due gruppi che rappresentano l'effettivo IPv4. Quindi, sostanzialmente, è possibile ricreare un IPv4 partendo da un IPv6.



Esempio: ::89AB:CDEF oppure ::137.171.205.239

Network		Subnet	Host
172.16.2.160	10101100	00010000	00000010 10100000
255.255.0.0	11111111	11111111	11111111 11000000
	10101100	00010000	00000000 00000000
Network Number	172	16	2 128

L'indirizzo di rete viene esteso di 10 bit a discapito degli hosts

128 240 248 252 254 255 256 240 248 254 255

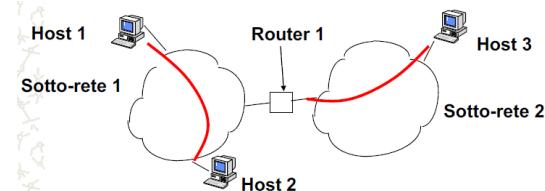
128 240 248 252 254 255 256 240 248 254 255

C'è un problema: l'IPv6 può spedire, instradare e ricevere pacchetti IPv4, ma IPv4 non è in grado di gestire pacchetti IPv6. Una soluzione sono i **DNS (Domain Name System)**, i quali **permettono di creare indirizzi IPv6 partendo da un IPv4**: il kernel capisce che si tratta di un indirizzo speciale ed usa la comunicazione IPv4.

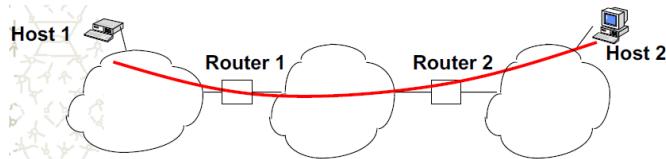
Livello Rete / Network: routing e forwarding

Riguardo il **routing IPv6**, può essere **diretto** o **indiretto**.

- **Instradamento diretto:** la trasmissione di un pacchetto avviene tra **due stazioni connesse nella stessa sottorete** senza coinvolgere alcun router intermedio.



- **Instradamento indiretto:** la trasmissione di un pacchetto avviene tra **due stazioni non situate nella stessa sottorete**, quindi la sottorete del mittente sarà diversa dalla sottorete del destinatario, coinvolgendo **router intermedi**: il router esamina il pacchetto ricevuto e, se l'host di destinazione non si trova in una sottorete a cui il router è direttamente connesso, **inoltrerà** il pacchetto **al router successivo**, il cui ripeterà tali controlli. Nel caso, invece, il destinatario si trovi nella stessa sottorete del router che sta esaminando il pacchetto, si individua l'indirizzo **MAC del destinatario** tramite procedura **ARP (Address Resolution Protocol)**.



Riassumendo, dato un pacchetto:

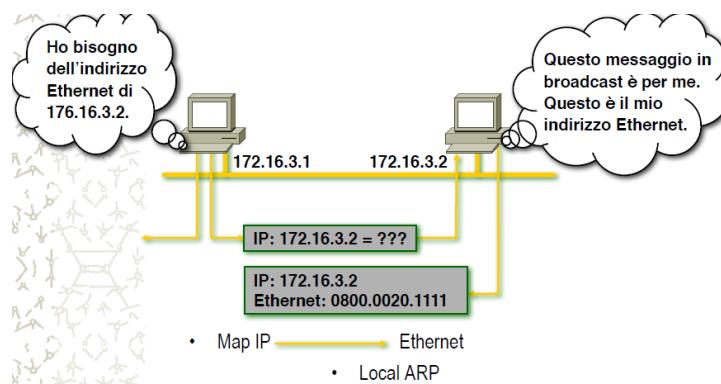
- Viene **estratto** il campo **destinazione**;
- Si cerca la **destinazione** nella **routing table**;
- Si trova la **prossima destinazione Z (hop)**, la quale può essere il dispositivo **destinatario** o il prossimo **router**;
- Il **pacchetto** viene **spedito** a Z.

Ad ogni "**hop**" viene **ricalcolata la strada** da seguire per tutti i pacchetti in transito, nel caso di **Connection Oriented**.

Il problema sta nel come fare a sapere a quale indirizzo MAC inviare il pacchetto, contando che l'host conosce solo l'indirizzo IP del destinatario. IP si appoggia, quindi, ad un protocollo chiamato **ARP(Address Resolution Protocol)**.

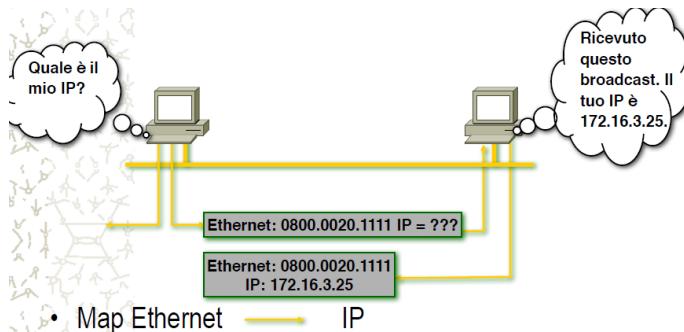
Vediamo come funziona **ARP**: poniamo di avere un host con indirizzo IP A1 e con indirizzo MAC MA1 il quale deve inviare un pacchetto IP ad un host con indirizzo IP A2 sulla stessa rete. ARP si procura le informazioni necessarie nel seguente modo:

- Viene **costruito un pacchetto data-link** (chiamato **ARP Request**) contenente A1, MA1, A2 e MA2, quest'ultimo contrassegnato da una serie di 0;
- Tale **pacchetto** viene **invia in broadcast** sulla **rete locale**;
- **Tutti ricevono** tale pacchetto **ARP**, ma solo l'**host** con MAC MA2 **lo processerà**;
- L'**host** di **destinazione creerà** un **pacchetto data-link** (chiamato **ARP Response**) nella quale inserirà il campo mancante. Tale pacchetto verrà trasmesso in maniera diretta e non in broadcast;
- Viene quindi **acquisito** il **MAC MA2** rilegato all'**indirizzo IP A2**.



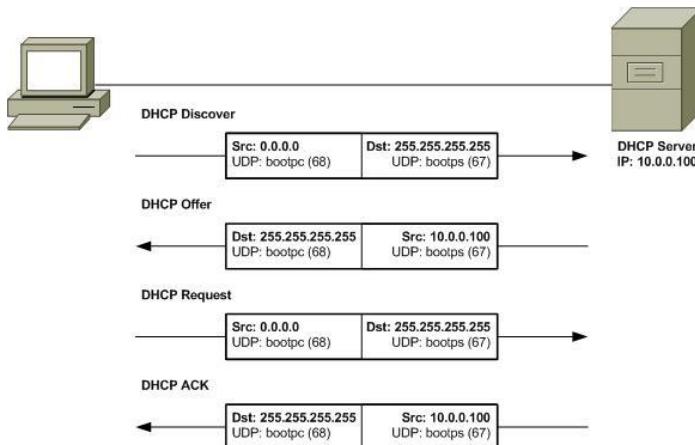
Ogni volta che viene rilevata una nuova associazione, essa viene memorizzata nella **cache**. Quando ARP rileva un indirizzo MAC, controlla la cache: se l'associazione è già presente, viene utilizzata senza mandare alcuna **ARP Request o Response**. Le **entry** nella cache di ARP hanno un **timer**, il quale alla scadenza eliminerà la entry. È possibile impostare anche delle entry senza una scadenza.

Nel caso fosse necessario, esiste una tecnica chiamata **ARP Reverse**, quale serve a trovare l'IP associato ad un indirizzo MAC.



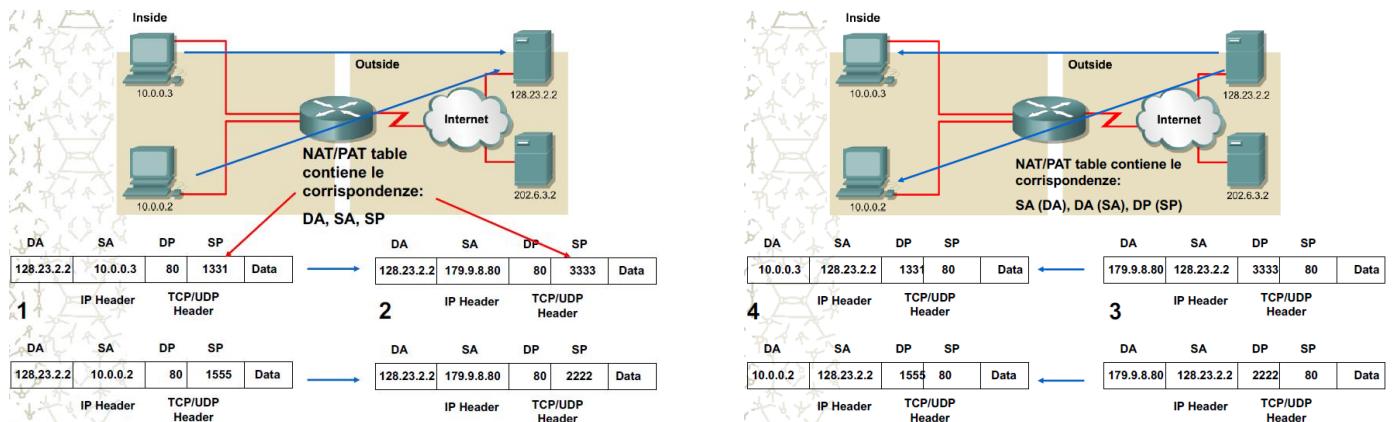
Livello Rete / Network: DHCP(Dynamic Host Configuration Protocol)

Il **DHCP** permette agli host, dopo lo startup, di ottenere un indirizzo IP da un server, evitando configurazioni manuali.



Livello Rete / Network: NAT

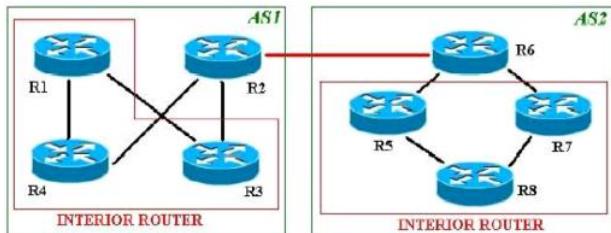
Il **Network Address Translation (NAT)** è un meccanismo che permette di mappare un indirizzo IP in un altro indirizzo IP. Le reti locali hanno diversi indirizzi **IP privati** che riguardano precisi dispositivi connessi alla rete. Attraverso il NAT, questi indirizzi privati vengono tradotti in un indirizzo **IP pubblico** quando le richieste in uscita vengono inviate ad Internet. Il processo inverso si verifica nel caso i dati siano in entrata.



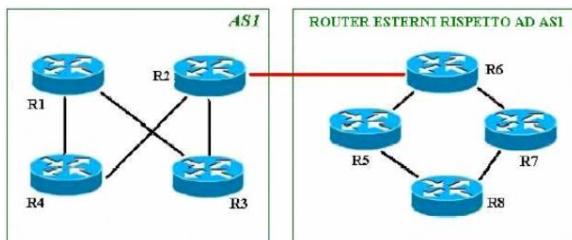
Livello Rete / Network: approfondimento sui router

Un router monta un **sistema operativo** nella sua memoria chiamato **Cisco IOS** ed è caratterizzato da **porte di ingresso ed uscita**, un **blocco di commutazione** che collega le porte di ingresso con quelle di uscita, un **processore di instradamento** che esegue protocolli di routing e aggiorna le tabelle di routing, **memoria ROM, RAM** (per le tabelle di routing), **NVRAM** (per le configurazioni di avvio), **Flash** (immagine IOS).

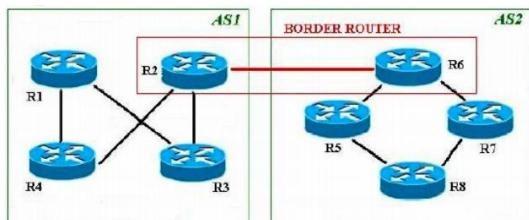
Il collegamento di più **reti sotto un unico dominio** prende il nome di **Autonomous System (AS)**. I router che instradano messaggi nello stesso **AS** e che non hanno diretta connessione con i router di altre reti sono detti **Interior Router**: scambiano informazioni di instradamento tramite un **Interior Gateway Protocol**;



I router che instradano messaggi tra **AS diversi** sono detti **Exterior Router**: scambiano informazioni utilizzando un **Exterior Gateway Protocol**.



I router che hanno la funzione di fare da ponte di collegamento tra **AS diversi** vengono detti **Border Router** (anche router di frontiera).



Livello Rete / Network: matching riguardo il routing

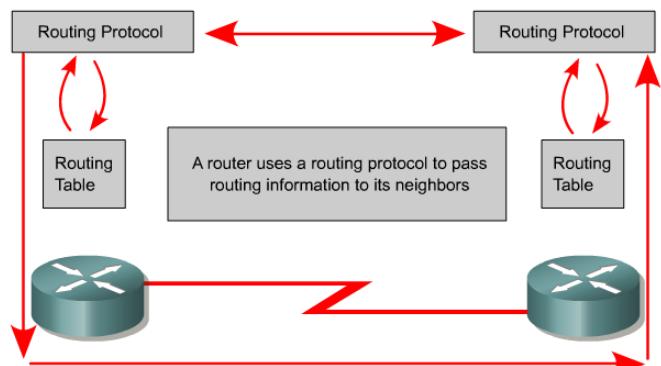
Per poter instradare i vari pacchetti, un router ha bisogno di alcune informazioni fondamentali, quali sono:

1. l'indirizzo IP dell'host di destinazione;
2. l'indirizzo dei router ad esso adiacenti;
3. i possibili percorsi alternativi per poter raggiungere reti remote.

Per valutare se un certo host con indirizzo X appartiene ad una sottorete con indirizzo Y/M, si effettua un'operazione di matching svolgendo $X \text{ AND } M = Y \text{ AND } M$: se il matching darà esito positivo per più righe nella tabella di routing, si attua la regola del **Longest Prefix Match**, cioè si utilizza la riga con il maggior numero di bit in comune X AND M. Quindi, sostanzialmente, dato un pacchetto con indirizzo di destinazione, viene effettuato il matching con tutti gli indirizzi IP nella tabella di **routing**: se la destinazione coincide con molteplici indirizzi della tabella di routing, allora tali indirizzi avranno maschere differenti. Verrà, quindi, scelto quello con maschera maggiore grazie al **Longest Prefix Match** e si proseguirà con la procedura **ARP**.

È possibile effettuare **due tipi di instradamento**:

1. **routing statico**, prevede il **calcolo dei percorsi offline**, quando la rete non è ancora attiva, a carico dell'operatore;
2. **routing dinamico**, permette ai **percorsi di cambiare dinamicamente** in base alle situazioni di traffico ed altre condizioni.



Il **routing statico** è ingestibile in condizioni di reti complesse e le destinazioni non possono cambiare, mentre con il **routing dinamico** vengono utilizzati protocolli per costruire le **tabelle di routing**: per costruire la tabella, ciascun router dovrà scambiare pacchetti informativi con i router ad esso collegati.

Livello Rete / Network: algoritmi basati sul percorso più breve

Una rete è rappresentabile sottoforma di grafo, dove ogni nodo rappresenta un router ed ogni arco rappresenta una linea di comunicazione, detta anche **canale**. Per scegliere un percorso/cammino tra due router, l'algoritmo cerca il più breve tra essi considerando le **metriche** possibili, quindi la distanza geografica, costi e capacità.

Introduciamo, quindi, la **metrica**, la quale offre una misurazione secondo un certo criterio: minore è la sua misurazione e più corto sarà il percorso. Uno degli algoritmi utilizzati per calcolare il percorso minimo è quello di Dijkstra (**Shorted Path First**): tale **algoritmo mantiene in una tabella la più piccola distanza conosciuta** per ogni destinazione e quale canale utilizzare per raggiungerla. Tali tabelle vengono aggiornate scambiando informazioni con i router vicini.

L'algoritmo di Dijkstra utilizza il protocollo **distance vector**: l'idea è quella di partire dal nodo sorgente e di guardare i nodi adiacenti assegnando loro il valore del costo per raggiungerli.

Facciamo un esempio: B raggiunge A in 5ms, C raggiunge A in 8ms e D raggiunge A in 4ms. Se Z raggiunge B in 2 ms, Z raggiunge C in 3ms e Z raggiunge D in 5ms, allora il percorso più breve sarà il seguente:

Z -> 2ms -> B -> 5ms -> A

Quindi Z -> 7ms -> A

Livello Rete / Network: algoritmi basati sul percorso più breve con “protocollo flooding”

Il **flooding** è un protocollo di instradamento usato dal router che **inoltra un pacchetto in ingresso su tutte le linee ad eccezione di quella da cui proviene** e viene solitamente usato per trovare il **percorso migliore**. Tale algoritmo genera un vasto numero di pacchetti duplicati, raggiungendo anche l'infinito, quindi si associa un contatore al fine di evitare ciò: se il contatore raggiunge lo 0, il pacchetto viene eliminato.

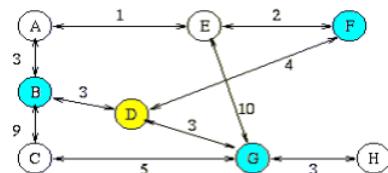
Gli aspetti negativi di questo algoritmo sono legati alla sua inefficienza, siccome manda ogni pacchetto su ogni rete provocando un utilizzo inefficiente della rete. Gli aspetti positivi riguardano il fatto che non c'è bisogno della conoscenza della topologia della rete che il pacchetto attraversa e che ogni pacchetto arriverà nel minor tempo possibile (siccome segue tutte le strade, quindi anche la più veloce).

Tale protocollo viene scarsamente utilizzato per via della sua inefficienza.

Livello Rete / Network: algoritmi di routing link state

Tali algoritmi nascono con l'intenzione di sostituire gli algoritmi **Distance Vector** e si basano sull'invio di pacchetti detti **Link State Packet (LSP)** contenenti le **informazioni** di **costo** e di **ritardo** di **ogni link uscente dal nodo** su cui si opera. La propagazione degli **LSP** avviene tramite **flooding**. Ogni nodo utilizza queste informazioni per calcolare il costo minimo verso tutti i nodi. Quindi, sostanzialmente, il router operante associa ad ogni destinazione un costo dipendente dalla linea (**link**) che collega i due nodi adiacenti.

Una volta ottenute le informazioni da tutti gli altri router della rete, si costruisce il grafo di rete e si utilizza **Dijkstra** per trovare il cammino minimo.



Gli algoritmi **LSP** non possono gestire qualsiasi rete di qualsiasi dimensione, quindi occorre realizzare il routing in modo gerarchico, suddividendo la rete in aree.

Livello Rete / Network: differenze tra link state e distance vector

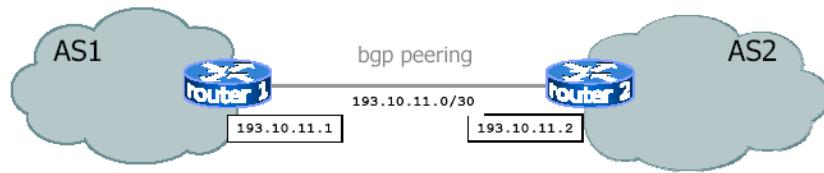
Sostanzialmente, i **protocolli distance vector** partono dal nodo sorgente e guardano i nodi adiacenti associando dei valori, quali saranno i costi per raggiungerlo. Si itera il ragionamento su ogni nodo.

I **protocolli link state**, invece, mandano informazioni in flooding sulla rete.

Riassumendo, i **link state** mandano informazioni a **tutti i router**, mentre i **distance vector** solo ai **nodi adiacenti**.

Livello Rete / Network: approfondimento sugli AS

Il **peering** è la connessione tra due **sistemi autonomi (AS)** appartenenti a **provider distinti**. L'instradamento tra due AS avviene sempre nello stesso modo che abbiamo discusso finora, con la differenza che esiste un unico algoritmo di instradamento tra organizzazioni adiacenti ed è necessario aggiornare le **tabelle di routing** manualmente aggiungendo **percorsi statici**. Ogni AS ha una topologia strana, composta da molteplici reti locali. Non tutte le reti locali sono connesse ad un **router di frontiera**, quindi è necessario comunicare all'esterno quali sono le reti raggiungibili. I router di frontiera permettono lo scambio di informazioni con altri router di frontiera, appartenenti ad AS differenti: ciò è possibile grazie al peering, il quale è la connessione tra due sistemi autonomi (AS) appartenenti a provider distinti.



Il **router di frontiera**, quindi, **fornisce comunicazione** tra AS differenti e si occupa dell'**instradamento**. Le informazioni tra due AS possono essere scambiare solo se la sessione **peering** è attiva, la quale è una connessione TCP.

Protocolli riguardanti i router presenti negli AS sono i seguenti:

- **IGP (Interior Gateway Protocol)**, protocolli che regolano l'instradamento dei pacchetti tra interior routers;
- **EGP (Exterior Gateway Protocol)**, protocolli che regolano l'instradamento dei pacchetti tra exterior routers;
- **BGP (Border Gateway Protocol)**, protocollo EGP che permette di collegare diversi border routers, quindi permette la comunicazione tra due o più AS differenti.

I protocolli IGP comprendono i protocolli **distance vector** e **link state**.

Protocolli Data-Link per WAN e LAN: IBSS, BSS ed ESS

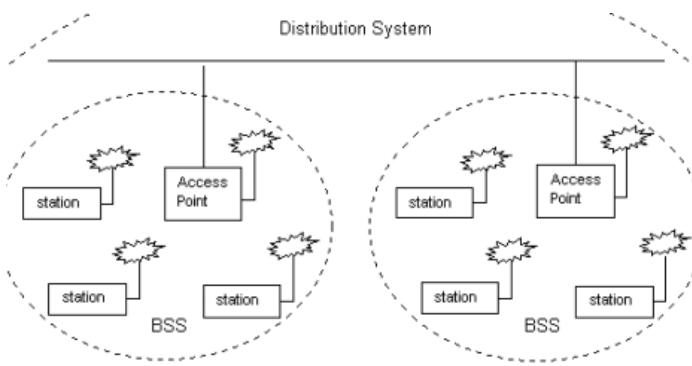
Nelle LAN i computer (o nodi) sono connessi tramite schede di rete e appositi cavi. Tra le reti LAN ci sono le **Wireless LAN (WLAN)**, nel caso si voglia utilizzare una rete senza fili. La WLAN è una rete locale in cui tutti i nodi comunicano tramite il canale radio, quindi senza fili. I vantaggi che offre sono molteplici, quali sono la **mobilità** (è possibile spostarsi con i dispositivi entro un certo range senza problemi); l'estensibilità ed il non utilizzo di cavi, in modo da potersi connettere anche dove non riuscirebbero ad arrivare i cavi; i costi sono relativamente bassi rispetto ad una LAN cablata; la configurazione è dinamica e vasta, quindi le WLAN offrono alta scalabilità. Le bande utilizzate nelle **WLAN** sono **2.4GHz** e **5GHz**.

Esistono due modalità di funzionamento per le WLAN: Ad **Hoc Network** (Independent Basic Service Set - IBSS) e **Infrastructure Mode** (Infrastructure Basic Service Set - BSS).

Le **IBSS** sono reti wireless in grado di connettere in maniera indipendente molteplici stazioni wireless tra loro senza l'utilizzo di un dispositivo centrale che faccia da tramite. Tale tipologia non è adatta in caso di reti con un gran numero di dispositivi.

Le **BSS** si basano su un **Access Point (AP) cablato** ad una LAN che funge da tramite per il traffico dei dispositivi wireless. Tale dispositivo permette l'accesso alla rete a dispositivi wireless. Nel caso si parli di AP pubblico, si parla allo stesso tempo di hotspot.

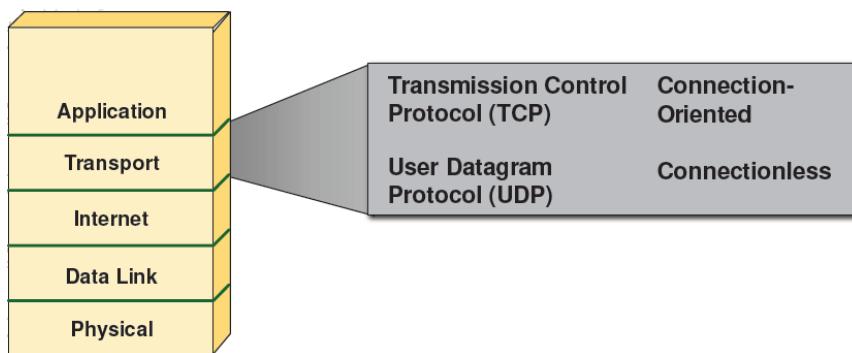
L'**AP** si interfaccia con il **Distribution System (DS)**, il quale non è altro che un dominio che **raccoglie molteplici BSS** che possono comunicare tra loro tramite i loro AP.



Altra tipologia di rete sono le **Extended Service Set (ESS)**, la quale si basa sul collegamento di molteplici **WLAN BSS** al fine di generare un'area di copertura notevolmente maggiore. Tale collegamento è dovuto dal DS. Gli elementi di rete al di fuori dell'ESS, vedono l'ESS come una singola rete con molteplici stazioni mobili.

Livello Trasporto

Il livello di trasporto ha lo scopo di fornire un **servizio di trasferimento end to end** al livello superiore, quindi si maschera il fatto che tra i due host che devono comunicare esista una rete di qualsiasi tipo, topologia e complessità. Per OSI lo strato superiore è il livello di sessione, mentre per TCP/IP è il livello di applicazione.



Livello Trasporto: TCP e UDP

Il livello di trasporto offre due importanti protocolli:

1. **Telegram Control Protocol (TCP)** riguarda la connessione **connection oriented**, quindi vengono effettuate consegne secondo il giusto ordine, viene controllata la congestione ed il flusso;
2. **User Datagram Protocol (UDP)** riguarda la connessione **connectionless**, quindi è considerabile inaffidabile siccome la consegna dei pacchetti viene effettuata senza alcun ordine e non usufruisce dei servizi di cui usufruisce TCP. Ergo, è un protocollo adatto nelle LAN ma non nelle WAN.

Andando nello specifico, **TCP** offre un flusso affidabile **end to end** con un funzionamento molto semplice: in **trasmissione** riceve il flusso di dati dall'applicazione, viene organizzato in pacchetti e spedito verso il destinatario; in **ricezione**, invece, viene ricostruito il flusso originale grazie ai pacchetti ricevuti.

UDP, invece, implementa un servizio di consegna inaffidabile siccome non si preoccupa di sapere nulla riguardo la corretta ricezione del pacchetto, permettendo però una **comunicazione molto più rapida** siccome non viene fatto alcun controllo riguardo la consegna dei pacchetti. Quindi, sostanzialmente, se si desidera la correttezza dei pacchetti si sceglie l'utilizzo di **TCP**; se si desidera una velocità maggiore con cui si comunica si sceglie l'utilizzo di **UDP**.

In generale, il **livello trasporto** su un **host** gestisce molteplici connessioni, **multiplando e demultiplando i pacchetti**: in **trasmissione** vengono raccolti i dati e vengono multiplati grazie ad un **multiplexer**, mentre in **ricezione** vengono demultiplati grazie ad un **demultiplexer** e consegnati alla socket corrispondente (la socket è la combinazione IP + porta).

Riguardo le socket, abbiamo la **socket UDP** e la **socket TCP**:

- **socket UDP** è identificata da due parametri, quali sono IP di destinazione e porta di destinazione;
- **socket TCP** è identificata da 4 parametri, quali sono IP di origine e destinazione, porta di origine e destinazione.

Livello Trasporto: le porte

Le applicazioni che utilizzano il **TCP/IP** si registrano ad un indirizzo detto **porta**. La **porta** è un indirizzo che indica l'applicazione remota a cui inviare i dati. Essa è un numero da 16 bit, quindi che va da 1 a 65535 (la porta 0 non viene utilizzata). Per connettersi ad un determinato servizio sul server, è necessario conoscere il numero di porta su cui il processo del server accetta le connessioni. Le porte inferiori alla 256 sono dette **porte ben note** e sono riservate a determinati servizi. Le porte ben note sono stabilite da un'autorità centrale, **IANA**.

Una particolare tipologia di porta è la “**effimera**”, la quale è una porta assegnata dinamicamente dal sistema operativo all'applicazione, sulla quale si riceveranno le risposte dal server, il quale utilizzerà le porte ben note.

Livello Trasporto: connessione con handshaking

Abbiamo, quindi, visto che con **TCP** è necessario instaurare una connessione al fine di poter proseguire con la spedizione dei pacchetti: come viene stabilita una connessione? Ebbene tramite una tecnica della **handshaking**: precisamente, quando si apre la connessione si usa la **3-way handshaking**, mentre quando la si vuole chiudere si usa la **4-way handshaking**. Vediamo ora come funziona precisamente:

L'handshaking non è altro che uno scambio di informazioni tra le due stazioni, il quale è accompagnato da un numero che indica quanti scambi vengono effettuati (ad esempio nel **3-way handshaking** vengono effettuati ben 3 scambi).

Poniamo di avere due DTE, detti A e B: A trasmette e B riceve.

Quando si apre la connessione si ricorre al **3-way handshaking**:

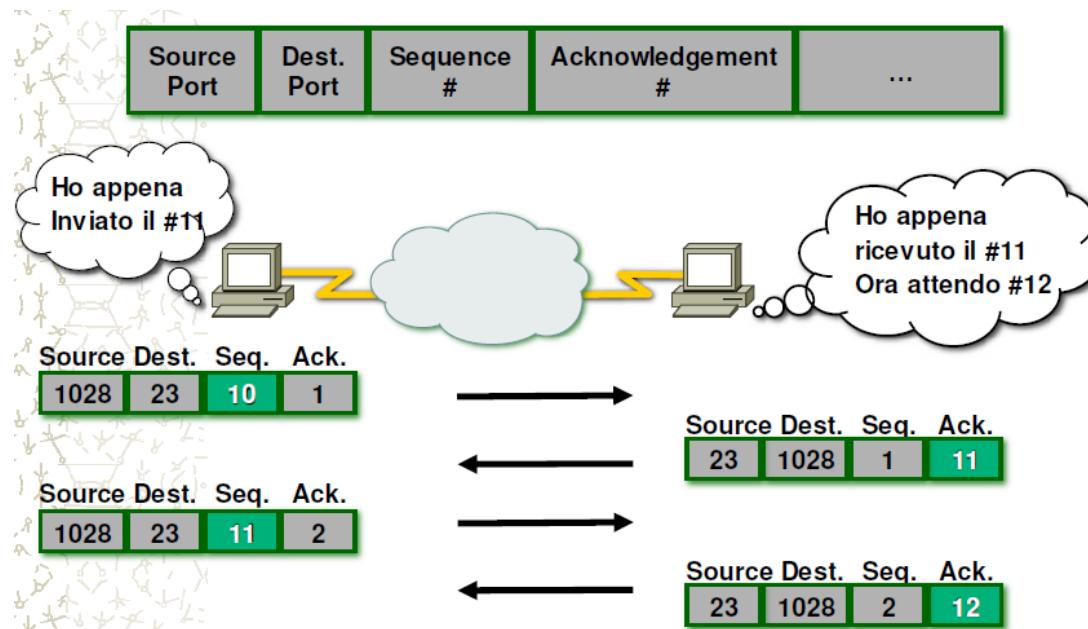
1. A vuole aprire la connessione, quindi manda una richiesta di connessione a B tramite un pacchetto TCP composto da un bit detto **SYN (flag di sincronizzazione)** e da un campo che indica il **numero di sequenza del pacchetto TCP**;
2. B riceve la richiesta di sincronizzazione, quindi deve notificare la corretta ricezione ad A: manda così un **pacchetto TCP con il SYN, il numero di sequenza del pacchetto TCP e l'ACK**, per notificare la corretta ricezione;
3. A riceve la notifica di B, quindi provvederà a sua volta con il mandare a B un **ACK** per notificare il corretto arrivo dell'**ACK referente al SYN**.

Effettuata tale procedura, le due stazioni saranno connesse e sincronizzate tra loro e potranno, quindi, trasmettere file.

Bisogna, però, considerare anche la chiusura della connessione, la quale ricorre al **4-way handshaking**. Essendo la connessione **full duplex**, dovrà essere chiusa indipendentemente in entrambi i nodi:

1. A vuole chiudere la connessione, quindi manda un **FIN (flag di chiusura)** a B, incluso **in un pacchetto TCP** contenente anche il **numero di sequenza** ed un **ACK** referente all'ultimo pacchetto ricevuto;
2. B riceve la notifica di chiusura e risponde con un ACK per notificare la corretta ricezione della notifica.

Come possiamo vedere, A ha fatto richiesta di chiusura, ma non B. In tal caso, B potrà ancora inviare dati ad A ma non viceversa, siccome la connessione è stata chiusa solo per A. Si itera, quindi, il ragionamento precedente anche per B.



Livello Trasporto: Approfondimento sul TCP

Abbiamo, quindi, capito che **TCP** offre una connessione affidabile **full duplex** con congestione e controllo del flusso. Vediamone ora il funzionamento in maniera più dettagliata: in **trasmissione** riceve un flusso di dati dall'applicazione che deve comunicare con il server, li organizza in unità lunghe al massimo 64Kb e le spedisce come **datagram IP**, il quale è il **pacchetto TCP**; in **ricezione**, invece, vengono ricevuti i **datagram IP**, vengono riordinati e viene ricostruito il flusso di byte originale nella sequenza corretta.

Riguardo la consegna dei pacchetti, **TCP** utilizza il meccanismo **sliding window** di tipo **Go-Back-N** con **timeout**: se il timeout scade, il pacchetto viene ritrasmesso. Sappiamo, come visto quando si parlò di sliding window, che è possibile che vengano recapitati molteplici pacchetti duplicati magari a causa di qualche ritardo: ebbene ciò è regolato dal **numero di sequenza** del pacchetto. Il tempo per cui si aspetta prima di dichiarare un timeout viene scelto da un algoritmo detto **RDT**. La dimensione della finestra da utilizzare è stabilita dal ricevente.

Riguardo il controllo del flusso, invece, bisogna prima sapere una cosa: gli **host** della **connessione TCP** riservano entrambi un buffer di ricezione, nei quali verranno collocati i dati correttamente ricevuti. L'applicazione leggerà, poi, quei dati direttamente dal **buffer**, ma non necessariamente nel preciso momento in cui verranno allocati nel buffer. Se l'applicazione che sta leggendo i dati è lenta, il sender potrebbe sovraccaricare l'applicazione che perderà i nuovi dati. La soluzione è proprio il controllo del flusso, il quale permette di adattare la velocità del mittente con quella del ricevente. Si introduce, quindi, la **finestra di congestione**, la quale impone un limite alla quantità di traffico che un **host** può inviare in una connessione. In particolare, la quantità di dati da inviare non può superare il minimo tra la **finestra di congestione** e quella di **ricezione**.

Livello Trasporto: Approfondimento su UDP

Riguardo **UDP**, abbiamo visto che offre una **connessione non affidabile** e che **non effettua controlli** riguardo il corretto arrivo dei pacchetti: difatti essi possono essere perduti durante il tragitto o addirittura consegnati fuori sequenza all'applicazione. I vantaggi che offre riguardano le prestazioni, permettendo elevata velocità di trasmissione. Inoltre, altra differenza importante riguarda i pacchetti: mentre **TCP suddivide il flusso di dati dell'applicazione** in molteplici pacchetti, **UDP cerca di maneggiarlo in un unico pacchetto**, in modo tale che in ricezione si riceva il flusso completo di dati dell'applicazione. Ovviamente se il flusso di dati è troppo grande per un unico pacchetto, viene spezzettato comunque in più pacchetti.

Nonostante sia inaffidabile, **UDP** offre caratteristiche appetibili: può utilizzare **trasmissione multicast** o **broadcast** a differenza di **TCP**, il quale non può gestire una comunicazione tra più di due entità; è molto più leggero ed efficiente grazie alla mancanza di controlli del pacchetto.

Livello Applicazione

La creazione di un'applicazione di rete consiste nella scrittura di un programma che giri su molteplici terminali diversi e che possa comunicare con altre istanze di sé stesso. Un esempio riguarda lo sviluppo web: il programma del server web comunica con i browser dei client.

Individuiamo, quindi, tre architetture utilizzabili nelle applicazioni di rete:

1. **Architettura client-server:** il **server** non è altro che un **host sempre attivo** con un **IP sempre attivo**, il quale mette a disposizione uno o più servizi per i client; il client comunica con il server e può farlo in qualsiasi momento, richiede e usa i servizi messi a disposizione del server, il suo IP è molto spesso dinamico, quindi varia.
2. **Architettura peer-to-peer (P2P):** non viene utilizzato alcun server, bensì **copie di host** comunicano tra loro. Tali host possono avere indirizzi **IP dinamici**, quindi che variano.
3. **Architettura ibrida (client-server e P2P):** un'implementazione di tale architettura è la **messaggistica istantanea**: la chat avviene in **P2P**, mentre l'individuazione del contatto da messaggiare avviene tramite l'architettura **client-server**, siccome il server conserva gli IP dei contatti da messaggiare.

Il programma in esecuzione su un host è detto **processo**, distinguibile in:

- **processo server**, in attesa di richieste;
- **processo client**, inizia la comunicazione.

Due processi sullo stesso host possono comunicare tra loro tramite schemi **interprocesso** prestabiliti dal sistema operativo; due processi su host diversi possono comunicare tra loro tramite scambio di messaggi. È importante notare che i programmi che utilizzano l'architettura P2P non usufruiscono di processi server, bensì solo di **processi client**.

I processi scambiano messaggi tramite **socket**, le quali sarebbero le porte, presupponendo che esista un'infrastruttura in grado di trasportare e recapitare i messaggi. Il mittente deve, quindi, conoscere l'indirizzo IP del destinatario, ma non solo: siccome sullo stesso host possono essere in esecuzione molti processi, è necessario conoscere anche il numero di porta utilizzato dal processo.

Livello Applicazione: DNS

Il DNS è un database distribuito che permette di convertire i nomi degli **host** in **indirizzi IP numerici**. Sappiamo che per comunicare su internet sono necessari gli IP, ma l'utilizzo di nomi rende più semplice la comunicazione: per esempio, è più semplice ricordare www.google.it rispetto a 64.233.167.99.

Il **nome dell'host** non è altro che un insieme di caratteri alfanumerici separati da un punto, ad esempio di.unisa.it: il segmento più a destra rappresenta il **top level domain (TLD)**, il quale identifica la **nazione** di appartenenza **dell'host** oppure la **categoria** a cui appartiene (it, net, edu, gov, com, ecc.); il secondo segmento indica l'organizzazione a cui **appartiene l'host** (università, azienda, ente, ecc.); i successivi segmenti indicano lo specifico **host**.

Vediamo come funziona in maniera molto generale la richiesta di un IP:

1. Il client invia un messaggio di **richiesta DNS** specificando il nome dell'host da trasformare in indirizzo IP;
2. Dopo un ritardo che va da msec a decine di secondi, il client riceve un messaggio di **risposta DNS** che fornisce la coppia nome dell'host (**hostname**) ed **IP**.

Le richieste e risposte IP sono gestite da un enorme database distribuito e gerarchico:

server DNS root

server DNS com | server DNS org | server DNS edu → TLD server

server DNS amazon.com | server DNS pbs.org | server DNS umass.edu

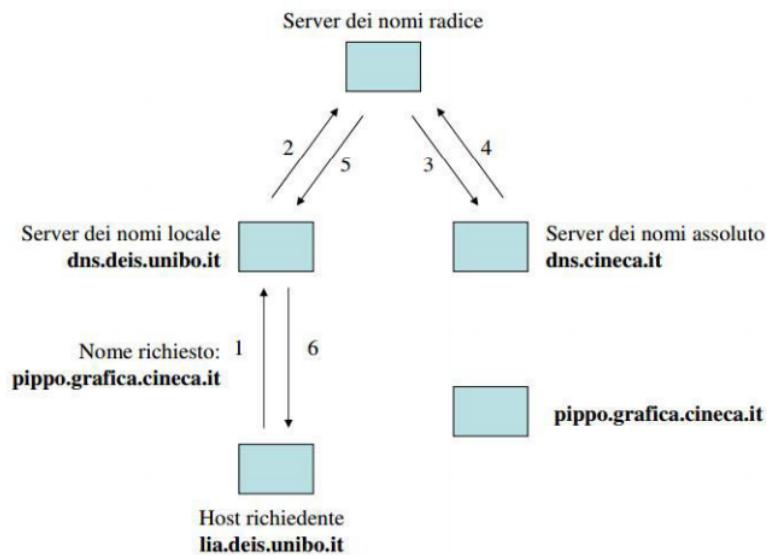
Facciamo un esempio: il client vuole l'IP di www.amazon.com:

1. Il client interroga il **server root** per trovare il server DNS **com**
2. Il client interroga il **server DNS com** per trovare il server DNS **amazon.com**
3. Il client interroga il **server DNS amazon.com** per ottenere l'IP di www.amazon.com

Il **DNS**, quindi, si basa sul **modello client/server** siccome, il client, richiede la traduzione del **nome** in **IP** al **server**. Quindi, quando un'applicazione deve convertire un nome di host nell'indirizzo IP desiderato o viceversa, chiama una procedura detta **resolver**, che contatta il **DNS server** del suo dominio.

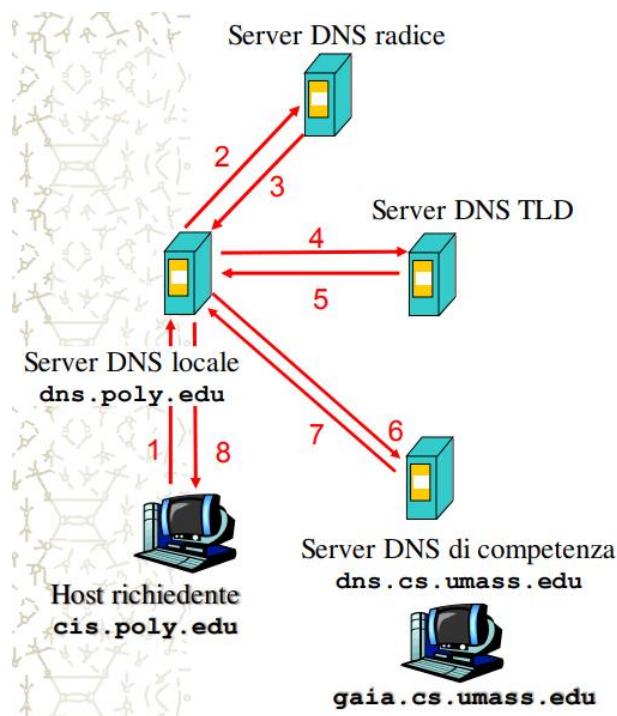
Il **server DNS** è quella parte dell'applicazione che stabilisce la corrispondenza tra **hostname** ed **indirizzo IP**. Ogni ISP (**Internet Service Provider**) possiede un server dei nomi locale (il quale non appartiene alla gerarchia vista prima), quindi la richiesta viene mandata prima al server dei nomi locali: se tale server non può soddisfare la richiesta di un host, si comporta come un **client DNS** ed inoltra la richiesta al root server. A questo punto subentrano due casi:

1. Nel primo caso il **root server** ha la corrispondenza hostname ed indirizzo IP, quindi invia la risposta al server dei nomi locale che la rinvierà al richiedente;
2. Nel secondo caso il **root server** non ha la corrispondenza giusta, quindi inoltrerà la richiesta verso un **server dei nomi assoluto** (un server dei nomi è detto assoluto se possiede la corretta correlazione).

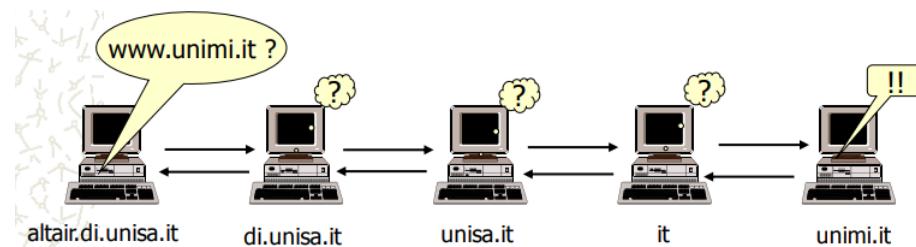


1. Viene fatta richiesta e controllato se pippo.grafica.cineca.it è presente nel server dei nomi locale dns.deis.unibo.it;
2. Non essendo presente nel server dei nomi locale, viene fatta richiesta al server dei nomi radice;
3. Il server radice non possiede la correlazione, quindi effettua una richiesta al server dei nomi assoluto (cioè il server dei nomi che possiederà sicuramente la correlazione);
4. /5/6. Il server dei nomi assoluto contiene la correlazione, quindi la manda al server dei nomi radice, il quale la invierà al server dei nomi locale per poi arrivare all'host richiedente.

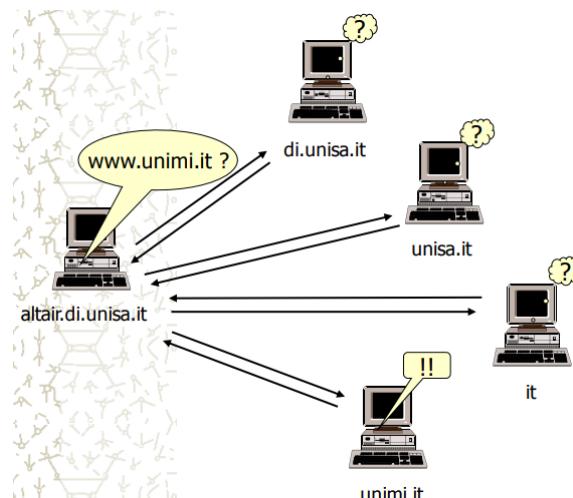
Precisamente, il server radice trova il server assoluto seguendo la normale prassi già vista, quindi passa per il server **TLD** per poi trovare il server di competenza.



Sostanzialmente, quindi, nel caso *l'hostname* sia relativo ad una zona “non coperta” dal server locale, viene effettuata una ricerca ricorsiva riguardo il server corretto: la richiesta viene propagata ai **name server** di livello superiore finché non si trova quello con la risposta giusta.



Altro modo di trovare il **name server** corretto è quello di affrontare un approccio iterativo.



Livello Applicazione: posta elettronica

La posta elettronica comprende ben tre agenti: *l'utente*; il **server di posta**; il **protocollo di trasferimento**, detto **SMTP** (*simple mail transfer protocol*).

L'utente, detto anche **mail reader**, usa programmi per leggere e gestire la propria posta.

Il server di posta contiene i messaggi in arrivo per gli utenti e una coda di messaggi da trasmettere. Utilizza il **protocollo SMTP** per inviare messaggi di posta elettronica: il client trasmette messaggi, mentre il server li riceve. Tale server è detto anche **relay agent** e fa da tramite nei trasferimenti: i client inviano la posta al **relay** che la invia al destinatario.

Il **protocollo SMTP** permette di trasferire messaggi, il che corrisponde ad un trasferimento di file. Tale protocollo usa **TCP** per trasferire in modo affidabile i messaggi siccome non sono ammesse perdite, utilizza la porta **25**. Il trasferimento avviene in tre fasi: **handshaking** per instaurare la connessione; viene trasmesso il messaggio; viene chiusa la connessione.

Ogni utente si identifica tramite l'indirizzo di posta elettronica, ovviamente univoco. Hanno il seguente formato:
username@host.domain

I messaggi sono formati da un'intestazione, nella quale troviamo mittente, destinatario e oggetto, ed il corpo, il quale conterrà il vero contenuto del messaggio.

Livello Applicazione: web

Il web non è altro che un enorme grafo di documenti distribuiti su server collegati ad internet. La maggior parte dei documenti sul web è in formato **HTML (HyperText Markup Language)**. Tali documenti possono essere elaborati tramite il protocollo **HTTP (HyperText Transfer Protocol)**: tale protocollo ha permesso la nascita del **World Wide Web (WWW)**, cioè l'enorme “ragnatela” quale è l'internet di oggi.

Una pagina web non è altro che un **documento HTML**, quindi un documento contenente informazioni. Tali pagine vengono lette dai browser, i quali rendono leggibili tali pagine all'utente finale interpretando l'HTML. Una pagina web ha uno o più **URL (Uniform Resource Locator)**, il quale permette il raggiungimento della pagina. L'URL è una sequenza di caratteri che identifica univocamente una risorsa nell'internet ed è composto da **protocollo** (http, https, ftp) e **hostname** (indirizzo fisico del server). Parametri opzionali dell'URL sono:

- **username:password@**, con tali parametri è possibile specificare l'autenticazione per l'accesso alla risorsa. Ovviamente i parametri inseriti, trovandosi nell'URL, saranno ben visibili. Ciò va posto tra protocollo ed hostname;
- **Porta**, necessaria da indicare quando il processo server è in ascolto su una porta non conforme allo standard;
- **Pathname o percorso**, da utilizzare per indicare una precisa risorsa come un'immagine o un qualsiasi file multimediale. Il percorso è posto sempre dopo l'hostname, ad esempio: <http://www.unisa.it/notizie/video.html>
- **Querystring**, permette di passare al server uno o più parametri. La querystring è sempre posta dopo hostname e percorso, ad esempio: <http://www.unisa.it/notizie.php?type=informatica>

Ogni sito web ha un processo server in ascolto su una porta TCP, il quale numero è 80: viene utilizzata per trasferire ogni tipo di risorsa sul WWW.

Introduzione

Componenti di una rete:

Hardware:

- Apparati di interconessione
- Apparati per il controllo della trasmissione

Software:

- Protocolli e Drivers:
 - codifica e formattazione dei dati
 - rilevamento di errori e correzione
 - controllo della congestione
 - Qualità del servizio

Funzionalità di una rete, è di fornire una comunicazione:

- Affidabile
- Efficiente
- Scalabile
- In grado di connettere ambienti applicativi diversi

Rileva e corregge automaticamente:

- Dati corrotti o persi
- Duplicazioni di dati (se si perde l'ack)
- Distribuzione con ordine diverso pacchetti

Trova cammini ottimali da una specifica sorgente a una specifica destinazione.

Una rete telematica è un insieme di dispositivi informatici mutuamente collegati tra di loro.

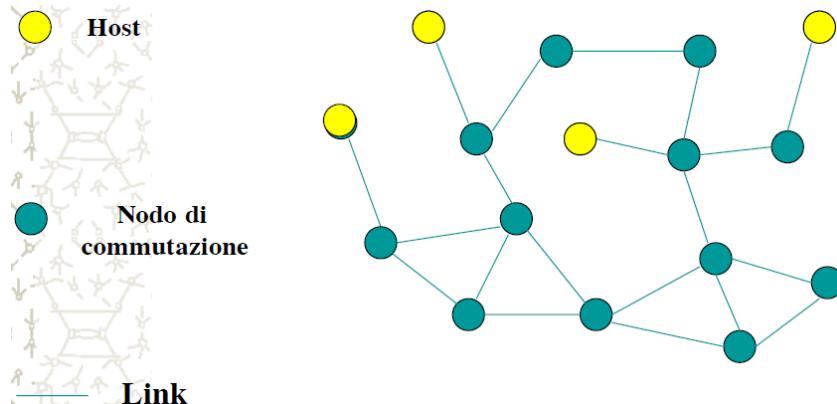
Una **Rete di Telecomunicazione** è definita come un sistema distribuito che permette la trasmissione di informazioni da un suo capo all'altro, consentendo un indirizzamento universale.

Quindi una rete deve implementare:

- funzionalità per il **trasporto dell'informazione**,
- funzionalità per l'**indirizzamento** e per la **commutazione (switching)**.

Un possibile modello fisico che implementa la definizione data di rete di telecomunicazione deve prevedere la presenza:

- **hosts (stazioni)** dispositivi autonomi connessi a una rete
- **links (collegamenti trasmissivi)**, **punto-a-punto**, interconnessi fra loro tramite nodi di commutazione
- **nodi di commutazione (Network switch)**, il cui compito è quello di riconoscere le richieste per l'apertura di una connessione e fare in modo che i dati, relativi a tale connessione, arrivino al nodo di destinazione.

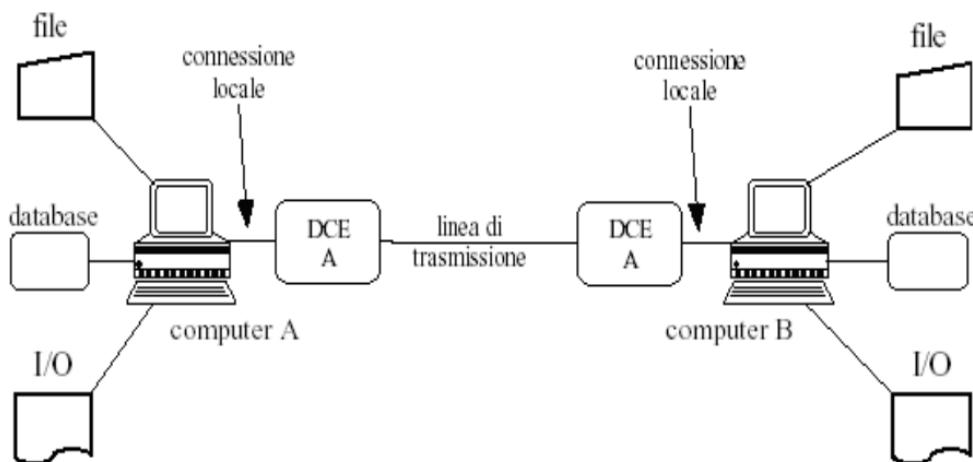


Terminologia

Il **Data Terminal Equipment** (abbreviata in **DTE**) è un qualunque dispositivo che svolge le funzioni di *sorgente* o *destinazione* di una comunicazione dati. Il **DTE** in trasmissione converte i dati dell'utente in segnali e in ricezione riconverte i segnali ricevuti.

Il **DTE** è collegato a un circuito di trasmissione dati tramite un **Data Communications Equipment (DCE)**. Di norma i dispositivi **DCE** forniscono il clock (timing interno), mentre i dispositivi **DTE** sincronizzano l'orologio fornito (timing esterno).

Generalmente il **DTE** è un *terminale* o un *personal computer*, mentre il **DCE** è un *modem*. Si può perciò affermare che lo scopo della rete è l'interconnessione dei vari DTE per la condivisione delle risorse, lo scambio di dati e la cooperazione tra i processi applicativi.

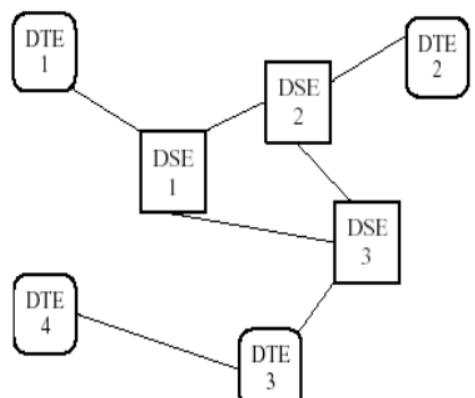


1. Il computer A e tutte le risorse (file database I/O) ad esso connesse costituisce il DTE A, mentre il computer B, con le proprie risorse, costituisce il DTE B.
2. Ciascun DTE è collegato alla linea di trasmissione mediante un apposito dispositivo, che prende il nome di Data CircuitTerminating Equipment (brevemente DCE).

Quando la linea di trasmissione è la normale linea telefonica, il DCE è un normale modem. Il DCE può essere uno switch o un router in ambito Ethernet.

Un **Data Switching Equipment** (brevemente **DSE**) o *nodo di commutazione* è un nodo intermedio della rete, senza alcuna funzione di supporto diretto agli utenti, la cui principale funzione è quella di commutare (switch) il traffico tra due o più DTE non direttamente collegati tra loro.

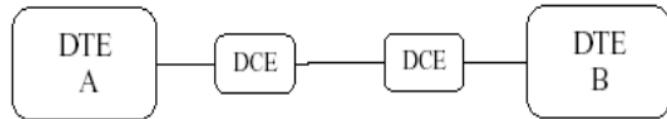
Un DSE sceglie dunque la strada (detta **percorso di rete**) che i messaggi devono seguire per arrivare alla loro destinazione



Modalità di trasmissione

- **Reti punto-a-punto:**

Un circuito fisico è detto **punto-a-punto** quando collega due soli DTE.



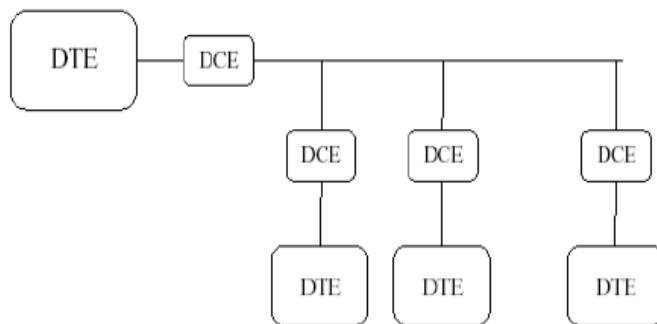
Il collegamento punto-a-punto è spesso utilizzato nella connessione tra due computer oppure in quella tra un computer ed un terminale. I principali vantaggi:

- *semplicità di gestione*: quello che viene trasmesso da un DTE è sempre diretto all'altro;
- *tempi di attesa nulli*: il DTE che deve trasmettere trova sempre il circuito disponibile, per cui può trasmettere ogni volta che ne ha bisogno.

Ma il costo della linea, specie se essa corre su una distanza notevole, può diventare elevato. Inoltre, una organizzazione che volesse collegare, al proprio mainframe, 10.000 terminali con questa tecnica, dovrebbe provvedere a installare 10.000 linee di collegamento.

- **Reti multipunto:**

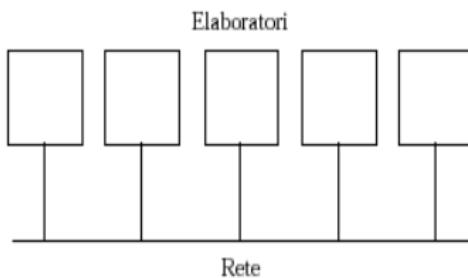
Un circuito fisico **multipunto** consiste nel mettere più di due DTE sulla stessa linea. Con DTE principale/master e secondari/slaves:



- **Reti broadcast:**

All'opposto delle reti multipunto e punto-a-punto si collocano le cosiddette **reti broadcast**: queste sono dotate di un unico canale di comunicazione che è condiviso da tutti gli elaboratori.

Brevi messaggi (spesso chiamati **pacchetti**) inviati da un elaboratore sono ricevuti da tutti gli altri elaboratori. Un indirizzo all'interno del pacchetto specifica il destinatario.



Funzionamento:

Quando un elaboratore riceve un pacchetto, esamina l'indirizzo di destinazione; se questo coincide col proprio indirizzo, il pacchetto viene elaborato, altrimenti viene ignorato.

Le reti broadcast, in genere, consentono anche di inviare un pacchetto a tutti gli elaboratori, usando un opportuno indirizzo. Si parla in questo caso di **broadcasting** (si pensi alla diffusione radio-televisiva). *In tal caso tutti prendono in considerazione il pacchetto.*

- **Reti multicast:**

Un'altra possibilità è inviare il pacchetto ad un sottoinsieme degli elaboratori: si parla in questo caso di **multicasting** e succede che solo gli elaboratori del suddetto sottoinsieme prendono in considerazione il pacchetto, che invece viene ignorato dagli altri.

In ciascun pacchetto è presente un bit che indica che si tratta di una trasmissione in **multicasting**, mentre i rimanenti bit contengono l'indirizzo del gruppo destinatario ed ovviamente i dati.

Flussi e circuiti

Il flusso trasmissivo, lungo una linea di comunicazione, può avvenire in 3 modi diversi:

- **Trasmissione simplex:**

I dati viaggiano, in questo caso, in una **sola direzione**. Esse le trasmissioni radio-televisive e le reti di comunicazione delle agenzie stampa.

Generalmente, il flusso trasmissivo di tipo simplex **non viene utilizzato** per la comunicazione dei dati, anche quando il flusso è unidirezionale: il motivo è che, nella comunicazione dei dati, è assolutamente necessario il *controllo della correttezza della ricezione*;

Questo controllo è possibile solo se l'utente, una volta ricevuti i dati inviati dalla sorgente, può a sua volta inviare alla sorgente un messaggio che indichi la corretta ricezione o, in caso contrario, che richieda la *ritrasmissione*.

- **Trasmissione half-duplex:**

Nella trasmissione **half-duplex**, invece, i dati possono viaggiare in **entrambe le direzioni, ma non contemporaneamente**.

È il modo classico di operare dei *terminali conversazionali*, che prevede l'invio di una richiesta, la ricezione della risposta e, sulla base di quest'ultima, l'invio di una ulteriore richiesta e così via.

- **Trasmissione full-duplex:**

Il modo più completo e anche più complesso è quello della trasmissione **full-duplex**: in questo caso, i dati possono **viaggiare, contemporaneamente, in entrambe le direzioni**.

Esempio classico è il colloquio tra due sistemi in cui mentre si trasmette un certo file in una direzione, ne viene trasmesso un altro nella direzione opposta.

Osserviamo che *il flusso full-duplex è particolarmente indicato per le reti a configurazione multipunto*: infatti, se la linea di trasmissione è di tipo full-duplex, è possibile che il *DTE master* riceva una richiesta da un *DTE slave* e, contemporaneamente, invii una risposta ad un altro *DTE slave*.

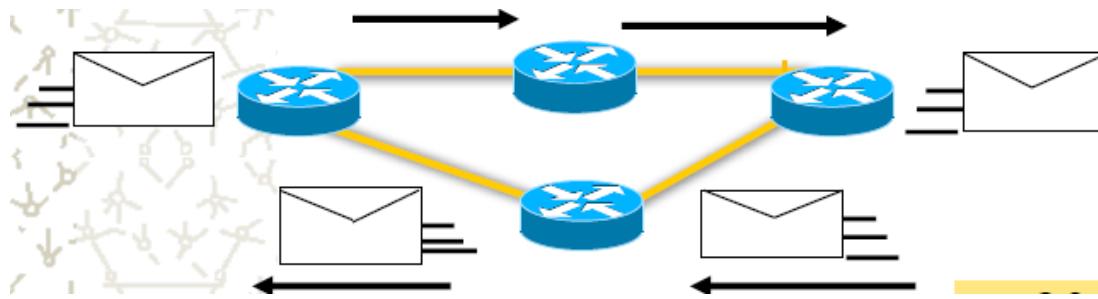
Trasmissione dei dati: la commutazione

È quell'operazione che predispone il percorso che le informazioni emesse dal mittente devono seguire per raggiungere il destinatario. Esistono fondamentalmente 2 tipi di commutazione:

- **Commutazione di pacchetto:**

Utilizzata per condividere un canale di comunicazione tra più nodi, suddividendo l'informazione da trasferire in pacchetti trasmessi individualmente e in sequenza, seguendo un meccanismo di instradamento dettato da relative tabelle di instradamento.

L'utilizzo ottimale delle risorse viene effettuato con il principio di **multiplazione statistica** (quando il canale è libero, viene usato da qualche altro per inviare altri pacchetti).



- **Commutazione a circuito:**

Avviene tramite commutatori (dispositivi di commutazione) che non sono altro che nodi intermedi (es. DSE o DCE) i quali determinano una connessione fisica diretta tra due stazioni che necessitano di comunicare. Questa connessione è assegnata alla coppia di stazione ed è mantenuta fino al termine della comunicazione.

Topologie di rete

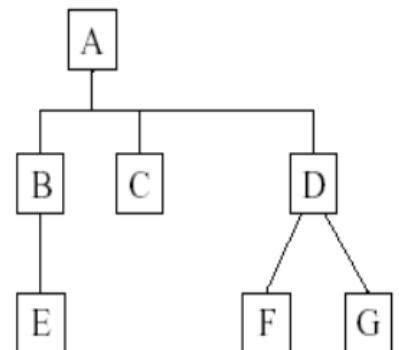
- **Rete gerarchica o ad albero:**

Questo tipo di configurazione è quella più comune. Il traffico di dati va dai nodi dei livelli più bassi verso i nodi intermedi o verso il nodo del livello più alto.

Quest'ultimo è in genere il nodo più potente dell'intera struttura, visto che deve provvedere alle richieste di tutta la rete.

Alcuni inconvenienti sono:

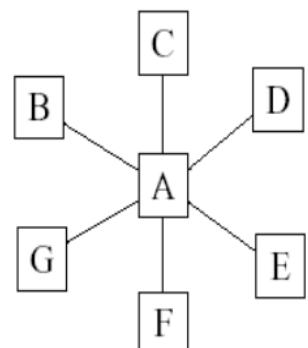
- il nodo principale, se è sovraccarico di lavoro, può diventare **un collo di bottiglia** per l'intera rete, il che comporta un rallentamento dei servizi per tutti gli utenti;
- inoltre, la caduta del nodo principale rende inoltre inutilizzabile l'intera rete.



Si può però ovviare adottando una **politica di back-up**: bisogna cioè mettere in grado uno o più altri nodi della rete di svolgere le stesse funzioni del nodo principale nel momento in cui questo dovesse venire a mancare.

- **Rete a stella:**

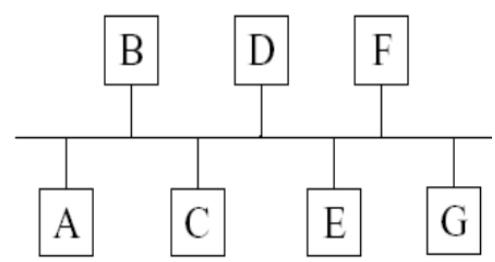
La configurazione a stella è simile a quella ad albero, con la fondamentale differenza che non c'è alcuna distribuzione funzionale, ossia tutte le funzioni riguardanti gli utenti periferici sono realizzate nel nodo centrale.



- **Rete a dorsale o bus condiviso:**

Questa configurazione è diventata popolare in quanto è adottata dalle reti locali di tipo **Ethernet**. La caratteristica è che c'è un unico cavo che collega tutte le stazioni.

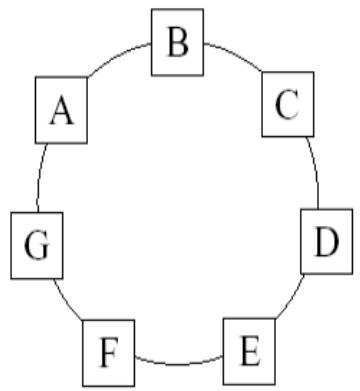
La trasmissione di una stazione viene ricevuta da tutte le altre. In ogni istante solo un elaboratore può trasmettere, mentre gli altri devono astenersi. È necessario un meccanismo di **arbitraggio** per risolvere i conflitti quando due o più elaboratori vogliono trasmettere contemporaneamente



- **Rete ad anello (ring):**

Questa configurazione è stata resa da popolare dalle prime **LAN** (*Local Area Network*) di tipo *Token-Ring*.

La trasmissione è in questo caso unidirezionale (i dati viaggiano cioè solo in un senso), ma, essendo l'anello un circuito chiuso su sé stesso, è possibile inviare un messaggio da qualsiasi stazione verso qualsiasi altra. Anche qui è necessario un meccanismo di **arbitraggio**

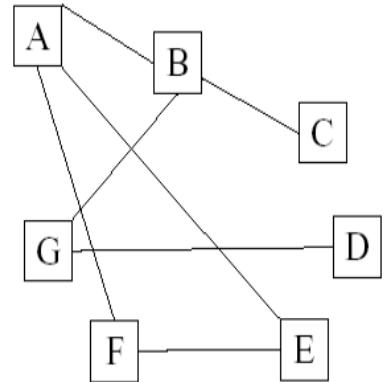


- **Rete a maglia o mesh:**

Quest'ultima topologia consiste nel collegare le varie stazioni con diversi circuiti.

Una topologia di questo tipo assicura buone prestazioni in quanto il traffico viene ripartito sui vari percorsi. Inoltre, essa conferisce una elevata affidabilità all'intera struttura, proprio grazie alla presenza di *percorsi multipli*.

Allo stesso tempo, però, i costi dei collegamenti possono anche essere elevati ed inoltre la gestione della struttura è chiaramente più complessa.



Protocolli

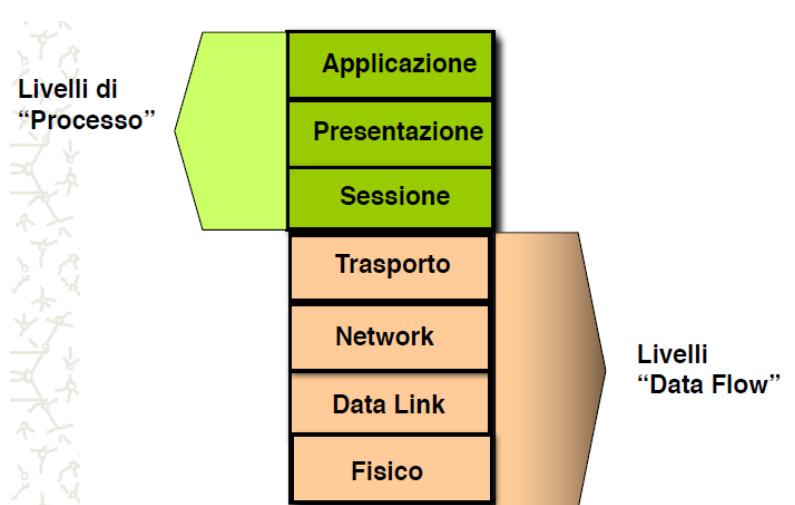
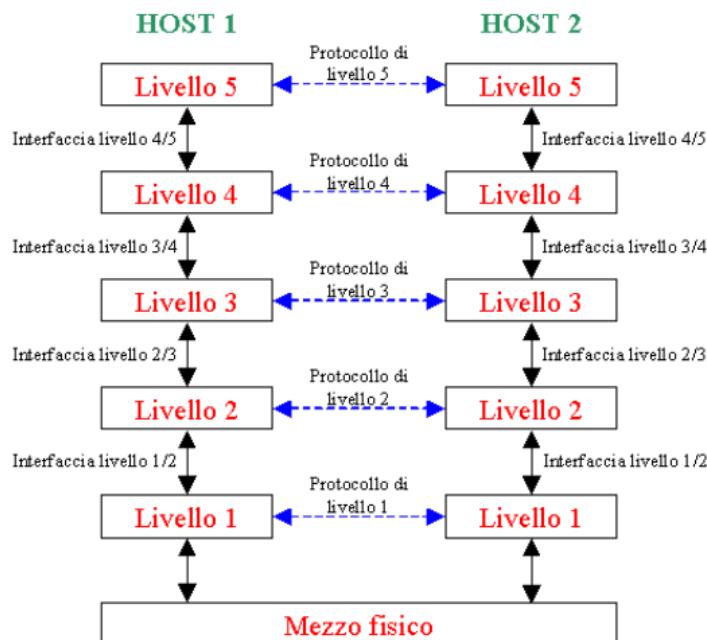
Serie di norme, convenzioni e tecniche per lo scambio di dati, comandi e informazioni di controllo tra due elementi.

Esistono molti **livelli** di protocolli: si va dal livello più basso, che regola il modo di trasmettere i segnali sulla linea (**protocollo di connessione**), al livello più alto, che indica come interpretare dati e comandi a livello applicativo.

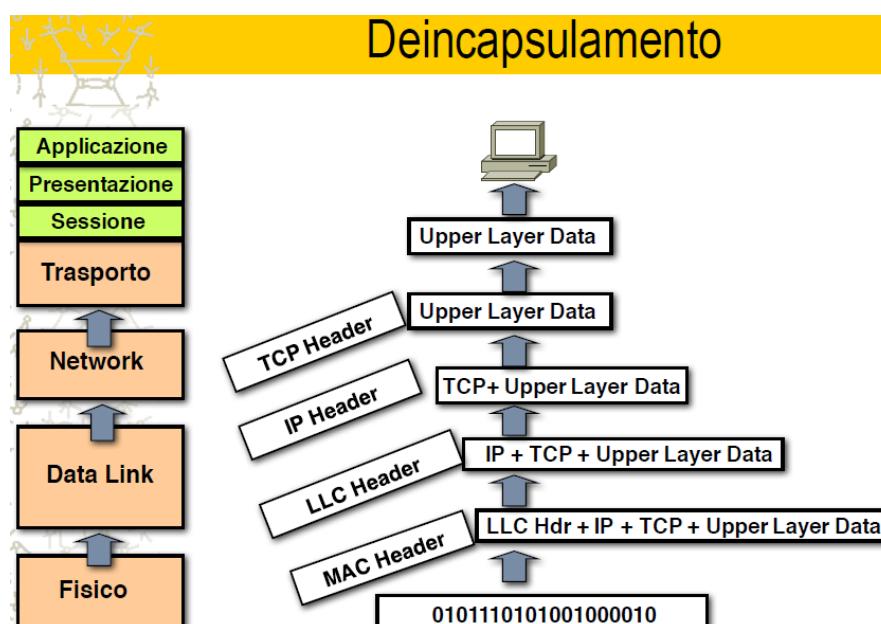
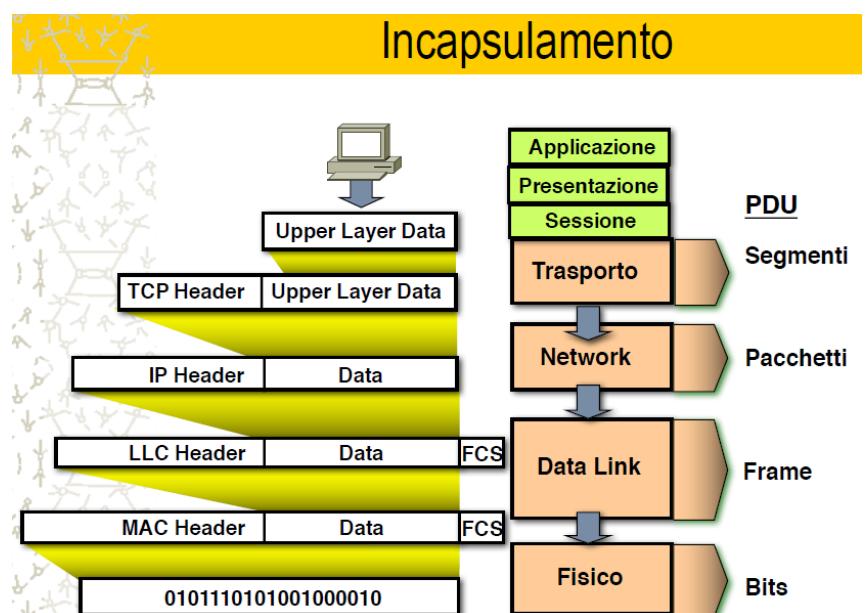
Modello ISO-OSI

ISO – International Standard Organization

OSI – Open System Interconnection



Applicazione	User Interface	Telnet HTTP
Presentazione	• How data is presented • Special processing such as encryption	ASCII EBCDIC JPEG
Sessione	Keeping different applications' data separate	Operating System/ Application Access Scheduling
Transport	• Reliable or unreliable delivery • Error correction before retransmit	TCP UDP SPX
Network	Provide logical addressing which routers use for path determination	IP IPX
Data Link	• Combines bits into bytes and bytes into frames • Access to media using MAC address • Error detection not correction	802.3 / 802.2 HDLC
Physical	• Move bits between devices • Specifies voltage, wire speed and pin-out cables	EIA/TIA-232 V.35



Tecniche di multiplazione

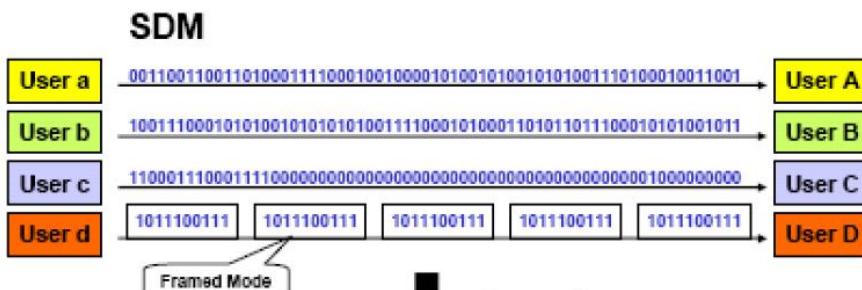
È una tecnica di trasmissione per cui più canali trasmissivi in ingresso condividono la stessa **capacità trasmittiva (larghezza di banda)** disponibile in uscita ovvero combinando **più segnali analogici** in un **solo segnale** (detto **multiplato**) trasmesso in uscita su uno stesso collegamento fisico.

La multiplazione permette di risparmiare sul cablaggio e sul numero di componenti. Ad esempio in elettronica il multiplexing permette a diversi segnali analogici di essere elaborati da un unico convertitore analogico-digitale.

Esistono diverse tecniche:

- **SDM (Space Division Multiplexing, divisione di spazio):**

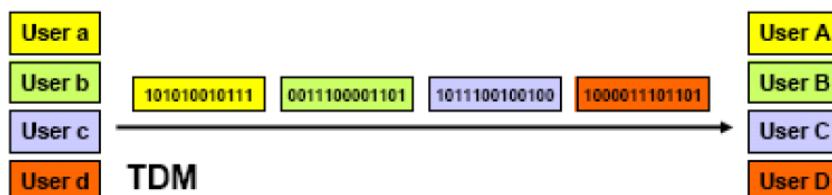
Ogni dispositivo ha uno spazio fisico separato rispetto agli altri. Es le stazioni radio FM che trasmettono solo in un determinato spazio (regione).



- **TDM (Time Division Multiplexing, divisione di tempo):**

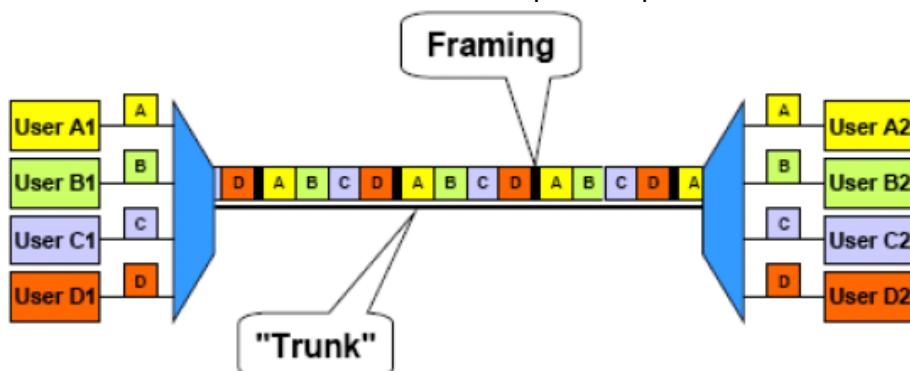
Il tempo di utilizzo del canale è organizzato in **frame** (organizzazione di flussi) tutti della stessa durata. Ciascuno di questi frame è ulteriormente suddiviso in **slot** (intervalli temporali).

Ogni canale logico occupa un intervallo di tempo, il MUX divide il canale in intervalli temporali che assegna poi ad ogni canale logico.



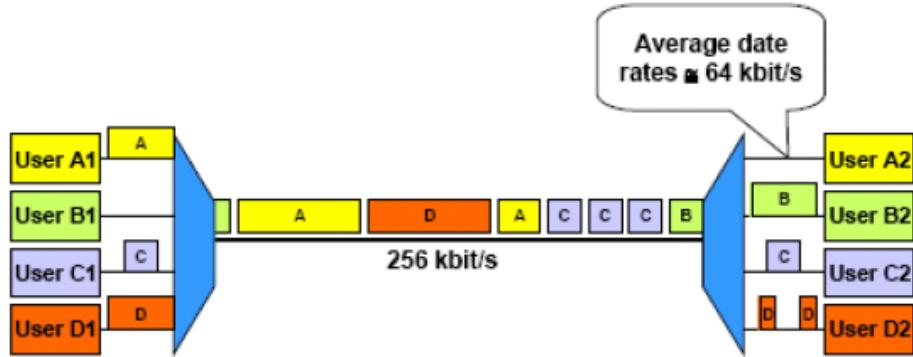
Esistono essenzialmente due metodi di multiplazione TDM:

- **Sincrono:** ogni canale di comunicazione è identificato dalla sua posizione in termini di slot temporali all'interno della **trama**. Questa correlazione fissa fra il canale di comunicazione e il relativo **timeslot** è il principale svantaggio del **TDM sincrono**: se il canale non è usato comunque occupa il **timeslot** inviando un **pattern idle**.



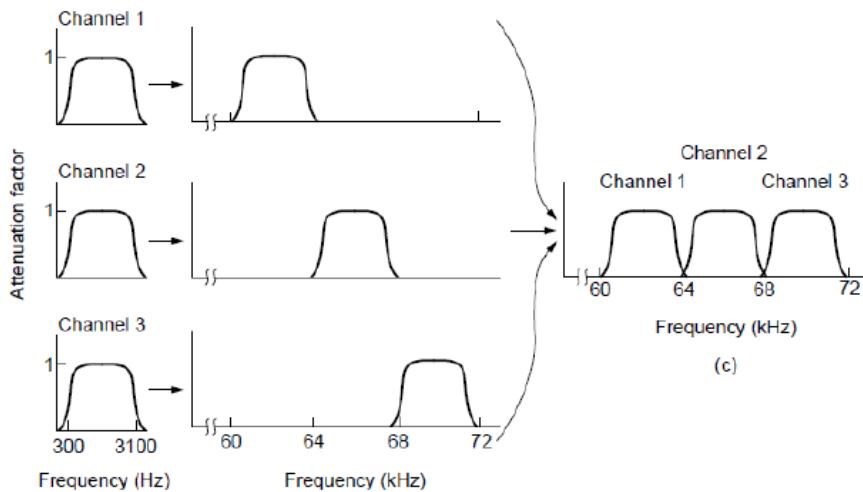
- **Statistico:** non esiste correlazione fra canale di comunicazione e relativo **timeslot**. La capacità del mezzo è distribuita statisticamente fra gli utenti che ne concorrono all'uso. È necessario uno schema separato di

tramatura e indirizzamento per garantire le associazioni dinamiche: se un canale non è usato gli altri canali possono disporre della sua capacità trasmissiva.



- **FDM (Frequency Division Multiplexing, divisione di frequenza):**

Ogni canale logico genera un segnale sulle stesse frequenze, questi segnali vengono modulati dal MUX in un unico segnale. Il DEMUX utilizza una serie di filtri per scomporre il segnale composto ricevuto nei segnali originali.



- **WDM (Wavelength Division Multiplexing):**

Simile FDM, le operazioni di divisione dei segnali luminosi possono essere facilmente effettuate attraverso un prisma che devia i raggi in base alla frequenza e all'angolo di incidenza, vengono raggruppati e divisi in questo modo.

- **CDM (Code Division Multiplexing):**

Un segnale a banda stretta viene sparso su una banda di frequenza più ampia, rendendo il segnale più tollerante alle interferenze, permettendo però a utenti diversi di condividere la stessa banda di frequenza (Es telefonia).

Livello Fisico – Lv 1

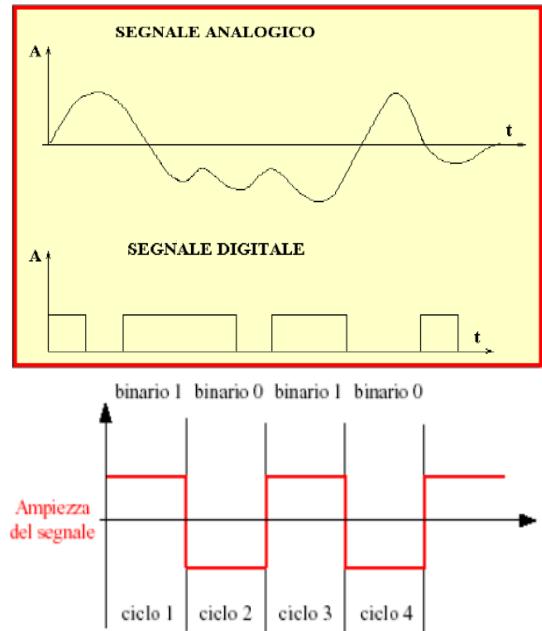
Nella trasmissione, questo livello riceve dal livello **datalink(Lv 2)** la sequenza di bit pacchettizzata da trasmettere sul canale e la converte in **segnali** adatti al mezzo trasmissivo come cavo coassiale (connettore BNC), doppino STP o UTP, fibre ottiche o onde radio.

I Segnali

I segnali sono variazioni di grandezze fisiche che trasportano informazioni. Le informazioni possono essere trasmesse tramite **segnali**, che possono essere di vario tipo: **acustico, elettrico, luminoso, elettromagnetico, ecc.**

I segnali elettrici trasmessi possono essere di due tipi:

- **ANALOGICI:** Sono analogici quei segnali che, al variare del tempo, possono assumere tutti i valori compresi fra i valori massimo e minimo consentiti dal canale di comunicazione.
- **DIGITALI:** Con il termine digitale, o **numerico**, si intende invece un segnale che può assumere solo due valori, o comunque soltanto un numero discreto di valori, come, ad esempio avviene per i dati che sono generati dai computer.



Per quanto riguarda la **trasmissione digitale dei dati**, questi valori discreti si ottengono facendo variare, nel modo quanto più brusco possibile, il valore del segnale da un livello all'altro.

Un **segnale analogico** è un'onda sinusoidale può essere rappresentata da tre parametri:

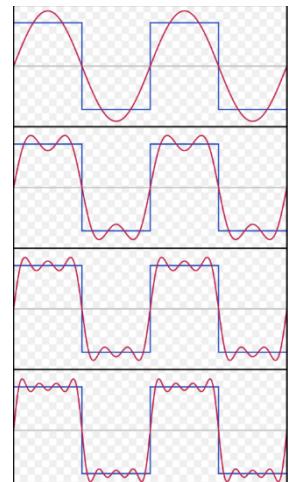
- **AMPIEZZA MASSIMA** di un segnale è il valore assoluto del segnale nella sua **intensità massima** (il picco massimo) ed è proporzionale all'energia trasportata dal segnale. Si rappresenta in VOLT.
- **FREQUENZA** è il numero di periodi in un 1s. Si indica con f: $1/T$. Dove T è il periodo (il tempo necessario affinché un segnale completa un ciclo). La frequenza quindi è la velocità con cui un segnale cambia rispetto al tempo. Cambiamenti veloci quindi implicano una frequenza alta, cambiamenti lenti implicano una frequenza bassa.
- **FASE** descrive la posizione dell'onda rispetto al tempo 0. Indica la posizione iniziale del primo ciclo. È misurata in gradi o radianti.

Serie di Fourier

Una funzione periodica **y(t)** è sviluppabile in una serie costituita da un termine costante **A₀** e da una somma di infinite sinusoidi:

$$y(t) = A_0 + \sum_{n=1}^{\infty} A_n \cos(n\omega_0 t) + \sum_{n=1}^{\infty} B_n \sin(n\omega_0 t)$$

Un'approssimazione di un'onda, mediante combinazioni lineari di funzioni sinusoidali.



Larghezza di banda di un canale

I canali di telecomunicazioni usati per trasmettere dati sono basati su mezzi trasmissivi quali: il rame, la fibra ottica, l'etero.

La larghezza di banda è la misura dell'ampiezza di banda dello spettro, misurata in Hertz. Può essere usata per valutare la larghezza di banda di un segnale informativo trasmesso dalla banda passante, disponibile o utilizzata, in un canale di comunicazione.

È definita pertanto larghezza di banda, l'insieme delle frequenze che un canale di telecomunicazioni fa passare.

Teorema del campionamento di Nyquist-Shannon

Il **campionamento** è una tecnica che consiste nel convertire un segnale continuo (nel tempo oppure nello spazio) in un segnale discreto, valutandone l'ampiezza a intervalli temporali o spaziali solitamente regolari, ottenendo così una stringa digitale che approssimi quella continua originaria.

Il teorema di **Nyquist-Shannon** definisce la minima frequenza (detta frequenza di Nyquist), necessaria per campionare un segnale analogico senza perdere informazioni, e per poter poi ricostruire il segnale analogico originario.

Capacità del mezzo trasmissivo

Affinché **l'informazione** viaggi a distanza, cioè in luoghi diversi, necessita di una elaborazione che la trasformi in **segnali elettrici** i quali, a loro volta, devono essere adattati ai canali utilizzati per il trasporto.

I canali trasmissivi utilizzati per la comunicazione dei dispositivi si suddividono in:

- **Canali ideali**: non causano distorsioni o ritardi nella propagazione dei segnali.
- **Canali non distorcenti**: causano solo un ritardo costante nella propagazione ed un'attenuazione costante in banda.
- **Canali distorcenti**: causano attenuazioni e ritardi, in funzione della frequenza dei segnali.

Il legame tra la **velocità di trasmissione (bit rate)** e la **larghezza di banda** è data dal **Teorema di Nyquist**.

Il massimo **bit rate**, ovvero la **capacità di canale**, relativo ai canali reali (con rumore termico) è dato dal **Teorema di Shannon** che considera anche il **rumore**.

Modulazione di un segnale

Una volta generato il segnale da trasmettere, questo può essere immesso direttamente sul canale. In questo caso si parla di trasmissione in **banda base**: il segnale che trasporta le informazioni ed il segnale sulla linea sono identici.

Vi sono diverse circostanze che rendono opportuno trasmettere il segnale in modo che occupi una **banda differente di frequenze**, questo tipo di trasmissione si realizza tramite un processo di **modulazione**.

La **modulazione** è un processo con il quale il **segnale da trasmettere** (segnaletico) viene utilizzato per **modificare nel tempo** le caratteristiche di un **segnale ausiliario sinusoidale (portante)**.

Utilizzando una **portante ad alta frequenza** si può quindi **spostare** la banda necessaria alla trasmissione delle informazioni in un intervallo **più opportuno** per la trasmissione stessa.

Un vantaggio è legato alla possibilità di trasmettere **più comunicazioni differenti** e contemporanee sullo **stesso mezzo**, trasferendo le bande relative alle diverse stazioni.

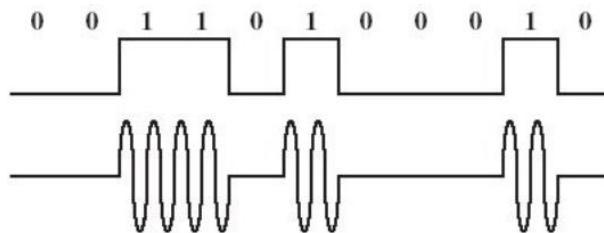
Tecniche di modulazione

Il **segnale modulante** viene utilizzato per modulare le caratteristiche della **portante**:

- **ampiezza**: il segnale viene utilizzato per modificare il valore della ampiezza della portante (modulazione di ampiezza)
- **frequenza**: il segnale modulante modifica istante per istante la frequenza della portante (modulazione di frequenza)
- **fase**: il segnale modulante cambia la fase della portante (modulazione di fase)

- **ASK: MODULAZIONE IN AMPIEZZA:**

Normalmente implementata usando solo 2 tipi di elementi del segnale, che possono assumere due forme: in una la sua ampiezza è **nulla**, nell'altra la sua ampiezza è uguale all'ampiezza **massima** del segnale portante. Questi due elementi rappresentano il valore del bit 0 o 1.

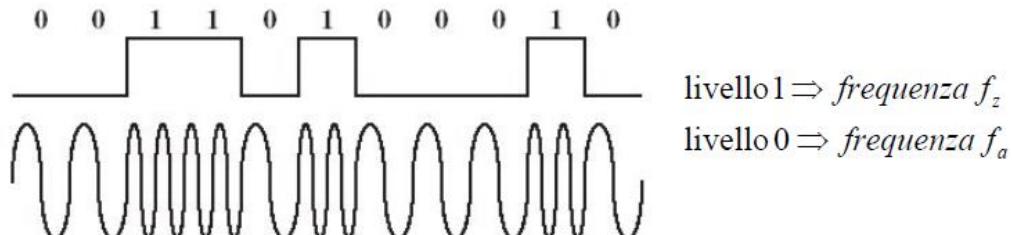


- **FSK: MODULAZIONE IN FREQUENZA:**

La modulazione FSK utilizza due frequenze portanti. A queste frequenze corrispondono i valori del bit 0 e 1.

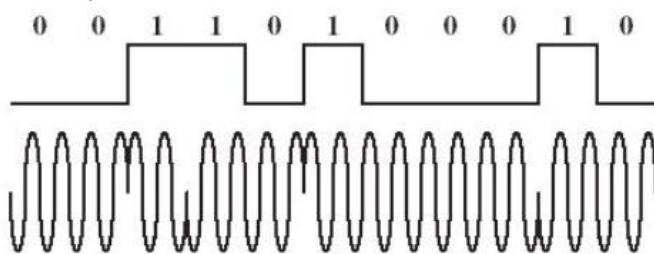
La prima frequenza portante viene utilizzata per il valore 0 e la seconda per il valore 1.

Un requisito importante nella FSK è la continuità di fase negli istanti di transizione da una frequenza all'altra.



- **PSK: MODULAZIONE IN FASE:**

La modulazione PSK è la più utilizzata rispetto alle ASK e FSK. Nella PSK è la fase a determinare il valore del bit. Al cambio di fase si associa il valore 1, viceversa si associa il valore 0.



- **QPSK: QUADRATURA PSK:**

La quadratura PSK utilizza 4 fasi diverse per ogni elemento del segnale, per questo si possono rappresentare 2 bit per ogni elemento del segnale. In pratica lo schema utilizza due modulazioni BPSK (Binary PSK) separate che poi vengono sommate in un unico segnale finale.

- **QAM: QUADRATURE AMPLITUDE MODULATION:**

Questa tecnica di modulazione mette insieme la modulazione ASK con la modulazione PSK. Si ottiene quindi una modulazione più efficiente rappresentando i bit con la variazione in contemporanea dell'ampiezza e della fase. Questa modulazione determina quindi una **grande velocità di trasmissione**.

I Mezzi Trasmissivi

È possibile classificare i mezzi fisici di trasmissione dei segnali in due categorie

- **Mezzi guidati:** elettrici, ottici
- **Mezzi non guidati:** onde radio, laser via etere

I mezzi trasmissivi elettrici rappresentano ancora oggi il mezzo più diffuso nell'ambito delle reti locali di edifici.

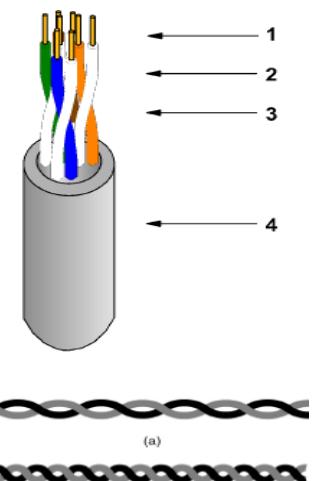
Dovendo trasportare il **segnale** in forma di **energia elettrica**, è necessario che le caratteristiche elettriche del mezzo siano tali da rendere massima la trasmissione dell'energia da un estremo all'altro e minima la dissipazione in altre forme (ad esempio calore, irradiazione elettromagnetica).

Doppino:

È il mezzo trasmissivo classico della telefonia e consiste in due fili di rame ricoperti da una guaina isolante e ritorti detti comunemente "coppia".

Nei doppini si usa una tecnica di **trasmissione bilanciata**. La binatura all'interno dei doppini serve a ridurre i disturbi elettromagnetici. Normalmente si utilizzano cavi con più coppie ed è allora necessario adottare passi di binatura differenti da coppia a coppia per ridurre la diafonia tra le coppie.

Infatti, se i passi di binatura fossero uguali, ogni conduttore di una coppia si troverebbe sistematicamente affiancato, ad ogni circuito, con uno dei due conduttori dell'altra coppia.



La Diafonia

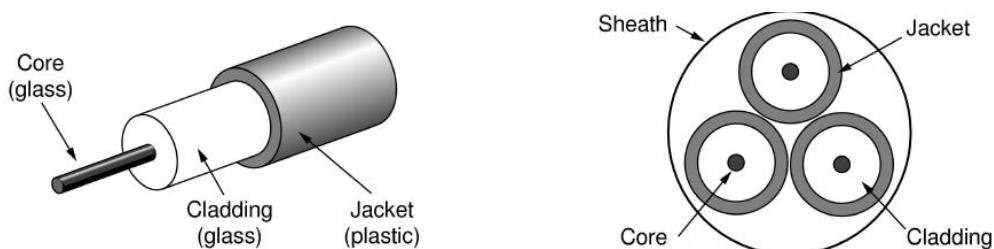
La **diafonia** è un fenomeno di accoppiamento elettrico tra mezzi trasmissivi vicini non isolati adeguatamente. Il segnale trasmesso su un cavo genera per induzione un segnale corrispondente nel cavo vicino, che si sovrappone al segnale trasmesso in quest'ultimo.

Si può verificare anche nella trasmissione con **mezzi non guidati**, quando un segnale emesso da una antenna si disperde durante la propagazione nell'aria, la parte dispersa può giungere in prossimità di un'altra antenna.

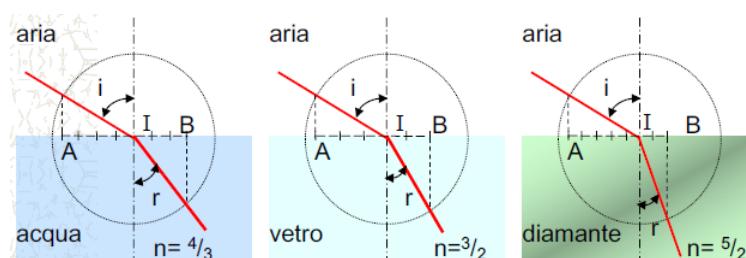
Fibre ottiche:

La fibra ottica sfrutta la proprietà della luce di non rifrangersi se l'angolo di incidenza è inferiore ad un certo limite. È rivestito con uno strato conduttivo e al centro il filo è costituito da una specie di vetro molto trasparente.

Le fibre sono normalmente raggruppate insieme intorno ad un filo di metallo che facilita la posa del cavo.

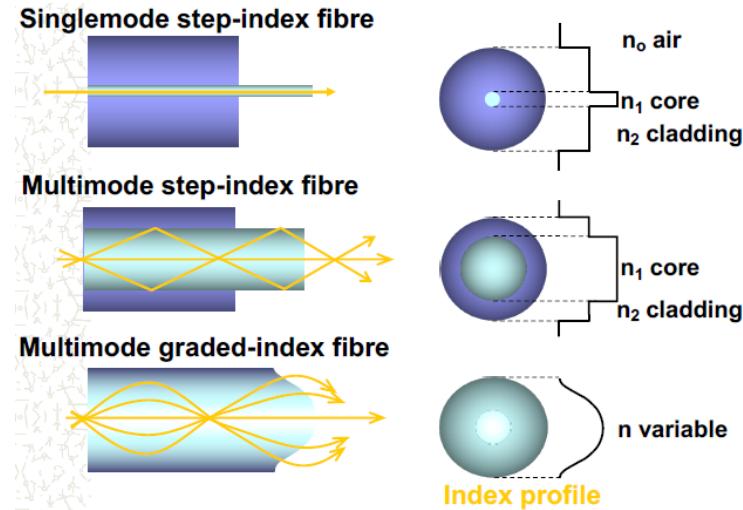


Si definisce **rifrazione**: "il fenomeno per cui un raggio luminoso passando da un mezzo trasparente ad un altro, anch'esso trasparente, cambia direzione nel punto in cui attraversa la superficie di separazione dei due mezzi".



Le reti in fibre possono essere strutturate ad anello o a stella passiva, ma il collegamento più usato è quello punto-a-punto unidirezionale. Ci sono due tipi di fibre ottiche:

- **Fibra multimodale:** È una fibra il cui nucleo è abbastanza ampio da permettere diversi angoli di rimbalzo della luce trasmessa.
- **Fibra monomodale:** È una fibra il cui nucleo permette il passaggio di poche lunghezze d'onda. Questo fa comportare la fibra come una semplice guida d'onda.



Parte dell'energia luminosa che si propaga lungo la fibra viene assorbita dal materiale o si diffonde in esso, costituendo quindi una perdita ai fini del segnale trasmesso. Il rapporto tra la potenza ottica trasmessa e quella ricevuta, dopo aver percorso una lunghezza di fibra di riferimento, definisce l'attenuazione della fibra stessa, in funzione della lunghezza d'onda e del tipo di fibra.

Minore è l'attenuazione maggiore la distanza utile per la trasmissione.

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.

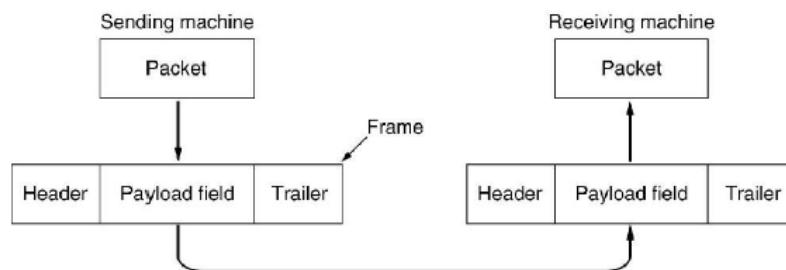
Il compito del data link layer è quindi quello di **organizzare** il trasferimento dei dati tra **due apparati adiacenti**, e di fornire una interfaccia definita per consentire allo strato di rete di **accedere ai servizi** offerti.

Per realizzare le sue funzioni il data link layer:

- Riceve 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:

- Riceve 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).

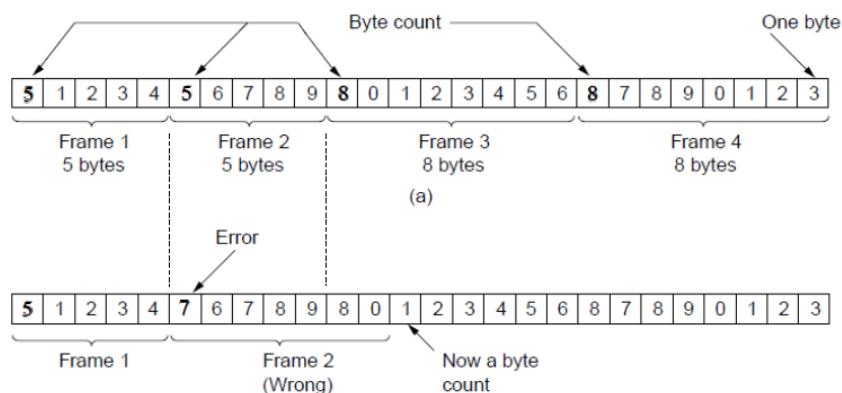
Al fine di fornire servizi al **livello di rete**, il **livello data link** deve usufruire dei servizi forniti dal **livello fisico**.

Lo strato fisico non può **garantire** il trasferimento **privò di errori**, che dovranno essere gestiti dal **DLL** (library 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:

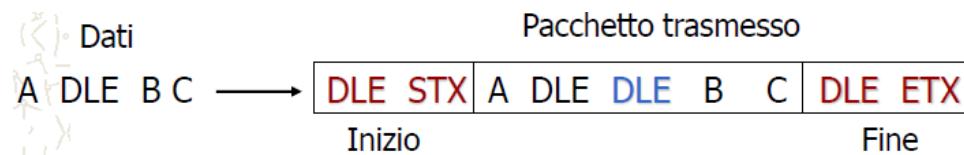
Un campo dell'intestazione indica il numero di caratteri nel pacchetto. Se si perde il sincronismo non si riesce a trovare l'inizio di un pacchetto successivo.



2. 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**.

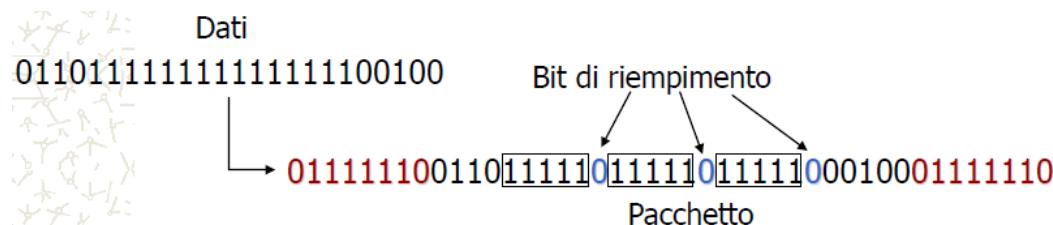
I dati nel pacchetto non possono contenere queste due sequenze. In trasmissione si duplica ogni **DLE** nei dati che poi si elimina in ricezione. Un **STX** o **ETX** preceduto da **due DLE** è un dato del pacchetto.



3. 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.

2. 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).

7 bit di dati	Byte con bit di parità	
	Bit di parità pari	Bit di parità dispari
1101001	01101001	11101001
1111111	11111111	01111111

3. 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

Start flag	type	seq	ack	Pacchetto (livello rete)	Check sum	End flag
------------	------	-----	-----	--------------------------	-----------	----------

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 **rtrasmette** 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.

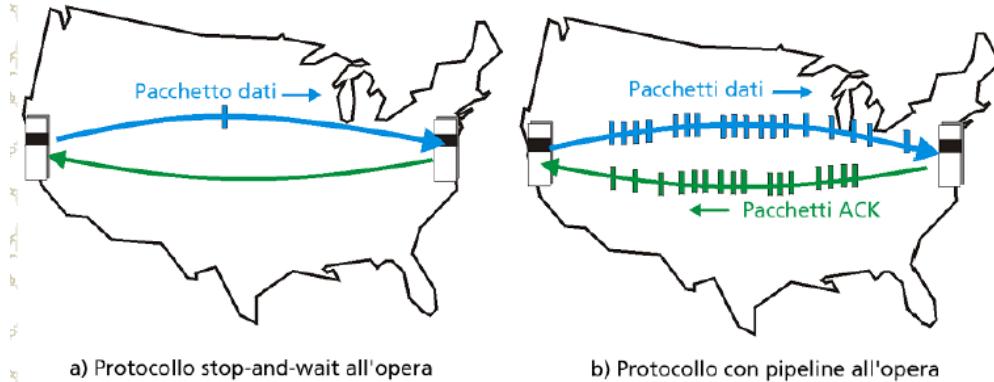
Il frame di riscontro si indica generalmente con il termine **ACK (ACKnowledge)** o **RR (Receiver Ready)**. Se i pacchetti contenenti ACK/NAK si danneggiano, bisogna fare il checksum a ACK/NAK.

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 necessita di maggiori risorse di **buffer**.

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

2. SELECTIVE REJECT (Ripetizione selettiva)

Il ricevente invia riscontri specifici per tutti i pacchetti ricevuti correttamente

