

# **COMPOSITION**

**TELECOMUNICAZIONI**

**ANNO 2020 - 2021**

**100 sheets • 200 pages  
9<sup>3</sup>/<sub>4</sub> x 7<sup>1</sup>/<sub>2</sub> in/24.7 x 19.0 cm  
wide ruled**

24/02/22

## Sviluppo in Serie di Fourier (Funzioni Periodiche Tempo Continuo) (Proprietà: pag. 13)

↓

FORMA ESPOENZIALE  $x(t) = x(t+T) \quad x \in C$

- $x(t) = \sum_{n=-\infty}^{\infty} c_n e^{j n \omega_0 t}$

- $f_0 = \frac{1}{T} \rightarrow$  FREQUENZA (DEI FASORI)  
FONDAMENTALE (IN Hz)

- $\omega_0 = 2\pi f_0 \rightarrow$  PULSAZIONE FONDAMENTALE

- $c_n = \frac{1}{T} \int_{-T/2}^{+T/2} x(t) e^{-jn\omega_0 t} dt$

### Tipi di Convergenza

→ PUNTUALE

$$\lim_{N \rightarrow \infty} \left\{ x(t) - \sum_{n=-N}^N c_n e^{j n \omega_0 t} \right\} = 0$$

→ MEDIA QUADRATICA

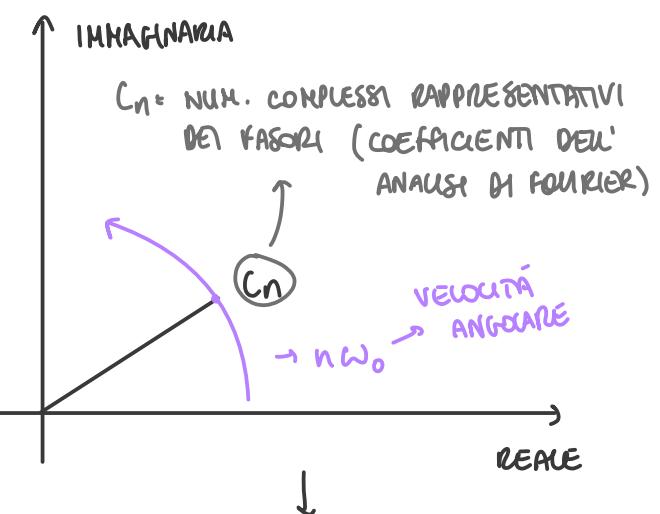
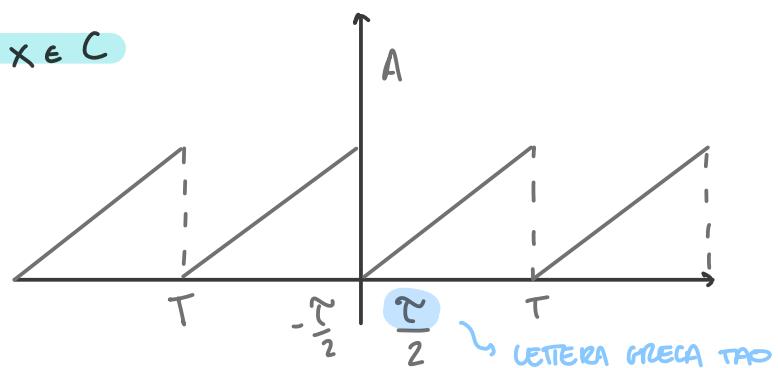
$$\lim_{N \rightarrow \infty} \int_0^T \left| x(t) - \sum_{n=-N}^N c_n e^{j n \omega_0 t} \right|^2 dt = 0$$

POICHÉ I COEFFICIENTI DEL SEGNALE SONO IN GENERALE COMPLESSI, SE NE GRAFICA DI SOTTO IL MODO NUOVO, DETTO SPETTO DI AMPIEZZA, E L'ARGOMENTO, DETTO SPETTO DI FASE.

QUESTI SONO Detti SPETRI 'A RIGHE' IN QUANTO PER CONVENZIONE SI USA CONGIUNGERE IL PUNTO RAPPRESENTANTE IL VALORE CON L'ASSE DELLE FREQUENZE.

DATO CHE LE PULSAZIONI POSSONO ESSERE SPA POSITIVE ( $n > 0$ ) CHE NEGATIVE ( $n < 0$ ) LO SPETTO SI DICE DI TIPO 'BILATERO'.

PULSAZIONE POSITIVA: ROTAZIONE ANTORARIO DEL FASORE  
PULSAZIONE NEGATIVA: ROTAZIONE ORARIA DEL FASORE



PIÙ IL SEGNALE SI RIPETE IN PRETTO NEL TEMPO PIÙ I FASORI DEVONO GIRARE VELOCEMENTE

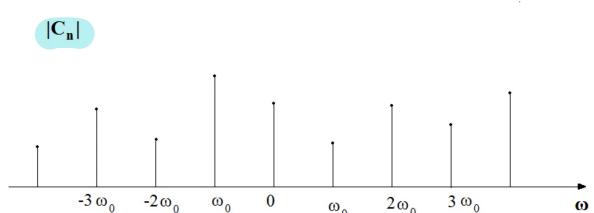


Fig.2 Spettro di ampiezza bilatero (a righe).

$\arg\{C_n\}$

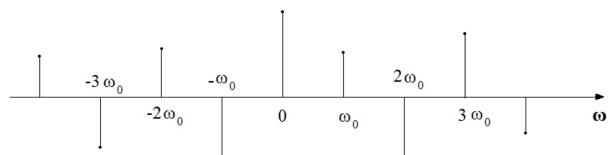


Fig.3 Spettro di fase bilatero (a righe).

## RAPPRESENTAZIONE MONOLATERALE (SEGNALI REAUX)

↓

SE  $x(+)$  È REALE IL COEFFICIENTE  $c_0$  È REALE IN QUANTO RAPPRESENTA IL VALORE MEDIO DELLA FUNZIONE; I COEFFICIENTI POSITIVI E NEGATIVI SONO INVECE LEGATI DA SIMMETRIA HERMITIANA.

$c_{-n} = c_n^*$  → IN PARTICOLARE PER UN SEGNALE REALE LO SPETTO DI AMPIEZZA RISULTA SIMMETRICO RISPETTO ALL'ORIGINE, E QUELLO DI FASE ANTISIMMETRICO.

$$|c_n| = |c_{-n}|$$

$$\arg\{c_n\} = -\arg\{c_{-n}\}$$

SFRUTTANDO LA SIMMETRIA HERMITIANA È POSSIBILE MANIPOLARE LA FUNZIONE DI SINTESI IN MODO DA UTILIZZARE SOLO LE FREQUENZE POSITIVE.

$$\begin{aligned} x(+)&= \sum_{n=-\infty}^{-1} c_n e^{jn\omega_0 t} + c_0 + \sum_{n=1}^{\infty} c_n e^{jn\omega_0 t} = \\ &= \sum_{n=1}^{\infty} c_{-n} e^{-jn\omega_0 t} + c_0 + \sum_{n=1}^{\infty} c_n e^{jn\omega_0 t} = \\ &= c_0 + \sum_{n=1}^{\infty} 2 \operatorname{Re}\{c_n e^{jn\omega_0 t}\} = c_0 + \sum_{n=1}^{\infty} \operatorname{Re}\{2c_n e^{jn\omega_0 t}\} \end{aligned}$$

SPEZZANDO LA SOMMATORIA IN TRE TERMINI, CAMBIANDO  $-n$  CON  $n$  NELLA PRIMA IN MODO CHE GLI ESTREMI COINCIDANO CON LA SECONDA. RACCOGLIAMO RICORDANDO CHE NUM. COMPL. + NUM. COMPL. = 2 RE

↓  
2 VOLTE LA LORO PARTE REALE

DEFINENDO GLI SPETTRI DI AMPIEZZA E FASE MONOLATERALI:

$$\left\{ \begin{array}{l} A_0 = c_0 \\ A_n = 2|c_n| \quad n > 0 \\ \text{REAL} \end{array} \right. \quad \varphi_n = -\arg\{c_n\} \quad n > 0 \quad A_n \neq 0$$

ED OPERANNO LE DUE SEGUENTI SOSTITUZIONI:

$$c_0 = A_0 \quad 2c_n = A_n e^{-j\varphi_n} \quad n > 0$$

SI OTTIENE LA FORMA IN SOLI COSENI DELLO SVILUPPO DI FOURIER: (SECONDA FORMA)

$$\boxed{x(+)=A_0 + \sum_{n=1}^{\infty} A_n \cos(n\omega_0 t - \varphi_n)}$$

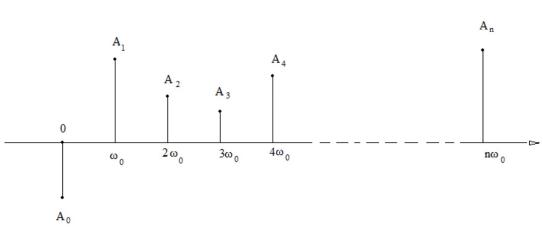


Fig. 4 Spettro di ampiezza monolaterale di un segnale periodico.

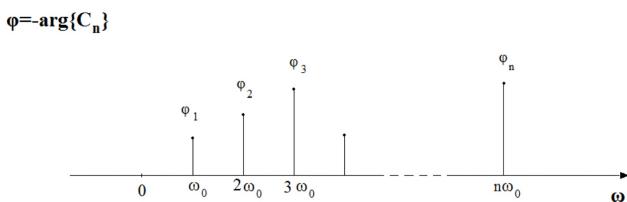
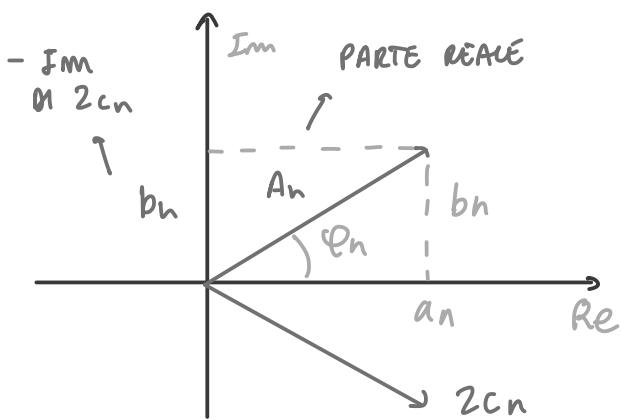


Fig. 5 Spettro di fase monolaterale di un segnale periodico.



→ RELAZIONE TRA I COEFFICIENTI DELLE TRE FORME DELLO SVILUPPO IN SERIE DI FOURIER

↓

1<sup>a</sup> FORMA: PIÙ GENERALE, PIÙ COMPLESSA

2<sup>a</sup> FORMA: SEGNALE PERIODICO = SOMMA DI SINUSOIDI, PIÙ SEMPLICE E PRATICA

3<sup>a</sup> FORMA: PIÙ IMMEDIATA

### TERZA FORMA DELLA SERIE DI FOURIER

$$\rightarrow x(t) = \frac{1}{2} a_0 + \sum_{n=1}^{\infty} a_n \cos(n\omega_0 t) + \sum_{n=1}^{\infty} b_n \sin(n\omega_0 t)$$

$$a_n = \begin{cases} 2c_0 & n=0 \\ \operatorname{Re}\{2c_n\} & n>0 \end{cases}$$

] COEFFICIENTI MONOLATERALI

$$b_n = -\operatorname{Im}\{2c_n\} \quad n>0$$

↓

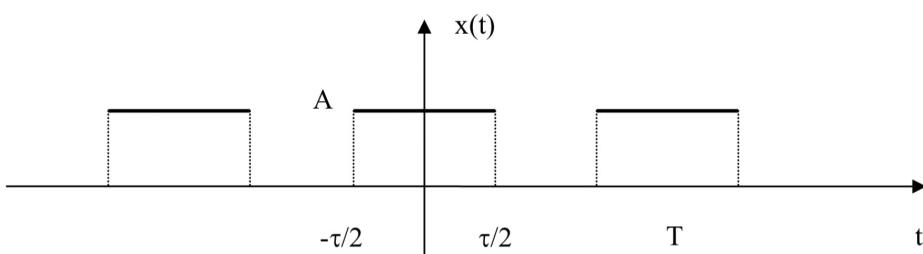
I COEFFICIENTI POSSONO ANCHE ESSERE RICAVATI DIRETTAMENTE

$$a_n = \frac{2}{T} \int_{-T/2}^{+T/2} x(t) \cos(n\omega_0 t) dt \quad n \geq 0$$

$$b_n = \frac{2}{T} \int_{-T/2}^{+T/2} x(t) \sin(n\omega_0 t) dt \quad n > 0$$

### ESERCIZIO 1 (PAGINA 25)

CALCOLARE GLI SPETTRI DI AMPIEZZA E FASE MONOLATERALI DI UNA SUCCESSIONE PERIODICA DI IMPULSI, AVENTI AMPIEZZA A, DURATA  $\tau$  E PERIODO T.



SI TRATTA DI UNA FUNZIONE REALE DISCRETA NEI VALORI E TEMPO-CONTINUA.

Fig.12 Successione periodica di impulsi rettangolari.

PER DETERMINARE GLI SPETTRI MONOCATERALI, CIOÉ  $A_n$  E  $\varphi_n$  CONVIENE PRIMA CALCOLARE I COEFFICIENTI  $c_n$  DELLA FORMA ESPONENZIALE.

$$\begin{aligned}
 c_n &= \frac{1}{T} \int_{-T/2}^{+T/2} x(t) e^{-jn\omega_0 t} dt \xrightarrow{\text{AMPIEZZA}} = \frac{A}{T} \int_{-\tau/2}^{+\tau/2} e^{-jn\omega_0 t} dt \xrightarrow{\text{DURATA}} = \frac{A}{T} \left[ \frac{e^{-jn\omega_0 \tau/2}}{-jn\omega_0} \right]_{-\tau/2}^{+\tau/2} = \\
 &= \frac{A}{T} \frac{e^{-jn\omega_0 \tau/2} - e^{jn\omega_0 \tau/2}}{-jn\omega_0} = \frac{A}{T} \frac{-2j \sin(n\omega_0 \frac{\tau}{2})}{-jn\omega_0} = \frac{A\tau}{T} \frac{\sin(n\omega_0 \frac{\tau}{2})}{n\omega_0 \frac{\tau}{2}} = \\
 &= \frac{I}{T} \operatorname{sinc}\left(\frac{n\pi}{T}\right) \rightarrow \text{FUNZIONE INTEGRAL SENO} \quad \operatorname{sinc} z = \frac{\sin \pi z}{\pi z}
 \end{aligned}$$

I COEFFICIENTI  $c_n$  SONO IN QUESTO CASO REALI E NON COMPLESSI, QUINDI UTILIZZIAMO LA 2<sup>a</sup> FORMA. GLI SPETTRI DI AMPIEZZA SONO QUINDI ESPRESI DA:

$$\begin{aligned}
 A_0 &= c_0 = \frac{I}{T} \\
 A_n &= |2c_n| = \frac{2I}{T} \left| \frac{\sin(n\omega_0 \frac{\tau}{2})}{n\omega_0 \frac{\tau}{2}} \right| = \frac{2I}{T} \left| \operatorname{sinc}(n \frac{\pi}{T}) \right|
 \end{aligned}$$

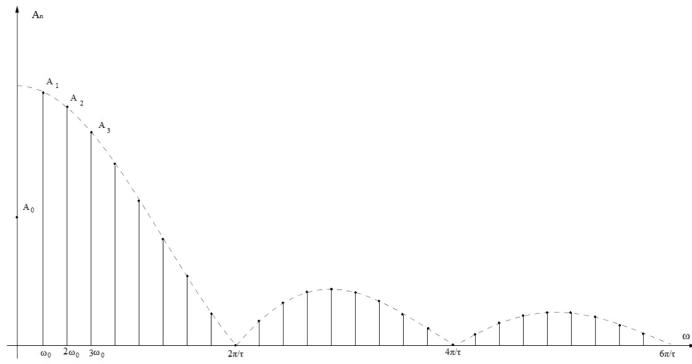


Fig. 13 Spettro di ampiezza a righe di una successione periodica di impulsi.

E QUELLI DI FASE DA:

$$\varphi_n = -\arg\{c_n\} = \begin{cases} 0 & c_n > 0 \\ \pi & c_n < 0 \end{cases} \rightarrow 2^a \text{ FORMA} \rightarrow \text{COEFFICIENTI REALI!!}$$

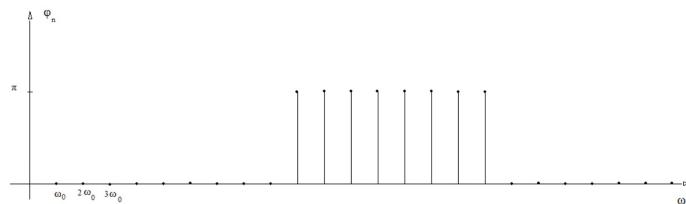


Fig. 14 Spettro di fase a righe di una successione periodica di impulsi.

SI OSSERVA CHE LA POSIZIONE DELLE RIGHE È SEMPRE A MULTIPLI DELLA PULSAZIONE FONDAMENTALE, QUINDI LO SPAZIO FRA DI ESSE È INVERSALEMENTE PROPORZIONALE AL PERIODO T. AL CONTRARIO IL LORO VALORE DIPENDE DALL'ANDAMENTO DEL SEGNALE.

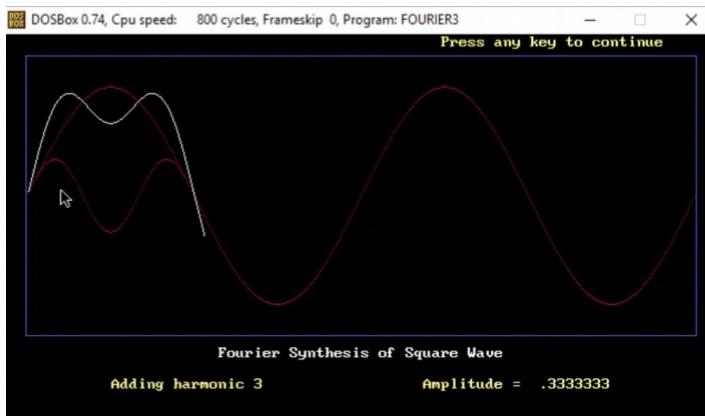
24/02/22

NEL CASO PARTICOLARE (OVIAMENTE NON IN GENERALE), LE RIGHE DELLO SPETTRO DI AMPIEZZA POSSONO ESSERE OTTENUTE "CAMPIONANDO" A MULITPLI DI  $\omega_0$  LA SEGUENTE FUNZIONE AUXILIARIA

$$Aux(\omega) = \frac{2I}{T} \left| \frac{\sin\left(\frac{\omega\tau}{2}\right)}{\frac{\omega\tau}{2}} \right| \longrightarrow A_0 = \frac{Aux(0)}{2} \quad A_n = Aux(n\omega_0)$$

## FOURIER3 - MSDOS: VISUALIZZAZIONE GRAFICA DELL'ANALISI DI FOURIER

- SQUARE WAVE (ONDA QUADRA)



→ IN ROSSO SONO RAPPRESENTATE LE ARMONICHE SINUSOIDALI, LA PRIMA È LA FONDAMENTALE.

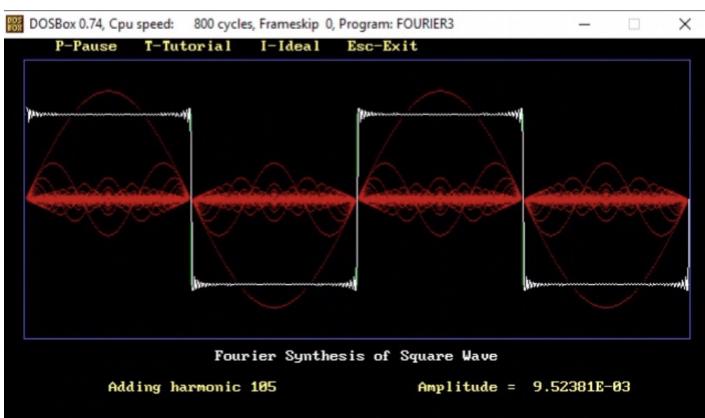
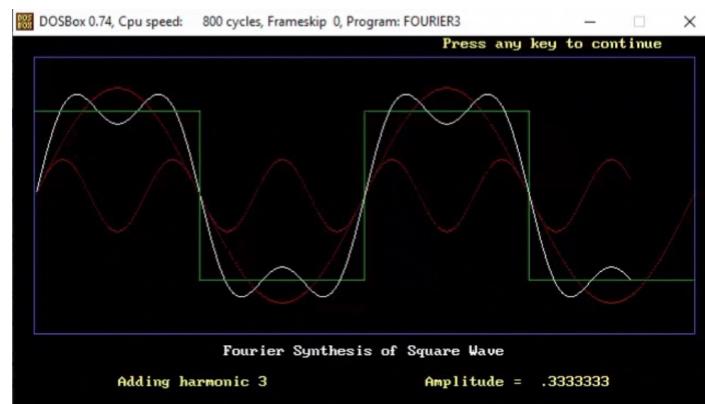
→ QUI SI AGGIUNGE LA TERZA ARMONICA, DOVE INFATTI CI SONO TRE PERIODI.



→ IN VERDE È RAPPRESENTATA LA  $x(t)$  COME SOMMA DI SINUSOIDI, L'ONDA CHE SI VUOLE OTTENERE.

$$x(t) = A_j \sum_{n=1}^a A_n c(n \omega_0 t - \varphi_n)$$

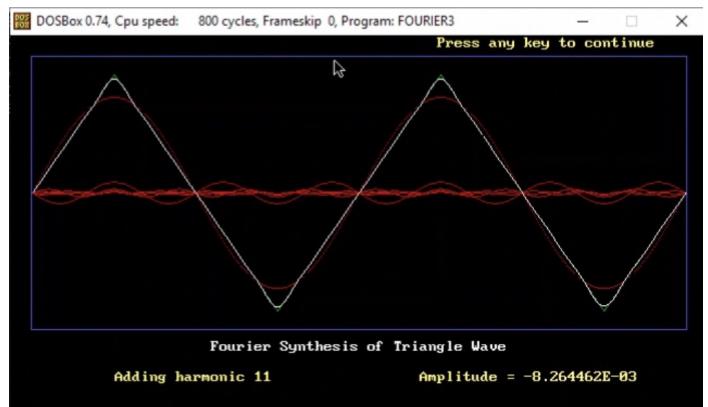
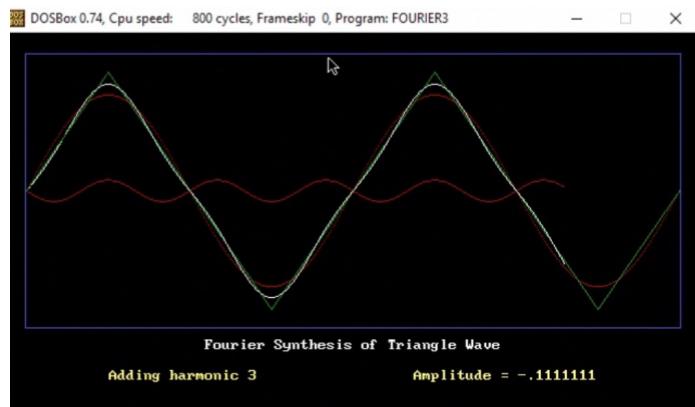
(FORMULA DELLA SQUARE WAVE)



→ IN BIANCO È RAPPRESENTATA LA SOMMA DELLE ARMONICHE (ROSSO)

PIÙ ARMONICHE SI AGGIUNGONO, PIÙ IL GRAFICO BIANCO ASSOMIGUA A QUELLO VERDE.

- TRIANGLE WAVE → CONTINUA, MA LA SUA DERIVATA PRIMA È DISCONTINUA.



↑ GIÀ CON POCHE ARMONICHE SI AVVICINA MOLTO ALLA SUA  $x(t)$

## RELAZIONE TRA MUSICA E ANALISI DI FOURIER

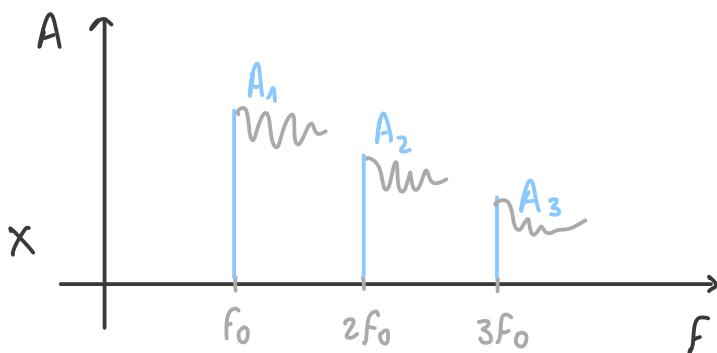
L'analisi di Fourier è indipendente dalla natura dei segnali a cui è applicata.

Non tutti gli strumenti musicali emettono dei suoni quasi-periodici (si intendono quasi-periodici i suoni che sono approssimativamente periodici ovviamente a meno del transitorio iniziale e finale, non potendo certo durare un suono all'infinito), ad esempio le maracas e le nacchere non li emettono.

Sono tuttavia generalmente quasi-periodici i suoni emessi dagli strumenti a corda, dai legni e dagli ottoni.

Consideriamo le note emesse da un pianoforte. Alla pressione di un tasto un martelletto percuote un gruppo di corde metalliche tese, facendole vibrare più o meno intensamente a seconda dell'intensità con la quale il pianista ha pigiato il tasto.

La vibrazione è libera (e quindi quasi-periodica) finché il tasto rimane pigiato, quindi viene smorzata al rialzarsi del tasto (a meno che non si voglia che gli smorzatori rimangano alzati, nel qual caso si preme l'apposito pedale). Lo strumento dispone generalmente di 88 tasti (sette ottave e una terza minore), 52 bianchi e 36 neri. Il punto di riferimento centrale della tastiera è il tasto do, chiamato per questo "do centrale". Grazie ai suoi 88 tasti, il piano raggiunge un'estensione di sette ottave.



$$\begin{aligned} \text{LA} &= 220 \text{ Hz} && \text{FONDAMENTALE} \\ \text{LA} &\rightarrow f_0 = 440 \text{ Hz} \rightarrow 4^{\text{a}} \text{ OTTAVA} \\ \text{LA} &\rightarrow f'_0 = 880 \text{ Hz} \rightarrow 8^{\text{a}} \text{ OTTAVA} \end{aligned}$$

Il "volume" con la quale una nota è emessa non è altro che la sensazione umana (soggettiva) legata all'intensità (oggettiva) del suono.

L'"altezza" della nota è legata alla frequenza del segnale emesso, ovvero alla frequenza della prima armonica, o armonica fondamentale. Ad ogni nota corrisponde quindi una ben specifica frequenza della fondamentale.

## RAPPRESENTAZIONE ORTONORMALE DEI SEGNALI (ARGOMENTO AVANZATO, FACOLTATIVO)

Esiste un'importante analogia fra segnali e vettori.

Come un vettore rappresentato in uno spazio a  $n$  dimensioni può essere scomposto su una base ortonormale  $n$ -dimensionale, cioè espresso come combinazione lineare dei suoi versori, così è possibile scomporre un segnale su una base ortonormale, in cui gli elementi della base non sono più versori, ma funzioni ortonormali. Sia il vettore che il segnale, può quindi essere definito dalla  $n$ -pla dei coefficienti della combinazione lineare.

Si definisce quindi **prodotto interno** fra due segnali  $x(t)$  e  $y(t)$ , (generalmente complessi), definiti sull'intervallo  $T$ , il seguente:

$$\langle x, y \rangle_T = \int_T x(t) y(t)^* dt$$

Così come due vettori sono **ortogonali** se e solo se il loro prodotto interno è nullo, così lo sono due segnali:

$$x \perp y \Leftrightarrow \langle x, y \rangle_T = 0$$

Così come per un vettore, la **norma** è data dalla **radice quadrata dell'energia**.

$$\|x\| = \sqrt{\langle x, x \rangle_T}$$

Così come due vettori sono **ortonormali** quando oltre ad essere ortogonali hanno norma unitaria, così è per i segnali.

$$x, y \text{ ORTONORMALI SE } \begin{cases} \langle x, y \rangle_T = 0 \\ \|x\| = \|y\| = 1 \end{cases}$$

Così come per i vettori esistono basi ortonormali, composte da versori, analogamente è possibile definire una **base ortonormale** per i segnali, composta da funzioni ortonormali, ad esempio

$$\{\psi_m(t)\} \text{ definite sull'intervallo } T.$$

Così come per i vettori un segnale è rappresentabile su una base ortonormale se è possibile scriverlo come **combinazione lineare degli n-elementi della base** (possono essere anche infiniti)

$$x(t) = x_1 \psi_1(t) + x_2 \psi_2(t) + \dots + x_n \psi_n(t)$$

Data una base ortonormale in grado di rappresentarlo, esiste una **corrispondenza biunivoca fra segnale ed n-pla dei coefficienti**.

$$x(t) \Leftrightarrow (x_1, x_2, \dots, x_n) \quad x_u = \langle x, \psi_u \rangle_T$$

Una base si dice completa per un dato insieme di segnali se è in grado di rappresentarli tutti.

## BASI ORTONORMALI E SVILUPPO IN SERIE DI FOURIER (ARGOMENTO AVANZATO, FACOLTATIVO)

Si consideri la serie (infiniti termini) delle funzioni esponenziali che compaiono nello sviluppo in serie di Fourier, limitate su un intervallo  $T$ :

$$\{\varphi_m(t)\} \quad \varphi_m(t) = e^{j m \omega_0 t}$$

Esse sono ortogonali, ma non ortonormali. Si consideri infatti il loro prodotto interno:

$$\langle \varphi_m, \varphi_n \rangle = \int_T e^{jm\omega_0 t} e^{-jn\omega_0 t} dt = \int e^{j(m-n)\omega_0 t} dt = \begin{cases} 0 & m \neq n \\ T & m = n \end{cases}$$

Dividendo le funzioni esponenziali per la loro norma è immediato ottenere la seguente serie di funzioni ortonormali:

$$\psi_m(t) = \frac{1}{\sqrt{T}} \varphi_m(t) \quad \text{PER ESSERE INFATI: } \langle \psi_m, \psi_n \rangle_T(t) = \begin{cases} 0 & m \neq n \\ 1 & m = n \end{cases}$$

Lo sviluppo in serie di Fourier rappresenta la scomposizione di un segnale su una base ortogonale dove le funzioni hanno energia T.

È sufficiente normalizzare le funzioni per ottenere una scomposizione su base ortonormale, i cui coefficienti differiscono da quelli dello sviluppo solo per essere moltiplicati dal fattore di normalizzazione.

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t} \xrightarrow{\text{MOLTIPLICO PER } \sqrt{T}} c'_n = c_n \sqrt{T} = \sum_{n=-\infty}^{\infty} c'_n \psi_n(t)$$

I coefficienti sono per definizione ottenibili come prodotto interno della funzione periodica con gli elementi della base ortonormale.

$$c'_n = \langle x, \psi_n \rangle_T = \frac{1}{\sqrt{T}} \int_T x(t) e^{-jn\omega_0 t} dt$$

Ne consegue che le formule dello sviluppo in serie, a meno del fattore di normalizzazione, non sono altro che un caso particolare di scomposizione su base ortonormale.

La base ortonormale individuata è completa per tutti i segnali periodici che soddisfano le condizioni di esistenza dello sviluppo in serie.

## TRASFORMATA DI FOURIER

La trattazione vista in precedenza per i segnali periodici può essere estesa, con le opportune variazioni, ai segnali aperiodici. Consideriamo una funzione  $x(t)$ , in generale complessa, aperiodica, definita su tutto l'asse  $t$ . Sotto condizioni analoghe a quelle viste in precedenza per le serie di Fourier, esiste la trasformata di Fourier:

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt$$

La cui formula di antitrasformazione è data dalla seguente:

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega$$

Analogamente a quanto visto per le serie, la formula di sintesi esprime il segnale  $x(t)$  come una somma di infiniti fasori, con pulsazioni (o frequenze) estese ad entrambi i semiassi. In questo caso però, si tratta di un'infinità non numerabile, e tutte le pulsazioni sono coinvolte, non solo quelle multiple di un determinato valore. Inoltre l'ampiezza di un fasore ad una determinata pulsazione è infinitesima.

Di conseguenza lo spettro di ampiezza bilatero, o meglio la densità spettrale di ampiezza,  $|X(\omega)|$  non è più a righe ma continuo.

Analogamente è continuo per lo spettro di fase bilatero, definito come  $\arg\{X(\omega)\}$

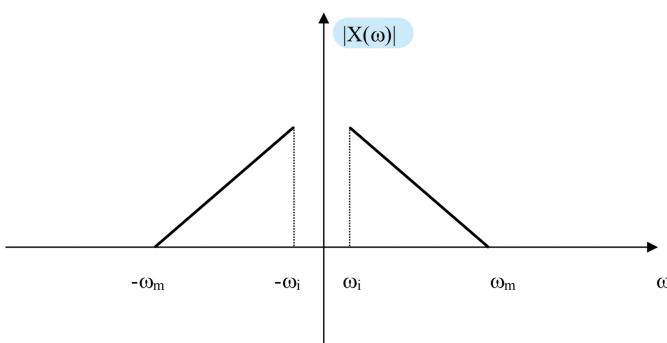


Fig.8 Spettro di ampiezza bilatero (modulo trasformata).

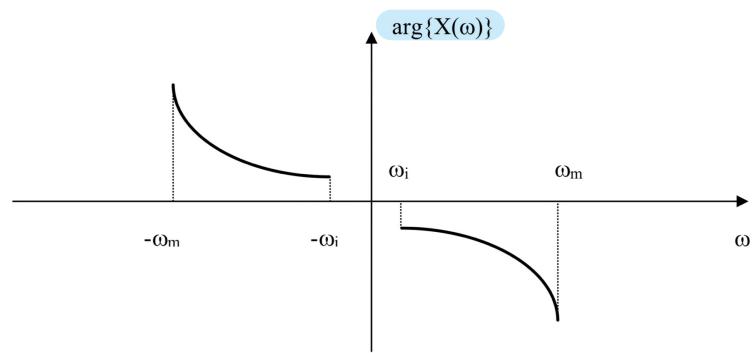


Fig.9 Spettro di fase bilatero (argomento trasformata).

## RAPPRESENTAZIONI MONOLATERE (SEGNALI REALI)

Nel caso di  $x(t)$  reale si possono dare delle espressioni alternative, monolatere, alla formula di sintesi.

Quando  $x(t)$  reale, vale la relazione (simmetria hermitiana):  $X(-\omega) = X^*(\omega)$

Procedendo in modo analogo a quanto fatto in precedenza per le serie è possibile arrivare alla seguente:

$$\begin{aligned} x(t) &= \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega = \frac{1}{2\pi} \int_0^{+\infty} 2\operatorname{Re}\left\{ |X(\omega)| e^{j\arg\{X(\omega)\}} e^{j\omega t} \right\} d\omega = \\ &= \int_0^{+\infty} \frac{|X(\omega)|}{\pi} \cos(\arg\{X(\omega)\} + \omega t) d\omega \end{aligned}$$

28/02/22

Da cui definito lo spettro (densità spettrale) di ampiezza monolatero come

$$V(\omega) = \frac{|X(\omega)|}{\pi} \quad \omega > 0$$

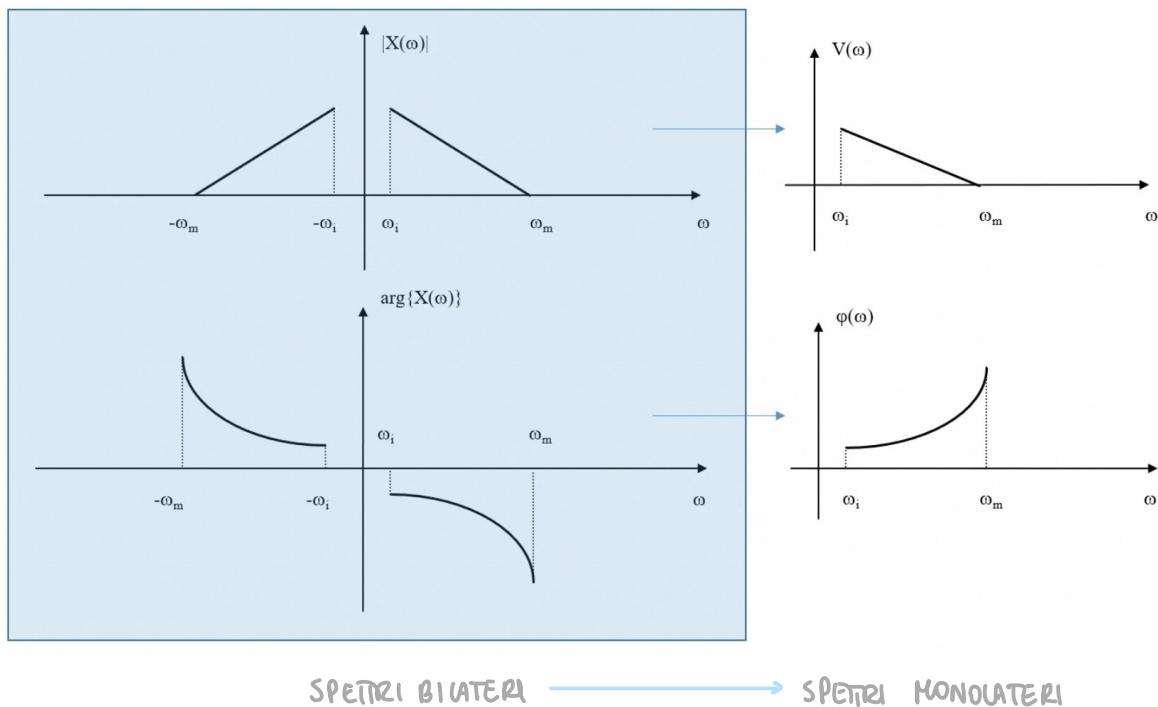
E quello monolatero di fase come

$$\varphi(\omega) = -\arg \{X(\omega)\} \quad \omega \geq 0, \quad V(\omega) \neq 0$$

Otteniamo infine l'espressione sotto, detta "integrale di Fourier":

$$x(t) = \int_0^{+\infty} V(\omega) \cos [\omega t - \varphi(\omega)] d\omega$$

L'integrale di Fourier è l'analogo dello sviluppo in serie in soli coseni; esso rappresenta  $x(t)$  come "somma" di infiniti termini sinusoidali, il generico dei quali ha pulsazione  $\omega$ , ampiezza (infinitesima)  $V(\omega) d\omega$  e fase  $\varphi(\omega)$ .



## BANDE

Essa è l'intervallo di pulsazioni (o frequenze) per le quali è diverso da zero lo spettro di ampiezza, eventualmente entro un determinato livello di approssimazione.

Per comodità, nel caso di segnali reali si considera sempre solo il semiasse positivo anche quando si utilizza uno spettro bilatero.

In casi particolari la banda può essere formata da più intervalli disgiunti, ma di norma lo spettro è su di un unico intervallo (almeno approssimativamente).

Quando invece la banda comprende lo 0 oppure è significativamente prossima ad esso, si parla di funzione passa-basso.

Quando invece la banda è "piccola" rispetto alla pulsazione centrale si parla di funzione passa-banda.

$$B_\omega = \omega_m - \omega_i$$

$$\omega_c = \frac{\omega_m + \omega_i}{2} \rightarrow \frac{B_\omega}{\omega_c} \ll 1$$

I concetti di banda appena visti si applicano anche ai segnali periodici.

Alcuni esempi di bande di segnali analogici (tutti passa-banda):

- Telefonica netta (standard europeo): 300-3400 Hz
- Audio AM (segnale modulante): 50-4500.
- Modulazione FM (segnale modulante): 30-15000 Hz.
- CD: circa 20-20000 Hz.

### PROPRIETÀ DELLA TRASFORMATA DI FOURIER

- CONIUGAZIONE:  $F[x^*(+)] = X^*(-\omega)$
- TRASLATORIALE TEMPORALE:  $F[x(t - t_0)] = X(\omega) e^{-j\omega t_0}$

DIMOSTRAZIONE  $\rightarrow F[x(t - t_0)] = \int_{-\infty}^{+\infty} x(t - t_0) e^{-j\omega t} dt = \int_{-\infty - t_0}^{+\infty - t_0} x(\xi) e^{-j\omega(\xi + t_0)} d\xi =$

$$= e^{-j\omega t_0} \int_{-\infty}^{+\infty} x(\xi) e^{-j\omega(\xi)} d\xi = X(\omega) e^{-j\omega t_0}$$

EPSILON  
 SOSTITUISCO  $(t - t_0) = \xi$

- DERIVAZIONE  $\rightarrow$  ESPRIMIAMO  $x(+)$  COME ANTITRASFORMATA DELLA SUA TRASFORMATA.

$$F[\dot{x}(+)] = j\omega X(\omega)$$

DIMOSTRAZIONE  $\rightarrow \dot{x}(+) = \frac{d}{dt} \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) \frac{d}{dt} e^{j\omega t} d\omega =$

$$= \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) j\omega e^{j\omega t} d\omega = F^{-1}[j\omega X(\omega)]$$

DERIVO ULLI  
 L'ANTITRASFORMATA

- INTEGRAZIONE:  $F \left[ \int_{-\infty}^t x(\xi) d\xi \right] = \frac{X(\omega)}{j\omega} \quad \text{se } X(0) = \left[ \int_{-\infty}^{+\infty} x(t) dt \right] = 0$

Le due proprietà precedenti permettono di trasformare un'equazione integro-differenziale (ad esempio quella di un circuito RLC serie) in una equazione algebrica, molto più facile da risolvere.

- CONVOLUZIONE: DATE DUE FUNZIONI  $x(t)$  E  $y(t)$  AVVENTI COME TRASFORMATE DI FOURIER  $X(\omega)$  E  $Y(\omega)$ , IL PRODOTTO DI CONVOLUZIONE FRA  $x(t)$  E  $y(t)$  (O PRODOTTO INTEGRALE) DEFINITO COME

$$x(t)^* y(t) = \int_{-\infty}^{+\infty} x(\tau) y(t - \tau) d\tau$$

HA COME TRASFORMATA IL PRODOTTO (SEMPLICE) DELLE TRASFORMATE  $X(\omega)$  E  $Y(\omega)$ .

$$\mathcal{F}[x(t)^* y(t)] = X(\omega) Y(\omega)$$

La proprietà permette di trasformare un prodotto di convoluzione, generalmente molto difficile da calcolare per via dell'integrale, in un prodotto semplice. E' fondamentale nello studio delle reti lineari.

DIMOSTRAZIONE → Si parte dalla definizione, si esprime quindi  $y(t)$  come antitrasformata della sua trasformata, si cambia l'ordine degli integrali ritrovando  $X(\omega)$ , quindi si nota che il risultato finale è la formula di antitrasformazione di  $X(\omega)Y(\omega)$ ; l'enunciato risulta quindi dimostrato per l'unicità della trasformata (corrispondenza biunivoca).

$$x(t)^* y(t) = \int_{-\infty}^{+\infty} x(\tau) y(t - \tau) d\tau = \int_{\tau} x(\tau) \frac{1}{2\pi} \int_{\omega} Y(\omega) e^{j\omega(t-\tau)} d\omega d\tau =$$

↓

$$y(t) = \mathcal{F}^{-1}[Y(\omega)]$$

CAMBIO L'ORDINE  
DEGLI INTEGRALI → =  $\frac{1}{2\pi} \int_{\omega} Y(\omega) \int_{\tau} x(\tau) e^{-j\omega\tau} d\tau e^{j\omega t} d\omega =$   
RITROVANDO  $X(\omega)$

$$= \frac{1}{2\pi} \int_{\omega} Y(\omega) X(\omega) e^{j\omega t} d\omega = \mathcal{F}^{-1}[Y(\omega) X(\omega)]$$

### Esercizio 2 - TRASFORMATA DI UN IMPULSO RETTANGOLARE (PAGINA 28)

Calcolare gli spettri di ampiezza e di fase di un impulso rettangolare di ampiezza A e durata  $\tau$ .

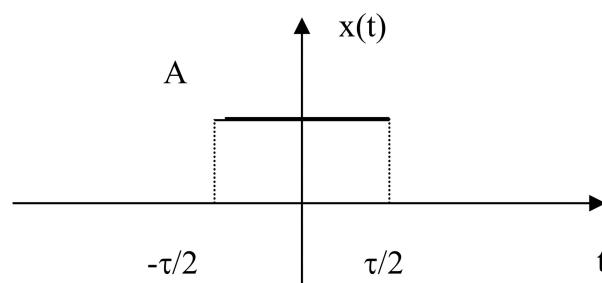


Fig.15 Impulso rettangolare.

28/02/22

Si osservi che si tratta ancora di una funzione reale pari, discreta nei valori e tempo-continua, ma di tipo aperiodico.

La sua trasformata secondo Fourier è espressa da:

$$\begin{aligned}
 X(\omega) &= \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt = A \int_{-\tau/2}^{+\tau/2} e^{-j\omega t} dt = A \left[ \frac{e^{-j\omega t}}{-j\omega} \right]_{-\tau/2}^{+\tau/2} = \\
 &= A \frac{e^{-j\omega\tau/2} - e^{j\omega\tau/2}}{-j\omega} = A \frac{-2j \sin(\omega\tau/2)}{-j\omega} = A\tau \frac{\sin(\omega\tau/2)}{\omega\tau/2} = \\
 &= I \frac{\sin(\omega\tau/2)}{\omega\tau/2} = I \frac{\sin(\omega\tau/2)}{\omega\tau/2} = I \operatorname{sinc}\left(\frac{\omega\tau}{2\pi}\right)
 \end{aligned}$$

IL SEGNALE VARIA IN  
MANIERA COSTANTE  
QUINDI POSSO PORTARE  
FUORI A

$\operatorname{sinc}z = \frac{\sin \pi z}{\pi z}$

Si noti che il risultato ottenuto è reale, come ci si doveva aspettare, visto che la funzione  $x(t)$  oltre ad essere reale è pari.

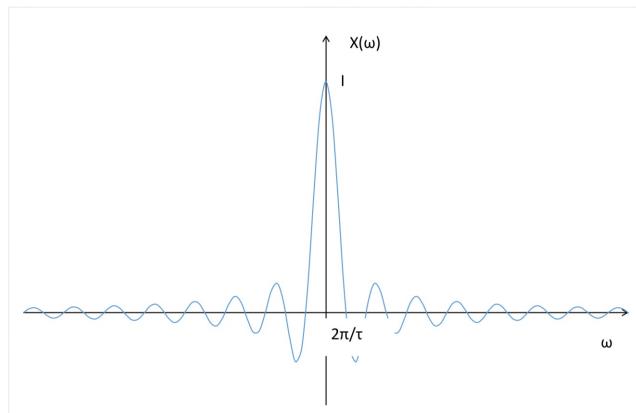
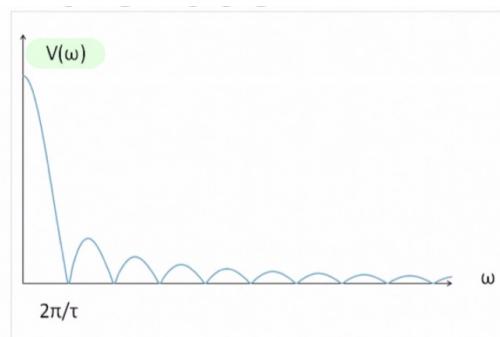


Fig.16 Trasformata di un impulso rettangolare

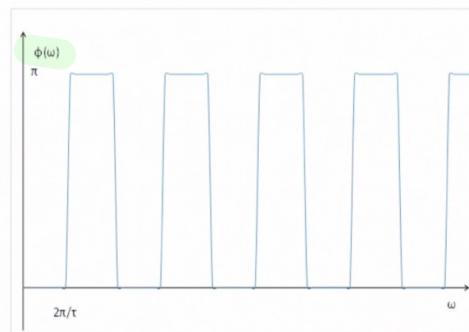
- IL GRAFICO È QUELLO DEL sinc (UNA SINUSOIDA SMORZATA)
- PIÙ  $\tau$  È PICCOLO PIÙ LA BANDA DEL GRAFICO (DISTANZA TRA I PERIODI) TENDE AD ALLARGARSI. SE L'IMPULSO ( $\tau$ ) È LARGO LA BANDA SI COMPRIME VERSO L'ORIGINE (SI RESTRINDE)

Ricordando le relazioni che definiscono gli spettri di ampiezza e di fase abbiamo:

$$\begin{aligned}
 V(\omega) &= \frac{|X(\omega)|}{\pi} = \frac{I}{\pi} \left| \frac{\sin(\omega\tau/2)}{\omega\tau/2} \right| = \\
 &= \frac{I}{\pi} |\operatorname{sinc}(\omega\tau/2\pi)| \quad \omega \geq 0
 \end{aligned}$$



$$\varphi(\omega) = -\arg\{X(\omega)\} = \begin{cases} 0 & X(\omega) > 0 \\ \pi & X(\omega) < 0 \end{cases}$$



Esercizio 2 - TRASFORMATA DI UN IMPULSO RETTANGOLARE (PAGNA 28) (CONTINUAZIONE)

Confrontiamo la trasformata dell'impulso con i coefficienti dello sviluppo in serie della successione di impulsi

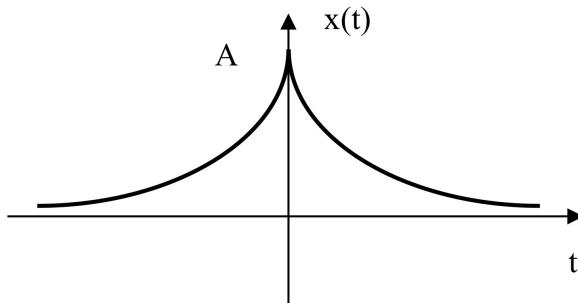
$$X(\omega) = I \frac{\sin(\omega T/2)}{\omega T/2} \quad c_n = \frac{I}{T} \frac{\sin(n\omega_0 T/2)}{n\omega_0 T/2}$$

Si ottiene che i coefficienti possono essere ottenuti campionando a multipli della fondamentale la trasformata (divisa per T).

$$c_n = \frac{1}{T} X(n\omega_0) \rightarrow \text{QUESTO RISULTATO È GENERALE!!}$$

Esercizio 3 - TRASFORMATA DI UN'ESPOENZIALE BIATERA (PAG. 30)

$$x(t) = A e^{-|t|/t_0} \quad t_0 > 0$$



Si tratta di una funzione reale pari, questa volta continua nei valori, tempo-continua di tipo aperiodico.

Fig.19 Esponenziale bilatera.

$$\begin{aligned} X(\omega) &= \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt = A \int_{-\infty}^0 e^{t/t_0} e^{-j\omega t} dt + A \int_0^{+\infty} e^{-t/t_0} e^{-j\omega t} dt = \\ &= A \int_{-\infty}^0 e^{t(1/t_0 - j\omega)} dt + A \int_0^{+\infty} e^{t(-1/t_0 - j\omega)} dt = \end{aligned}$$

RISULTATO UTILE  
PER IL PROSSIMO  
ESEMPIO

$$\begin{aligned} &= A \left[ \frac{e^{t(1/t_0 - j\omega)}}{1/t_0 - j\omega} \right]_{-\infty}^0 + A \left[ \frac{e^{t(-1/t_0 - j\omega)}}{-1/t_0 - j\omega} \right]_0^{+\infty} = \frac{At_0}{1 - j\omega t_0} + \frac{At_0}{1 + j\omega t_0} = \\ &= \frac{2At_0}{1 + \omega^2 t_0^2} \end{aligned}$$

ESERCIZIO 4 - TRASFORMATA DI UN'ESPONENZIALE MONOLATERA (PAG. 32)

$$x(t) = \begin{cases} 0 & t < 0 \\ A e^{-\frac{t}{t_0}} & t \geq 0 \end{cases} \quad t_0 > 0$$

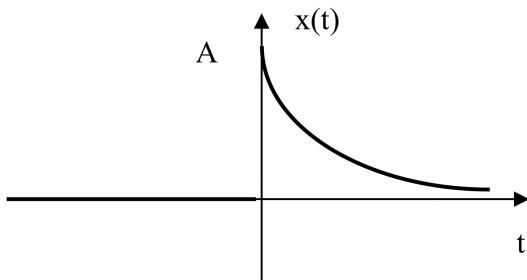


Fig.21 Esponenziale monolatera.

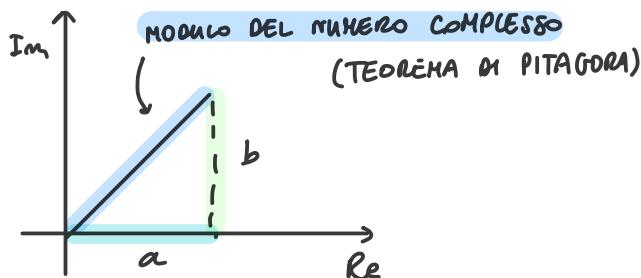
Il calcolo della trasformata è immediato in quanto l'espressione della trasformata coincide con il primo dei due integrali dell'esercizio precedente, quindi basta richiamare il risultato ottenuto.

$X(\omega)$  RISULTA COMPLESSO, PERCHÉ  $x(t)$  È REALE MA NON È PARI.

$$X(\omega) = \int_{-\infty}^{+\infty} x(t) e^{-j\omega t} dt = A \int_0^{+\infty} e^{-t/t_0} e^{-j\omega t} dt = \frac{At_0}{1+j\omega t_0}$$

SPETTO DI AMPIEZZA MONOLATERO:

$$V(\omega) = \frac{|X(\omega)|}{\pi} = \frac{At_0}{\pi} \frac{1}{\sqrt{1+\omega^2 t_0^2}} \quad \omega \geq 0$$

SPETTO DI FASE MONOLATERO:

$$\begin{aligned} \phi(\omega) &= -\arg\{X(\omega)\} = -(0 - \arctg \omega t_0) = \\ &= \arctg \omega t_0 \end{aligned}$$

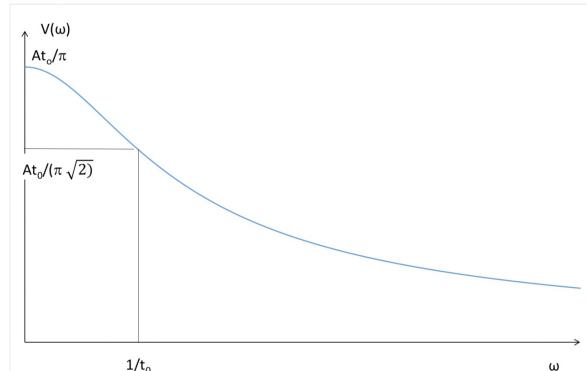


Fig.22 Spettro (densità spettrale) di ampiezza dell'esponenziale monolatera.

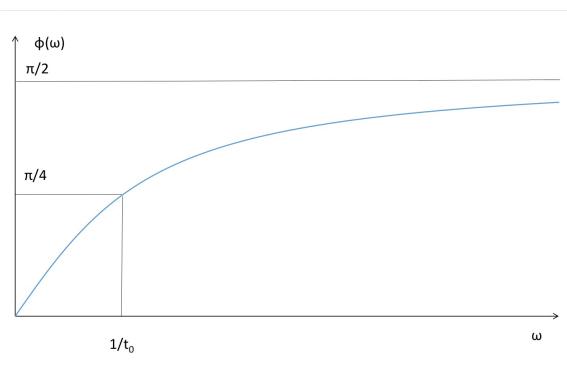


Fig.23 Spettro di fase dell'esponenziale monolatera.

Esercizio 5 - ANTITRASFORMATA DI UN IMPULSO NEI FREQUENZE (PAG. 33)

$$X(\omega) = \begin{cases} X_0 & |\omega| < \omega_m \\ 0 & |\omega| > \omega_m \end{cases} \quad X_0 > 0$$

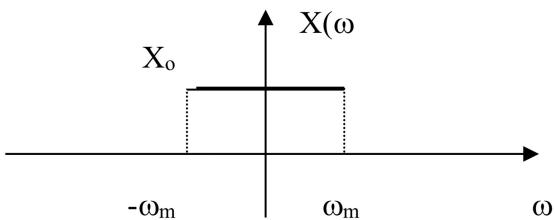


Fig. 24 Impulso nelle frequenze.

Questo esercizio è il duale dell'esercizio 2. Questa volta si vuole \$x(t)\$, quindi si procede dalle frequenze ai tempi.

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega = \frac{X_0}{2\pi} \int_{-\omega_m}^{+\omega_m} e^{j\omega t} d\omega = \frac{X_0}{2\pi} \left[ \frac{e^{j\omega t}}{jt} \right]_{-\omega_m}^{+\omega_m} =$$

$$\begin{aligned} X_0 &= \frac{X_0 \omega_m}{\pi} \frac{\sin(\omega_m t)}{\omega_m t} = X_0 \frac{\sin(\omega_m t)}{\omega_m t} = X_0 \operatorname{sinc}\left(\frac{\omega_m t}{\pi}\right) \end{aligned}$$

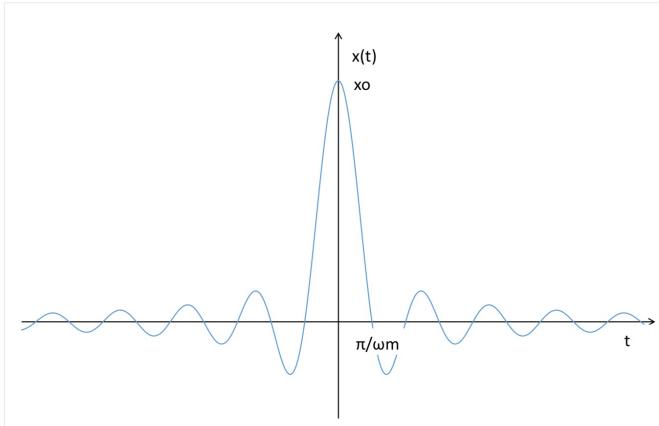


Fig. 25 Antitrasformata di un impulso nelle frequenze.

All'impulso nelle frequenze corrisponde un sinc nei tempi, viceversa ad un impulso nei tempi corrisponde un sinc nelle frequenze.

Maggiore la banda, più stretto il sinc. \$(\omega\_m)\$

**Sinc \$\leftrightarrow\$ IMPULSO RETTANGOLARE**

Esercizio 5 - VARIANTE

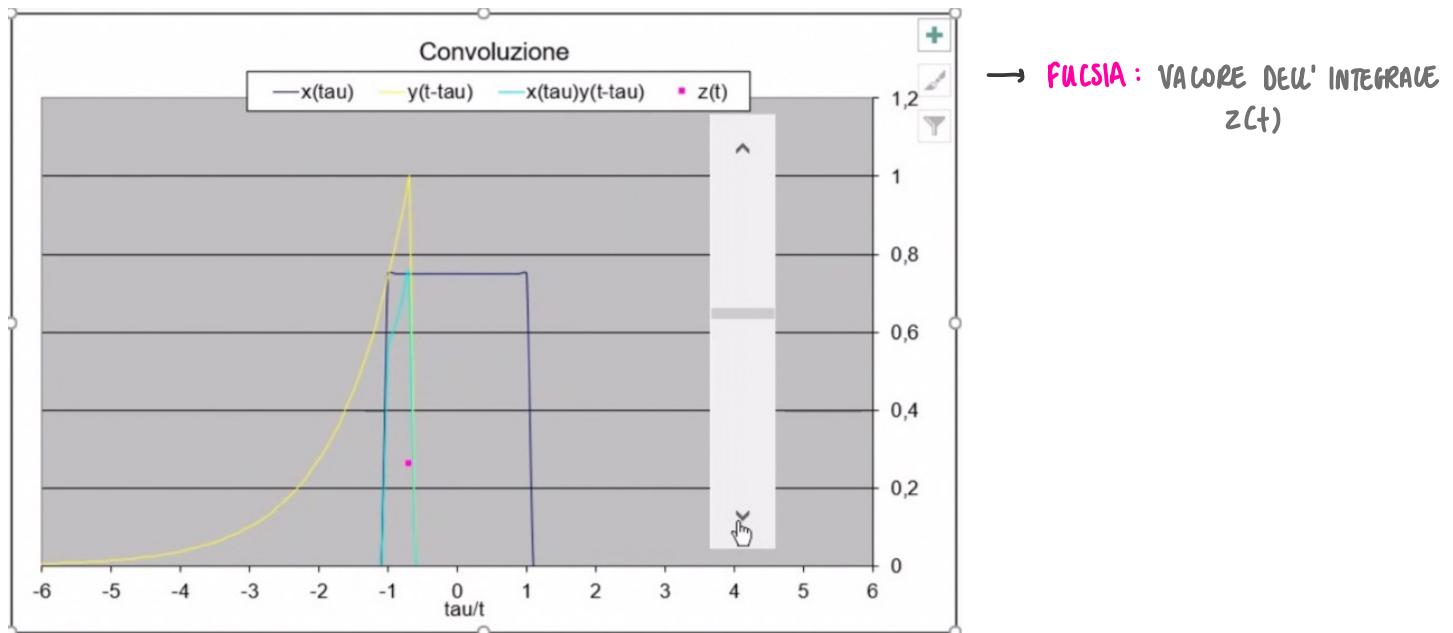
$$X(\omega) = \begin{cases} X_0 e^{-j\omega t_0} & |\omega| < \omega_m \\ 0 & |\omega| > \omega_m \end{cases} \quad X_0 > 0$$

Come esercizio precedente, ma aggiunta una caratteristica di fase proporzionale a \$\omega\$.

Il risultato è immediato, ricordando la proprietà dei segnali ritardati

$$x(t) = x_0 \frac{\sin[\omega_m(t-t_0)]}{\omega_m(t-t_0)} = x_0 \operatorname{sinc}\left[\frac{\omega_m(t-t_0)}{\pi}\right]$$

## GRAFICO DELLA PROPRIETÀ DI CONVOLUZIONE DELLA TRASFORMATA DI FOURIER



## CAPITOLO 3 - TRASFORMATE AL LIMITE

## LA FUNZIONE GENERALIZZATA DELTA DI DIRAC

Prima di definire la delta di Dirac conviene ricordare le seguenti definizioni:

- **funzione ordinaria:** associa ad un numero un altro numero ed uno soltanto (ad esempio la funzione coseno);
- **funzionale:** associa ad una funzione un numero (ad esempio il valore medio);
- **operatore:** associa ad una funzione un'altra funzione (ad esempio l'operatore derivata).

Ciò premesso le distribuzioni, o funzioni generalizzate, sono dei funzionali dotati delle proprietà di essere lineari e continui, operanti su delle funzioni dotate di particolari proprietà di regolarità dette funzioni di prova.

Il numero che una distribuzione  $T$  associa alla funzione di prova  $\varphi$  è indicato con  $\langle \varphi, T \rangle$ . Si noti che  $\langle \cdot \rangle$  è lo stesso simbolo utilizzato normalmente per il prodotto scalare, o prodotto interno, di due funzioni ordinarie.

La distribuzione delta di Dirac non è funzione ordinaria.

Data una funzione  $x(t)$  continua in  $t=0$  (funzione di prova), la distribuzione delta di Dirac associa ad essa il valore  $x(0)$ , ovvero campiona la funzione  $x(t)$  nell'origine.

$$\langle x, \delta \rangle = x(0)$$

Analogamente si può definire la distribuzione  $\delta_{t_0}$  come la distribuzione che associa a  $x(t)$ , continua in  $t_0$ , il valore  $x(t_0)$ , ovvero che campiona la funzione in  $t_0$ .

$$\langle x, \delta_{t_0} \rangle = x(t_0)$$

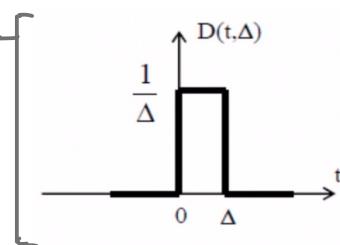
Il valore  $x(0)$  di una funzione  $x(t)$  continua nell'origine può essere ricavato anche facendo ricorso a famiglie di funzioni ausiliarie  $f_\Delta(t)$  per le quali vale la seguente espressione:

$$x(0) = \lim_{\Delta \rightarrow 0} \int_{-\infty}^{+\infty} x(t) f_\Delta(t) dt$$

ESEMPIO: FUNZIONE AUSILIARIA  $D(t, \Delta)$

IMPULSO RETTANGOLARE, NON CENTRATO

$I = 1$  (INTENSITÀ)



Se vale questo limite, si suole indicare:

$$\delta(t) = \lim_{\Delta \rightarrow 0} f_\Delta(t)$$

da interpretare non in senso ordinario, ma in senso integrale, ovvero sulla base della seguente

$$\int_{-\infty}^{+\infty} x(t) \delta(t) dt = \lim_{\Delta \rightarrow 0} \int_{-\infty}^{+\infty} x(t) f_\Delta(t) dt = x(0)$$

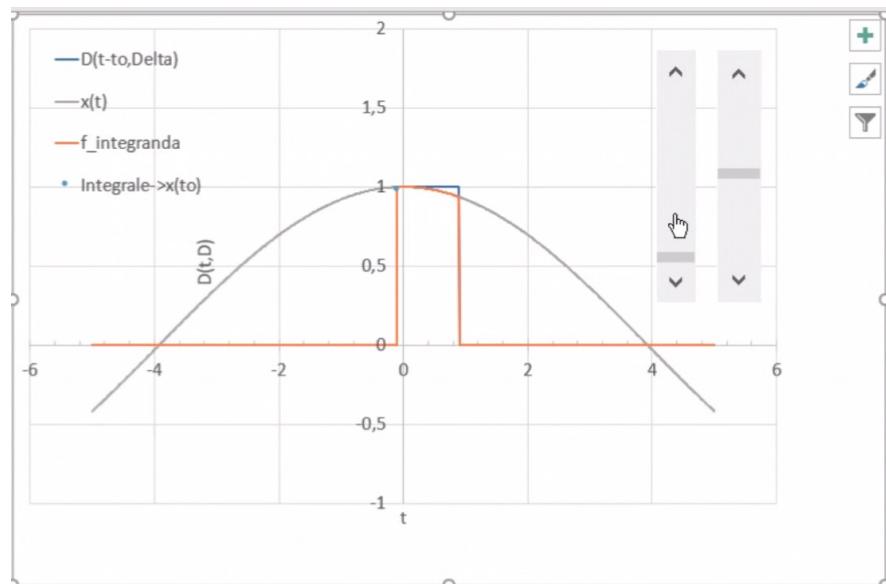
Confrontando infine l'espressione sopra con la definizione di distribuzione delta si ha

$$\langle x, \delta \rangle = x(0) = \int_{-\infty}^{+\infty} x(t) \delta(t) dt$$

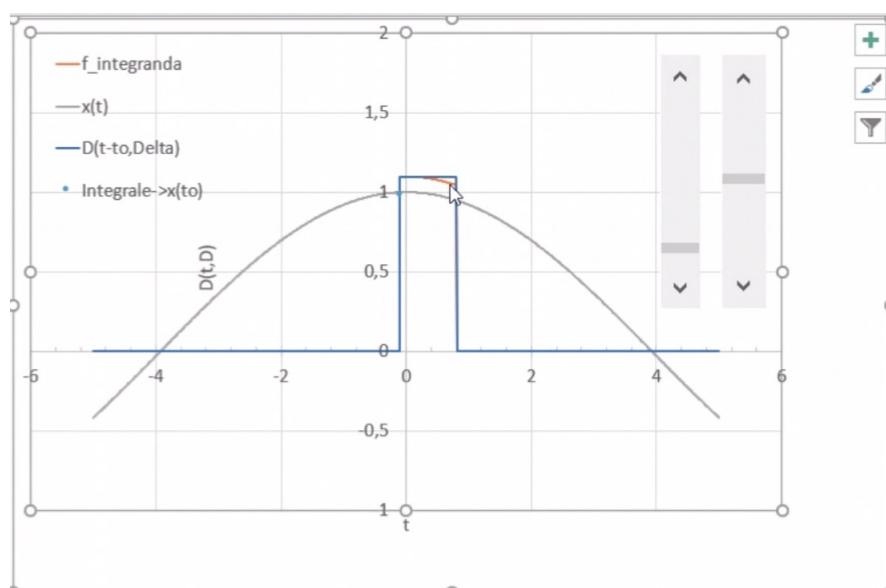
La prima uguaglianza rappresenta la definizione della distribuzione  $\delta$ , la seconda un modo alternativo di ottenere il medesimo risultato, utile perché permette di operare con il simbolo  $\delta(t)$  in modo analogo ad una funzione ordinaria. Si noti inoltre la corrispondenza formale fra il simbolo  $\langle \cdot, \cdot \rangle$  e l'integrale a destra, identico al prodotto interno di due funzioni.

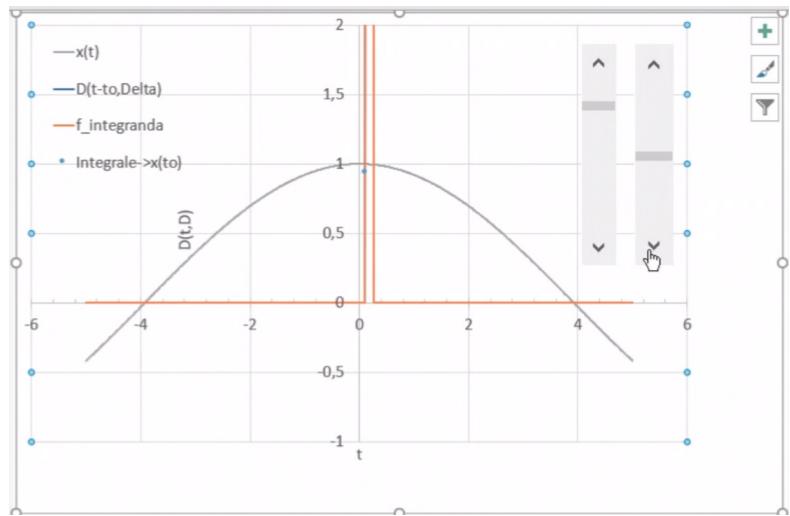
Quanto detto si estende ovviamente anche alle distribuzioni delta ritardate, per le quali

$$\langle x, \delta_{t_0} \rangle = x(t_0) = \int_{-\infty}^{+\infty} x(t) \delta(t - t_0) dt = \lim_{\Delta \rightarrow 0} \int_{-\infty}^{+\infty} x(t) f_\Delta(t - t_0) dt$$



→ RAPPRESENTAZIONE GRAFICA  
DI QUANTO DETTO FIN'ORA





### PROPRIETÀ DELLA FUNZIONE DELTA DI DIRAC

- PARITÀ →  $\int_{-\infty}^{+\infty} x(t) \delta(t - t_0) dt = \int_{-\infty}^{+\infty} x(t) \delta(t_0 - t) dt$

DIMOSTRAZIONE →  $\int_{-\infty}^{+\infty} x(t) \delta(t_0 - t) dt = \lim_{\Delta \rightarrow 0} \int_{-\infty}^{+\infty} x(t) D(t_0 - t, \Delta) dt =$

 $= \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \int_{t_0 - \Delta}^{t_0} x(t) dt = x(t_0)$

- CONVOLUZIONE (ELEMENTO NEUTRO)

La funzione generalizzata delta rappresenta l'elemento neutro dell'integrale di convoluzione.

$$x(t) = \int_{-\infty}^{+\infty} x(\tau) \delta(t - \tau) d\tau = x(t)^* \delta(t)$$

LA DIMOSTRAZIONE È IDENTICA A QUELLA DI PARITÀ, BASTA SOSTITUIRE  $t$  CON  $\tau$  E  $t_0$  CON  $t$ .

- CAMBIO DI ARGOMENTO →  $\delta(\alpha t) = \frac{\delta(t)}{|\alpha|} \quad \alpha \neq 0$

DIMOSTRAZIONE →  $\int_{-\infty}^{+\infty} \delta(\alpha t) dt = \int_{-\infty}^{+\infty} \frac{\delta(\xi)}{|\alpha|} d\xi \cdot \int_{-\infty}^{+\infty} \frac{\delta(t)}{|\alpha|} dt$

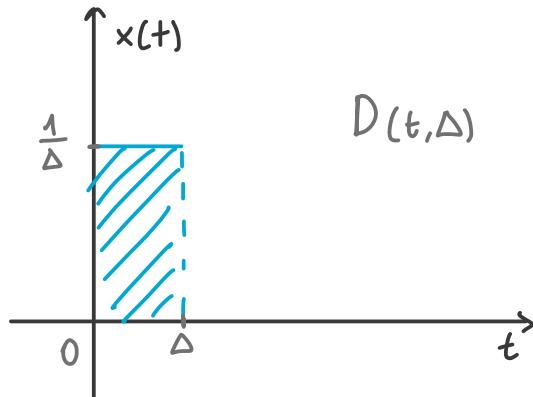
La presente proprietà è utile nel passaggio dalle descrizioni nelle pulsazioni a quelle in frequenza, o viceversa, per le quali vale la seguente

$$\delta(\omega) = \delta(2\pi f) = \frac{\delta(f)}{2\pi}$$

- TRASFORMATA DI FOURIER

$$\int_{-\infty}^{+\infty} \delta(t) e^{-j\omega t} dt = 1$$

Dall'espressione sopra deriva che la trasformata di una delta nell'origine dei tempi corrisponde ad una costante nel dominio delle frequenze.



### DIMOSTRAZIONE:

Immediata dalla definizione di distribuzione, in quanto la funzione esponenziale campionata nell'origine vale 1.

### Verifica

Si noti, come verifica, che lo stesso risultato poteva essere ottenuto utilizzando la funzione ausiliaria  $D(t, \Delta)$ . Ricordando che

$$F[D(t, \Delta)] = \frac{\sin \frac{\omega \Delta}{2}}{\frac{\omega \Delta}{2}} e^{-j \frac{\omega \Delta}{2}}$$

$$\text{SI HA } \rightarrow \lim_{\Delta \rightarrow 0} \int_{-\infty}^{+\infty} D(t, \Delta) e^{-j\omega t} dt = \lim_{\Delta \rightarrow 0} F[D(t, \Delta)] = 1$$

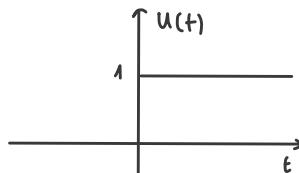
**PROPRIETÀ: GRADINO UNITARIO (o di HEAVISIDE)**

**L'integrale di una delta nell'origine è il gradino unitario.**

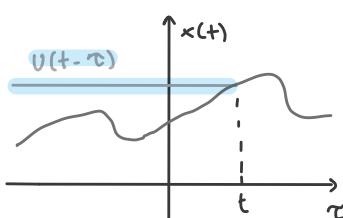
È evidente che la delta non è una funzione ordinaria: l'integrale di funzioni ordinarie è sempre una funzione continua.

$$U(t) = \int_{-\infty}^t \delta(\tau) d\tau$$

$$U(t) = \begin{cases} 1 & t > 0 \\ 0 & t < 0 \end{cases}$$



**DIMOSTRAZIONE:** Poiché l'integrale di una funzione  $x(t)$  è dato dalla convoluzione della funzione stessa con la funzione gradino,



$$y(t) = \int_{-\infty}^t x(\tau) d\tau = \int_{-\infty}^{+\infty} x(\tau) U(t-\tau) d\tau = x(t) * U(t)$$

Sostituendo  $x(t)$  con  $\delta(t)$ , e ricordando che la Delta di Dirac è l'elemento neutro del prodotto di convoluzione, si ottiene immediatamente il risultato.

**ALTRI RELAZIONI CON LE FUNZIONI A GRADINO:**

$$\frac{dU(t)}{dt} = \delta(t) \rightarrow$$

La derivata del gradino esiste (anche nel punto di discontinuità), ma è una distribuzione.

ANALOGAMENTE

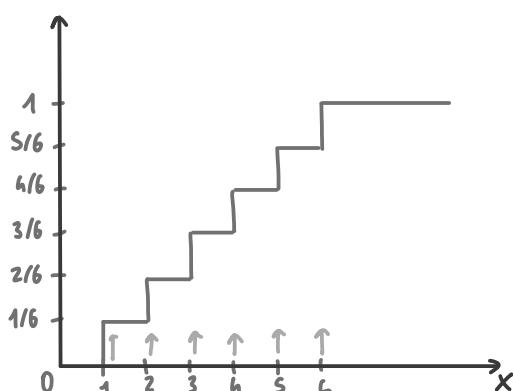
$$U(t-t_0) = \int_{-\infty}^{t-t_0} \delta(\tau) d\tau$$

$$\frac{dU(t-t_0)}{dt} = \delta(t-t_0)$$

ALTRI GRADINI:

$$\ell(t) = \begin{cases} +\frac{1}{2} & t > 0 \\ -\frac{1}{2} & t < 0 \end{cases} = U(t) - \frac{1}{2}$$

$$\text{sign}(t) = \begin{cases} 1 & t > 0 \\ -1 & t < 0 \end{cases} = 2U(t) - 1 = 2\ell(t)$$



## TRASFORMATE NELLE FREQUENZE:

Dato che in letteratura vengono utilizzate sia le rappresentazioni nelle pulsazioni che nelle frequenze, conviene avere un minimo di familiarità con entrambe.

A questo scopo si ricordano le definizioni di trasformata nelle frequenze (indicata qui con un pedice f) ed antitrasformata:

$$X_f(f) = \int_{-\infty}^{+\infty} x(t) e^{-j2\pi ft} dt \quad x(t) = \int_{-\infty}^{+\infty} X_f(f) e^{j2\pi ft} df$$

Vale la seguente relazione fra trasformate nelle frequenze e trasformate nelle pulsazioni  $X(\omega)$ :

$$X_f(f) = X(2\pi f) = X(\omega)$$

## TRASFORMATE ELEMENTARI (NOTAZIONE FREQUENZE)

La trasformata di una costante nei tempi è una delta nell'origine delle frequenze.

La trasformata di un esponenziale nei tempi è una delta ritardata.

Utilizzando le frequenze si ha:

$$F_f[1] = \delta(f) \quad F_f[e^{j2\pi f_0 t}] = \delta(f - f_0)$$

**VERIFICA** → Queste proprietà non sono ottenute in modo diretto, applicando la trasformata al primo membro, ma in maniera inversa, cioè inserendo i secondi membri nelle formule di antitrasformazione:

$$\int_{-\infty}^{+\infty} \delta(f) e^{j2\pi ft} df = 1 \quad \int_{-\infty}^{+\infty} \delta(f - f_0) e^{j2\pi ft} df = e^{j2\pi f_0 t}$$

In generale, per passare da una notazione all'altra (tra frequenze e pulsazioni) vale la seguente formula:

$$\rightarrow \delta(\omega) = \frac{\delta(f)}{2\pi}$$

## TRASFORMATA DI UN SEGNALE SVILUPPABILE IN SERIE DI FOURIER

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t}$$

Grazie alle trasformate elementari, è possibile trovare la trasformata di un segnale periodico. Naturalmente non è una funzione ordinaria.

$$X(\omega) = \sum_{n=-\infty}^{\infty} c_n 2\pi \delta(\omega - n\omega_0)$$

$$X_f(f) = \sum_{n=-\infty}^{\infty} c_n \delta(f - n f_0)$$

TRASFORMATA DI COSENO E SENOCOSENO

$$\cos \omega_o t = \frac{e^{j\omega_o t} + e^{-j\omega_o t}}{2}$$

SENO

$$\sin \omega_o t = \frac{e^{j\omega_o t} - e^{-j\omega_o t}}{2j}$$

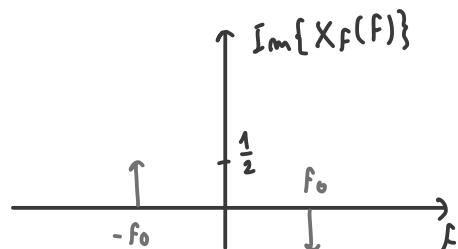
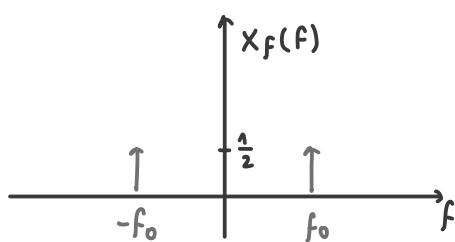
$$X(\omega) = \frac{1}{2} \delta(\omega - \omega_o) 2\pi + \frac{1}{2} \delta(\omega + \omega_o) 2\pi$$

$$X(\omega) = \frac{1}{2j} \delta(\omega - \omega_o) 2\pi - \frac{1}{2j} \delta(\omega + \omega_o) 2\pi =$$

$$X_f(f) = \frac{1}{2} \delta(f - f_o) + \frac{1}{2} \delta(f + f_o)$$

$$= -j \frac{1}{2} \delta(\omega - \omega_o) 2\pi + j \frac{1}{2} \delta(\omega + \omega_o) 2\pi$$

$$X_f(f) = -j \frac{1}{2} \delta(f - f_o) + j \frac{1}{2} \delta(f + f_o)$$

TRASFORMATA DI FOURIER DUE FUNZIONI A GRADINO

A partire dalla trasformata del gradino  $U(t)$  (formula non dimostrata)

$$F[U(t)] = \frac{1}{j\omega} \quad F_f[U(t)] = \frac{1}{j2\pi f}$$

Dalle definizioni di  $U(t)$  e  $\text{sign}(t)$  segue

$$\begin{aligned} F[U(t)] &= \frac{1}{j\omega} + \frac{2\pi\delta(\omega)}{2} & F_f[U(t)] &= \frac{1}{j2\pi f} + \frac{\delta(f)}{2} \\ F[\text{sign}(t)] &= \frac{2}{j\omega} & F_f[\text{sign}(t)] &= \frac{1}{j\pi f} \end{aligned}$$

TRASFORMATA DI FOURIER DI UN INTEGRALE

POICHÉ  $y(t) = \int_{-\infty}^t x(\tau) d\tau = \int_{-\infty}^{+\infty} x(\tau) U(t - \tau) d\tau = x(+)* U(t)$

$$F[U(t)] = \frac{1}{j\omega} + \frac{2\pi\delta(\omega)}{2}$$

Trasformando il prodotto di convoluzione e facendo uso della (3.46) si ottiene dalla  $\text{espressione generale della trasformata di un integrale}$ ,

$$Y(\omega) = X(\omega) F[U(t)] = \frac{X(\omega)}{j\omega} + \frac{2\pi\delta(\omega)X(\omega)}{2} = \frac{X(\omega)}{j\omega} + \frac{2\pi\delta(\omega)X(0)}{2} \rightarrow Y(\omega) = \frac{X(\omega)}{j\omega} \quad \text{PER } X(0) = 0$$

ESERCIZIO 6 - TRASFORMATA DI UN SEGNALE MODULATO PER UNA SINUSOIDA (PAG. 35)

$$s(t) = x(t) \cos \omega_0 t$$

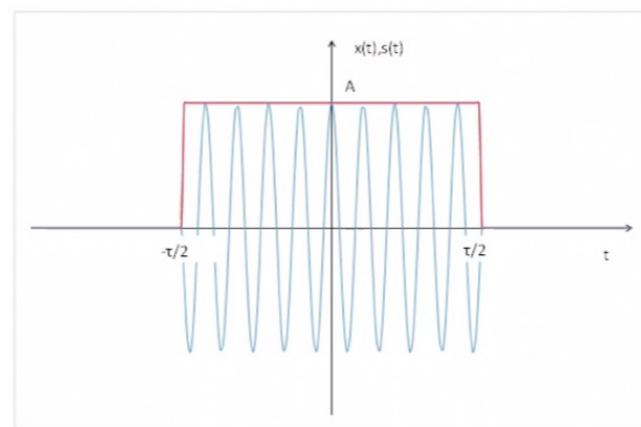
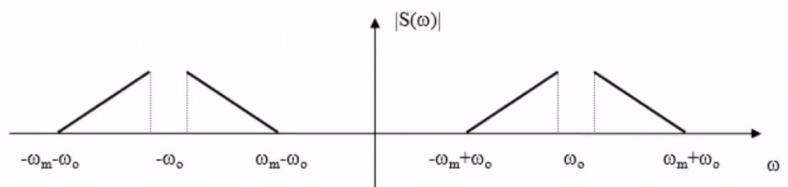
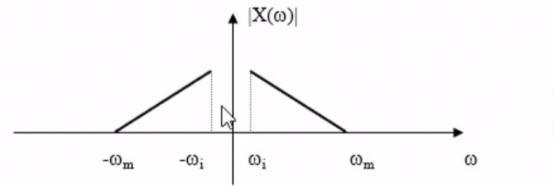
Si esprime il coseno come somma di due esponenziali complessi coniugati

$$s(t) = \frac{1}{2} x(t) e^{j\omega_0 t} + \frac{1}{2} x(t) e^{-j\omega_0 t}$$

Quindi si ottiene la trasformata con semplici passaggi

$$\begin{aligned} S(\omega) &= \frac{1}{2} \int_{-\infty}^{+\infty} x(t) e^{j\omega_0 t} e^{-j\omega t} dt + \frac{1}{2} \int_{-\infty}^{+\infty} x(t) e^{-j\omega_0 t} e^{-j\omega t} dt = \\ &= \frac{1}{2} \int_{-\infty}^{+\infty} x(t) e^{-j(\omega-\omega_0)t} dt + \frac{1}{2} \int_{-\infty}^{+\infty} x(t) e^{-j(\omega+\omega_0)t} dt = \\ &= \frac{1}{2} X(\omega - \omega_0) + \frac{1}{2} X(\omega + \omega_0) \end{aligned}$$

Si noti che se per  $\omega_0 \gg \omega_m$  si ottiene un segnale passa-banda a partire da un segnale passa-basso.



→ CASO PARTICOLARE, TRONCO DI SINUSOIDA

$$\begin{aligned} S(\omega) &= \frac{1}{2} X(\omega - \omega_0) + \frac{1}{2} X(\omega + \omega_0) = \\ &= \frac{I}{2} \operatorname{sinc} \frac{(\omega - \omega_0)\tau}{2\pi} + \frac{I}{2} \operatorname{sinc} \frac{(\omega + \omega_0)\tau}{2\pi} \end{aligned}$$

**Un tronco di sinusoida non è una sinusoida!**

La sua trasformata sul lato positivo è data da sinc centrato sulla frequenza della sinusoida. Maggiore la durata della finestra, più stretto il sinc (più simile alla riga della sinusoida).

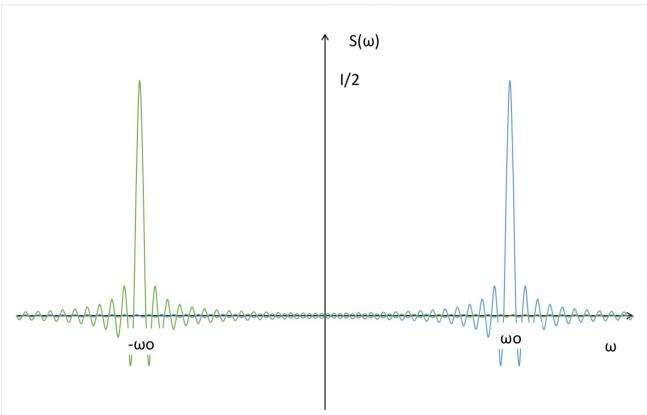
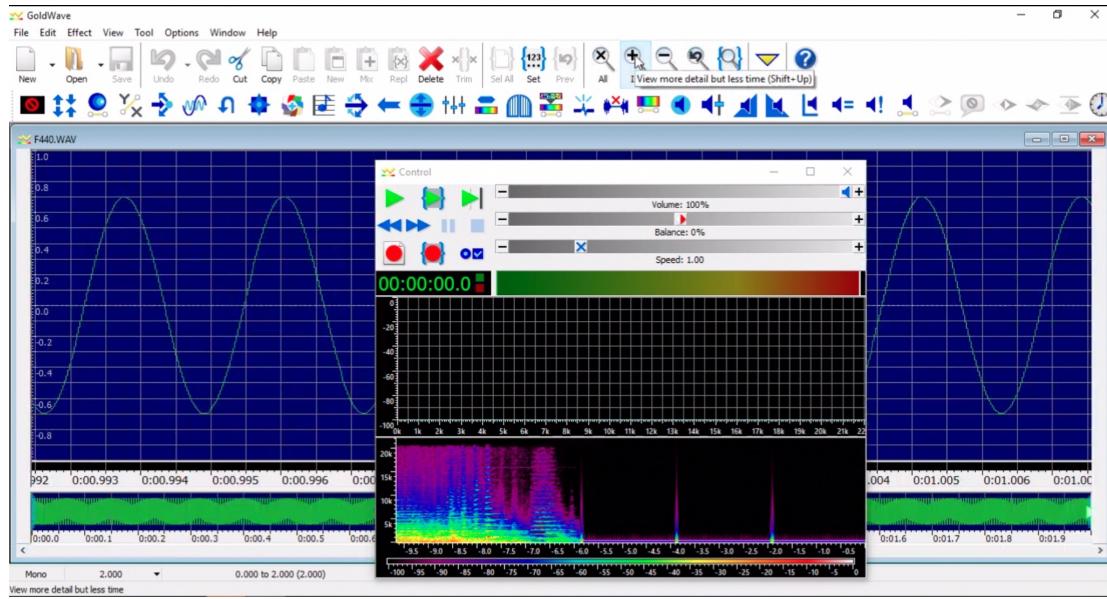


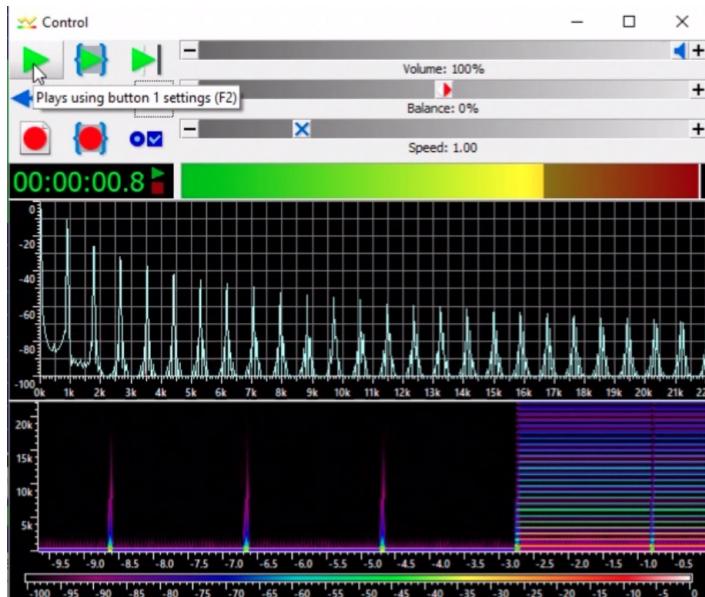
Fig.29 Trasformata di un tronco di sinusoida

04/03/22

## PROGRAMMA UTILIZZATO: GOLDWAVE

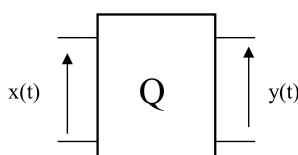


→ RAPPRESENTAZIONE  
AUDIO-VISIVA DI  
UN'ONDA PERIODICA  
SINUOIDALE



→ SIMILE AD UNO SPETTRO DI AMPIEZZA

→ SPETTOGRAMMA

CAPITOLO 5 - SISTEMI UNEARI

$$y(+)=Q[x(+)]$$

- **Causale:** se l'uscita da un determinato istante dipende dai valori passati e da quello attuale dell'ingresso, ma non dai valori futuri.
- **Algebrico:** se l'uscita dipende solo dal valore attuale dell'ingresso (il sistema non ha memoria).
- **Lineare:** vale la "sovraposizione degli effetti".

$$Q[c_1x_1(+) + c_2x_2(+)] = c_1Q[x_1(+)] + c_2Q[x_2(+)]$$

Il sistema è detto **tempo-invariante** se la risposta al segnale ritardato è la risposta ritardata, qualunque sia il ritardo  $t_0$ :

$$y(t-t_0) = Q[x(t-t_0)]$$

RISPOSTA IMPULSIVA DI UN SISTEMA UNEARE TEMPO-INVARIANTE

La risposta impulsiva  $h(t)$  è definita come la risposta della rete all'impulso di Dirac  $\delta(t)$  (distribuzione  $\delta$ ).

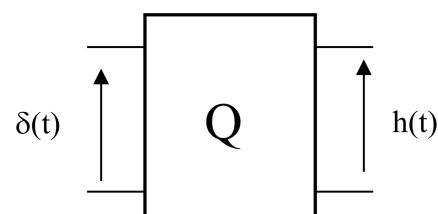
Un sistema lineare tempo invariante può essere completamente caratterizzato dal dominio dei tempi dalla sua  $h(t)$ .

$h(t)$  può essere reale o complessa, noi ci limiteremo al caso reale.

DEFINIZIONE ALTERNATIVA (OPERATIVA)

$$h(t) = \lim_{\Delta \rightarrow 0} y_\Delta(t)$$

CONDIZIONE CAUSALITÀ  $\rightarrow h(t) = 0 \quad t < 0$



La risposta impulsiva consente di esprimere l'uscita della rete quando al suo ingresso è presente un generico segnale  $x(t)$ .

$$y(t) = x(t) * h(t)$$

**DIMOSTRAZIONE:** La dimostrazione di questo fondamentale risultato si basa sul fatto che la Delta di Dirac è l'elemento neutro del prodotto di convoluzione e sul principio di sovrapposizione degli effetti valido per le reti lineari. Per prima cosa si esprime l'ingresso come prodotto di convoluzione di se stesso con la Delta:  $x(t) = x(t) * \delta(t) = \int_{-\infty}^{+\infty} x(\tau) \delta(t - \tau) d\tau$

PER TEMPO INVARIANZA  $\rightarrow$  1)  $\delta(t - \tau) \rightarrow h(t - \tau)$

PER LINEARITÀ (PRODOTTO  $\rightarrow$  2)  $x(\tau) \delta(t - \tau) \rightarrow x(\tau) h(t - \tau)$   
DI UNA COSTANTE)

PER LINEARITÀ (SOMMA)  $\rightarrow$  3)  $\int_{-\infty}^{+\infty} x(\tau) \delta(t - \tau) d\tau \rightarrow \int_{-\infty}^{+\infty} x(\tau) h(t - \tau) d\tau$

DEF. CONVOLUZIONE  $\rightarrow$  4)  $x(t) \rightarrow x(t) * h(t)$

### FUNZIONE DI TRASFERIMENTO DI UNA RETE LINEARE:

La caratterizzazione nel dominio delle frequenze di una rete lineare è data dalla funzione di trasferimento  $H(\omega)$ .

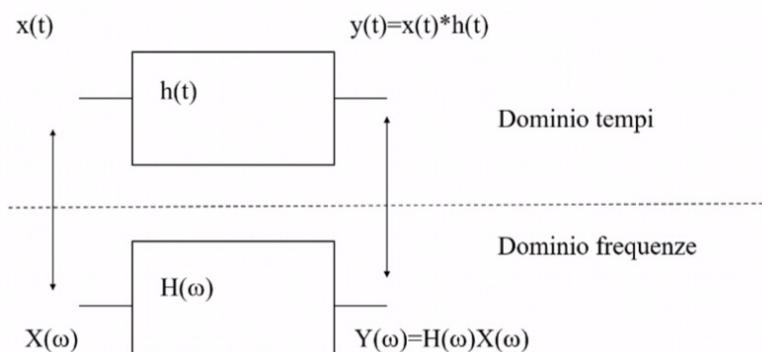
La definizione più immediata di funzione di trasferimento è come trasformata di Fourier della risposta impulsiva:

$$H(\omega) = F[h(t)]$$

Data questa definizione, dalla relazione  $y(t) = x(t) * h(t)$  è immediato, ricordando il teorema della convoluzione, ricavare la seguente

$$Y(\omega) = X(\omega) H(\omega)$$

Essa permette di ricavare la risposta noto l'ingresso nel dominio delle frequenze.



→ QUESTE DUE FORMULE HANNO LE STESSE GENERALITÀ? NO.  
UNA È UNA TRASFORMATO

→ ANCHE SE LA  $h(t)$  È REALE, LA  $H(\omega)$  È GENERALMENTE COMPLESSA

Nel caso in cui  $h(t) \in \mathbb{R}$ , cioè la rete è reale, vale la simmetria hermitiana:  $H(\omega) = H^*(-\omega)$

## CARATTERISTICHE DI AMPIEZZA E FASE

La funzione di trasferimento è in generale complessa; il medesimo contenuto informativo può essere dato da due funzioni reali, le caratteristiche di ampiezza e fase, prese insieme:

$$\begin{cases} T(\omega) = |H(\omega)| \\ \beta(\omega) = -\arg\{H(\omega)\} \end{cases}$$

Esse sono definite su tutto l'asse delle pulsazioni (a differenza degli spettri monolateri dei segnali).

Se la rete è reale:

$$\begin{cases} T(\omega) = T(-\omega) \\ \beta(\omega) = -\beta(-\omega) \end{cases}$$

PROPRIETÀ:

- RISPOSTA AD UN FASORE

Ad un fasore in ingresso una rete lineare tempo-invariante risponde con un fasore in uscita, avente medesima frequenza angolare e diverso numero complesso rappresentativo.

$$x(t) = c_x e^{j\omega_1 t} \xrightarrow{\text{SENALE IN INGRESSO}} y(t) = c_y e^{j\omega_1 t}, \quad c_y = c_x H(\omega_1) \xrightarrow{\text{RISPOSTA}}$$

DIMOSTRAZIONE: RICORDANDO CHE  $F(e^{j\omega_0 t}) = 2\pi\delta(\omega - \omega_0)$  SI HA

$$Y(\omega) = X(\omega)H(\omega) = c_x 2\pi\delta(\omega - \omega_1)H(\omega) = c_x H(\omega_1) 2\pi\delta(\omega - \omega_1) =$$

$$= c_y 2\pi\delta(\omega - \omega_1)$$

$$\text{DA CUI ANTITRASFORMANDO HO } y(t) = c_y e^{j\omega_1 t}$$

DIMOSTRAZIONE COMPLETA

$$\begin{aligned} Y(\omega) &= X(\omega)H(\omega) = c_x 2\pi\delta(\omega - \omega_1)H(\omega) \\ y(t) &= \frac{1}{2\pi} \int_{-\infty}^{\infty} c_x 2\pi\delta(\omega - \omega_1)H(\omega)e^{j\omega t} d\omega = \\ &= c_x \int_{-\infty}^{\infty} \delta(\omega - \omega_1)H(\omega)e^{j\omega t} d\omega = \\ &= c_x H(\omega_1)e^{j\omega_1 t} \end{aligned}$$

- **RISPOSTA AD UNA SINUSOIDA** → RISPONDE CON UNA SINUSOIDA CON STESSA FREQUENZA DELLA SINUS. IN INGR.

Ad una sinusoida in ingresso una rete lineare tempo-invariante reale risponde con una sinusoida in uscita, avente medesima frequenza angolare e diversa ampiezza e fase. → **LA RISPOSTA DEVE ESSERE REALE (CONVOLZIONE DI  $x(t) \in h(t)$  ENTRAMBI REAU)**

$$x(t) = A_x \cos(\omega_1 t - \varphi_x) \longrightarrow y(t) = A_y \cos(\omega_1 t - \varphi_y)$$

↓  
SEGNALE IN INGRESSO

↓  
dove  $A_y = A_x T(\omega_1)$  → AMPIZZA DELLA SINUSOIDA  
DI USCITA

$\varphi_y = \varphi_x + \beta(\omega_1)$  → FASE DELLA SINUSOIDA  
DI USCITA

DIMOSTRAZIONE "MONOCATERA":

$$x(t) = A_x \cos(\omega_1 t - \varphi_x) = \operatorname{Re} \left\{ \dot{A}_x e^{j\omega_1 t} \right\}, \quad \dot{A}_x = A_x e^{-j\varphi_x}$$

$$\dot{A}_x e^{j\omega_1 t} \longrightarrow \dot{A}_x H(\omega_1) e^{j\omega_1 t} = \dot{A}_y e^{j\omega_1 t}$$

$$\operatorname{Re} \left\{ \dot{A}_x e^{j\omega_1 t} \right\} + j \operatorname{Im} \left\{ \dot{A}_x e^{j\omega_1 t} \right\} \longrightarrow \operatorname{Re} \left\{ \dot{A}_y e^{j\omega_1 t} \right\} + j \operatorname{Im} \left\{ \dot{A}_y e^{j\omega_1 t} \right\}$$

$$\operatorname{Re} \left\{ \dot{A}_x e^{j\omega_1 t} \right\} \xrightarrow{h(t) \in R} \operatorname{Re} \left\{ \dot{A}_y e^{j\omega_1 t} \right\}$$

$$\operatorname{Re} \left\{ \dot{A}_y e^{j\omega_1 t} \right\} = A_y \cos(\omega_1 t - \varphi_y)$$

IN PIÙ  
(NON  
NECESSARIE  
PER L'ESAME)

DIMOSTRAZIONE ALTERNATIVA "BILATERA":

$$x(t) = A_x \cos(\omega_1 t - \varphi_x) = \operatorname{Re} \left\{ \dot{A}_x e^{j\omega_1 t} \right\} = \frac{1}{2} \dot{A}_x e^{j\omega_1 t} + \frac{1}{2} \dot{A}_x^* e^{-j\omega_1 t}, \quad \dot{A}_x = A_x e^{-j\varphi_x}$$

$$y(t) = \frac{1}{2} \dot{A}_x H(\omega_1) e^{j\omega_1 t} + \frac{1}{2} \dot{A}_x^* H(-\omega_1) e^{-j\omega_1 t} =$$

$$= \operatorname{Re} \left\{ \dot{A}_x H(\omega_1) e^{j\omega_1 t} \right\} = \operatorname{Re} \left\{ \dot{A}_y e^{j\omega_1 t} \right\} =$$

$$= A_y \cos(\omega_1 t - \varphi_y)$$

- **RISPOSTA A SEGNALI SVILUPPABILI IN SERIE DI FOURIER O TRASFORMABILI SECONDO FOURIER**

Riassumendo, la conoscenza della funzione di trasferimento permette di ricavare la risposta nel dominio delle frequenze indipendentemente dal tipo di rappresentazione del segnale in ingresso. Si ricordano le seguenti formule specifiche, di immediata derivazione, utili per gli esercizi.

$$X(\omega)$$

$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t}$$

$$x(t) = A_o + \sum_{n=1}^{\infty} A_n \cos(n\omega_0 t - \varphi_n) \quad y(t) = A_o H(0) + \sum_{n=1}^{\infty} A_n T(n\omega_0) \cos[n\omega_0 t - \varphi_n - \beta(n\omega_0)]$$

$$Y(\omega) = X(\omega)H(\omega)$$

$$y(t) = \sum_{n=-\infty}^{\infty} c_n H(n\omega_0) e^{jn\omega_0 t}$$

$$x(t) = \int_0^{+\infty} V(\omega) \cos[\omega t - \varphi(\omega)] d\omega \quad y(t) = \int_0^{+\infty} V(\omega) T(\omega) \cos[\omega t - \varphi(\omega) - \beta(\omega)] d\omega$$

## DEFINIZIONI ALTERNATIVE DI TRASFORMATA (ALTRI CORSI)

Per non partire dalla risposta impulsiva  $h(t)$ , che richiede la definizione di  $\delta(t)$ , in altri corsi potete trovare come definizione della  $H(\omega)$  quelle che per noi sono proprietà derivate. Alcuni partono dalla formula che lega fasore di uscita e fasore di ingresso, per definire  $H(\omega)$  come rapporto dei due numeri complessi rappresentativi dei fasori.

$$x(t) = c_x e^{j\omega_1 t} \quad y(t) = c_y e^{j\omega_1 t}$$

$$H(\omega_1) \equiv \frac{c_y}{c_x}$$

Altri dalla formula che lega sinusoide di uscita e sinusoide di ingresso, per definire  $H(\omega)$  come rapporto fra i due numeri complessi rappresentativi delle sinusoidi.

$$x(t) = A_x \cos(\omega_1 t - \varphi_x) = \operatorname{Re} \{ \dot{A}_x e^{j\omega_1 t} \}$$

$$y(t) = A_y \cos(\omega_1 t - \varphi_y) = \operatorname{Re} \{ \dot{A}_y e^{j\omega_1 t} \}$$

$$H(\omega) \equiv \frac{\dot{A}_y}{\dot{A}_x}$$

Altri ancora la definiscono come rapporto fra trasformate:

$$H(\omega) \equiv \frac{Y(\omega)}{X(\omega)}$$

Tutte queste definizioni non sono sbagliate ma hanno dei limiti.

La prima a volte viene scritta come rapporto fra fasori, anziché fra i loro numeri complessi rappresentativi.

Non è analiticamente sbagliata, ma estremamente pericolosa.

Attenzione, se uno scrive, confondendo dominio dei tempi con quello delle frequenze l'espressione sotto, termina l'esame.

**La funzione di trasferimento è nel dominio delle frequenze NON in quello dei tempi.**

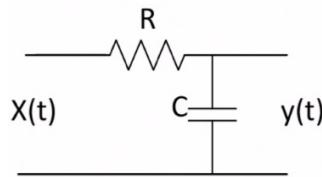


$$H(\omega) \equiv \frac{y(t)}{x(t)} \rightarrow \text{ERRORE !!!}$$

RISPOSTA IMPULSIVA E FUNZIONE DI TRASFERIMENTO DI UNA RETE RC (PAGINA 72-73)

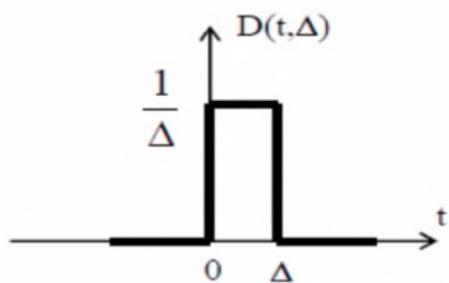
Calcoliamo la  $h(t)$  dalla definizione operativa:

$$h(t) = \lim_{\Delta \rightarrow 0} y_\Delta(t)$$



Come primo passo occorre quindi calcolare la risposta alla funzione ausiliaria  $D(t, \Delta)$ . Ricordando le formule dei transitori di carica e scarica si ha:

$$y_\Delta(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{\Delta} \left( 1 - e^{-\frac{t}{RC}} \right) & 0 \leq t \leq \Delta \\ \frac{1}{\Delta} \left( 1 - e^{-\frac{\Delta}{RC}} \right) e^{-\frac{t-\Delta}{RC}} & t \geq \Delta \end{cases}$$



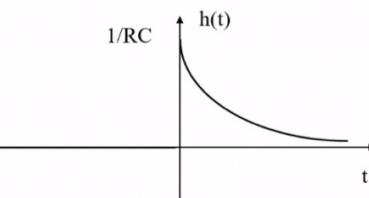
Come secondo passo occorre calcolare il limite:

$$h(t) = \lim_{\Delta \rightarrow 0} y_\Delta(t)$$

Ricordando che si ha:  $\lim_{x \rightarrow 0} e^x = 1 + x$

$$\lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \left( 1 - e^{-\frac{\Delta}{RC}} \right) = \lim_{\Delta \rightarrow 0} \frac{1}{\Delta} \left( 1 - 1 + \frac{\Delta}{RC} + \dots \right) = \frac{1}{RC}$$

$$h(t) = \begin{cases} 0 & t < 0 \\ \frac{1}{RC} e^{-\frac{t}{RC}} & t \geq 0 \end{cases}$$



Nota la  $h(t)$ , si ottiene la funzione di trasferimento facendone la trasformata. Senza fare nessun calcolo, ricordando l'esercizio della trasformata di un'esponenziale monolatera.

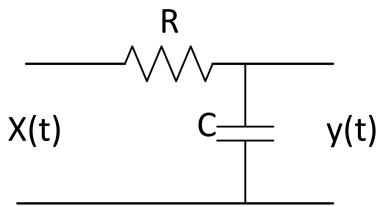
$$x(t) = \begin{cases} 0 & t < 0 \\ Ae^{-\frac{t}{t_0}} & t \geq 0 \end{cases} \quad X(\omega) = \frac{At_0}{1 + j\omega t_0}$$

Dal confronto fra  $x(t)$  ed  $h(t)$ , si osserva che esse coincidono se  $t_0 = RC$  e  $A = 1/RC$ . Ponendo quindi questi valori nell'espressione di  $X(\omega)$ , si ottiene:

$$H(\omega) = \frac{1}{1 + j\omega RC}$$

### FUNZIONE DI TRASFERIMENTO RETE RC (PAGINA 75)

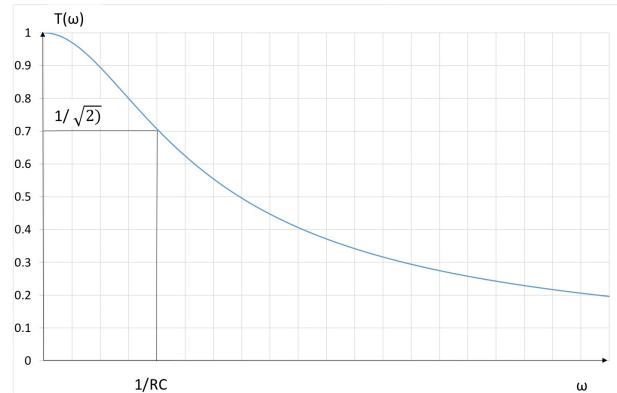
Calcolare la funzione di trasferimento e le caratteristiche di ampiezza e di fase della rete RC di figura, utilizzando le formule per il partitore non resistivo.



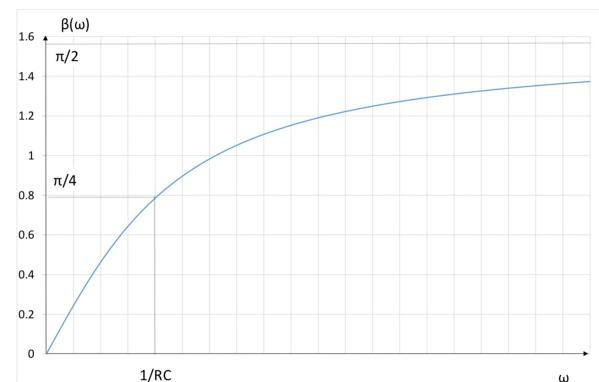
$$H(\omega) = \frac{Z_2}{Z_1 + Z_2} = \frac{\frac{1}{j\omega C}}{R + \frac{1}{j\omega C}} = \frac{1}{1 + j\omega CR}$$

Le caratteristiche di ampiezza e fase sono date da:

$$T(\omega) = |H(\omega)| = \frac{1}{|1 + j\omega CR|} = \frac{1}{\sqrt{1 + (\omega CR)^2}}$$



$$\begin{aligned}\beta(\omega) &= -\arg\{H(\omega)\} = \arg\{1 + j\omega CR\} - \arg\{1\} = \\ &= \arctg(\omega CR)\end{aligned}$$



La rete si comporta come un filtro passa-basso non ideale. La pulsazione di taglio è:

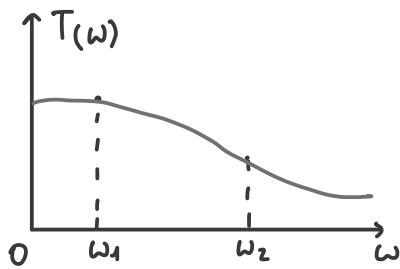
$$\omega_t = \frac{1}{RC}$$

ESERCIZIO TIPO DA ESAME ORALE

TROVARE LA RISPOSTA A DUE SINUSOIDI.

$$x(t) = V_1 \cos(\omega_1 t - \varphi_1) + V_2 \cos(\omega_2 t - \varphi_2)$$

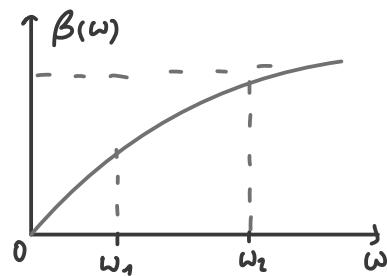
$$y(t) = ?$$



$$y(t) = V_1 T(\omega_1) \cos \left[ \omega_1 t - \varphi_1 - \beta(\omega_1) \right] +$$

$$+ V_2 T(\omega_2) \cos \left[ \omega_2 t - \varphi_2 - \beta(\omega_2) \right]$$

$$y(t) = V_1 T(\omega_1) \cos \left[ \omega_1 t - \varphi_1 - \omega_1 \frac{\beta(\omega_1)}{\omega_1} \right]$$

FUNZIONI DI TRASFERIMENTO DI SISTEMI ELEMENTARI

- LINEA DI RITARDO

$$y(t) = x(t - t_0) \quad H(\omega) = \frac{Y(\omega)}{X(\omega)} = e^{-j\omega t_0}$$

- DERIVATORE

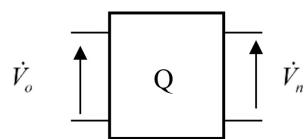
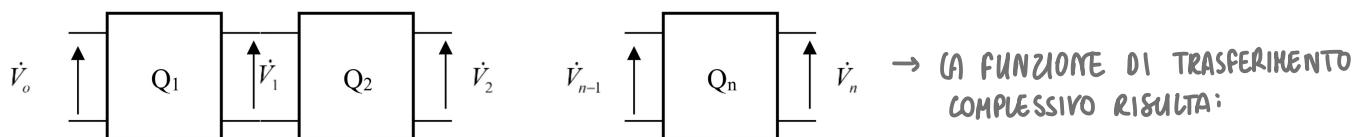
$$y(t) = \dot{x}(t) \quad H(\omega) = j\omega$$

- INTEGRATORE

$$y(t) = \int_{-\infty}^t x(\tau) d\tau \quad H(\omega) = \frac{1}{j\omega}$$

SISTEMI IN CASCATA

Consideriamo più sistemi lineari in cascata: la funzione di trasferimento della cascata è eguale al prodotto delle funzioni di trasferimento.



$$H(\omega) = \frac{V_n}{V_o}$$



E PER LE FUNZIONI DEI VARI BLOCCHI

$$H_1(\omega) = \frac{\dot{V}_1}{\dot{V}_0} \quad H_2(\omega) = \frac{\dot{V}_2}{\dot{V}_1} \quad \rightarrow \quad H_N(\omega) = \frac{\dot{V}_N}{\dot{V}_{N-1}}$$

↓

DA CUI È IMMEDIATO:

$$H(\omega) = H_1(\omega) H_2(\omega) \dots H_N(\omega)$$

Si faccia tuttavia attenzione, nel caso i sistemi lineari siano reti elettriche (quadripoli), che la funzione di trasferimento di ognuno di essi deve essere calcolata tenendo conto del carico effettivamente presente ai morsetti di uscita.

### CONDIZIONI DI NON DISTORSIONE

Si dice che il segnale  $y(t)$  riproduce indistorto  $x(t)$  quando differisce da esso solo per una costante moltiplicativa ed un ritardo:

$$y(t) = A_x(t - t_0) \quad \text{Dove in generale tutte le grandezze sono complesse.}$$

Quando  $x(t)$  e  $y(t)$  sono l'ingresso e l'uscita di una rete lineare, tale rete è detta non distorcente (rispetto a  $x(t)$ ) nel caso in cui valga la condizione sopra.

Trasformando questa secondo Fourier si ha:

$$\begin{aligned} Y(\omega) &= A X(\omega) e^{-j\omega_0 t} \\ &\downarrow \\ H(\omega) &= A e^{-j\omega_0 t_0} \quad \forall \omega: X(\omega) \neq 0 \end{aligned}$$

La condizione sopra è necessaria perché la rete non distorca l'ingresso (cioè se  $y(t)$  riproduce indistorto  $x(t)$  essa deve valere sempre).

Essa è anche condizione sufficiente (se vale allora  $y(t)$  riproduce indistorto  $x(t)$ ) come si può dimostrare immediatamente procedendo a ritroso.

Restringendosi al caso di ingresso e rete reali, anche l'uscita sarà reale. (convoluzione di due funzioni reali)

Assumendola positiva e chiamandola  $T_0$ , dalla condizione di non distorsione enunciata per  $H(\omega)$  si ricavano le equivalenti condizioni di non distorsione per le caratteristiche di ampiezza e fase, avendo indicato con  $B_x$  la banda di  $x(t)$ :

$$\left\{ \begin{array}{l} T(\omega) = T_0 \\ \beta(\omega) = \omega t_0 \end{array} \right. \quad |\omega| \in B_x$$

- La prima richiede la costanza della caratteristica di ampiezza: tutte le componenti in frequenza del segnale devono essere trattate allo stesso modo, cioè amplificate ed attenuate allo stesso valore di  $T_0$ ;
- La seconda richiede la diretta proporzionalità: tutte le componenti in frequenza del segnale devono essere trattate allo stesso modo, cioè ritardate dello stesso valore  $t_0$ .

Sono richieste su tutta la banda del segnale in ingresso (e soltanto su di essa). Possiamo dire che le condizioni di non distorsione sono **condizioni di equità** nel dominio delle frequenze rispetto alle componenti dell'ingresso.

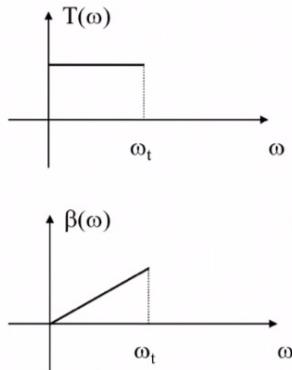
### FILTRI IDEALI

Un **filtro ideale** è un sistema lineare avente la proprietà di fare passare alcune componenti in frequenza, senza distorcerle, eliminando le altre.

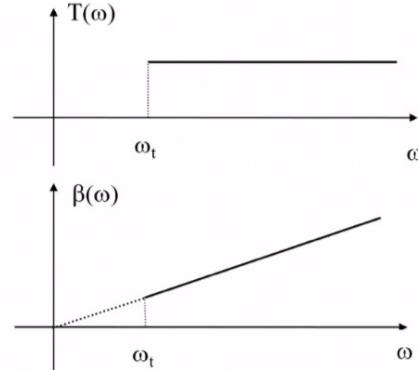
Essi sono caratterizzati da una o più bande passanti, dove valgono le condizioni di non distorsione, e da una o più bande attenuate, dove la caratteristica di ampiezza è nulla. Il passaggio dalle une alle altre avviene alle cosiddette frequenze di taglio.

L'**idealità** nel dominio della frequenza consiste nelle discontinuità della caratteristica di ampiezza nelle transizioni fra bande passanti e bande attenuate (il taglio è verticale, cioè ideale). Essa si riflette nel dominio dei tempi, come vedremo più avanti, in una risposta impulsiva di tipo anticipativo.

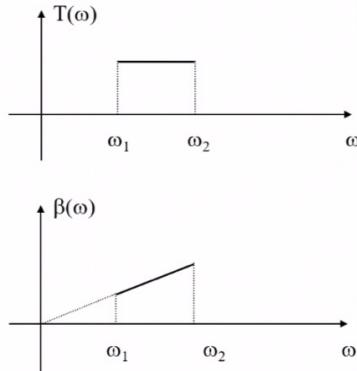
• Passa-basso 



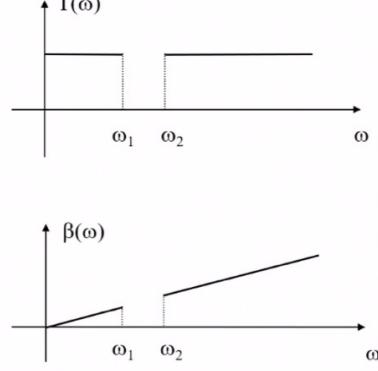
• Passa-alto 



• Passa-banda 



• Elimina-banda (notch filter) 

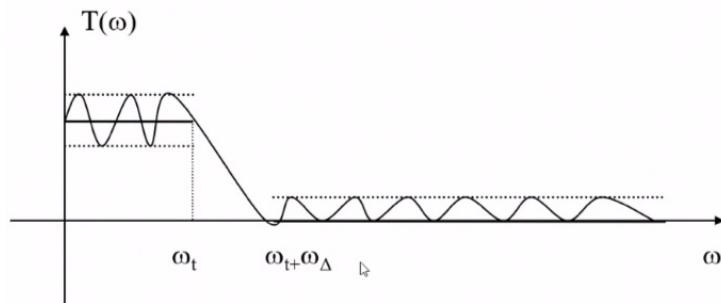


## FILTRI NON IDEALI

I filtri non ideali hanno solo approssimativamente:

- Caratteristica di ampiezza in banda passante costante, nulla in banda attenuata;
- Caratteristica di fase solo approssimativamente direttamente proporzionale in banda passante;
- Taglio discontinuo;

Esistono più "famiglie"; all'interno di una famiglia maggiore è la complessità, maggiore il grado di idealità del filtro.



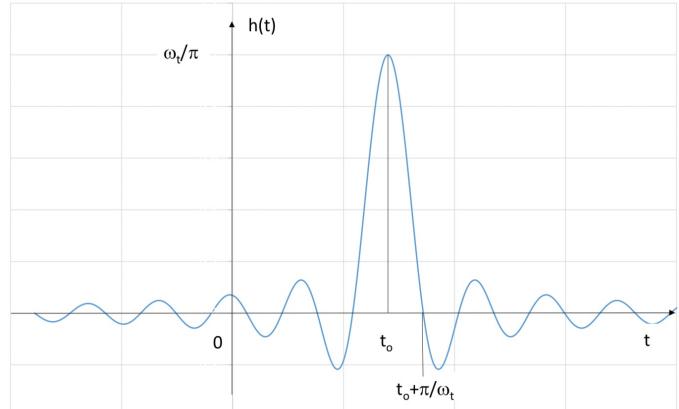
## RISPOSTA IMPULSIVA DI UN FILTRO PASSA-BASSO IDEALE (PAGINA 92)

La funzione di trasferimento è data da:

$$H(\omega) = \begin{cases} e^{-j\omega t_0} & |\omega| \leq \omega_t \\ 0 & |\omega| > \omega_t \end{cases}$$

Ricordando l'esercizio sulla antitrasformata di un impulso nelle frequenze, e la proprietà delle funzioni ritardate si ha:

$$h(t) = F^{-1}[H(\omega)] = \frac{\omega_t}{\pi} \frac{\sin \omega_t (t - t_0)}{\omega_t (t - t_0)}$$

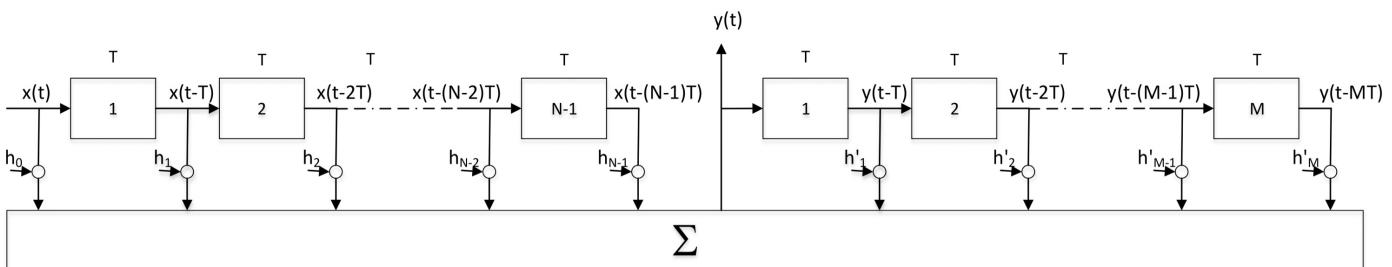


Si noti che  $h(t)$  è **anticipativa**.

La condizione di fisica realizzabilità  $h(t)=0$  per  $t<0$  non è verificata.

## FILTRI RICORRENTI

Consideriamo lo schema a blocchi del sistema rappresentato nella figura sotto: esso è costituito da due catene di blocchi di ritardo (pari a T), una diretta ed una in retroazione. La prima è formata da  $N-1$  blocchi e da N prese che prelevano versioni diversamente ritardate del segnale d'ingresso; esse moltiplicate per i coefficienti  $h_k$  entrano nel sommatore in basso.



Da esso viene prelevata l'uscita  $y(t)$  che viene posta in ingresso alla catena di retroazione, simile alla prima ma con  $M$  elementi di ritardo ed  $M$  prese, da cui vengono prelevate le versioni diversamente ritardate dell'uscita. Una volta moltiplicate per i coefficienti  $h'_k$  vengono anche esse inserite nel sommatore.

$$y(t) = \sum_{k=0}^{N-1} h_k \times (t - kT) + \sum_{k=1}^M h'_k y(t - kT)$$

Applicando la trasformata di Fourier ad entrambi i membri e ricordando la trasformata di un segnale ritardato si ha:

$$\begin{aligned} Y(\omega) &= \sum_{k=0}^{N-1} h_k X(\omega) e^{-j\omega kT} + \sum_{k=1}^M h'_k Y(\omega) e^{-j\omega kT} = \\ &= X(\omega) \sum_{k=0}^{N-1} h_k e^{-j\omega kT} + Y(\omega) \sum_{k=1}^M h'_k e^{-j\omega kT} \end{aligned}$$

Riordinando è immediato ricavare la funzione di trasferimento come rapporto delle trasformate:

$$H(\omega) = \frac{Y(\omega)}{X(\omega)} = \frac{\sum_{k=0}^{N-1} h_k e^{-j\omega kT}}{1 - \sum_{k=1}^M h'_k e^{-j\omega kT}}$$

La funzione di trasferimento dipende dal ritardo  $T$ ; dalle lunghezze delle linee dirette e di retroazione,  $N-1$ ,  $M$ ; dai valori dei coefficienti.

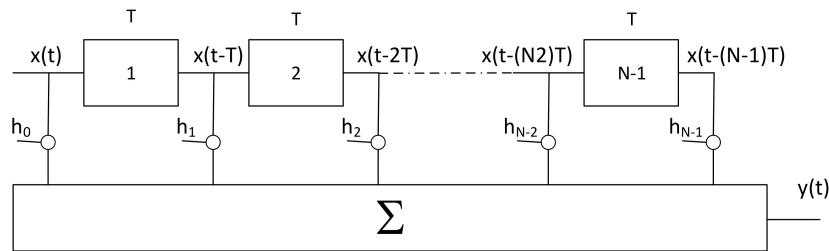
Al variare di tutti questi parametri è possibile fare assumere alla funzione di trasferimento andamenti diversi, mantenendo inalterato lo schema.

La complessità del filtro dipende invece dal numero di prese, cioè dalla lunghezza delle linee di ritardo.

La funzione di trasferimento è periodica nelle pulsazioni con periodo  $\omega_0 = \frac{2\pi}{T}$

## FILTRI TRASVERSALI

Nei filtri trasversali manca il ramo di retroazione, ovvero i coefficienti  $h_k$  sono tutti nulli.



La relativa funzione di trasferimento è:

$$H(\omega) = \sum_{k=0}^{N-1} h_k e^{-j\omega kT}$$

## FILTRI PURAMENTE RICORRENTI

In essi manca la catena diretta e l'ingresso viene portato direttamente al sommatore, cioè  $h_0=1$  mentre sono tutti nulli gli altri coefficienti  $h_k$ .

La relativa funzione di trasferimento è:

$$H(\omega) = \frac{1}{1 - \sum_{k=1}^N h'_k e^{-j\omega kT}}$$

## CRITERI DI PROGETTO DI FILTRI TRASVERSALI A COEFFICIENTI SIMMETRICI

Il progetto dei filtri ricorrenti non verrà esaminato in dettaglio. Tuttavia, per dare almeno un'indicazione di massima considereremo qui un caso particolare, quello di filtri trasversali a coefficienti simmetrici, impiegati per la realizzazione di un filtro passa-basso.

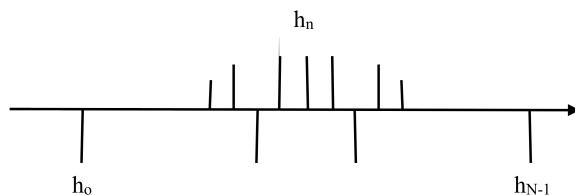
Si tratta di rielaborare la  $H(\omega)$ , già nota, per metterla in una forma che aiuti a comprendere i criteri di progetto:

$$h_k = h_{N-1-k} \quad k = 0, 1, \dots, N-1$$

Consideriamo per semplicità solo il caso di  $N$  dispari (esiste un coefficiente centrale).

$$n = \frac{N-1}{2} \quad h_{n-r} = h_{n+r}$$

$$r = 1, 2, \dots, n$$



Conviene quindi estrarre dalla funzione di trasferimento l'esponenziale  $e^{-j\omega nT}$  e scrivere  $N-1$  come  $2n$ :

$$H(\omega) = \sum_{k=0}^{N-1} h_k e^{-j\omega kT} = e^{-j\omega nT} \sum_{k=0}^{2n} h_k e^{-j\omega(k-n)T}$$

Si cambia indice della sommatoria, ponendo  $n = k - r$  per renderla simmetrica; quindi si procede in modo simile a quanto fatto per introdurre la II forma dello sviluppo in serie di Fourier,

$$H(\omega) = e^{-j\omega nT} \sum_{r=-n}^n h_{n+r} e^{-j\omega rT} = e^{-j\omega nT} \left( h_n + \sum_{r=1}^n 2h_{n+r} \cos \omega rT \right)$$

Ponendo

$$G(\omega) = h_n + \sum_{r=1}^n 2h_{n+r} \cos \omega rT \quad \longrightarrow \quad \text{Si ha infine la forma voluta:}$$

$$H(\omega) = G(\omega) e^{-j\omega nT}$$

$G(\omega)$  è funzione reale, pari e periodica con periodo  $2\alpha / T$

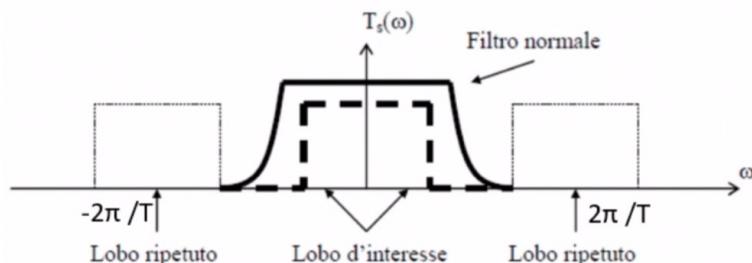
Il suo modulo rappresenta la caratteristica di ampiezza; la caratteristica di fase, quando  $G(\omega) > 0$ , è proporzionale a  $\omega$ .

Il progetto dovrà fare in modo che  $G(\omega)$  sia circa costante nelle bande passanti e circa zero in quelle attenuate.

In realtà, a causa della periodicità di  $G(\omega)$ , le bande passanti desiderate si ripetono con periodo  $2\pi/T$ , dando origine a bande passanti spurie (si veda l'esempio nella figura sotto, riferito ad un passa-basso).

Possono essere eliminate con un filtro passa-basso reale tipo RC.

Il taglio ripido è a carico del trasversale!



### PROCEDIMENTO SUB-OTTIMO PER LA REALIZZAZIONE DI UN PASSA-BASSO

Esistono algoritmi ottimali che, date le specifiche di tolleranza, realizzano l'andamento voluto della  $G(\omega)$  con andamenti di tipo oscillatorio attorno ai valori desiderati 1 e 0.

Un procedimento sub-ottimo viene suggerito immediatamente dall'espressione della  $G(\omega)$ . Se fosse  $n = \infty$  questa relazione altro non sarebbe che lo sviluppo in serie di Fourier di una funzione periodica e pari (sviluppo in soli coseni, seconda forma).

Lo sviluppo in questione ci darebbe quindi i coefficienti cercati:

$$h_n = \frac{a_0}{2} \quad h_{n+r} = h_{n-r} = \frac{a_r}{2}$$

Con infiniti termini sarebbe possibile realizzare qualunque caratteristica di ampiezza, anche ideale, con discontinuità in corrispondenza alle pulsazioni di taglio.

Ciò richiederebbe però un filtro di lunghezza infinita, non fisicamente realizzabile, che peraltro introdurrebbe anche un ritardo infinito.

Ovviamente occorre quindi limitare il numero dei termini.

Un procedimento sub-ottimo per realizzare, nell'ambito di una prefissata approssimazione, una qualunque caratteristica di ampiezza potrebbe allora essere il seguente: dapprima si sviluppa in serie di Fourier la caratteristica assegnata, quindi si provvede poi a troncare tale sviluppo in modo da mantenere un numero di termini sufficiente ad approssimare tale caratteristica secondo le specifiche stabilite.

La lunghezza del filtro risulta superiore a quella a cui condurrebbero procedimenti ottimali.

## RISPOSTA IMPULSIVA DI UN FILTRO TRASVERSALE

La risposta impulsiva di un filtro trasversale non esiste nell'ambito delle funzioni ordinarie. Infatti, ponendo in ingresso al filtro  $\delta(t)$  anziché un generico  $x(t)$  si ottiene immediatamente la risposta impulsiva

$$h(t) = \sum_{k=0}^{N-1} h_k \delta(t - kT)$$

Essa risulta quindi formata da un pettine di N delta di Dirac.



Si noti che detta risposta impulsiva, essendo costituita da un numero finito di funzioni impulsive, ha durata finita. Per questo motivo si usa anche la denominazione di filtri FIR (Finite Impulse Response) per indicare i filtri trasversali.

I filtri ricorrenti presentano invece come risposta impulsiva una successione di infinite funzioni impulsive, e quindi con durata infinita: da qui la denominazione di filtri IIR (Infinite Impulse Response).

**CAPITOLO 4 - ANALISI DI FOURIER DI SEGNAI TEMPO-DISCRETI****TRASFORMATURA DI FOURIER DI SERIE TEMPORALE**

Le funzioni tempo discrete sono anche chiamate **serie temporali** (se formate da un numero infinito di termini). Esse possono rappresentare segnali che hanno già origine in tale forma oppure essere ottenute da una funzione tempo continua mediante lettura dei valori da essa assunti in istanti che si succedono con un intervallo T.

Questa operazione è detta **campionamento** e la corrispondente funzione tempo discreta è anche chiamata **funzione campionata**.

Consideriamo ora la serie temporale:

$$\{x_n\} = \{\dots, x_{-2}, x_{-1}, x_0, x_1, x_2, \dots\}$$

i cui elementi si succedono con intervallo T sull'asse dei tempi e possono essere sia continui che discreti nei valori.

In modo analogo a quanto fatto per le funzioni tempo continue, sotto opportune condizioni, si può definire la trasformata di Fourier della serie temporale mediante la seguente relazione:

$$X_s(\omega) = \sum_{n=-\infty}^{\infty} x_n e^{-jn\omega T}$$

La formula di antitrasformazione, che dalla trasformata  $X_s(\omega)$  permette di ritornare agli elementi  $x_n$  della serie è la seguente:

$$x_n = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} X_s(\omega) e^{jn\omega T} d\omega \quad n = \dots, -2, -1, 0, 1, 2, \dots$$

**DIMOSTRAZIONE DELLA FORMULA DI ANTITRASFORMAZIONE**

La trasformata di una serie è una funzione periodica con periodo  $\omega_p = \frac{2\pi}{T}$

$$X_s(\omega) = X_s(\omega + \omega_p)$$

Essa può essere espressa mediante uno sviluppo in serie di Fourier di tipo esponenziale. A tale scopo occorre prestare la massima attenzione al significato dei simboli, in quanto la funzione periodica è in  $\omega$  e non in t.

$$X_s(\omega) = \sum_{n=-\infty}^{\infty} c_n e^{+jn\frac{2\pi}{\omega_p}\omega} = \sum_{n=-\infty}^{\infty} c_n e^{+jn\frac{2\pi}{T}\omega} = \sum_{n=-\infty}^{\infty} c_n e^{+jn\omega T}$$

$$c_n = \frac{1}{\omega_p} \int_{-\frac{\omega_p}{2}}^{+\frac{\omega_p}{2}} X_s(\omega) e^{-jn\frac{2\pi}{\omega_p}\omega} d\omega = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} X_s(\omega) e^{-jn\omega T} d\omega$$

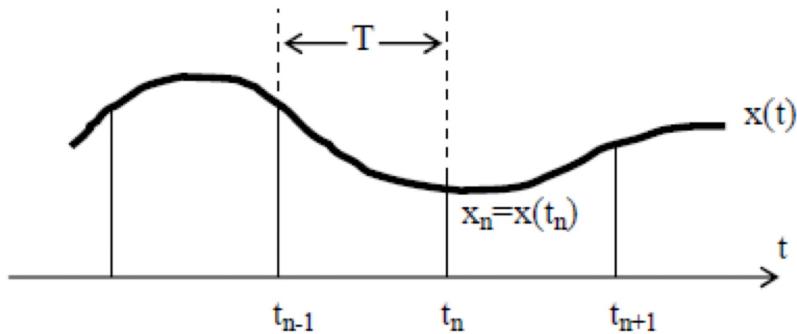
Confrontando lo sviluppo in serie con la definizione di trasformata, si ottiene la seguente relazione, che dimostra la formula di antitrasformazione:

$$x_n = C_{-n}$$

CAMPIONAMENTO

L'operazione di campionamento consiste nel leggere i valori di una funzione tempo continua ad intervalli regolari.

Converte una funzione tempo continua in una serie temporale.



I valori  $x_n = x(t_n)$  sono detti valori campionati, gli istanti di lettura  $t_n$  istanti di campionamento, l'intervallo costante  $T$  che li separa è detto intervallo di campionamento e il suo inverso,  $1/T = f_0$ , frequenza di campionamento.

Senza perdere la generalità possiamo porre  $t_n = nT$  (a tal fine basta assumere l'origine dei tempi in uno degli istanti di campionamento) ottenendo la seguente serie temporale:

$$x_n = x(nT) \quad n = \dots, -2, -1, 0, 1, 2, \dots$$

RELAZIONE FRA TRASFORMATE

Fra la trasformata di una serie temporale ottenuta per campionamento e la trasformata della funzione campionata esiste la seguente importantissima relazione:

$$X_s(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X(\omega + k\omega_0)$$

**Dimostrazione:**

si basa sulla "doppia natura" dei campioni ed in particolare sul confronto fra le due seguenti formule di antitrasformazione.

$$x_n = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} X_s(\omega) e^{j\omega nT} d\omega \quad X(nT) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega nT} d\omega$$

La dimostrazione procede manipolando la seconda formula per rendere l'integrale che in essa compare simile a quello presente nella prima formula.

- 1 passaggio:** spezzo l'integrale in infiniti integrali di ampiezza uguale a quello della prima formula

$$X(nT) = \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \int_{-\frac{\pi}{T} + k\omega_0}^{+\frac{\pi}{T} + k\omega_0} X(\omega) e^{j\omega nT} d\omega \quad \omega_0 = \frac{2\pi}{T}$$

- **2 passaggio:** per ognuno degli infiniti integrali effettuo il cambiamento di variabile

$$\xi = \omega - k\omega_0 \quad \text{ottenendo una somma di integrali tutti definiti sullo stesso intervallo di integrazione della prima formula}$$

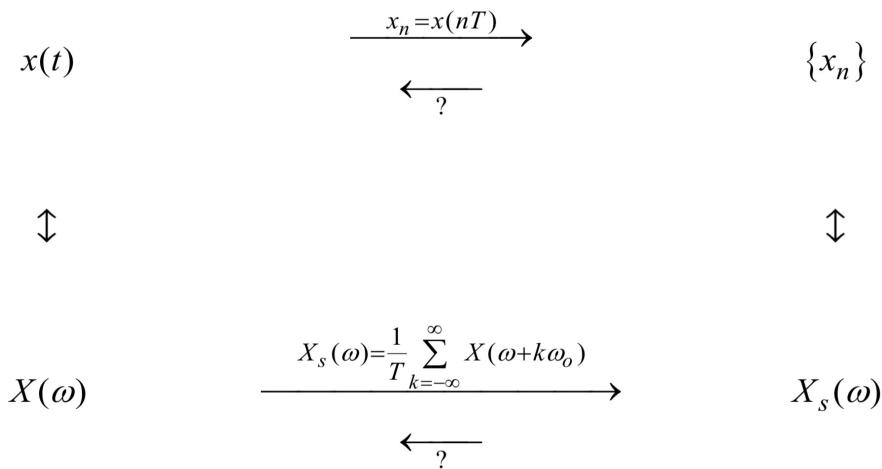
$$\begin{aligned} x(nT) &= \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} X(\xi + k\omega_0) e^{j\xi nT} e^{jk\omega_0 nT} d\xi = \\ &= \frac{1}{2\pi} \sum_{k=-\infty}^{\infty} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} X(\xi + k\omega_0) e^{j\xi nT} d\xi \end{aligned}$$

- **3 passaggio:** chiamando nuovamente  $\omega$  la variabile di integrazione, e portando la sommatoria all'interno dell'integrale si ottiene

$$x(nT) = \frac{1}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} \sum_{k=-\infty}^{\infty} X(\omega + k\omega_0) e^{j\omega nT} d\omega$$

Dal confronto dell'espressione ottenuta con quello della prima formula (anti trasformazione di una serie), dato che l'uguaglianza vale qualunque sia l'intero  $n$ , discende la validità della relazione fra le trasformate.

### L'OPERAZIONE DI CAMPIONAMENTO È REVERSIBILE?



### TEOREMA DEL CAMPIONAMENTO NEL DOMINIO DEI TEMPI (SHANNON ET ALII)

Se fosse possibile passare da  $X_s(\omega)$  a  $X(\omega)$  da questa si potrebbe risalire a  $x(t)$  antitrasformando. In generale non è possibile, ma, se i termini della ripetizione non sono sovrapposti certamente sì.

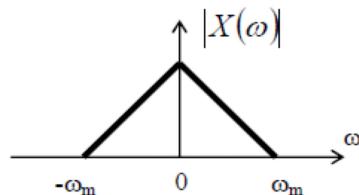
#### Enunciato del teorema di Shannon:

dato un segnale passa-basso, condizione sufficiente perché la conoscenza dei valori campionati sia equivalente alla conoscenza della funzione campionata  $x(t)$  è che la frequenza di campionamento sia maggiore del doppio della massima frequenza di  $x(t)$ .

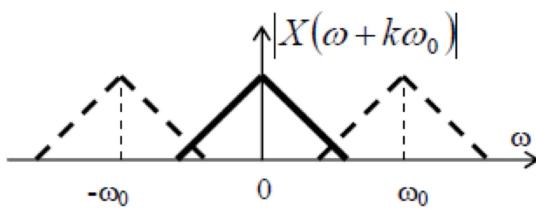
$$\omega_0 > 2\omega_m$$

**DIMOSTRAZIONE GRAFICA**

Modulo della trasformata del segnale passa-basso considerato.

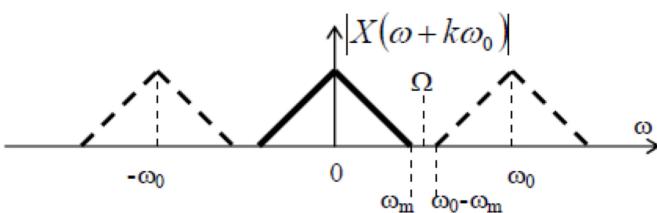


Moduli della trasformata del segnale passa-basso considerato.  
Aliasong nelle frequenze.



Moduli della trasformata del segnale passa-basso considerato.  
Assenza di aliasing.

$$\omega_0 > 2\omega_m$$

**Frequenza di Nyquist:**

È la metà della frequenza di campionamento; rappresenta il limite superiore delle frequenze di un segnale passa-basso rappresentabili con una determinata frequenza di campionamento.

**Segnali passa-banda:**

Il teorema di campionamento può essere esteso a segnali passa-banda. In questo caso, nella condizione è sufficiente sostituire alla frequenza massima del segnale la sua larghezza di banda.

**Importanza:**

Fondamentale per la conversione A/D, ovvero per la rappresentazione digitale dei segnali (CD, ecc.)

**Cenni storici:**

Formulato da Shannon nel 1949, il teorema è stato preceduto da formulazioni simili di altri autori, quindi compare sotto vari nomi. Quelle più sate sono teorema di Shannon (sul campionamento) o di Nyquist-Shannon.

## Sviluppo in serie di Shannon

Sotto la condizione sufficiente del teorema di Shannon, la conoscenza dei campioni equivale alla conoscenza della funzione campionata.

Lo sviluppo in serie ci permette di esprimere analiticamente  $x(t)$  in funzione dei suoi valori campionati  $x(nT)$ .

A tal fine, ricordando che la trasformata della serie equivale alla ripetizione periodica della trasformata del segnale, a meno di una costante moltiplicativa, e scegliendo la frequenza di Nyquist, per isolare il termine centrale della ripetizione, abbiamo

$$X(\omega) = \begin{cases} TX_s(\omega) & |\omega| < \omega_0/2 \\ 0 & \text{ALTROVE} \end{cases}$$

L'espressione sopra deve essere inserita nella formula di antitrasformazione del segnale  $x(t)$ :

$$x(t) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X(\omega) e^{j\omega t} d\omega = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} X_s(\omega) e^{j\omega t} d\omega$$

Ricordando la definizione di trasformata di Fourier di una serie, sostituendo, e scambiando l'ordine di serie ed integrale, si ottiene:

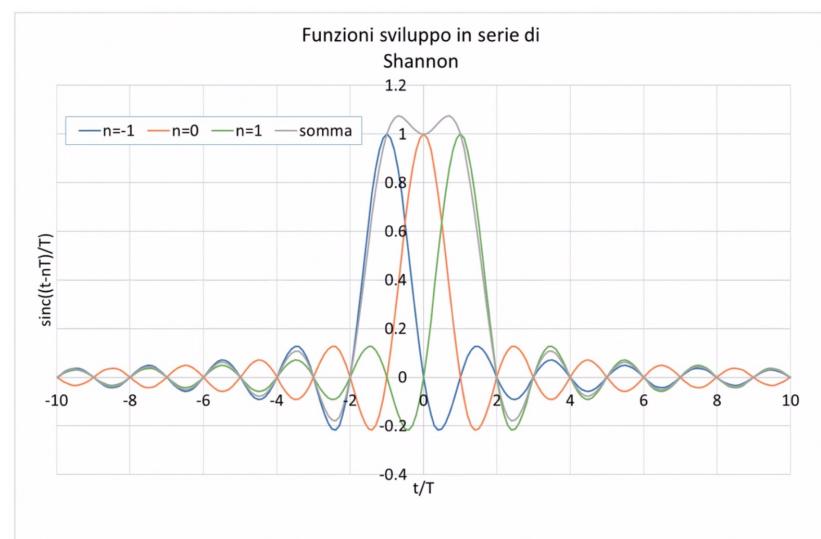
$$x(t) = \frac{T}{2\pi} \sum_{n=-\infty}^{\infty} x_n \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} e^{j\omega t} e^{-jn\omega T} d\omega$$

Con semplici passaggi (si veda l'esercizio sull'antitrasformata dell'impulso nelle frequenze) si arriva allo sviluppo in serie di Shannon:

$$x(t) = \sum_{n=-\infty}^{\infty} x_n \frac{\sin \frac{\pi}{T} (t - nT)}{\frac{\pi}{T} (t - nT)} = \sum_{n=-\infty}^{\infty} x_n \operatorname{sinc}\left(\frac{t - nT}{T}\right)$$

Le funzioni che compaiono nello sviluppo in serie sono fra loro ortogonali. I coefficienti dello sviluppo in serie sono quindi i valori campionati.

- Se cambio un campione non posso riottenere lo stesso segnale  $x(t)$  cambiando gli altri campioni;
- La funzione  $\operatorname{sinc}(t/T)$  vale 1 nell'origine e zero per  $t=nT$  con  $n \neq 0$ ;
- Segue che per  $t=nT$  solo la  $n$ -esima funzione è diversa da zero, e vale 1.



## ALCUNE PROPRIETÀ DELLA TRASFORMATA DI UNA SERIE TEMPORALE

### • SERIE RITARDATA

SE  $F[\{x_n\}] = X_s(\omega)$ , ALLORA

$$F[\{x_{n-m}\}] = X_s(\omega) e^{-j\omega mT} \longrightarrow$$

Si noti che la formula è la stessa delle funzioni tempo continue dato che a m posizioni corrisponde il ritardo mT.

### • CONVOLUZIONE FRA SERIE TEMPORALI

Date due serie temporali, la loro convoluzione definisce una nuova serie temporale i cui termini sono espressi da:

$$z(t) = \int_{-\infty}^{+\infty} x(\tau) y(t - \tau) d\tau \quad z_n = \sum_{i=-\infty}^{\infty} x_i y_{n-i}$$

Anche in questo caso l'analogia con il caso tempo continuo è ampia.

La trasformata del prodotto di convoluzione è come nel caso continuo data dal prodotto delle trasformate:

$$Z_s(\omega) = X_s(\omega) Y_s(\omega)$$

La dimostrazione è la stessa del caso tempo continuo, cambia solo la notazione:

$$\begin{aligned} z_n &= \sum_{i=-\infty}^{\infty} x_i y_{n-i} = \sum_{i=-\infty}^{\infty} x_i \underbrace{\frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} Y_s(\omega) e^{j(n-i)\omega T} d\omega}_{\hookrightarrow y_{n-i}} = \\ &= \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} Y_s(\omega) \sum_{i=-\infty}^{\infty} x_i e^{-j i \omega T} e^{j n \omega T} d\omega = \frac{T}{2\pi} \int_{-\frac{\pi}{T}}^{+\frac{\pi}{T}} Y_s(\omega) X_s(\omega) e^{j n \omega T} d\omega \end{aligned}$$

### • CONVOLUZIONE FRA UNA SERIE TEMPORALE ED UNA FUNZIONE TEMPO-CONTINUA

Data la serie temporale  $\{x_n\}$  e la funzione tempo-continua  $g(t)$ , la loro convoluzione definisce una funzione tempo continua espressa da:

$$y(t) = \{x_n\} * g(t) = \sum_{n=-\infty}^{\infty} x_n g(t - nT)$$

La trasformata del prodotto di convoluzione è al solito il prodotto delle trasformate:

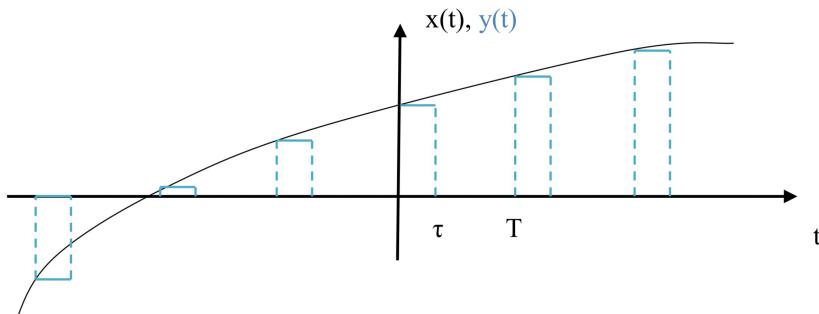
$$Y_s(\omega) = X_s(\omega) G_s(\omega)$$

**Dimostrazione:** si applica direttamente a  $y(t)$  l'operatore trasformata di Fourier.

$$\begin{aligned} Y(\omega) &= \int_{-\infty}^{+\infty} \sum_{n=-\infty}^{\infty} x_n g(t - nT) e^{-j\omega t} dt = \sum_{n=-\infty}^{\infty} x_n \int_{-\infty}^{+\infty} g(t - nT) e^{-j\omega t} dt = \\ &= G(\omega) \sum_{n=-\infty}^{\infty} x_n e^{-j\omega nT} = G(\omega) X_s(\omega) \end{aligned}$$

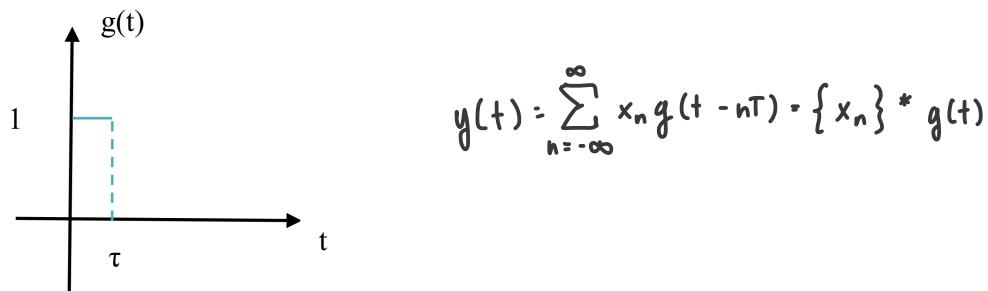
## TRASFORMATA DI UN SEGNALE PAM OTTENUTO DA UNA SERIE DI CAMPIONI

Si consideri la successione di impulsi rettangolari ottenuta campionando una funzione tempo-continua  $x(t)$  con intervallo  $T$  e mantenendo i valori campionati per un intervallo  $\tau$  con  $\tau < T$ .



La successione di impulsi in esame, aventi intervallo di ripetizione e durata costanti, ma ampiezze variabili, costituisce una successione di impulsi modulata in ampiezza e viene denominata segnale PAM (Pulse Amplitude Modulation).

Il segnale PAM in questione può essere visto come convoluzione fra la serie temporale e l'impulso rettangolare  $g(t)$ , di ampiezza unitaria e durata  $\tau$ .



Poiché l'operatore trasformata di Fourier trasforma i prodotti di convoluzione in prodotti semplici e ricordando il legame fra trasformata di una serie ottenuta per campionamento e trasformata della funzione campionata si ottiene:

$$Y(\omega) = X_s(\omega) G(\omega) = \frac{1}{T} \sum_{n=-\infty}^{\infty} X(\omega + n\omega_0) G(\omega)$$

Si noti infine che per esprimere la trasformata  $G(\omega)$  basta ricordare l'esercizio sulla trasformata di un impulso rettangolare:

$$G(\omega) = \tau \frac{\sin(\omega\tau/2)}{\omega\tau/2} e^{-j\frac{\omega\tau}{2}}$$

CONVERSIONE ANALOGICO DIGITALE TECNICA PCM

La conversione di un segnale analogico in un segnale digitale (A/D), ed in particolare in una serie di bit, è di importanza fondamentale non solo nei sistemi di telecomunicazione moderni, ma anche nell'industria discografica, in quella fotografica, nella realizzazione di strumenti di misura e in moltissimi altri campi.

I vantaggi dei segnali digitali sono molteplici, ma forse il più importante è la molto maggiore facilità di elaborazione e di memorizzazione, attuabile con tecniche di tipo informatico, cioè con algoritmi operanti su hardware generico o su appositi processori (DSP, Digital Signal Processor) e su supporti dello stesso tipo (CD, DVD, HD, SSD...).

Le tecniche di conversione analogico digitale sono diverse; qui esamineremo quella di base, detta anche PCM (Pulse Code Modulation) per motivi storici.

Da un punto di vista logico la conversione analogico-digitale PCM prevede tre passaggi: campionamento, quantizzazione e codifica, riassunti nello schema a blocchi della figura sotto.

1) CAMPIONAMENTO

Trasforma il segnale tempo continuo in un segnale tempo-discreto, cioè in una serie temporale.

Il teorema di Shannon ci assicura che non si ha perdita di informazione, cioè l'operazione è reversibile, se la frequenza di campionamento è superiore al doppio della massima frequenza  $f_m$  a cui è apprezzabile lo spettro del segnale stesso.

$$f_0 > 2f_m$$

La condizione sopra è condizione sufficiente per evitare l'aliasing nel dominio delle frequenze e per poter quindi ricostruire il segnale originario. Il rispetto della condizione può avvenire in due modi: o aumentando la frequenza di campionamento fino a soddisfare la condizione (con un certo margine, necessario per l'operazione inversa), oppure, nel caso in cui la frequenza di campionamento sia fissa, anteponendo al campionatore un filtro passa-basso in grado di ridurre la frequenza massima del segnale.

## 2) QUANTIZZAZIONE

Trasforma la serie dei campioni, continua nei valori, nella serie dei valori quantizzati, discreta nei valori.

L'operazione non è reversibile, in quanto non è possibile risalire da un valore quantizzato al campione che l'ha generato.

Assumendo  $x(t)$  bilanciato, con valori compresi nell'intervallo  $[-M, M]$ , i campioni risulteranno anch'essi compresi in detto intervallo, potendo assumere qualsiasi valore all'interno di esso. Per poter procedere, è necessario ridurre il numero dei valori da infinito a finito, passando da una serie temporale continua nei valori ad una discreta nei valori.

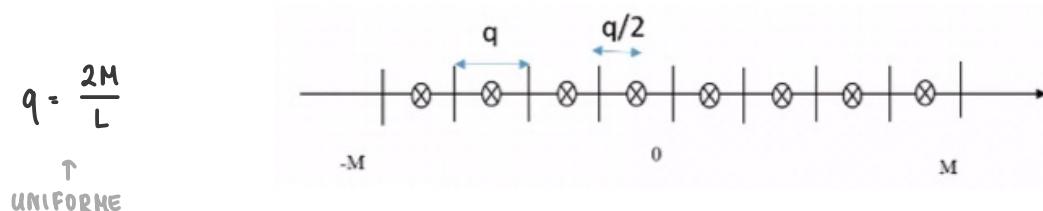
L'operazione viene detta **quantizzazione**, e consiste in una **approssimazione dei valori ottenuti**. L'intervallo di variabilità  $[-M, M]$  dei valori campionati viene suddiviso in un numero finito  $L$  di intervalli (intervalli di quantizzazione) e tutti i valori interni a ciascuno di questi vengono identificati con uno di essi, che indichiamo con  $q_n$ .

L'intervallo su cui opera il quantizzatore,  $[-M, M]$  viene detto "**dinamica del quantizzatore**", ed è opportuno che coincida con quella del segnale.

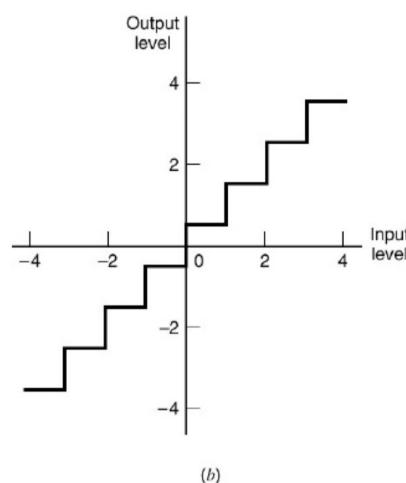
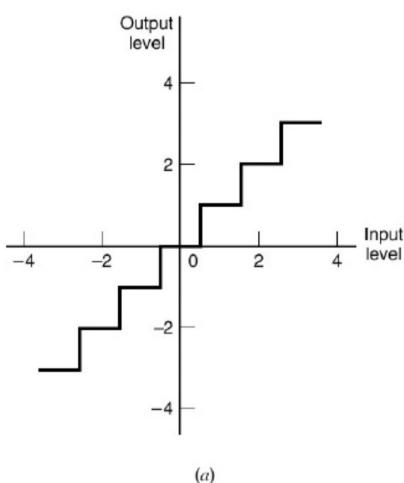
Se la dinamica del quantizzatore è suddivisa in intervalli uguali, il quantizzatore si dice uniforme, altrimenti non-uniforme.

La differenza fra campione e valore quantizzato corrispondente si dice "errore" di quantizzazione  $e_n = x_n - q_n$

Il valore rappresentativo di ogni intervallo è arbitrario, ma spesso viene scelto per simmetria il valore centrale; minimizza l'errore massimo ( $q/2$ , con  $q$  la larghezza dell'intervalle di quantizzazione).



Two types of quantization: (a) midtread and (b) midrise.



La rappresentazione in due dimensioni mette in evidenza il legame funzionale fra campioni e valori quantizzati, detto "**legge di quantizzazione**"

$$q_n = f(x_n)$$

I valori assumibili dalla variabile  $q_n$  vengono anche detti "livelli" di quantizzazione, per motivi ovvi guardando l'andamento a scala della funzione di quantizzazione.

La scelta del quantizzatore ottimo, cioè della miglior legge di quantizzazione, intendendo per migliore quella che minimizza gli effetti dell'errore di quantizzazione, dipende dalla statistica dei campioni (*teoria dell'informazione*).

Per la sua semplicità, viene spesso utilizzato il quantizzatore uniforme, anche quando non ottimo. Risulta evidente che maggiore è il numero di livelli  $L$ , minore sarà in generale l'errore di quantizzazione.

L'aumento di  $L$  comporta un aumento dei bit necessari a rappresentare il segnale, per cui si impone un compromesso fra qualità e numero di bit.

### 3) CODIFICA

Trasforma la serie dei valori quantizzati in una serie di bit.

La codifica associa ad ognuno degli  $L$  livelli di quantizzazione una parola formata da un certo numero  $l$  di bit, in modo da avere una corrispondenza biunivoca fra valori ed "etichette" binarie.

Non considerando le codifiche entropiche per semplicità, si assume che tutte le etichette siano formate dallo stesso numero di bit,  $l$ , che dovrà essere quindi messo in relazione al numero di livelli  $L$ .

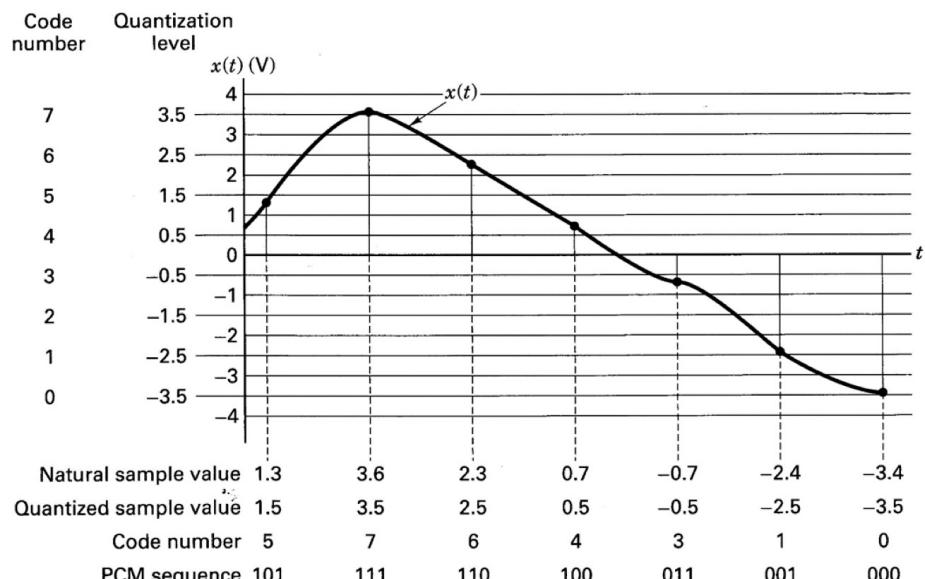
$$l \geq \log_2 L \quad \text{TEMPO DI BIT} \rightarrow T_b = \frac{T}{l} \quad \text{FREQUENZA DI BIT} \rightarrow f_b = \frac{1}{T_b} = f_0 l$$

**ESEMPI** • Nel caso di segnale telefonico la banda netta è di 300-3400 Hz (standard europeo). Gli standard richiedono  $f_0=8000$  Hz e prevedono 255 livelli ( $=2^8-1$ ) di quantizzazione (quantizzatore non uniforme "midstep") cioè 8 bit per campione. La frequenza di bit risulta  $f_b=64000$  bit/s.

Nel caso di segnale CD (mono) lo standard richiede  $f_0=44100$  Hz e 65536 livelli di quantizzazione (quantizzatore uniforme "midriser") cioè 16 bit per campione. La frequenza di bit risulta  $f_b=705600$  bit/s, cioè più di un ordine di grandezza superiore rispetto al caso telefonico. Ovviamente la qualità, sia in termini di banda riproducibile, sia in termini di rapporto segnale rumore di quantizzazione è molto più elevata.

La **conversione A/D** appena vista prende il nome di **PCM, Pulse Code**

**Modulation** perché in origine era associata alla creazione di un segnale PAM binario (con impulso rettangolare RZ o NRZ, vedi in seguito) a partire dalla serie di bit ottenuta nella conversione A/D.

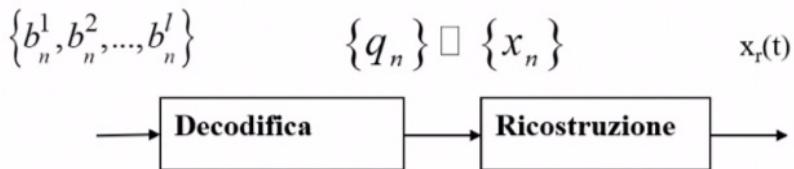


## CONVERSIONE DIGITALE ANALOGICA (D/A)

La conversione digitale analogica (D/A) consiste nella ricostruzione del segnale originario  $x(t)$  a partire dal corrispondente messaggio numerico, cioè dai bit che lo rappresentano.

Essa prevede due soli passi: **decodifica e ricostruzione del segnale**.

Si noti che la quantizzazione non ha inverso, essendo irreversibile.



### DECODIFICA

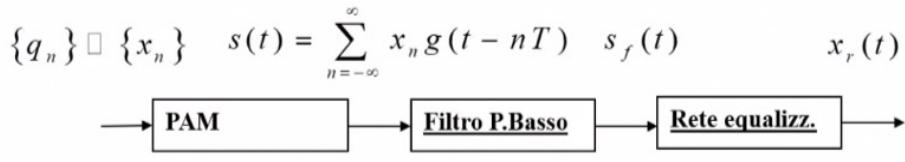
È ovviamente il processo inverso della codifica, mediante il quale si ricostruiscono i valori quantizzati a partire dalle parole di codice.

Essendo la quantizzazione non reversibile, la serie dei valori quantizzati deve necessariamente essere trattata come se fosse la serie dei valori campionati.

L'errore di quantizzazione farà però sì che il segnale ricostruito differisca da quello originale: la differenza prenderà il nome di rumore di quantizzazione:

$$e(t) = x_r(t) - x(t)$$

### RICOSTRUZIONE DEL SEGNALE

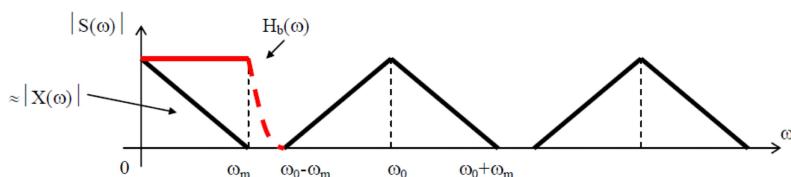


- GENERAZIONE DEL SEGNALE PAM

$$s(t) = \{x_n\} * g(t) = \sum_{n=-\infty}^{\infty} x_n g(t - nT)$$

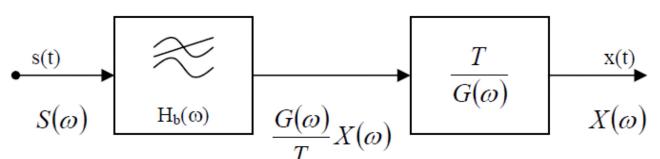
- FILTRATURA PASSA-BASSO ( $\omega_t = \omega_m$ )

$$S(\omega) = X_s(\omega)G(\omega) = \frac{1}{T} \sum_{k=-\infty}^{\infty} X(\omega + k\omega_0)G(\omega) \quad S_f(\omega) = X(\omega) \frac{G(\omega)}{T}$$



- EQUALIZZAZIONE

$$H_e(\omega) = \begin{cases} \frac{T}{G(\omega)} & |\omega| \leq \omega_m \\ \text{qualsiasi} & |\omega| > \omega_m \end{cases}$$



## RAPPORTO SEGNALE RUMORE QUANTIZZATORE UNIFORME (NON NECESSARIA PER L'ESAME)

L'ipotesi di partenza è che i valori campionati siano uniformemente distribuiti all'interno di ogni intervallo di quantizzazione. Ciò è certamente verificato se:

- la densità di probabilità  $p(x_n)$  è uniforme nell'intervallo  $[-M, M]$ ;
- in modo approssimato, se sono soddisfatte entrambe le condizioni seguenti:
  - gli intervalli sono piccoli (ovvero  $L$  è grande);
  - la distribuzione  $p(x_n)$ , è regolare all'interno di  $[-M, M]$ .

Nel caso la potenza del rumore di quantizzazione è data da:

$$E[e^2] = \int_{-q/2}^{q/2} e^2 p(e) de = \frac{1}{q} \int_{-q/2}^{q/2} e^2 de = \frac{q^2}{12}$$

Il rapporto segnale **rumore di picco** è dato da:  $SQNR_{\max} = \frac{M^2}{E[e^2]} = \frac{12M^2}{q^2} = \frac{12M^2}{4M^2} = 3 \cdot 2^{2b}$

Il rapporto segnale **rumore medio** è dato da:

$$SQNR = \frac{E[x^2]}{E[e^2]} = \frac{M^2}{F_c E[e^2]} = \frac{1}{F_c} 3 \cdot 2^{2b} \quad F_c = \frac{M^2}{E[x^2]}$$

$$SQNR(dB) = 10 \log_{10}(SQNR) = 10 \log_{10}(3) + 2b \underbrace{10 \log_{10}(2)}_{\approx 3} + 10 \log_{10}(F_c) = \rightarrow \text{DECIBEL: UNITÀ LOGARITMICA}$$

Le prestazioni dipendono dalla densità di probabilità dei campioni,  $p(x)$ , attraverso il fattore di cresta  $F_c$ .

Ogni bit di quantizzazione in più o in meno determina una variazione di 6 dB del rapporto segnale rumore; questa proprietà è spesso comune anche ad altri tipi di quantizzazione.

## TRASFORMATA DI FOURIER DISCRETA (DFT) (PAGINA 62)

La trasformata di Fourier discreta (**DFT** = Discrete Fourier Transform), si applica non più a una serie di infiniti termini ma ad una  $n$ -pla, cioè ad un vettore, costituito in generale da componenti complesse. Più precisamente la DFT stabilisce una corrispondenza biunivoca fra  $n$ -ple di numeri, in generale complessi:

$$(x_0, x_1, \dots, x_{N-1}) \xleftrightarrow{\text{DFT}} (X_0, X_1, \dots, X_{N-1})$$

Si noti che i termini sono numerati a partire da 0. L'elemento  $q$ -esimo dell' $n$ -pla di arrivo è definito come (formula di trasformazione):

$$X_q = \sum_{n=0}^{N-1} x_n e^{-j \frac{2\pi}{N} nq}$$

La formula di antitrasformazione (**IDFT** = Inverse Discrete Fourier Transform), che ci restituisce un termine della  $n$ -pla di partenza a partire da quella di arrivo è la seguente:

$$x_n = \frac{1}{N} \sum_{q=0}^{N-1} X_q e^{j \frac{2\pi}{N} nq}$$

### DIMOSTRAZIONE DELLA ANTITRASFORMATA

Dal secondo membro della formula di antitrasformazione, sostituendo  $X_q$  con la sua definizione (si noti il pedice  $k$  anziché  $n$  nella sommatoria, per non confonderlo con il pedice di arrivo), cambiando l'ordine delle sommatorie, si ha:

$$\frac{1}{N} \sum_{q=0}^{N-1} X_q e^{j \frac{2\pi}{N} n q} = \frac{1}{N} \sum_{q=0}^{N-1} \underbrace{\sum_{k=0}^{N-1} x_k e^{-j \frac{2\pi}{N} k q}}_{X_q} e^{j \frac{2\pi}{N} n q} = \frac{1}{N} \sum_{k=0}^{N-1} x_k \underbrace{\sum_{q=0}^{N-1} e^{j \frac{2\pi}{N} q(n-k)}}_{X_q}$$

Esaminiamo separatamente la seconda sommatoria:

$$\sum_{q=0}^{N-1} e^{j \frac{2\pi}{N} q(n-k)} = \sum_{q=0}^{N-1} \left( e^{j \frac{2\pi}{N} (n-k)} \right)^q$$



Essa può essere vista come la somma parziale di una serie geometrica, di ragione  $x$ . Dall'analisi sappiamo che:

$$\sum_{n=0}^{N-1} x^n = \begin{cases} \frac{1-x^N}{1-x} & x \neq 1 \\ N & x = 1 \end{cases}$$

Applicando alla nostra serie troncata:

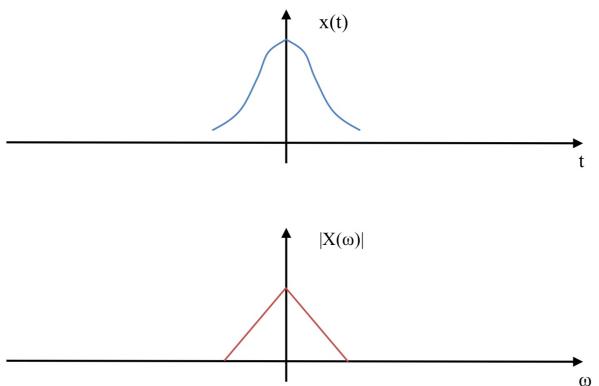
$$\sum_{q=0}^{N-1} \left( e^{j \frac{2\pi}{N} (n-k)} \right)^q = \begin{cases} \frac{1-e^{j 2\pi (n-k) N}}{1-e^{j 2\pi (n-k)}} = 0 & n \neq k \\ N & n = k \end{cases}$$

Sostituendo, si ottiene infine:

$$\frac{1}{N} \sum_{q=0}^{N-1} X_q e^{j \frac{2\pi}{N} n q} = \frac{1}{N} N x_n = x_n$$

### LEGAMETRA TRASFORMATA DI FOURIER DISCRETA E CONTINUA

La trasformata di Fourier discreta costituisce un elemento fondamentale dell'elaborazione digitale dei segnali (Digital Signal Processing). Essa può in particolare essere utilizzata per calcolare, in modo approssimato, la trasformata di Fourier di segnali tempo continui.

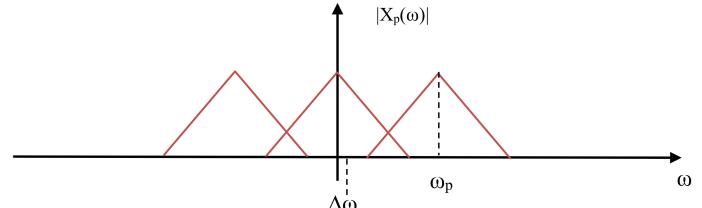
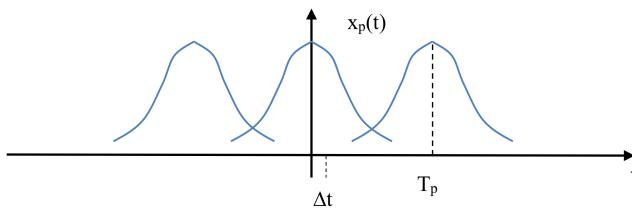


Risulterebbe comodo che prendendo N campioni di  $x(t)$  ed eseguendo la DFT si ottengessero N campioni della trasformata  $X(\omega)$ .

Il legame è tuttavia più complesso, e riguarda le ripetizioni periodiche di  $x(t)$  e di  $X(\omega)$ . Costruiamo pertanto le due ripetizioni periodiche:

$$x_p(t) = \sum_{i=-\infty}^{\infty} x(t - iT_p)$$

$$X_p(\omega) = \sum_{i=-\infty}^{\infty} X(\omega - i\omega_p)$$



I loro **periodi** sono arbitrari, ma legati dal vincolo:  $T_p \omega_p = 2\pi N$

Definisco gli **incrementi** nel tempo e nelle pulsazioni:

$$\Delta t \equiv \frac{T_p}{N} = \frac{2\pi}{\omega_p} \quad \Delta \omega \equiv \frac{\omega_p}{N} = \frac{2\pi}{T_p}$$

Si noti che risultano vincolati:  $\Delta t \Delta \omega = \frac{2\pi}{N}$

Prendiamo quindi N campioni della ripetizione nei tempi a partire dall'origine con intervallo  $\Delta t$ . Si noti che l'ultimo punto di campionamento,  $(N-1)\Delta t$  sarà appena prima del periodo, essendo  $T_p = N\Delta t$ .

$$x_n = x_p(n\Delta t) \quad n = 0, 1, \dots, N-1$$

Analogamente nelle pulsazioni:

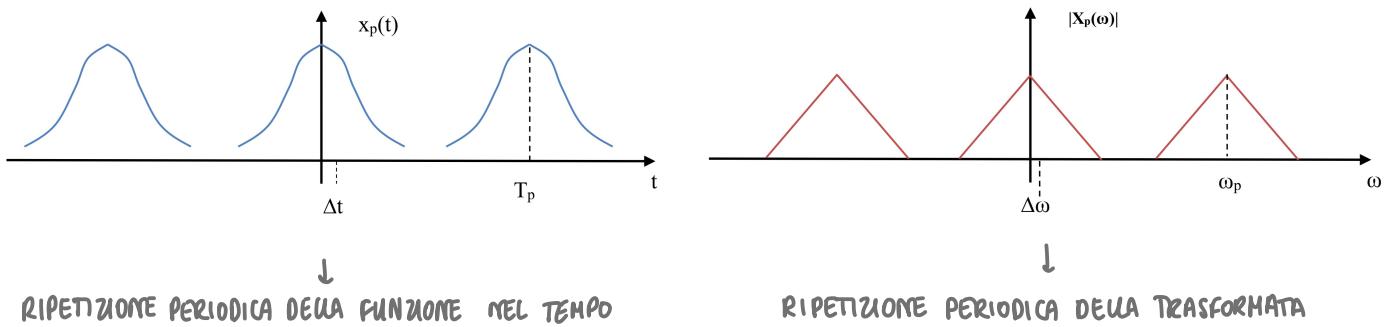
$$X_n = X_p(n\Delta \omega) \quad n = 0, 1, \dots, N-1$$

Moltiplichiamo quindi per  $\Delta t$  la n-pla nei tempi; si può dimostrare che essa è legata alla n-pla nelle frequenze dalla trasformazione discreta di Fourier:

$$(x_0 \Delta t, x_1 \Delta t, \dots, x_{N-1} \Delta t) \xleftrightarrow{\text{DFT}} (X_0, X_1, \dots, X_{N-1})$$

Il legame è quindi (a meno della costante) fra N campioni nei tempi ed N campioni nelle frequenze, non di  $x(t)$  e  $X(\omega)$ , ma delle loro ripetizioni periodiche (in particolare con i periodi di ripetizione legati fra loro dal numero di punti N).

Quando i termini delle due ripetizioni non si sovrappongono in modo significativo (N sufficientemente grande), la DFT può essere utilizzata per calcolare la trasformata continua.



DFT (DISCRETE FOURIER TRANSFORM) E FFT (FAST FOURIER TRANSFORM)

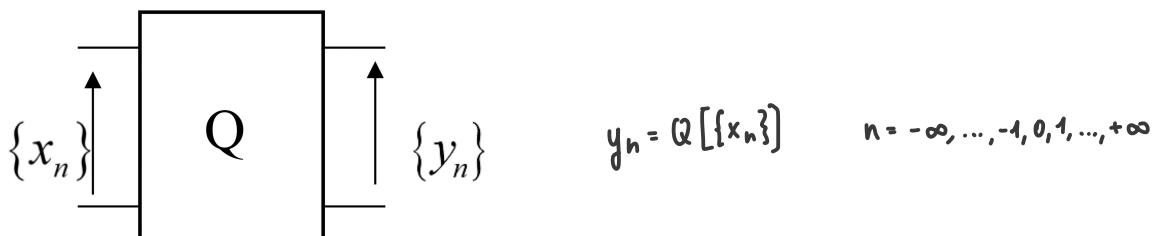
Le formule DFT e IDFT richiedono un numero di moltiplicazioni pari ad  $N^2$ . Infatti sono necessarie  $N$  moltiplicazioni per ognuno degli  $N$  elementi della  $n$ -pla di arrivo. Poiché il tempo di calcolo è praticamente determinato da tali moltiplicazioni, possiamo dire che esso cresce proporzionalmente ad  $N^2$ .

Tuttavia, nel caso di  $N$  potenza di 2, è possibile sfruttare alcune simmetrie nel calcolo ed ottenere lo stesso risultato della DFT con un tempo di calcolo proporzionale a  $N$ , conseguendo una fondamentale riduzione.

Gli algoritmi che permettono di conseguire tale risultato sono molteplici: ad essi viene attribuita la denominazione di trasformata di Fourier veloce (FFT = Fast Fourier Transform). Il campo di applicazione della FFT è vastissimo.

SISTEMI LINEARI TEMPO-DISCRETI (PAGINA 93)

La definizione dei sistemi lineari tempo discreti ricalca quella dei sistemi lineari tempo continui, a meno della simbologia.



Il generico termine  $y_n$  dipende in generale dall'intera serie d'ingresso. Nel caso in cui dipenda solo dai valori passati e presente, cioè se  $y_n$  è indipendente da  $x_m$  per  $m > n$  il sistema si dice **causale**, in quanto esiste una relazione causa effetto fra ingresso ed uscita, come sempre accade nei sistemi fisicamente realizzabili.

Altrimenti il sistema è **anticipativo** (possono essere anticipativi alcuni sistemi ideali). Se la dipendenza si riduce al solo valore  $x_n$  presente in ingresso, il sistema si dice **privo di memoria**.

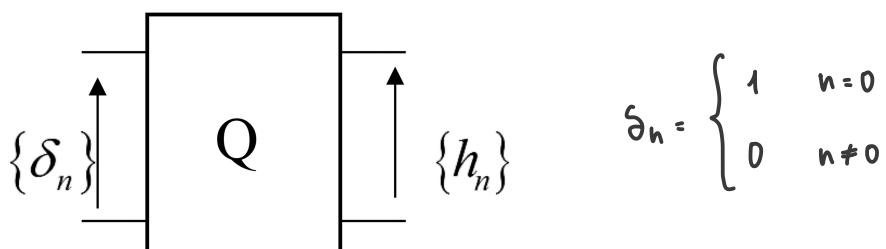
Il sistema è detto lineare se vale la "sovraposizione degli effetti":

$$Q[\{c_1 x_n^1 + c_2 x_n^2\}] = c_1 Q[\{x_n^1\}] + c_2 Q[\{x_n^2\}]$$

Il sistema è detto **tempo-invariante** se la risposta alla serie ritardata è la risposta ritardata, qualunque sia il ritardo ( $iT$ ):

$$y_{n-i} = Q[\{x_{n-i}\}]$$

La **risposta impulsiva** di un sistema discreto lineare tempo-invariante è la risposta alla serie temporale, formata da tutti "0" tranne un "1" nell'origine (corrispondente tempo discreto del Delta di Dirac):



La risposta impulsiva descrive completamente il comportamento della rete lineare. In particolare consente di esprimere la risposta ad una qualsiasi serie temporale d'ingresso rimanendo nel dominio dei tempi.

$$x_n = \sum_{i=-\infty}^{\infty} x_i \delta_{n-i} = \{x_n\} * \{\delta_n\}$$

Dalle condizioni di linearità e tempo-invarianza, in modo analogo a quanto visto per i sistemi tempo continui, discende la relazione:

$$y_n = \sum_{i=-\infty}^{\infty} x_i h_{n-i} = \{x_n\} * \{h_n\}$$

Si definisce **funzione di trasferimento** di un sistema discreto lineare tempo-invariante la trasformata secondo Fourier della risposta impulsiva discreta:

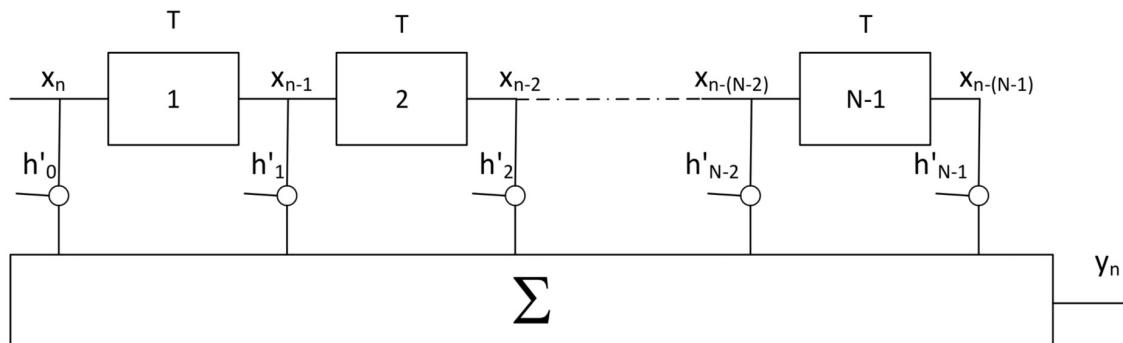
$$H_s(\omega) = F[\{h_n\}] = \sum_{n=-\infty}^{\infty} h_n e^{-jn\omega T}$$

Se le serie temporali di ingresso e di uscita sono dotate di trasformata secondo Fourier, per il teorema della convoluzione vale la relazione:

$$Y_s(\omega) = X_s(\omega) H_s(\omega)$$

### FILTRI TRASVERSALE TEMPO DISCRETO

Un filtro trasversale tempo-discreto è costituito da un registro a scorrimento (shift-register) e da un sommatore. Di fatto lo schema è lo stesso del caso tempo continuo, tuttavia in ingresso ed in uscita si hanno in questo caso delle serie temporali. Si noti che alle prese dei coefficienti è stato aggiunto un apice per evitare confusione con i termini della risposta impulsiva.



Il filtro trasversale è un sistema discreto causale con memoria finita, in quanto non essendoci rami in retroazione, il termine n-esimo della serie temporale d'uscita dipende solamente dal valore in ingresso e dagli N - 1 termini precedenti.

La relazione ingresso-uscita è data da:

$$y_n = \sum_{k=0}^{N-1} h'_k x_{n-k}$$

Risposta impulsiva:

$$\{h_n\} = \begin{cases} 0 & n < 0 \\ h'_n & 0 \leq n \leq N-1 \\ 0 & n > N-1 \end{cases}$$

Funzione di trasferimento:

$$H_s(\omega) = F[\{h_n\}] = \sum_{k=0}^{N-1} h'_k e^{-jk\omega T}$$

ESEMPIO SUI FILTRI TRASVERSALI TEMPO DISCRETI (PAGINA 95)

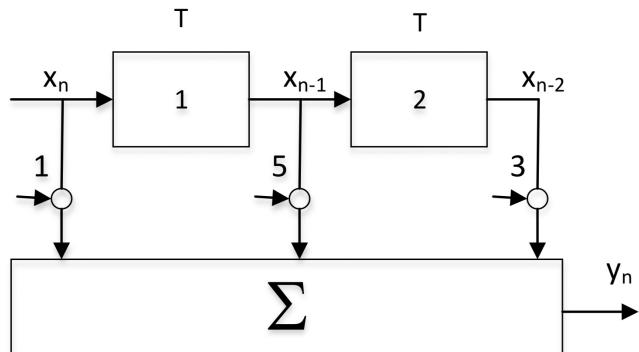
Dato un sistema lineare tempo invariante, la cui relazione ingresso-uscita è data da:

$$y_n = x_n + 5x_{n-1} + 3x_{n-2}$$

Calcolare la risposta impulsiva discreta, la funzione di trasferimento e la sequenza di uscita quando in ingresso è presente la sequenza:

$$x_n = \begin{cases} 1 & 0 \leq n \leq 5 \\ 0 & \text{ALTROVE} \end{cases}$$

Per prima cosa occorre riconoscere dalla relazione ingresso uscita che la rete è un filtro trasversale tempo-discreto



Si possono quindi richiamare i risultati già noti dalla teoria per le prime due domande. In particolare si ha:

$$\{h_n\} = \begin{cases} 0 & n < 0 \\ 1 & n = 0 \\ 5 & n = 1 \\ 3 & n = 2 \\ 0 & n > 2 \end{cases}$$

$$H_s(\omega) = \sum_{k=0}^{N-1} h_k e^{-jk\omega T} = 1 + 5e^{-j\omega T} + 3e^{-j2\omega T}$$

Le terza domanda si risolve utilizzando la relazione ingresso uscita data nel testo. Conviene scrivere una tabella con tante colonne quante prese ed una riga per ogni stato del registro a scorrimento in successione temporale.

<b>1</b>	<b>5</b>	<b>3</b>
1	0	0
1	1	0
1	1	1
1	1	1
1	1	1
1	1	1
0	1	1
0	0	1

→ Moltiplicando i termini di ciascuna riga per i coefficienti e sommando si ottiene:

$$\{y_n\} = \begin{cases} 0 & n < 0 \\ 1 & n = 0 \\ 6 & n = 1 \\ 9 & n = 2 \\ 9 & n = 3 \\ 9 & n = 4 \\ 9 & n = 5 \\ 8 & n = 6 \\ 3 & n = 7 \\ 0 & n > 7 \end{cases}$$

Si noti che la lunghezza è di 8 elementi, pari alla lunghezza della sequenza di ingresso più N-1.

## CAPITOLO 7 - INTRODUZIONE ALLA TEORIA DELLA MODULAZIONE

In alcuni casi, ed in particolare in quello radio, può capitare che si debba trasmettere un segnale su un canale di tipo passa-banda, mentre il segnale è di tipo passa-basso. Senza entrare nei dettagli, esistono almeno tre motivazioni per dover trasformare un passa-basso in un passa-banda:

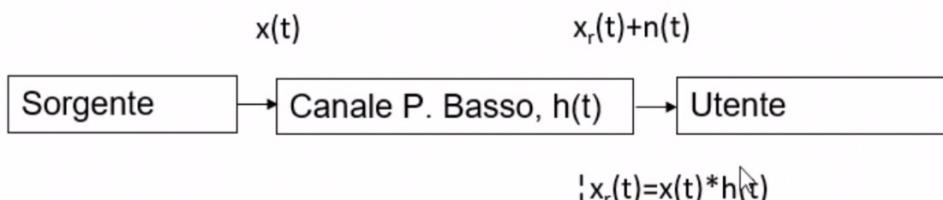
- La dimensione delle antenne deve essere confrontabile con la lunghezza d'onda. Questo implicherebbe antenne gigantesche per le basse frequenze.
- In un segnale passa-basso, il rapporto fra la frequenza massima e quella minima dello spettro è molto alto (circa 10 nel caso telefonico, circa 1000 in un segnale audio qualità CD), quando addirittura non infinito. Un'antenna progettata per lavorare su una frequenza estrema lavorerebbe malissimo sull'altra.
- Interferirebbe facilmente con altri segnali della stessa natura trasmessi contemporaneamente nelle vicinanze.

Occorre associare al segnale  $x(t)$  un segnale passa-banda, trasmissibile sul canale. Tale associazione deve naturalmente essere invertibile, e si ottiene ricorrendo alla modulazione di una oscillazione sinusoidale, a frequenza sufficientemente elevata, detta **"portante"**. Il segnale  $x(t)$  si dice allora segnale **"modulante"**, in quanto modula, cioè modifica, le caratteristiche della portante (ampiezza e/o argomento), mentre il segnale  $s(t)$  così ottenuto è detto **oscillazione "modulata"**.

Una modulazione si dice analogica o digitale a seconda che sia analogico o digitale (tipo PAM) il segnale modulante.

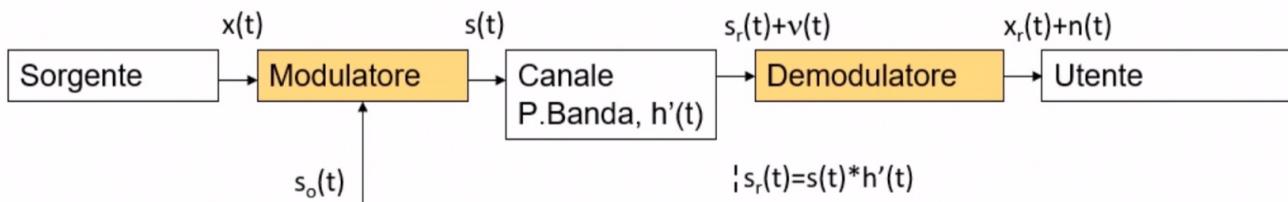
### MODelli SISTEMI TRASMISSIONE ANALOGICA:

#### Passa-basso



- Nel caso analogico il canale non deve distorcere; il segnale  $x_r(t)$  deve riprodurre indistorto  $x(t)$ .
- Si aggiunge un rumore  $n(t)$ ; occorre "rapporto segnale rumore" S/N sufficientemente elevato.

#### Passa-banda



La portante, come ogni sinusoide, è caratterizzata dai parametri ampiezza, pulsazione e fase.

$$s_0(t) = V_0 \cos(\omega_0 t - \varphi_0)$$

Per trasmettere informazione bisogna variare (modulare) uno o più dei parametri sopra menzionati. Tali variazioni debbono inoltre essere "lente" rispetto alla rapidità di variazione della portante; si ottiene in tal modo un'oscillazione sinusoidale modulata  $s(t)$  la cui espressione più generale può essere posta nella forma:

$$s(t) = V(t) \cos(\varphi(t)) \quad V(t) \geq 0$$

In relazione all'espressione sopra, si danno le seguenti definizioni:

- $V(t)$  ampiezza istantanea;
- $\varphi(t)$  fase istantanea (= intero argomento del coseno);
- $\omega(t) = \dot{\varphi}(t)$  pulsazione istantanea.

Queste definizioni sono coerenti con quelle già note di ampiezza, fase istantanea e pulsazione di una sinusoide non modulata. Infatti, riferendole alla oscillazione portante si ha:

- $V(t) = V_0$  ampiezza (costante);
- $\varphi(t) = \omega_0 t - \varphi_0$  argomento del coseno (lineare in  $t$ );
- $\omega(t) = \omega_0$  pulsazione (costante).

### DEVIAZIONI

Sono definite come differenza fra i parametri dell'oscillazione modulata ed i corrispondenti della portante; Sono cioè le differenze introdotte dalla modulazione. Si chiamano istantanee, perché dipendono da  $t$ .

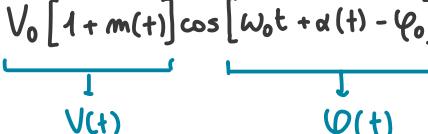
- deviazione istantanea di ampiezza:  $V(t) - V_0$
  - deviazione (istantanea) relativa di ampiezza; poiché  $V(t) \geq 0$ , segue  $m(t) \geq -1$ ;
- $$m(t) = \frac{V(t) - V_0}{V_0}$$
- deviazione istantanea di fase:  $\alpha(t) = \varphi(t) - (\omega_0 t - \varphi_0)$
  - deviazione istantanea di pulsazione:  $\Delta\omega(t) = \omega(t) - \omega_0$

Le deviazioni di fase e di pulsazione risultano legate:

$$\Delta\omega(t) = \alpha(t) \quad \alpha(t) = \int_{-\infty}^t \Delta\omega(\tau) d\tau$$

Si ottiene la (seconda) espressione dell'oscillazione modulata:

$$s(t) = V_0 [1 + m(t)] \cos [\omega_0 t + \alpha(t) - \varphi_0]$$



## PRINCIPALI MODULAZIONI ANALOGICHE

- MODULAZIONE DI AMPIEZZA (AM, AMPLITUDE MODULATION)

$$\text{AM} \left\{ \begin{array}{l} m(t) = kx(t) \\ \alpha(t) = 0 \end{array} \right.$$

Viene modificata quindi solo l'ampiezza dell'oscillazione portante (solo la modulazione in ampiezza è presente). Inoltre in AM  $kx(t) \geq -1$  a causa dell'analogico vincolo su  $m(t)$ , da cui la condizione a destra nella formula sotto.

$$s(t) = V_0 [1 + kx(t)] \cos[\omega_0 t - \varphi_0] \quad \text{con } V_0 [1 + kx(t)] \geq 0$$

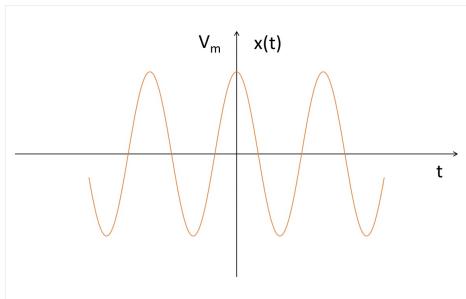
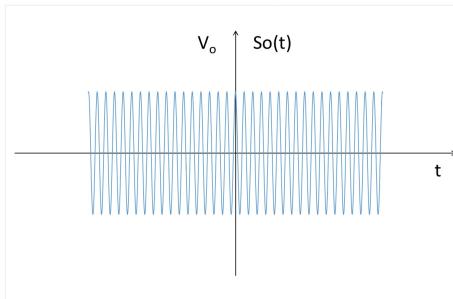
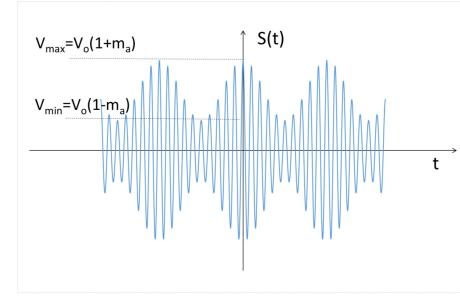
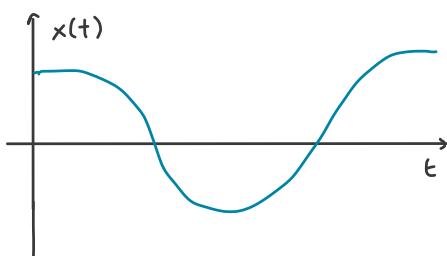
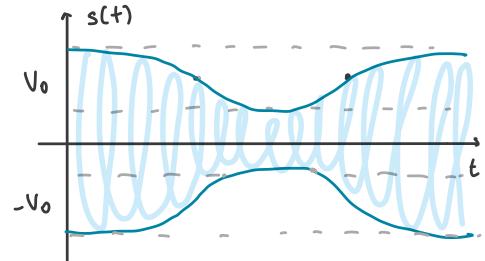


Fig. 98 Oscillazione modulante.

Fig. 99 Oscillazione portante (si noti che  $f_o >> f_m$ ).Fig. 100 Oscillazione modulata AM ( $m_a = 0.5$ ).

→ SEGNALE MODULANTE  
↓  
OSCILLAZIONE MODULATA →



- MODULAZIONE DI FASE (PM, PHASE MODULATION)

$$\text{PM} \left\{ \begin{array}{l} m(t) = 0 \\ \alpha(t) = kx(t) \end{array} \right. \quad \Delta\omega(t) = \dot{\alpha}(t) = k\dot{x}(t)$$

Viene modificato solo l'argomento dell'oscillazione portante (solo la modulazione in angolo è presente).

$$s(t) = V_0 \cos[\omega_0 t + kx(t) - \varphi_0]$$

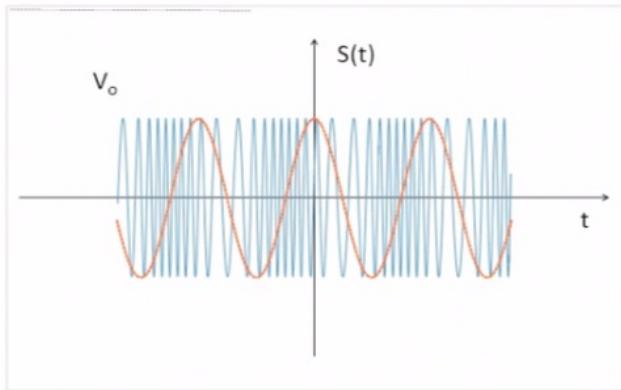
- MODULAZIONE DI FREQUENZA (FM, FREQUENCY MODULATION)

$$\text{FM} \quad \begin{cases} m(t) = 0 \\ \Delta\omega(t) = kx(t) \end{cases} \quad \alpha(t) = k \int_{-\infty}^t x(\tau) d\tau$$

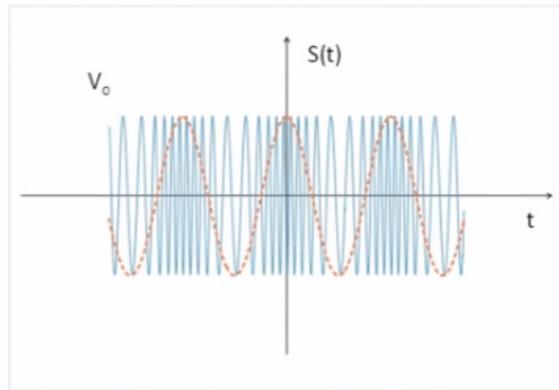
Viene modificato solo l'argomento dell'oscillazione portante (solo la modulazione in angolo è presente).

$$s(t) = V_0 \cos \left[ \omega_0 t + k \int_{-\infty}^t x(\tau) d\tau - \varphi_0 \right]$$

PM



FM

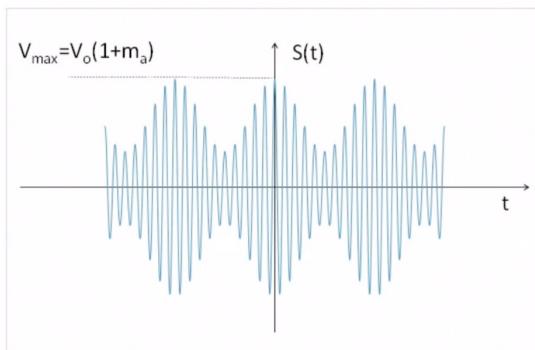
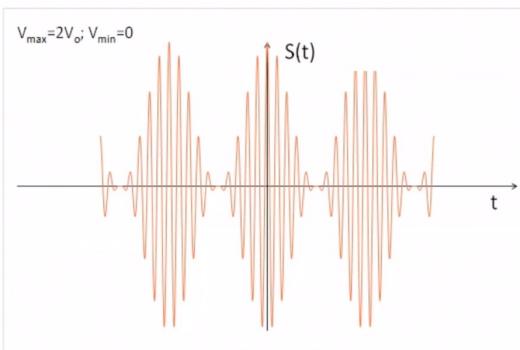


### INDICE DI MODULAZIONE

- INDICE DI MODULAZIONE AM (E SOVRAMODULAZIONE)

$$m_a = \max(|m(t)|) \quad m_a \in [0, 1]$$

L'indice di modulazione indica il livello di modulazione (cioè l'entità della trasformazione) che viene impressa alla portante dal segnale modulante. L'indice assume il valore 0 in assenza totale di modulazione, ed il valore 1 quando si ha il massimo della modulazione. La limitazione a 1 dipende dal fatto che  $m(t) \geq -1$  per definizione, essendo  $V(t) \geq 0$ .

 $m_a=0.5$  $m_a=1 \rightarrow \text{MASSIMO}$ 

A partire dal segnale modulante in ingresso  $x(t)$  il modulatore AM genera un segnale del tipo:

$$s(t) = V_0 [1 + kx(t)] \cos[\omega_0 t - \varphi_0]$$

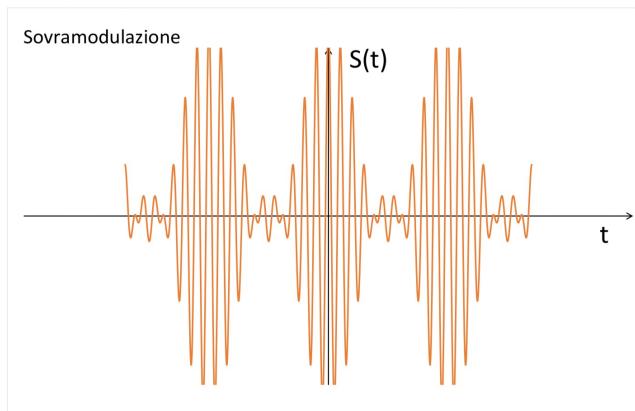
Occorre prestare attenzione a non avere  $kM > 1$  (o impropriamente  $m > 1$ ) altrimenti il segnale va in **"sovramodulazione"**.  
(La modulazione diventa "ibrida").

↓  
SI DEVE AVERE IN INGRESSO

$$M \leq \frac{1}{k}$$

**SOVRAMODULAZIONE AM**

$$\left\{ \begin{array}{l} V(t) = V_0 |1 + kx(t)| \\ \alpha(t) = \begin{cases} 0 & 1 + kx(t) > 0 \\ \pi & 1 + kx(t) < 0 \end{cases} \end{array} \right.$$



- 1) Disegnare  $v(t)$   
2) Disegnare  $-v(t)$   
3) Fare il disegno nel mezzo

- INDICE DI MODULAZIONE D'ANGOLI

$$m = \max(|\alpha(t)|)$$

- Quando  $m = 0$  l'oscillazione modulata coincide con la portante.
- Per  $m > 0$  la modulazione aumenta all'aumentare dell'indice.

Si noti che non esiste alcuna limitazione superiore a differenza dell'AM.

## INVILUPPO COMPLESSO RAPPRESENTATIVO (O EQUIVALENTE PASSA-BASSO) DI OSCILLAZIONI MODULATE

Per le oscillazioni modulate è possibile introdurre un'estensione del **metodo simbolico di Steinmetz**. L'oscillazione modulata è ancora vista come parte reale di un fasore, al posto del numero complesso rappresentativo si ha una funzione complessa del tempo,  $i(t)$ , detta inviluppo; complesso rappresentativo, o equivalente passa-basso.

$$s(t) = V(t) \cos[\omega_0 t + \alpha(t) - \varphi_0] \quad \rightarrow \quad s(t) = \operatorname{Re} \{ i(t) e^{j\omega_0 t} \} \quad i(t) = V(t) e^{j[\alpha(t) - \varphi_0]}$$

- Nota la frequenza della portante, l'oscillazione è completamente individuata da  $i(t)$ .
- Si dice complesso perché in generale  $i(t)$  è una funzione complessa.
- $i(t)$  è passa-basso anziché passa-banda.
- In assenza di modulazione,  $i(t)$  diventa costante, ricadendo nel caso del metodo simbolico classico.

Possiamo immaginare l'oscillazione modulata come una sinusoide i cui parametri variano nettamente rispetto alla portante.

Prendiamo ora due oscillazioni modulate diverse, ma con la stessa pulsazione della portante:

$$\begin{aligned} s_1(t) &= \operatorname{Re} \{ i_1(t) e^{j\omega_0 t} \} & i_1(t) &= V_1(t) e^{j[\alpha_1(t) - \varphi_{01}]} \\ s_2(t) &= \operatorname{Re} \{ i_2(t) e^{j\omega_0 t} \} & i_2(t) &= V_2(t) e^{j[\alpha_2(t) - \varphi_{02}]} \end{aligned}$$

La somma è una oscillazione modulata alla medesima frequenza, avente inviluppo complesso somma degli inviluppi:

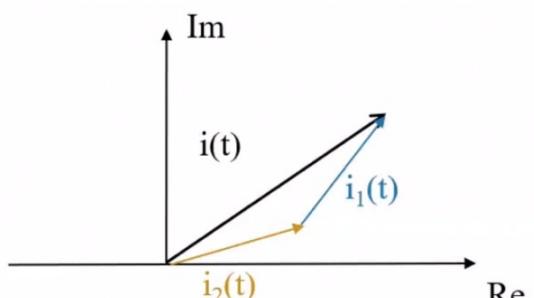
$$i(t) = i_1(t) + i_2(t) \quad \begin{cases} V(t) = |i(t)| \\ \alpha(t) = \arg \{ i(t) \} \end{cases}$$

### Dimostrazione:

$$\begin{aligned} s(t) &= s_1(t) + s_2(t) = \operatorname{Re} \{ i_1(t) e^{j\omega_0 t} \} + \operatorname{Re} \{ i_2(t) e^{j\omega_0 t} \} : \\ &= \operatorname{Re} \{ [i_1(t) + i_2(t)] e^{j\omega_0 t} \} = \operatorname{Re} \{ i(t) e^{j\omega_0 t} \} \end{aligned}$$

↓

LA SOMMA DI DUE PARTI REALE È LA PARTE  
REALE DELLA SOMMA.



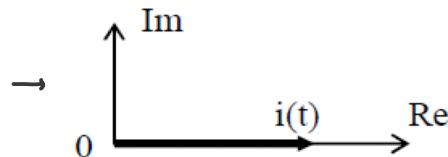
## CARATTERISTICHE SPECTRALI DI UNA OSCILLAZIONE AM

Allo scopo di determinare lo spettro di una modulazione AM consideriamo di nuovo la sua espressione, assunta per semplicità nulla la fase della portante:

$$s(t) = V_0 [1 + k_x(t)] \cos \omega_0 t \quad \text{con } V_0 [1 + k_x(t)] \geq 0$$

Il suo inviluppo complesso è dato da:

$$i(t) = V_0 [1 + k_x(t)] \quad \text{con } V_0 [1 + k_x(t)] \geq 0$$



Esso, assunta nulla la fase della portante, è reale positivo o nullo.

Separando i termini di  $s(t)$  possiamo vedere immediatamente che è dato dalla somma della portante e di un termine che, a meno di una costante, è il prodotto fra il segnale modulante e la portante.

$$s(t) = V_0 \cos \omega_0 t + V_0 k_x(t) \cos \omega_0 t$$

La trasformata di  $s(t)$  può essere calcolata direttamente ricordando la trasformata di una sinusoide e il teorema fondamentale della modulazione. Da essa è possibile ottenere immediatamente gli spettri di ampiezza e fase bilateri.

In alternativa, è possibile seguire un procedimento basato fin dall'inizio sulla rappresentazione monolatera dei segnali. Il vantaggio di questo secondo metodo è che permette, una volta ottenuto il risultato voluto per l'AM, di introdurre più facilmente alcune modulazioni derivate dall'AM, prendendone di volta in volta solo alcuni componenti. A questo scopo supponiamo che il segnale modulante  $x(t)$ , di tipo passa-basso, sia rappresentabile mediante l'integrale di Fourier:

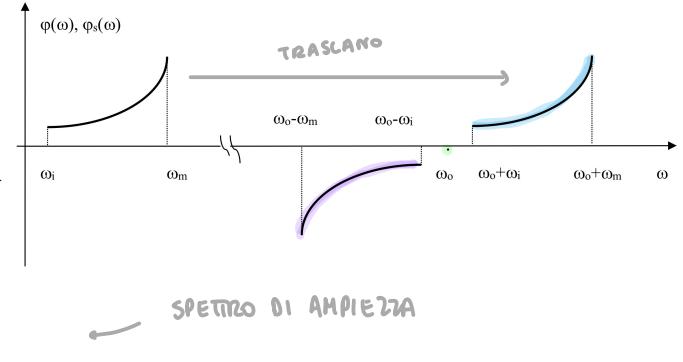
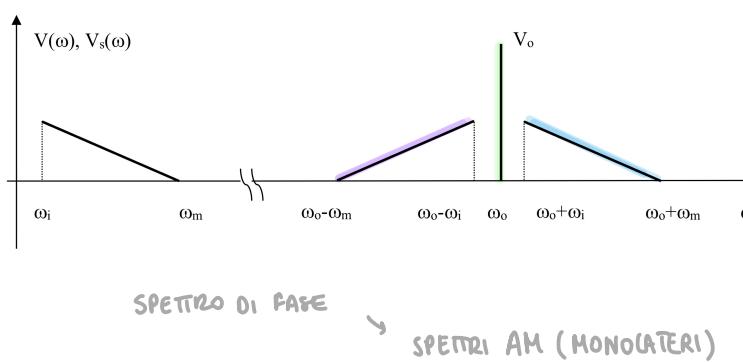
$$s(t) = \int_{\omega_i}^{\omega_m} V(\omega) \cos [\omega t - \varphi(\omega)] d\omega \rightarrow \text{Sostituendo nell'espressione dell'AM:}$$

$$s(t) = V_0 \cos \omega_0 t + V_0 k_x(t) \cos \omega_0 t$$

$$\cos(\alpha) \cos(\beta) = \frac{1}{2} \cos(\alpha+\beta) + \frac{1}{2} \cos(\alpha-\beta)$$

È quindi possibile scomporre l'AM nella somma di tre termini: **portante**, **banda laterale superiore**, **banda laterale inferiore**:

$$s(t) = V_0 \cos \omega_0 t + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 + \omega) t - \varphi(\omega)] d\omega + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 - \omega) t - \varphi(\omega)] d\omega$$

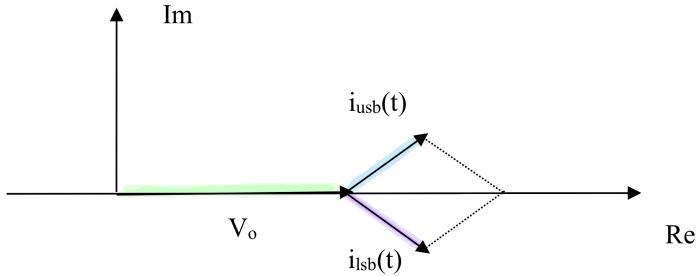


## INVILUPPO COMPLESSO AM

$$s(t) = V_0 \cos \omega_0 t + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 + \omega)t - \varphi(\omega)] d\omega + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 - \omega)t - \varphi(\omega)] d\omega$$

Ad ognuno dei tre termini corrisponde un termine dell'**inviluppo complesso**. La modulazione è solo in ampiezza.

$$i(t) = V_0 + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) e^{j[\omega t - \varphi(\omega)]} d\omega + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) e^{-j[\omega t - \varphi(\omega)]} d\omega$$



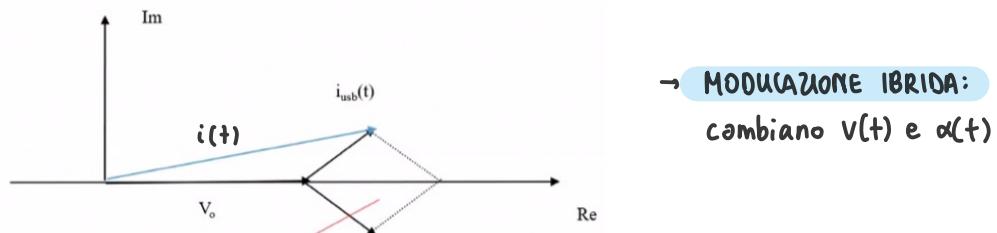
## SSB (SINGLE SIDE BAND)

Poiché ciascuno dei due segnali corrispondenti alla banda laterale superiore od inferiore contiene la stessa informazione (nota una delle due bande laterali, è immediato ricavare l'altra), è possibile eliminare una delle due bande laterali con l'evidente vantaggio di dimezzare la banda impegnata dal relativo segnale modulato, il cui spettro rimane costituito dalla riga corrispondente alla portante e dalla banda laterale non eliminata.

Il segnale modulato che così si ottiene viene chiamato a banda laterale singola ed indicato con la sigla **SSB (Single Side Band)**. Esso può essere generato a partire da una oscillazione AM, che viene filtrata con un filtro passa-banda che provveda ad eliminare la banda indesiderata.

$$s(t) = V_0 \cos \omega_0 t + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 + \omega)t - \varphi(\omega)] d\omega + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 - \omega)t - \varphi(\omega)] d\omega$$

$$i(t) = V_0 + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) e^{j[\omega t - \varphi(\omega)]} d\omega + \cancel{\frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) e^{-j[\omega t - \varphi(\omega)]} d\omega}$$



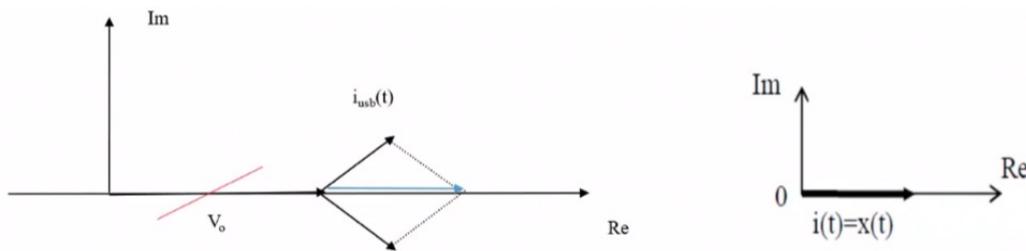
- DSB-SC (DSB suppressed carrier)

Una seconda modulazione derivata dall'AM è la **DSB-SC (DSB Suppressed Carrier)**. In essa vengono mantenute entrambe le bande laterali, ma viene eliminata la portante, per risparmiare potenza, senza perdere il contenuto informativo relativo al segnale modulante. Di fatto si ottiene una modulazione a prodotto (esaminata in maggior dettaglio poco più avanti). Le espressioni della DSB-SC sono:

$$s(t) = \cancel{V_0 \cos \omega_0 t} + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 + \omega)t - \varphi(\omega)] d\omega + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 - \omega)t - \varphi(\omega)] d\omega$$

$$i(t) = \cancel{V_0} + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) e^{j[\omega t - \varphi(\omega)]} d\omega + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) e^{-j[\omega t - \varphi(\omega)]} d\omega$$

L'eliminazione della portante rende la DSB-SC ibrida. Infatti, pur rimanendo  $i(t)$  reale (sotto l'ipotesi di fase della portante nulla), esso può essere positivo o negativo, rendendo non identicamente nulla la deviazione istantanea di fase.



**MODULAZIONE IBRIDA:** l'inviluppo è reale ma è negativo quando  $x(t) < 0$

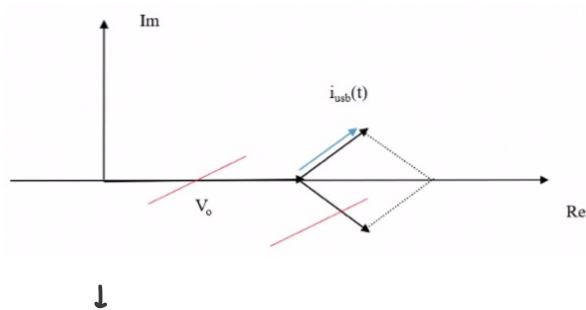
$$\alpha(t) = \pi \text{ se } x(t) < 0$$

- SSB-SC (SSB - Suppressed Carrier)

Se a partire da una DSB-SC si elimina anche una delle due bande laterali mediante un filtro, si ottiene una modulazione **SSB-SC (SSB - Suppressed Carrier)**. In questo caso il vantaggio è il dimezzamento della banda occupata, da sommare al risparmio in potenza già ottenuto nella DSB eliminando la portante. Le espressioni della SSB-SC sono:

$$s(t) = \cancel{V_0 \cos \omega_0 t} + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 + \omega)t - \varphi(\omega)] d\omega + \cancel{\frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 - \omega)t - \varphi(\omega)] d\omega}$$

$$i(t) = \cancel{V_0} + \frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) e^{j[\omega t - \varphi(\omega)]} d\omega + \cancel{\frac{kV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) e^{-j[\omega t - \varphi(\omega)]} d\omega}$$



**MODULAZIONE IBRIDA:** cambiano  $v(t)$  e  $\alpha(t)$

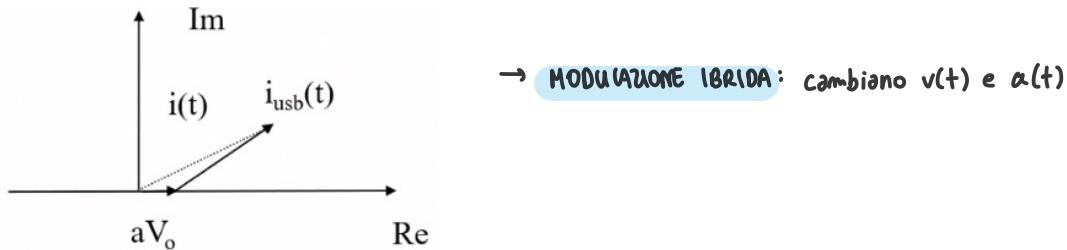
La SSB-SC equivale ad una conversione di frequenza in salita, in quanto gli spettri di ampiezza e fase dell'oscillazione modulata coincidono con quelli del segnale modulante, traslati di  $\omega_0$ .

### • SSB (PORTANTE PARZIALMENTE SOPPRESSA)

$$s(t) = aV_0 \cos \omega_0 t + \frac{aV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) \cos [(\omega_0 + \omega)t - \varphi(\omega)] d\omega$$

$$i(t) = aV_0 + \frac{aV_0}{2} \int_{\omega_i}^{\omega_m} V(\omega) e^{j[\omega t - \varphi(\omega)]} d\omega$$

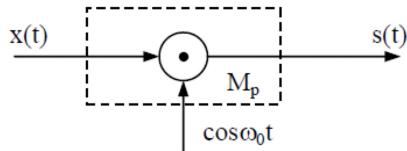
È una variante dell'SSB-SC; la portante, fortemente attenuata, può essere utile per l'operazione inversa (conversione in discesa).



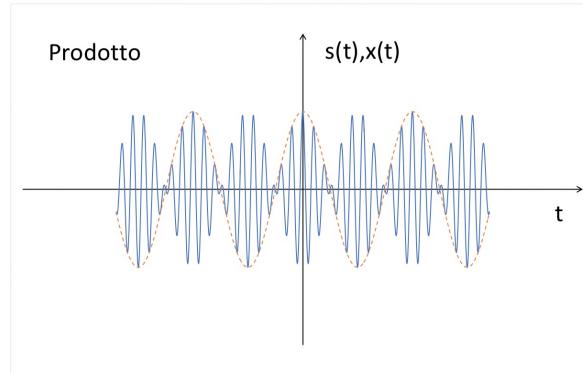
### MODULAZIONE A PRODOTTO E QAM

La **modulazione a prodotto** può essere ottenuta direttamente, senza passare dall'AM, facendo il prodotto del segnale modulante  $x(t)$  per una sinusoida tramite un circuito detto modulatore a prodotto o "mixer", ottenendo:

$$s(t) = x(t) \cos \omega_0 t$$



La modulazione a prodotto è importantissima nella sua versione digitale (ASK, L-ASK).



L'inviluppo complesso è estremamente semplice in quanto coincide con il segnale modulante:  $i(t) = x(t)$

La modulazione è ibrida. Infatti sia l'ampiezza istantanea che la deviazione istantanea di fase variano nel tempo:

$$V(t) = |x(t)|$$

$$\alpha(t) : \begin{cases} 0 & x(t) > 0 \\ \pi & x(t) < 0 \end{cases}$$

Gli spettri si ottengono immediatamente dall'espressione della trasformata:

$$S(\omega) = \frac{1}{2} X(\omega - \omega_0) + \frac{1}{2} X(\omega + \omega_0)$$

La banda di  $s(t)$  è doppia rispetto a quella di  $x(t)$ . Definita come "efficienza" in frequenza della modulazione il rapporto fra banda del segnale modulante e banda dell'oscillazione modulata, essa vale 1/2 per la modulazione a prodotto.

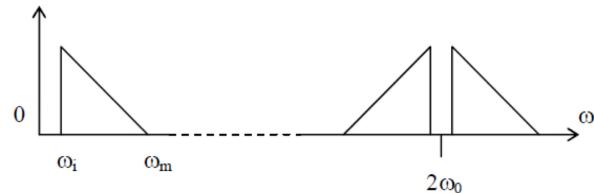
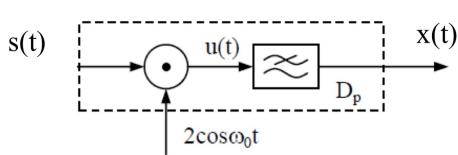
$$\eta_f = \frac{B_x}{B_s} = \frac{\omega_m}{2\omega_m} = \frac{1}{2}$$

## DEMODULATORE

La demodulazione si ottiene rimoltiplicando l'oscillazione modulata per la portante (moltiplicata per due per convenienza formale):

$$u(t) = 2s(t) \cos \omega_0 t = 2x(t) \cos^2 \omega_0 t = x(t) + x(t) \cos 2\omega_0 t$$

da cui filtrando passa-basso:  $x_d(t) = x(t)$



Il demodulatore richiede la "ricostruzione" della portante (frequenza e fase comprese), per cui è detto **coerente**.

Il problema è dato dalla ricostruzione della portante, ovvero come è possibile estrarre la portante da  $s(t)$ .

La ricostruzione implica un blocco non lineare in quanto il segnale  $s(t)$  non ha una componente alla pulsazione della portante (una rete lineare tempo invariante non può dare in uscita componenti in frequenza che non siano già presenti in ingresso).

La portante deve essere ricostruita non solo in frequenza ma anche in fase.

L'**errore di fase** provoca un'attenuazione del segnale demodulato, che sarà:

$$x_d(t) = x(t) \cos \Delta$$

Se infatti la portante ricostruita è sfasata in ritardo si ha:

$$u(t) = 2s(t) \cos(\omega_0 t - \Delta) = 2x(t) \cos \omega_0 t \cos(\omega_0 t - \Delta) = x(t) \cos \Delta + x(t) \cos(2\omega_0 t - \Delta)$$

$$\cos(\alpha) \cos(\beta) = \frac{1}{2} \cos(\alpha + \beta) + \frac{1}{2} \cos(\alpha - \beta)$$

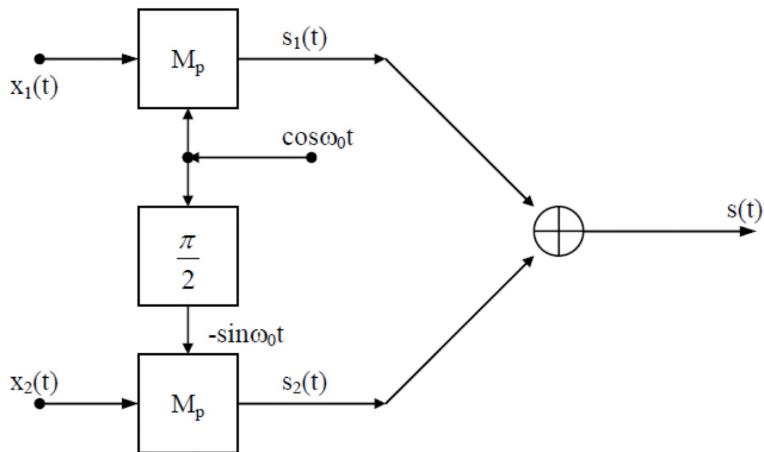
## MODULAZIONI QAM (QUADRATURE AMPLITUDE MODULATION)

- MODULATORE

È la somma di due modulazioni a prodotto (non AM!) con portanti in “quadratura”.

Vi sono due segnali modulanti  $x_1(t)$  e  $x_2(t)$ , aventi le stesse caratteristiche spettrali; devono inoltre essere “indipendenti”, cioè non legati fra di loro da nessuna relazione.

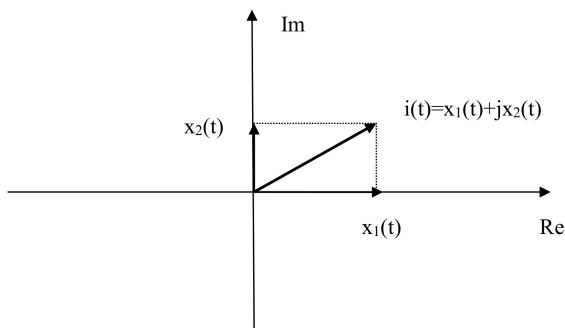
$$s(t) = x_1(t) \cos \omega_0 t - x_2(t) \sin \omega_0 t$$



Essa infatti consiste di due modulazioni a prodotto con portanti in quadratura, cioè con la seconda portante sfasata in anticipo di  $\pi/2$ .

**L'inviluppo complesso** è la somma dei due inviluppi complessi fase è quadratura (indipendenti!)

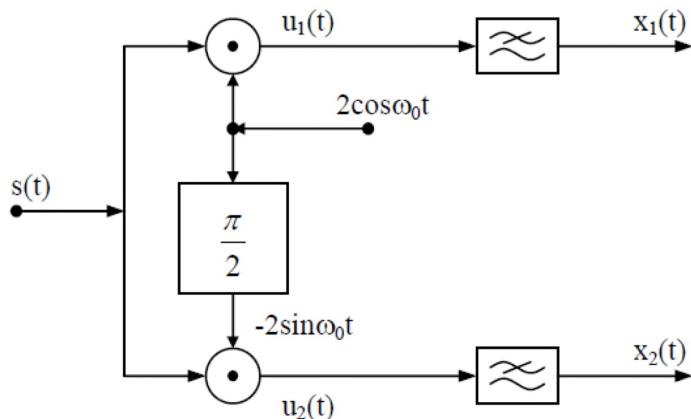
$$i(t) = x_1(t) + jx_2(t) = \operatorname{Re} \{ (x_1(t) + jx_2(t)) (\sim \omega_0 t + j \sin \omega_0 t) \}$$



**Lo spettro** è dato dalla sovrapposizione degli spettri di due modulazioni a prodotto. **Efficienza in frequenza** doppia (uno) rispetto a quella della modulazione a prodotto:

$$\eta_f = \frac{2B_x}{B_s} = \frac{2\omega_m}{2\omega_m} = 1 \rightarrow \text{Grande vantaggio della QAM!}$$

- DEMODULATORE



Per il segnale in uscita al primo (componente in fase) si ha:  $\text{sen}(\alpha + \beta) = \text{sen}\alpha \cos\beta + \text{sen}\beta \cos\alpha$

$$\begin{aligned} u_p(t) &= 2s(t)\cos\omega_0t = 2x_1(t)\cos^2\omega_0t - 2x_2(t)\sin\omega_0t\cos\omega_0t = \\ &= x_1(t) + x_1(t)\cos 2\omega_0t - x_2(t)\sin 2\omega_0t \end{aligned}$$

La prima delle tre componenti è l'unica voluta e viene isolata dal filtro passa-basso.

$$x_{\text{pol}}(t) = x_1(t)$$

Analogamente per la via in quadratura.

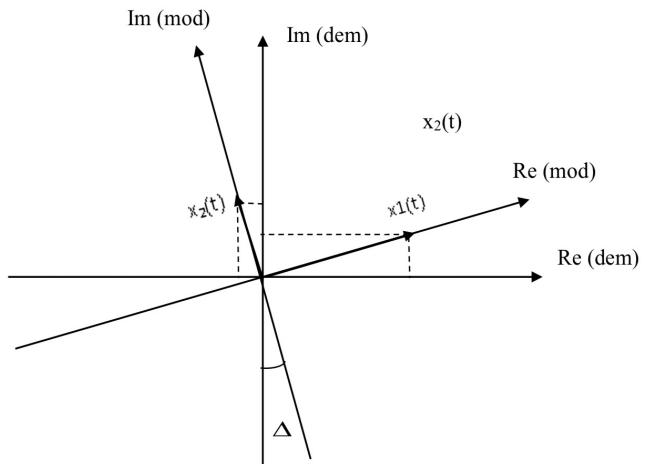
Nel caso del QAM un **errore di fase** provoca non solo un'attenuazione del segnale utile, ma anche un'**interferenza della via in quadratura**.

Infatti nel caso di una sfasatura in ritardo si ha:

$$x_{\text{pol}}(t) = x_1(t)\cos\Delta - x_2(t)\sin\Delta$$

$$x_{\text{qol}}(t) = x_2(t)\cos\Delta - x_1(t)\sin\Delta$$

Le relazioni sopra possono essere trovate per via analitica, o, molto più semplicemente, per via grafica, facendo ruotare in senso antiorario il sistema di coordinate (Re, Im) del modulatore e sovrapponendolo a quello, non ruotato, del demodulatore. Le componenti in uscita al demodulatore sono le proiezioni sugli assi di quest'ultimo del vettore  $i(t)$  disegnato sul primo.



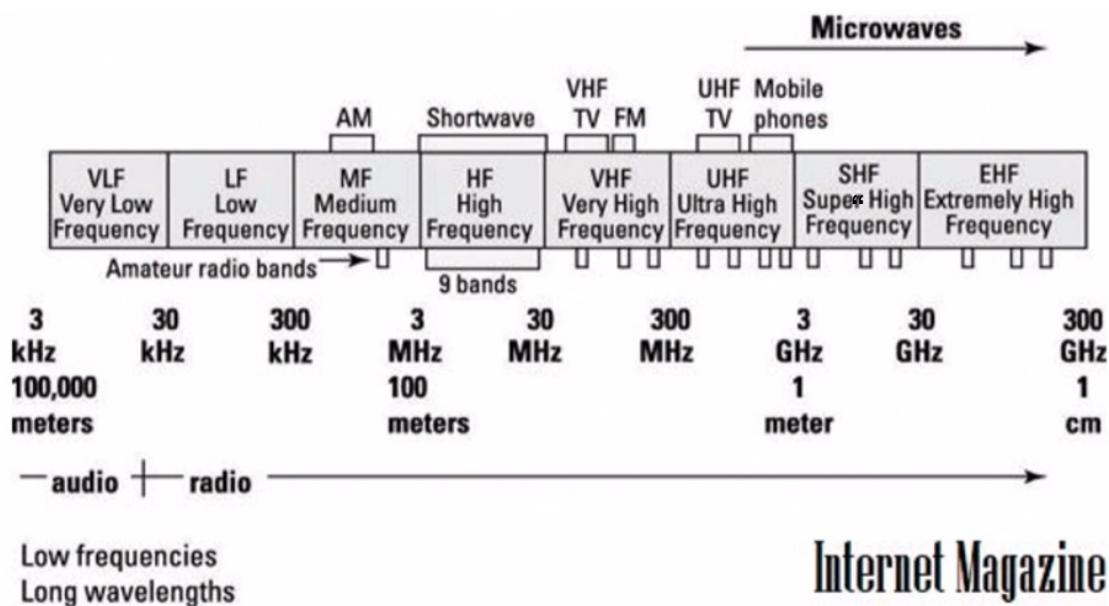
## SPIETRO SEGNALI MODULATI IN ANGOLO: FORMULA DI CARSON

Il calcolo dello spettro delle oscillazioni modulate in angolo (PM, FM) è possibile solo con segnale modulante sinusoidale. Esso richiede la conoscenza delle funzioni di Bessel, non viste in altri corsi. Qui ci limitiamo alla semplice enunciazione della **formula di Carson** che fornisce la banda occupata da un'oscillazione analogica modulata in angolo.

Si noti che in assenza di modulazione è uguale alla formula per l'AM, la modulazione a prodotto, la QAM:

$$B_\omega = 2(\omega_m + \Delta\omega_{\max}) \quad \Delta\omega_{\max} = \max(\Delta\omega(t))$$

## RADIO DIFFUSIONE



### AM:

- $x(t)$  banda 50-4500 Hz;  $B = 9$  kHz
- $f_0$  (dipende dalla banda)
- Mono
- MF abbandonata da RAI

### FM:

- $x(t)$  banda 30-15000 Hz;  $B = 180$  kHz
- $f_0$  87.5-108 MHz
- Mono e stereo
- In Italia è in corso di affiancamento il DAB+, radio digitale. Non si prevede vengano eliminate a breve le radio analogiche.

$$\Delta f_{\max} = 75 \text{ kHz}$$

Banda	Utilizzo	Frequenze
onde lunghe (LW)	AM	153 – 279 kHz
onde medie (MW)	AM	525 – 1615 kHz
onde corte (SW)	AM	2310 kHz - 30 MHz
FM	FM	87,5 - 108 MHz
DAB Band III	DAB	174 - 240 MHz

## CAPITOLO 8 - SEGNALI AD ENERGIA E A POTENZA FINITA

ENERGIA E POTENZA DI UN SEGNALE

Dato un segnale  $x(t)$ , in generale complesso, si definisce “potenza istantanea”:

$$p(t) = x^*(t)x(t) = |x(t)|^2$$

Da cui derivano le definizioni di

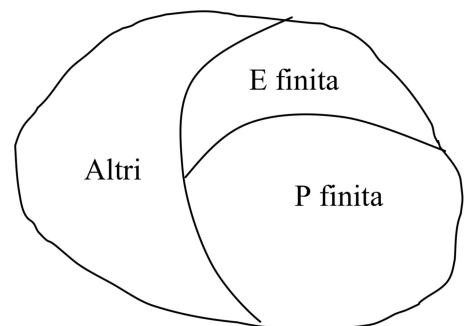
- “Energia” —  $E = \int_{-\infty}^{+\infty} |x(t)|^2 dt$
- “Potenza (media)” —  $P \leq |x(t)|^2 \geq \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} |x(t)|^2 dt$

Si noti che i termini “energia” e “potenza” sono utilizzati in senso lato. Ad esempio, supponendo che  $x(t)$  rappresenti una tensione, la “potenza istantanea” e la “potenza media” andrebbero misurate in V<sup>2</sup> e non in W. Analogamente l’“energia” andrebbe misurata in V<sup>2</sup>s e non in J. La denominazione deriva per analogia dalla osservazione che se la tensione fosse applicata ai capi di un resistore R, la potenza istantanea (in W) erogata al carico sarebbe data da:

$$P_W(t) = \frac{|x(t)|^2}{R} \quad (\text{W})$$

Supponendo quindi R=1 Ohm, si otterebbe la coincidenza numerica fra  $p(t)$  e  $P_W(t)$ . Allo stesso risultato si perviene supponendo che  $x(t)$  rappresenti una corrente anziché una tensione. Il contesto permette di risolvere ogni potenziale ambiguità.

Un segnale si dice ad energia finita se l’integrale che ne rappresenta l’energia converge (condizione fra l’altro sufficiente all’esistenza della sua trasformata di Fourier), mentre è a potenza finita nel caso in cui converga, ad un valore diverso da zero, l’integrale che ne rappresenta la potenza. Si noti che per un segnale ad energia finita la potenza tende a zero, per cui un segnale non può appartenere ad entrambe le categorie. Inoltre esistono segnali che non sono né ad energia né a potenza finita, come ad esempio  $x(t)=eat$ , per cui la situazione può essere riassunta dalla figura seguente, in cui l’insieme dei segnali è suddiviso in tre sottoinsiemi disgiunti.



**VALORE EFFICACE** → Il valore efficace di una costante coincide con la costante stessa.

$$\downarrow \\ x_{\text{eff}} = \sqrt{p} \quad \rightarrow \text{PER UNA SINUSOIDA} \quad x(t) = A \cos(\varphi t - \varphi)$$

$$V_{\text{effsen}} = \frac{A}{\sqrt{2}}$$

ANALISI GENERALIZZATA

I segnali ad energia finita ammettono trasformata di Fourier, mentre quelli a potenza finita in generale no, ad eccezione dei segnali periodici. Verrà introdotta tramite gli spettri di energia e di potenza una rappresentazione “energetica” dei segnali nel dominio delle frequenze. Questa rappresentazione si aggiunge per i segnali ad energia finita alla rappresentazione usuale mediante trasformata di Fourier, mentre spesso è l'unica possibile per i secondi, ad esclusione dei segnali periodici. Si parla quindi di analisi di Fourier generalizzata, per indicare che la possibilità di una rappresentazione nel dominio delle frequenze viene estesa anche a molti segnali che non ammettono trasformata di Fourier.

SEGNALI AD ENERGIA FINITAFUNZIONI DI CROSSCORRELAZIONE E AUTOCORRELAZIONE

Dati due segnali in generale complessi,  $x(t)$  ed  $y(t)$ , si definisce **funzione di crosscorrelazione** il coniugato del prodotto interno di un uno di essi per la versione anticipata dell'altro:

$$\dot{\varphi}_{xy}(\tau) = \langle x, y_{\tau} \rangle^* = \int_{-\infty}^{+\infty} x^*(t) y(t + \tau) dt$$

*c'è  $\tau$  per via del pedice*

Nel caso particolare in cui  $y(t)=x(t)$ , la funzione di crosscorrelazione prende il nome di **funzione di autocorrelazione**, che risulta quindi così definita:

$$\dot{\varphi}_x(\tau) = \langle x, x_{\tau} \rangle^* = \int_{-\infty}^{+\infty} x^*(t) x(t + \tau) dt$$

Si noti che l' autocorrelazione calcolata nell'origine rappresenta l'energia di un segnale.

$$\dot{\varphi}_x(0) = \int_{-\infty}^{+\infty} x^*(t) x(t) dt = \int_{-\infty}^{+\infty} |x(t)|^2 dt = E_x$$

PROPRIETÀ DELLE FUNZIONI DI CROSS ED AUTOCORRELAZIONE

1)  $\dot{\varphi}_{xy}(\tau) = \dot{\varphi}_{y^*x}^*(-\tau)$  → (SUPER) SIMMETRIA HERMITIANA CON INVERSIONE DEI PEDICI

DIMOSTRAZIONE → 
$$\dot{\varphi}_{yx}^*(-\tau) = \left[ \int_{-\infty}^{+\infty} y^*(t) x(t - \tau) dt \right]^* =$$

$$= \int_{-\infty}^{+\infty} y(t) x^*(t - \tau) dt = \text{ (cambiando } t' = t - \tau \text{ e quindi di nuovo } t = t') \\ = \int_{-\infty}^{+\infty} x^*(t') y(t' + \tau) dt'$$

2) Analogamente per la funzione di autocorrelazione si ha (simmetria hermitiana):

$$\dot{\varphi}_x(\tau) = \dot{\varphi}_x(-\tau)$$

3) Nel caso particolare di segnali reali (**proprietà di parità**):  $\dot{\varphi}_x(\tau) = \dot{\varphi}_x(-\tau)$

4) Applicando la diseguaglianza di Schwarz alla definizione di crosscorrelazione si ha:

$$|\dot{\Phi}_{xy}(\tau)|^2 \leq \dot{\Phi}_x(0) \dot{\Phi}_y(0) = E_x E_y \rightarrow |\dot{\Phi}_x(\tau)| \leq \dot{\Phi}_x(0) = E_x$$

DIMOSTRAZIONE → SI APPLICA LA DISEGUAGLIANZA DI SCHWARTZ

$$|\dot{\Phi}_{xy}(\tau)|^2 = \left| \int_{-\infty}^{+\infty} \underbrace{x^*(t)}_{f_1} \underbrace{y(t+\tau)}_{f_2} dt \right|^2 \leq \int_{-\infty}^{+\infty} \left| \underbrace{x^*(t)}_{f_1} \right|^2 dt \int_{-\infty}^{+\infty} \left| \underbrace{y(t+\tau)}_{f_2} \right|^2 dt = E_x E_y$$

La relazione appena trovata stabilisce che il massimo (in modulo) della funzione di autocorrelazione è nell'origine, e che esso coincide con l'energia.

5)  $\dot{\Phi}_{xy}(\tau) = x^*(-\tau)^* y(\tau)$

DIMOSTRAZIONE →  $\dot{\Phi}_{xy}(\tau) = \int_{-\infty}^{+\infty} x^*(t) y(t+\tau) dt =$  (cambiando variabile  $t' = -t$ )  
 $= \int_{-\infty}^{+\infty} x^*(-t') y(\tau - t') dt' =$   
 $= x^*(-\tau)^* y(\tau)$

Analogamente per la funzione di autocorrelazione:  $\dot{\Phi}_x(\tau) = x^*(-\tau)^* x(\tau)$

### TEOREMA DI PARSEVAL GENERALIZZATO

Per le funzioni ad energia finita è garantita l'esistenza della trasformata di Fourier.

Applicando l'operatore trasformata di Fourier alla funzione di crosscorrelazione si ottiene:

$$\dot{\Phi}_{xy}(\omega) = F[\dot{\Phi}_{xy}(\tau)] = F[x^*(-\tau)] F[y(\tau)] = X^*(\omega) Y(\omega)$$

da cui si deduce che è garantita anche l'esistenza della trasformata della funzione di crosscorrelazione. Ora è possibile scrivere la funzione di crosscorrelazione come antitrasformata della sua trasformata:

$$\dot{\Phi}_{xy}(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \dot{\Phi}_{xy}(\omega) e^{j\omega\tau} d\omega$$

e valutare il valore assunto nell'origine nel modo seguente:

$$\dot{\Phi}_{xy}(0) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} \dot{\Phi}_{xy}(\omega) d\omega = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X^*(\omega) Y(\omega) d\omega$$

Si ottiene quindi l'enunciato del teorema di Parseval generalizzato:

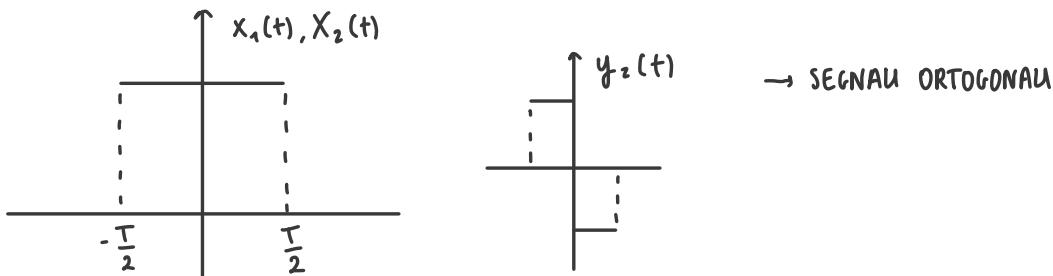
$$\int_{-\infty}^{+\infty} x^*(t) y(t) dt = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X^*(\omega) Y(\omega) d\omega \quad \text{OPPURE IN FORMA PIÙ COMPATTA} \rightarrow \langle x, y \rangle^* = \frac{1}{2\pi} \langle X, Y \rangle^*$$

Dal teorema di Parseval generalizzato si deduce che se due segnali sono ortogonali, lo devono per forza essere anche le relative trasformate, e viceversa.

Inoltre si possono fare le seguenti **osservazioni**:

- condizione sufficiente affinché due segnali siano ortogonali è che essi non si sovrappongano nel tempo (dal primo integrale);
- condizione sufficiente affinché due segnali siano ortogonali è che essi non si sovrappongano in frequenza (dal secondo integrale);
- due segnali che si sovrappongono sia nel tempo che in frequenza possono ancora essere ortogonali.

Le osservazioni sopra riportate sono alla base delle tecniche di multiplazione a divisione di tempo (TDM), di frequenza (FDM) e di codice (CDM).



### TEOREMA DI PARSEVAL E DENSITÀ SPECTRALE DI ENERGIA

Se  $y(t)$  e  $x(t)$  coincidono, si ottiene l'enunciato del teorema di Parseval:

$$E_x = \dot{\varphi}_x(0) = \underbrace{\int_{-\infty}^{+\infty} x(t)^* x(t) dt}_{E_x} = \frac{1}{2\pi} \int_{-\infty}^{+\infty} X^*(\omega) X(\omega) d\omega$$

$$\text{Poiché } F[\dot{\varphi}_x(\tau)] = |X(\omega)|^2 \rightarrow E_x = \int_{-\infty}^{+\infty} \frac{F[\dot{\varphi}_x(\tau)]}{2\pi} d\omega$$

La funzione integrando al secondo membro prende il nome di **densità spettrale di energia**. Occorre precisare che essa è bilatera e riferita alle pulsazioni. Si noti che essa lega, mediante trasformata di Fourier, la funzione di autocorrelazione allo spettro di energia.

$$E_{bil}(\omega) = \frac{F[\dot{\varphi}_x(\tau)]}{2\pi} = \frac{|X(\omega)|^2}{2\pi}$$

Se  $x(t)$  è reale si può definire una densità spettrale di energia **monolatera** come:

$$E(\omega) = \begin{cases} 2F_{bil}(\omega) & \omega > 0 \\ E_{bil}(\omega) & \omega = 0 \end{cases}$$

L'energia in questo caso si ottiene limitando l'integrazione al solo semiasse positivo.

$$E = \int_0^{+\infty} E(\omega) d\omega \quad \text{PROPRIETÀ LOCALE} \cdot E_{\omega_1, \omega_2} = \int_{\omega_1}^{\omega_2} E(\omega) d\omega$$

Se si preferisce riferirsi alle frequenze anziché alle pulsazioni si ha:

$$E_{f,bil}(f) = 2\pi E_{bil}(2\pi f) \quad \text{PER LA QUALE} \rightarrow E = \int_{-\infty}^{+\infty} E_{f,bil}(f) df$$

Analogamente per la versione monolatera, possibile solo per segnali reali.

## SEGNALI A POTENZA FINITA

Le definizioni delle funzioni di **crosscorrelazione** ed **autocorrelazione** per le funzioni a potenza finita si possono ottenere con la seguente sostituzione formale:

$$\int_{-\infty}^{+\infty} dt \rightarrow \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} dt$$

$\Phi_{xy}(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x^*(t) y(t+\tau) dt$  CROSCORRELAZIONE

$\Phi_x(\tau) = \lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x^*(t) x(t+\tau) dt$  AUTOCORRELAZIONE

(uguali a quelle a energia finita: cambia il puntino sul  $\Phi$  per distinguerle)

Per le funzioni appena definite valgono le seguenti proprietà, analoghe a quelle viste per i segnali ad energia finita.

- $\Phi_{xy}(\tau) = \Phi_{yx}^*(-\tau)$
- $\Phi_x(\tau) = \Phi_x^*(-\tau)$
- $\Phi_x(\tau) = \Phi_x(-\tau)$  se  $x(t)$  è reale
- $|\Phi_{xy}(\tau)|^2 \leq \Phi_x(0) \Phi_y(0) = P_x P_y$
- $|\Phi_x(\tau)| \leq \Phi_x(0) = P_x$

L'ultima relazione in particolare afferma che il massimo (in modulo) della funzione di autocorrelazione è nell'origine, e che esso coincide con la potenza.

## DENSITÀ SPETTRALE DI POTENZA

Mentre per le funzioni ad energia finita è garantita l'esistenza della trasformata della funzione di autocorrelazione, ciò non accade per le funzioni a potenza finita. Supponendo tuttavia che esista tale trasformata, operando in modo analogo a quanto visto in precedenza per le funzioni ad energia finita si ottiene:

$$\Phi_x(\tau) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F[\Phi_x(\tau)] e^{j\omega\tau} d\omega \rightarrow \Phi_x(0) = \frac{1}{2\pi} \int_{-\infty}^{+\infty} F[\Phi_x(\tau)] d\omega \rightarrow P_x = \int_{-\infty}^{+\infty} \frac{F[\Phi_x(\tau)]}{2\pi} d\omega$$

La funzione integranda prende il nome di densità spettrale di potenza (bilatera, riferita alle pulsazioni):

$$G_{BIL}(\omega) = \frac{F[\Phi_x(\tau)]}{2\pi}$$

In modo del tutto analogo a quanto visto in precedenza, si può ottenere la densità spettrale riferita alle frequenze:

$$G_{f,BIL}(f) = 2\pi G_{BIL}(2\pi f)$$

e le versioni monolatere (indicate senza pedice "bil"), valide nel solo caso di segnali reali.

Anche nel caso di funzioni a potenza finita, la denominazione di spettro di potenza è giustificata dalla proprietà locale per cui:  $P_{\omega_1, \omega_2} = \int_{\omega_1}^{\omega_2} G(\omega) d\omega$

Nei segnali a energia finita esiste  $x(\omega)$ , in quelle a potenza finita no, possiamo rappresentarli solo con lo spettro di potenza.

ENERGIA FINITA  $x(t) \rightarrow \Phi_x(\tau) \leftrightarrow \dot{E}_{BIL}(\omega) = \frac{|X(\omega)|^2}{2\pi}$   
 POTENZA FINITA  $x(t) \rightarrow \Phi_x(\tau) \leftrightarrow G_{BIL}(\omega)$

Per i segnali a potenza finita, non periodici, lo spettro di potenza costituisce l'unica rappresentazione spettrale possibile.

## SEGNALI A POTENZA FINITA PERIODICI

I segnali periodici rappresentano un caso particolare di segnali a potenza finita, in quanto per essi è possibile avere altre rappresentazioni nel dominio delle frequenze (sviluppi in serie di Fourier, trasformata generalizzata). Risulta interessante individuare comunque anche il loro spettro di potenza, per collegarlo alle altre rappresentazioni.

Si noti innanzitutto che le funzioni di cross ed autocorrelazione di segnali periodici di uguale periodo  $T$ , possono essere calcolate su un periodo. Applicando quindi la seguente sostituzione formale:

$$\lim_{T \rightarrow \infty} \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} dt \rightarrow \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} dt \quad \left[ \begin{array}{l} \varphi_{xy}(\tau) = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x^*(t) y(t+\tau) dt \quad \text{CROSSCORRELAZIONE} \\ \varphi_x(\tau) = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} x^*(t) x(t+\tau) dt \quad \text{AUTOCORRELAZIONE} \end{array} \right]$$

Per individuare lo spettro di potenza, conviene dimostrare che la funzione di autocorrelazione è anch'essa una funzione periodica dello stesso periodo  $T$ . A questo scopo conviene esprimere  $x(t)$  ed il suo coniugato come sviluppi in serie di Fourier:

$$x^*(t) = \sum_{n=-\infty}^{\infty} c_n^* e^{-jn\omega_0 t} \quad x(t) = \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0 t}$$

$$\varphi_x(\tau) = \frac{1}{T} \int_{-\frac{T}{2}}^{+\frac{T}{2}} \sum_{n=-\infty}^{\infty} c_n^* e^{-jn\omega_0 t} \sum_{k=-\infty}^{\infty} c_k e^{jk\omega_0(t+\tau)} dt = \frac{1}{T} \sum_{n=-\infty}^{\infty} \sum_{k=-\infty}^{\infty} c_n^* c_k e^{jk\omega_0 \tau} \int_{-\frac{T}{2}}^{+\frac{T}{2}} e^{-j(n-k)\omega_0 t} dt$$

Poiché (ortogonalità delle funzioni dello sviluppo in serie di Fourier):

$$\int_{-\frac{T}{2}}^{+\frac{T}{2}} e^{-j(n-k)\omega_0 t} dt = \begin{cases} 0 & n \neq k \\ T & n = k \end{cases} \quad \varphi_x(\tau) = \sum_{n=-\infty}^{\infty} c_n^* c_n e^{jn\omega_0 \tau} = \sum_{n=-\infty}^{\infty} |c_n|^2 e^{jn\omega_0 \tau}$$

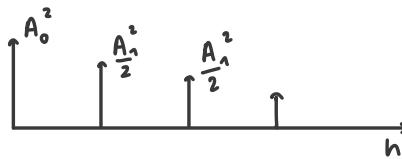
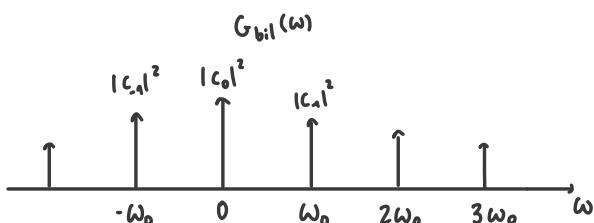
Dalla funzione di autocorrelazione si ottiene lo spettro di potenza:

$$G_{bil}(\omega) = \frac{F[\varphi_x(\tau)]}{2\pi} = \sum_{n=-\infty}^{\infty} |c_n|^2 \delta(\omega - n\omega_0)$$

Nel caso di segnali reali essa può anche essere scritta in forma monolatera utilizzando le relazioni che legano i coefficienti  $A_n$  e  $c_n \rightarrow A_0 = c_0$ ,  $A_n = 2|c_n| \quad n > 0$

$$G(\omega) = A_0^2 \delta(\omega) + \sum_{n=1}^{\infty} \frac{A_n^2}{2} \delta(\omega - n\omega_0)$$

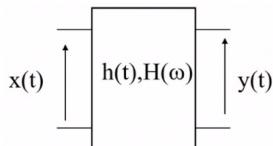
Si noti che l'espressione sopra non rappresenta altro che uno spettro di potenza a righe, i cui termini corrispondono alla potenza delle varie componenti sinusoidali, prese singolarmente. Questo risultato poteva anche essere ottenuto direttamente ricordando l'ortogonalità delle funzioni che compaiono nello sviluppo in serie di Fourier.



## TRASFORMAZIONI LINEARI TEMPO INVARIANTI DI SPETTRI DI ENERGIA E DI POTENZA

Si supponga che i segnali  $y(t)$  ed  $x(t)$  si trovino rispettivamente in ingresso ed in uscita ad una rete lineare tempo invariante, avente risposta impulsiva  $h(t)$  e funzione di trasferimento  $H(\omega)$ . Gli spettri di energia o di potenza dei due segnali sono legati dal modulo della  $H(\omega)$  al quadrato.

$$\text{SEGNALI AD ENERGIA FINITA} \rightarrow E_{y,\text{bil}}(\omega) = |H(\omega)|^2 E_{x,\text{bil}}(\omega)$$



DIMOSTRAZIONE: ricordando che  $Y(\omega) = H(\omega)X(\omega)$   
si ha  $\frac{|Y(\omega)|^2}{2\pi} = |H(\omega)|^2 = \frac{|X(\omega)|^2}{2\pi}$

La relazione è valida anche per tutte le altre forme di densità spettrale di energia (monolaterale, riferite alle frequenze) e permette di dimostrare facilmente la proprietà locale della densità spettrale di energia.

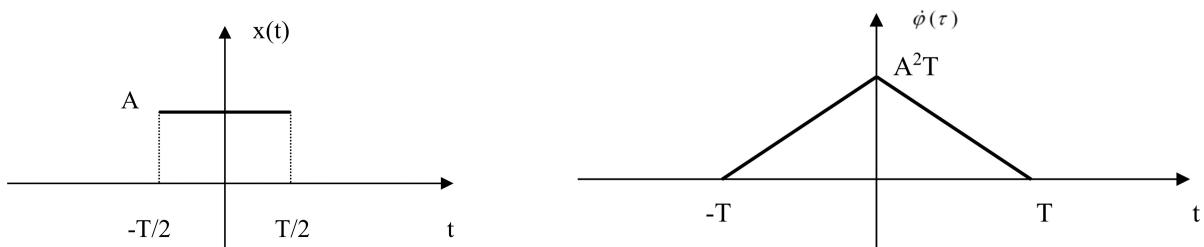
Dato un filtro ideale con caratteristica di ampiezza unitaria in banda passante, il segnale in uscita,  $x_f(t)$  avrà densità spettrale coincidente con quella del segnale in ingresso in banda passante, nulla in banda attenuata.

La sua energia, coincidente con l'energia delle componenti in banda passante, sarà data da:

$$E_{w_1, w_2} = \int_0^\infty |H(\omega)|^2 E_x(\omega) d\omega = \int_{w_1}^{w_2} E_x(\omega) d\omega$$

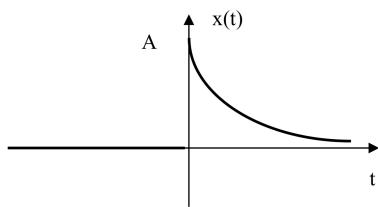
$$\text{SEGNALI A POTENZA FINITA} \rightarrow G_{y,\text{bil}}(\omega) = |H(\omega)|^2 G_{x,\text{bil}}(\omega) \quad (\text{DIMOSTRAZIONE NON NECESSARIA})$$

### ESERCIZIO 1 - FUNZIONE DI AUTOCORRELAZIONE DI UN IMPULSO RETTANGOLARE



Per prima cosa conviene calcolare l'energia, che è evidentemente uguale a  $A^2 T$ . Quindi notare che  $x(t)$  e  $x(t+\tau)$  si sovrappongono solo se  $|\tau| < T$ , quindi al di fuori di questo intervallo la funzione di autocorrelazione è nulla perché è nulla la funzione integranda. Per  $|\tau| < T$  la funzione integranda, dove diversa da zero, vale  $A^2$  e l'intervallo di integrazione cresce linearmente da  $-T$  a 0 e quindi decresce sempre linearmente da 0 a  $T$ . Il massimo si ha nell'origine e vale l'energia.

## Esercizio 2 - Energia e spettro di energia dell'esponenziale unilaterale

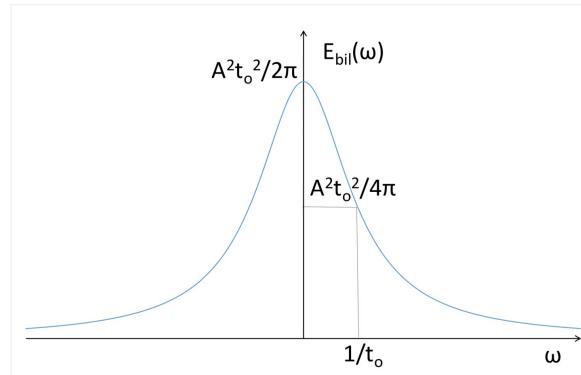


$$x(t) = \begin{cases} 0 & t < 0 \\ Ae^{-\frac{t}{t_0}} & t \geq 0 \end{cases} \quad t_0 > 0$$

$$E = \int_{-\infty}^{+\infty} |x(t)|^2 dt = A^2 \int_0^{+\infty} e^{-2t/t_0} dt = A^2 \left[ \frac{e^{-2t/t_0}}{-2/t_0} \right]_0^{+\infty} = \frac{A^2 t_0}{2} \quad \rightarrow \text{ricordando la trasformata di } x(t): X(\omega) = \frac{A t_0}{1 + j\omega t_0}$$

$$E_{\text{bil}}(\omega) = \frac{|X(\omega)|^2}{2\pi} = \frac{A^2 t_0^2}{2\pi} \frac{1}{1 + \omega^2 t_0^2} \quad \rightarrow$$

Densità spettrale di energia

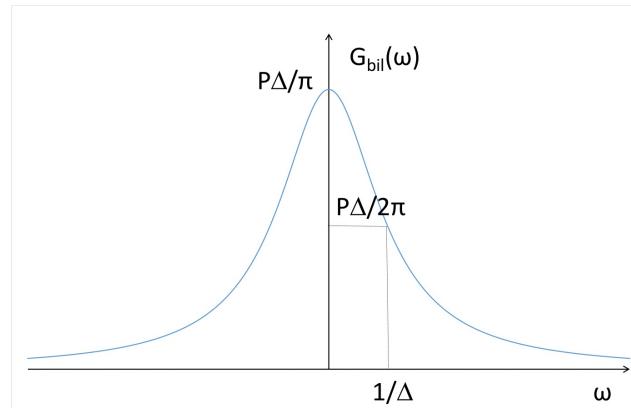
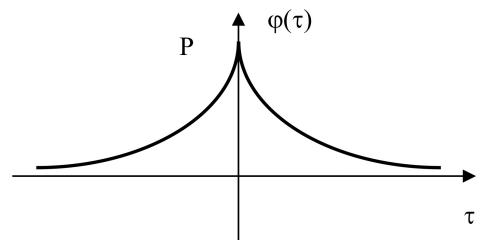


## Esercizio 3 - Calcolo spettro di potenza a partire dalla funzione di autocorrelazione

Si faccia attenzione che in questo esercizio non viene data l'espressione di una funzione  $x(t)$ , che resta incognita, ma quella di una funzione di autocorrelazione. Si noti che più funzioni  $x(t)$  possono avere la medesima funzione di autocorrelazione e quindi lo stesso spettro di potenza. La funzione di autocorrelazione ha l'andamento dell'esponenziale bilatera già studiato (solo i simboli sono diversi, si noti in particolare che il valore nell'origine è indicato con  $P$  perché deve essere uguale alla potenza):

$$\varphi(\tau) = P e^{-\frac{|\tau|}{\Delta}} \quad \Delta > 0 \quad \text{per definizione si ottiene lo spettro di potenza:}$$

$$G_{\text{bil}} = \frac{F[\varphi_x(\tau)]}{2\pi} = \frac{1}{2\pi} \frac{2P\Delta}{1 + \omega^2 \Delta^2} = \frac{P\Delta/\pi}{1 + \omega^2 \Delta^2}$$



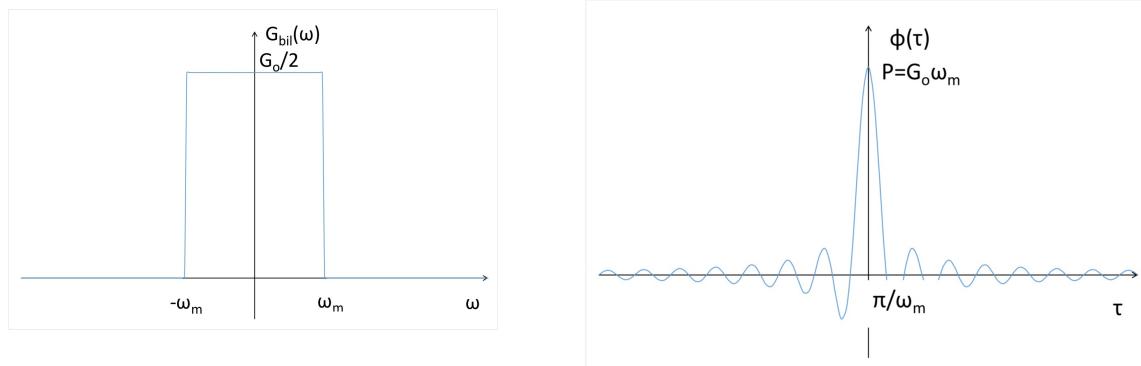
## ESEMPIO 4 - CALCOLO DELLA FUNZIONE DI AUTOCORRELAZIONE A PARTIRE DALLO SPETTO DI POTENZA

In questo caso si deve procedere all'inverso, ovvero noto lo spettro di potenza occorre ricavare la funzione di autocorrelazione, utilizzando l'antitrasformata di Fourier. Lo spettro di potenza viene dato in forma monolatera. Si ha infatti:

$$G(\omega) = \begin{cases} G_0 & 0 \leq \omega \leq \omega_m \\ 0 & \omega > \omega_m \end{cases} \rightarrow \text{ci riportiamo a uno spettro bilatero} \quad G_{\text{bil}} = \begin{cases} G_0/2 & 0 \leq |\omega| \leq \omega_m \\ 0 & |\omega| > \omega_m \end{cases}$$

Ricordando l'antitrasformata di un impulso nelle frequenze:  $x(t) = \frac{X_0 \omega_m}{\pi} \frac{\sin(\omega_m t)}{\omega_m t}$

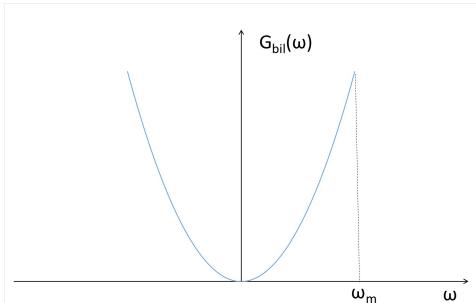
$$\Phi_x(\tau) = F^{-1}[2\pi G_{\text{bil}}(\omega)] = 2\pi \frac{G_0 \omega_m}{2\pi} \frac{\sin \omega_m \tau}{\omega_m \tau} = P \frac{\sin \omega_m \tau}{\omega_m \tau}$$



## ESEMPIO 5 - CALCOLO DELLO SPETTO DI POTENZA IN USCITA AD UN DERIVATORE

All'ingresso di una rete derivatrice si suppone presente un segnale  $x(t)$  con spettro di potenza analogo a quello dell'esercizio precedente. Quello in uscita si ottiene moltiplicandolo per il modulo al quadrato della funzione di trasferimento:

$$G_{x,\text{bil}}(\omega) = |H(\omega)|^2 G_{\text{bil}}(\omega) = |\omega|^2 G_{\text{bil}}(\omega) = \begin{cases} \omega^2 G_0/2 & 0 \leq |\omega| \leq \omega_m \\ 0 & |\omega| > \omega_m \end{cases}$$



Il valore efficace richiede il calcolo della potenza, ottenibile integrando lo spettro di potenza:

$$\dot{x}_{\text{eff}}^2 = \int_0^{\omega_m} G_x(\omega) d\omega = G_0 \int_0^{\omega_m} \omega^2 d\omega = G_0 \frac{\omega_m^3}{3} = P \frac{\omega_m^2}{3} = x_{\text{eff}}^2 \frac{\omega_m^2}{3}$$

$$\dot{x}_{\text{eff}} = x_{\text{eff}} \frac{\omega_m}{\sqrt{3}}$$

### DEFINIZIONI DI ENERGIA E POTENZA DI UN SEGNALE TEMPO DISCRETO

Definita la potenza istantanea come  $|x_n|^2$ , modulo al quadrato dell'n-esimo termine della serie temporale, derivano le seguenti definizioni di “energia” e “potenza media” (media temporale della potenza istantanea), o semplicemente potenza:

$$E = \sum_{n=-\infty}^{\infty} |x_n|^2 \quad P = \langle |x_n|^2 \rangle = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^{+N} |x_n|^2$$

Per le serie ad energia finita esiste la trasformata di Fourier. La funzione di autocorrelazione ha l'espressione:

$$c_k = \sum_{n=-\infty}^{+\infty} x_n^* x_{n+k}$$

Per le serie a potenza finita non esiste di norma la trasformata di Fourier. La funzione di autocorrelazione è data dalla seguente media temporale:

$$c_k = \langle x_n^* x_{n+k} \rangle = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^{+N} x_n^* x_{n+k}$$

CAPITOLO 9 - CENNI SUI SEGNALI DIGITALISPETTRI DI SEGNALI PAM DETERMINISTICI

I **segnali PAM** (Pulse Amplitude Modulation) sono esprimibili come convoluzione fra una serie temporale ed un impulso ad energia finita  $g(t)$ :

$$s(t) = \sum_{n=-\infty}^{\infty} a_n g(t-nT) = \{a_n\} * g(t)$$

Il segnale PAM può essere **analogico**, se la serie è analogica, ovvero se ogni termine della serie può assumere con continuità infiniti valori all'interno di un prefissato intervallo, **digitale** se invece può assumere solo un numero finito di valori.

Se la serie temporale è ad **energia finita**, allora anche il segnale PAM è ad energia finita, e la sua trasformata è data dal prodotto delle trasformate  $S(\omega) = A_s(\omega) G(\omega)$ .

Se invece la serie temporale è a **potenza finita**, il segnale PAM è anch'esso a potenza finita e non è trasformabile secondo Fourier, per cui occorre passare ad un'analisi generalizzata, ovvero calcolare il suo spettro di potenza.

Consideriamo qui il caso dei segnali PAM a potenza finita per analizzarne lo spettro. Assumiamo la serie a valore medio nullo per semplicità; si può dimostrare che la sua funzione di autocorrelazione è data dal prodotto di convoluzione delle funzioni di autocorrelazione della serie e dell'impulso, a meno di una costante  $1/T$ .

$$\varphi_s(\tau) = \frac{1}{T} \{c_n\} * \varphi_g(\tau) \quad ; c_k = \langle x_n^* x_{n+k} \rangle = \lim_{N \rightarrow \infty} \frac{1}{2N+1} \sum_{n=-N}^{+N} x_n^* x_{n+k}$$

Lo spettro di potenza del segnale PAM è dato dalla trasformata della funzione di autocorrelazione (nell'ipotesi che esista):

$$G_{s,bil}(\omega) = \frac{F[\varphi_s(\tau)]}{2\pi} \cdot \frac{1}{2\pi T} F[\varphi_g(\tau)] F[\{c_n\}] = \frac{|G(\omega)|^2}{2\pi T} \sum_{n=-\infty}^{\infty} c_n e^{-jknT}$$

Si noti che lo spettro è dato dal prodotto di due termini: il primo dipende dalla **forma dell'impulso**, il secondo dalla **autocorrelazione della serie**.

Se la serie temporale è reale, la sua autocorrelazione è reale e pari, per cui la sua trasformata può essere scritta con solo riferimento agli indici positivi:

$$\sum_{k=-\infty}^{\infty} c_n e^{-jknT} = c_0 + 2 \sum_{k=1}^{\infty} c_k \cos k\omega T \rightarrow G_{s,bil}(\omega) = \frac{|G(\omega)|^2}{2\pi T} \left[ c_0 + 2 \sum_{k=1}^{\infty} c_k \cos k\omega T \right]$$

Infine se la autocorrelazione è nulla per  $k$  diverso da zero, come spesso in pratica, lo spettro di potenza del segnale PAM coincide con lo spettro di energia dell'impulso, a meno di una costante moltiplicativa.

$$G_{s,bil}(\omega) = \frac{c_0 |G(\omega)|^2}{2\pi T}$$

## SPEZI DI SEGNALI PAM ALEATORI

Un segnale è aleatorio quando il suo andamento nel tempo è aleatorio, ovvero è una funzione aleatoria (detta anche processo stocastico). I segnali aleatori sono sempre modellati come segnali a potenza finita. Un segnale PAM è aleatorio se lo è la serie temporale. Per una serie aleatoria non è nota a priori la successione dei valori; tuttavia ciò non significa che la conoscenza della serie sia nulla, infatti la serie può sempre essere descritta in termini statistici, come successione di infinite variabili aleatorie; in particolare se la serie è stazionaria tutte le valutazioni statistiche sono indipendenti dall'indice  $n$  che denota la posizione della variabile all'interno della sequenza.

Ad esempio, se la serie temporale fosse data dai lanci successivi di un dado, il risultato dell' $n$ -esimo lancio sarebbe dato dalla variabile aleatoria  $a_n$ , la cui ben nota descrizione statistica non dipenderebbe dalla posizione per il semplice motivo che il dado non cambia fra un lancio e l'altro. A titolo di esempio, si riporta la formula del valore medio statistico di una serie stazionaria:

$$E[a_n] = \sum_{i=1}^L a^i P(a^i) \quad \text{dove gli } a^i \text{ rappresentano i valori che possono essere assunti e le } P(a^i) \text{ le rispettive probabilità.}$$

Poiché queste ultime non dipendono da  $n$ , per la stazionarietà, anche il valor medio è indipendente da  $n$ .

Per le serie aleatorie è possibile definire una autocorrelazione statistica (a priori) come media statistica del prodotto delle coppie di valori posti a distanza  $k$ , ovvero come  $E[a_n^* a_{n+k}]$ ; se la serie è stazionaria, come supporremo d'ora in avanti, la probabilità della coppia (congiunta del secondo ordine) non dipende dalla posizione delle due variabili aleatorie all'interno della sequenza, cioè dal pedice " $n$ " che rappresenta la posizione del primo elemento, ma solo dalla distanza fra gli elementi, cioè da " $k$ ":

$$C_{\text{stat},k} = E[a_n^* a_{n+k}] = \sum_{i=1}^L \sum_{\ell=1}^L (a^i)^* a^\ell P(a^i, a^\ell, k)$$

Per  $k = 0$  coincide con il momento del secondo ordine (**potenza statistica**):

$$C_{\text{stat},0} = E[a_n^* a_n] = \sum_{i=1}^L (a^i)^* a^i P(a^i) = \sum_{i=1}^L |a^i|^2 P(a^i)$$

Le variabili aleatorie a distanza  $k$  sono **incorrelate** se:

$$C_{\text{stat},k} = E[a_n^* a_{n+k}] = \begin{cases} E[|a_n|^2] & k=0 \\ E[a_n^*] E[a_{n+k}] - E[a_n^*] E[a_n] & k \neq 0 \end{cases}$$

Se il valor medio è nullo, le variabili sono incorrelate, se e solo se l'autocorrelazione è nulla per  $k$  diverso da zero. Infine se la serie è "**ergodica**" i valori medi statistici coincidono con gli analoghi temporali: quelli statistici di ordine  $n$  sono calcolabili a priori, note le probabilità congiunte di ordine  $n$ , quelli temporali sono calcolati a posteriori sulle «realizzazioni» del processo stocastico, cioè sulle serie temporali che si ottengono come risultato dell'esperimento; nel caso del dado, una realizzazione è l'effettiva sequenza dei valori ottenuta lanciando il dado consecutivamente.

Per l'ergodicità è sufficiente che la serie, oltre ad essere stazionaria, sia a memoria finita, cioè le variabili indipendenti per valori di  $k$  sufficientemente grandi.

$c_n = \langle x_n^* x_{n+u} \rangle$      $c_{n,\text{stat}} = E[a_n^* a_{n+u}]$      $\rightarrow$  Tutti i lanci del dado hanno la stessa funzione di autocorrelazione, quindi hanno tutti lo stesso spettro di potenza. Basta quindi conoscere la statistica di un segnale aleatorio, non i suoi valori.

L'autocorrelazione statistica e quella temporale coincidono per i processi ergodici:

- la prima è nota a priori ed è unica del processo;
- la seconda solo a posteriori, essendo di una realizzazione.       $c_{\text{stat},u} = c_u$

Si hanno una serie di conseguenze a catena:

- tutte le realizzazioni di un processo ergodico, anche se ovviamente diverse fra loro, hanno la stessa autocorrelazione temporale, dovendo questa coincidere con quella statistica, che è unica;
- tutte le realizzazioni dei segnali PAM, pur diverse fra loro in quanto è diversa la successione dei simboli della serie, hanno lo stesso spettro di potenza; esso si ottiene dalla formula per i segnali deterministicamente sostituendo alla autocorrelazione temporale quella statistica, data dalla:

$$G_{s,\text{bil}}(\omega) = \frac{|G(\omega)|^2}{2\pi T} \left[ c_{\text{stat},0} + 2 \sum_{k=1}^{\infty} c_{\text{stat},k} \cos k\omega T \right]$$

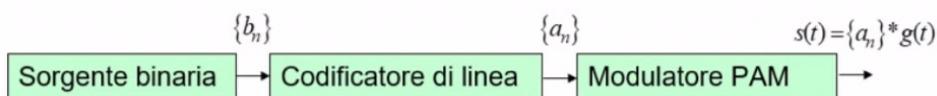
In particolare, se gli elementi della serie aleatoria sono incorrelati (sufficiente è che siano indipendenti) si ottiene infine:

$$G_{s,\text{bil}}(\omega) = \frac{E[|a_n|^2] |G(\omega)|^2}{2\pi T}$$

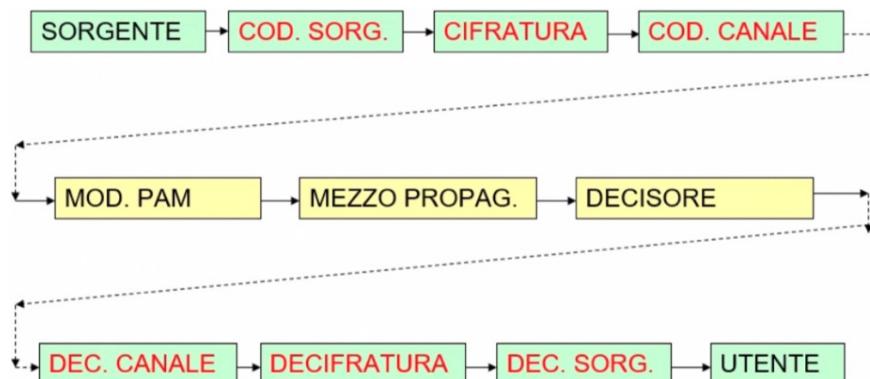
### CENNI SUI SEGNALI DIGITALI ALEATORI IN BANDA BASE

#### MODELLO SEMPLIFICATO DI UNA SORGENTE DIGITALE IN BANDA BASE

- **Sorgente binaria:** si assume che emetta bit equiprobabili e indipendenti
- **Codificatore di linea:** trasforma la serie temporale dei bit in una serie di simboli digitali; possibili scopi: eliminare il valore medio, ridurre la frequenza di simbolo, aiutare il recupero del sincronismo in ricezione, evidenziare errori, ecc.
- **Modulatore PAM:** convolve la serie dei simboli con l'impulso  $g(t)$ .



#### MODELLO COMPLETO TRASMISSIONE BANDA BASE (NON NECESSARIO PER L'ESAME)



→ I passaggi in rosso sono studiati dalla Teoria dell'Informazione, fondata da C. Shannon, che ne individuò i limiti teorici.

↓  
La teoria dei codici studia i metodi pratici.

## CODIFICA BINARIA

Questa codifica si limita a trasformare i bit a “0” in “-1”, allo scopo di rendere a valore medio nullo la serie temporale dei simboli, che ovviamente risulta reale.

$$E[a_n] = 0$$

Dato che si ha un simbolo per ogni bit, allora il tempo di simbolo  $T$  coincide con il tempo di bit  $T_b$ .  $T = T_b$

Analogamente per i loro inversi, cioè le frequenze di simbolo  $f$ , e di bit,  $f_b$ .

$$a_n \text{ indipendenti: poiché i bit sono indipendenti} \quad c_{\text{stat},k} = \begin{cases} E[a_n^2] = 1 & k=0 \\ E[a_n]E[a_n] = 0 & k \neq 0 \end{cases}$$

## CODIFICA MULTILIVELLO

La codifica multilivello è un'estensione della codifica bipolare, nella quale viene emesso un simbolo ogni 1 bit (fondamentale nelle modulazioni digitali); lo scopo è di ridurre la frequenza di simbolo.

$$T = \ell T_b \quad f_s = \frac{1}{T} = \frac{f_b}{\ell}$$

I simboli possono assumere i seguenti  $L = 2^i$  valori:  $a^i = \pm 1, \pm 3, \dots, \pm (L-1)$

- Il caso bipolare è un caso particolare di codifica multilivello ( $L=2$ ).
  - Numeri dispari per avere la stessa distanza (2) dal valore successivo.
  - Non è obbligatorio, ma conviene mappare gruppi di 1 bit in un simbolo adottando una codifica di Gray; a simboli contigui corrispondono 1-ple che differiscono di un solo bit.

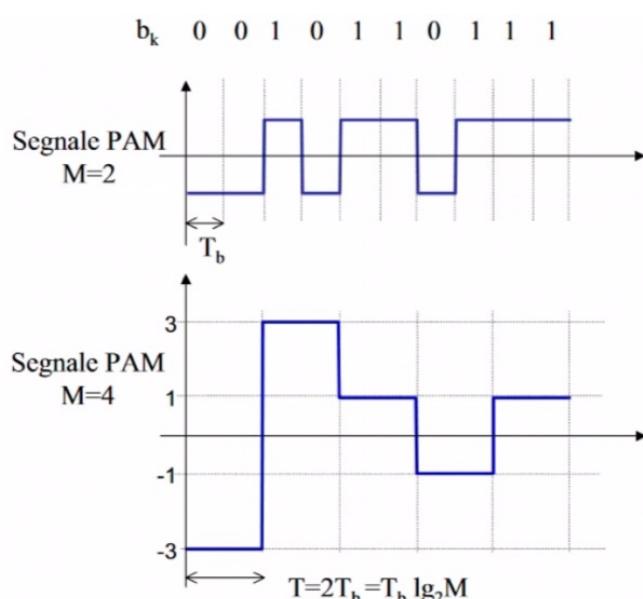
<b>I-pla</b>	<b>Simbolo</b>
000	+7
001	+5
011	+3
010	+1
110	-1
111	-3
101	-5
100	-7

PAM CODICE MULTILEVELLO ( $L=2$  E 4,  $a(t) = NRZ$ )

#### Il modulatore PAM multilevel (4)

Immaginiamo di voler trasmettere la sequenza binaria [0010110111], utilizzando un sistema di trasmissione PAM a **M = 4** livelli.

$b_{k-1} b_k$	a
0 0	-3
0 1	-1
1 1	+1
1 0	+3



Nella codifica multilivello i simboli  $a_n$  sono indipendenti ed equiprobabili, ottenuti da gruppi di  $l$  bit distinti (senza bit in comune). Il valore medio è quindi nullo mentre la funzione di autocorrelazione statistica è data da:

$$c_{\text{stat},k} = \begin{cases} E[a_n^2] = \frac{l^2 - 1}{3} & k = 0 \\ E[a_n]E[a_n] = 0 & k \neq 0 \end{cases}$$

### SPETTRI DEI SEGNALI PAM ALEATORI: CODIFICA MULTILIVELLO E IMPULSO RETTANGOLARE NRZ

Nel caso di codifica multilivello (comprendente anche il caso bipolare), la autocorrelazione è nulla tranne che nell'origine. Lo spettro del segnale PAM diventa quindi:

$$G_{s,\text{bil}} = \frac{E[a_n^2]|G(\omega)|^2}{2\pi T} \quad E[a_n^2] = \frac{l^2 - 1}{3}$$

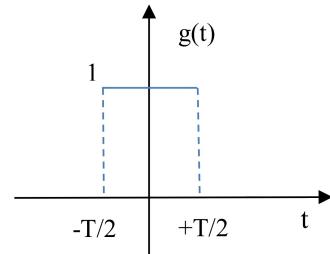
Come impulso  $g(t)$  si può prendere l'impulso rettangolare di durata  $T$  (detto anche **NRZ, No Return to Zero**);

La sua trasformata è data da:

$$G(\omega) = T \frac{\sin \omega T/2}{\omega T/2} \rightarrow G_{s,\text{bil}}(\omega) = \frac{E[a_n^2]T}{2\pi} \left| \frac{\sin \omega T/2}{\omega T/2} \right|^2$$

Conviene passare alle frequenze:

$$G_{s,f,\text{bil}}(f) = 2\pi G_{s,\text{bil}}(2\pi f) = E[a_n^2]T \left| \frac{\sin 2\pi f T/2}{2\pi f T/2} \right|^2 = E[a_n^2]T |\text{sinc } f T|^2 = \frac{E[a_n^2]}{f_s} \left| \text{sinc } \frac{f}{f_s} \right|^2$$



### SPETTO PAM: CODIFICA MULTILIVELLO E NRZ

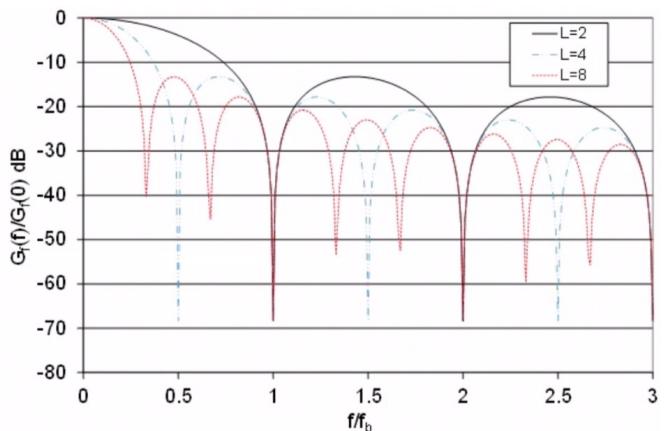
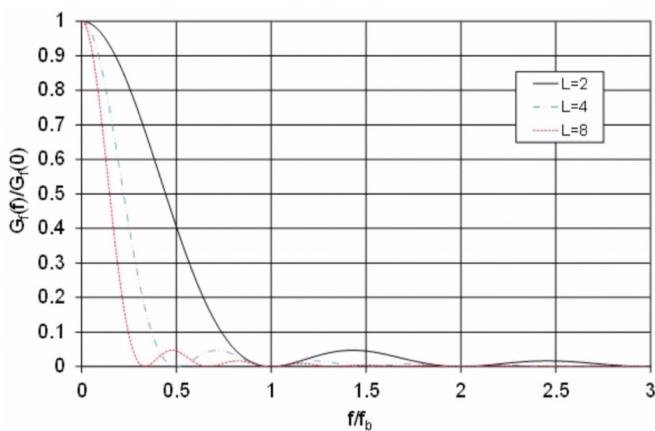
- La frequenza di simbolo coincide con la frequenza di bit nel caso bipolare ( $l=1$ ), altrimenti è ridotta di un fattore  $l$ .
- La banda del primo lobo è pari alla frequenza di simbolo.

Vantaggio del multilivello: a parità di frequenza di bit la banda è ridotta di un fattore  $l$ ;

Svantaggio: più alta probabilità d'errore a parità di altre condizioni.

PER PASSARE DA PULSAZIONI A FREQUENZE.

- 1) Sostituire gli  $\omega$  con  $2\pi f$ ;
- 2) Moltiplicare per  $2\pi$ .



## CODIFICA AMI (ALTERNATE MARKED INVERSION) (AVANZATO, FACOLTATIVO)

La codifica AMI è una codifica usata in ambito telefonico: lascia invariati i bit a zero; alterna il segno dei bit a 1. Lo scopo dell'alternanza è quello di rilevare il tasso di errore (violazione dell'alternanza dei segni).

$$a^i = 0, \pm 1 \quad P(0) = \frac{1}{2} \quad P(1) = \frac{1}{4} \quad P(-1) = \frac{1}{4}$$

$$E[a_n] = \sum_{i=1}^L a^i P(a^i) = 0 \cdot \frac{1}{2} + 1 \cdot \frac{1}{4} - 1 \cdot \frac{1}{4} = 0$$

$b_n$	$a_n$
0	0
1	+1, -1 alternati

$$C_{\text{stat},k} = \begin{cases} E[a_n^k] = \frac{1}{2} & k=0 \\ E[a_n a_{n+k}] = -\frac{1}{4} & k=1 \\ E[a_n a_{n+k}] = 0 & k>1 \end{cases}$$

Le variabili aleatorie  $a_n$  e  $a_{n+1}$  non sono indipendenti per via dell'alternanza del segno.

$$C_{\text{stat},1} = E[a_n^* a_{n+1}] = \sum_{i=1}^3 \sum_{\ell=1}^3 (a^i)^* a^\ell P(a^i, a^\ell, 1) = \\ = (1 \cdot -1) \cdot \frac{1}{8} + (-1 \cdot +1) \cdot \frac{1}{8} = -\frac{1}{4}$$

$b_n b_{n+1}$	$a_n a_{n+1}$	P
00	00	1/4
01	01, 0-1	1/8
10	10, -10	1/8
11	1-1, -11	1/8

$a^i a^\ell$	$P(a^i, a^\ell; 1)$	$P(a^i)P(a^\ell)$
00	1/4	1/2x1/2=1/4
01	1/8	1/2x1/4=1/8
0-1	1/8	1/2x1/4=1/8
10	1/8	1/4x1/2=1/8
1-1	1/8	1/4x1/4=1/16
11	0	1/4x1/4=1/16
-10	1/8	1/4x1/2=1/8
-11	1/8	1/4x1/4=1/16
-1-1	0	1/4x1/4=1/16

Valori di tutte le possibili combinazioni.

Calcolo  $C_{\text{stat},2} \rightarrow$  le variabili aleatorie  $a_n$  e  $a_{n+2}$  sono indipendenti.  $\rightarrow P(a^i, a^\ell, 2) = P(a^i)P(a^\ell)$

$b_n b_{n+1} b_{n+2}$	$a_n a_{n+1} a_{n+2}$	P
000	000	1/8
010	010, 0-10	1/16
001	001, 00-1	1/16
011	01-1, 0-11	1/16
100	100, -100	1/16
110	1-10, -110	1/16
101	10-1, -101	1/16
111	1-11, -1-1	1/16

$$C_{\text{stat},2} = E[a_n^* a_{n+2}] = \sum_{i=1}^3 \sum_{\ell=1}^3 (a^i)^* a^\ell P(a^i, a^\ell, 2) = \\ = (1 \cdot -1) \cdot \frac{1}{16} + (1 \cdot 1) \cdot \frac{1}{16} + (-1 \cdot 1) \cdot \frac{1}{16} + (-1 \cdot -1) \cdot \frac{1}{16} = 0$$

INDIPENDENTI!

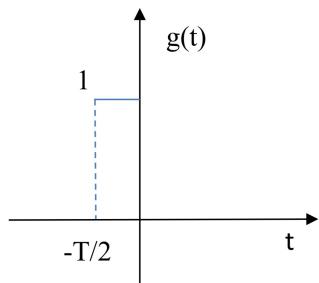
$a^i a^\ell$	$P(a^i, a^\ell; 2)$	$P(a^i)P(a^\ell)$
00	1/8+1/16+1/16=1/4	1/2x1/2=1/4
01	1/16+1/16=1/8	1/2x1/4=1/8
0-1	1/16+1/16=1/8	1/2x1/4=1/8
10	1/16+1/16=1/8	1/4x1/2=1/8
1-1	1/16	1/4x1/4=1/16
11	1/16	1/4x1/4=1/16
-10	1/16+1/16=1/8	1/4x1/2=1/8
-11	1/16	1/4x1/4=1/16
-1-1	1/16	1/4x1/4=1/16

## SPEZI DEI SEGNALI PAM ALEATORI: CODIFICA AMI E IMPULSO RETTANGOLARE RZ

$$G_{s,\text{bil}}(\omega) = \frac{|G(\omega)|^2}{2\pi T} \left[ c_{\text{stat},0} + 2 \sum_{k=1}^{\infty} c_{\text{stat},k} \cos k\omega T \right]$$

- PER AMI:

$$E[a_n^2] + 2 \sum_{n=1}^{\infty} E[a_n a_{n+k}] \cos k\omega T = \frac{1}{2} + 2 \left( -\frac{1}{4} \right) \cos \omega T = \frac{1 - \cos \omega T}{2} = \sin^2(\omega T / 2)$$



- PER RZ (RETURN TO ZERO):

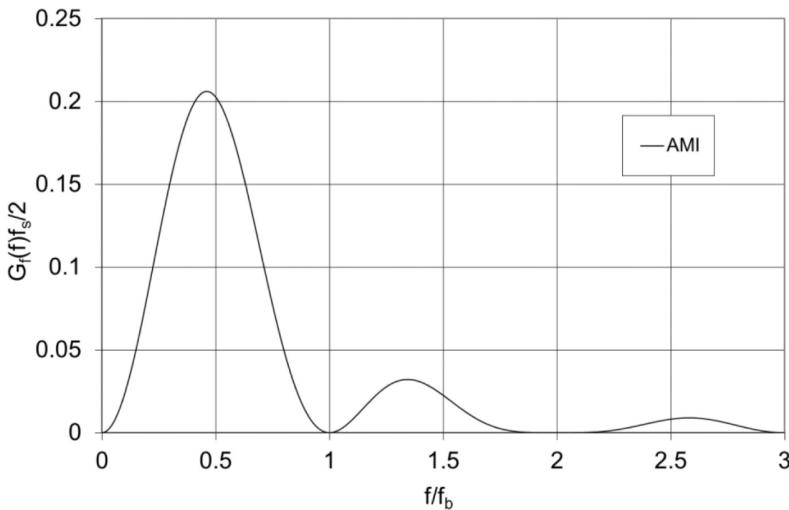
$$G(\omega) = \frac{T}{2} \frac{\sin \omega T / 4}{\omega T / 4} e^{j \omega T / 4} \rightarrow |G(\omega)|^2 = \frac{T^2}{4} \left| \frac{\sin \omega T / 4}{\omega T / 4} \right|^2$$

$$G_{s,\text{bil}}(\omega) = \frac{T^2 / 4}{2\pi T} \left| \frac{\sin \omega T / 4}{\omega T / 4} \right|^2 \sin^2(\omega T / 2) = \frac{T / 4}{2\pi} \left| \frac{\sin \omega T / 4}{\omega T / 4} \right|^2 \sin^2(\omega T / 2)$$

$$G_{s,f,\text{bil}}(f) = 2\pi G_{s,\text{bil}}(2\pi f) = \frac{T}{4} \left| \frac{\sin^2 \pi f T / 4}{2\pi f T / 4} \right|^2 \sin^2 \left( \frac{2\pi T}{2} \right) = \frac{1}{4f_s} \operatorname{sinc}^2 \left( \frac{f}{2f_s} \right) \sin^2 \left( \pi \frac{f}{f_s} \right)$$

$\nearrow$  nulli a multipli di  $f_s$        $\nwarrow$  nulli a multipli di  $2f_s$

Si noti che il sinc ha punti di nullo a multipli di due volte la frequenza di simbolo, mentre il seno a multipli di una volta sola. In altre parole, anche se un impulso RZ ha il primo lobo al doppio della frequenza di simbolo, questo non comporta un allargamento della banda del primo lobo del segnale PAM grazie alla funzione di autocorrelazione della codifica AMI.



## CENNI SULLE MODULAZIONI DIGITALI (SEGNALI DIGITALI ALEATORI PASSA-BANDA)

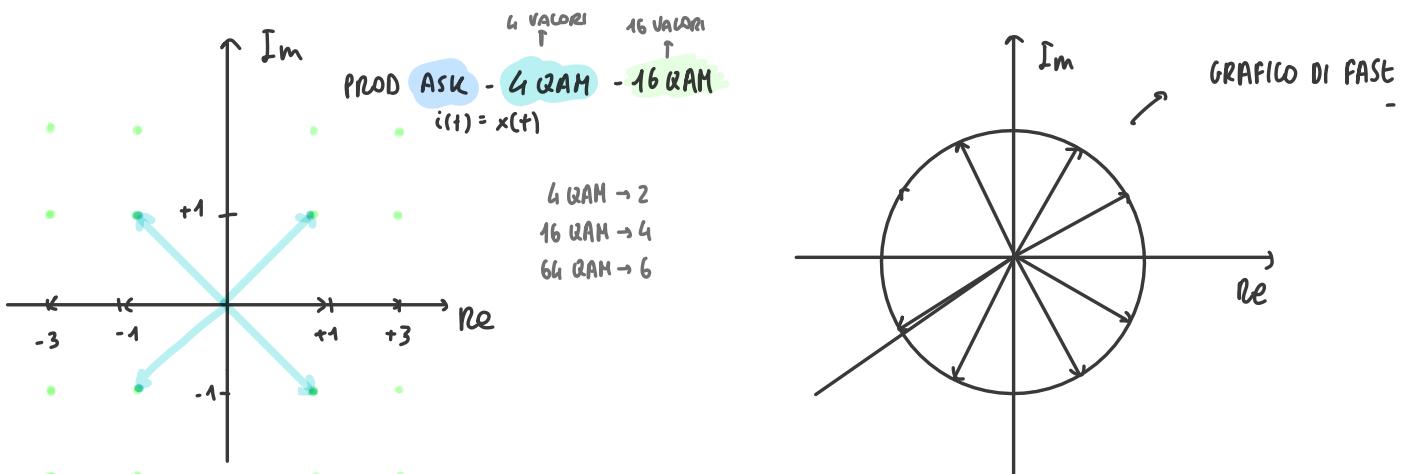
In linea di principio le modulazioni digitali si ottengono dalle modulazioni analogiche sostituendo al segnale modulante analogico un segnale modulante digitale, cioè di solito un segnale PAM a L livelli con codifica NRZ.

Esistono però modulazioni analogiche che non hanno un corrispondente digitale (l'AM), e viceversa, come l'FSK.

Nella tabella sotto vengono mostrate alcune corrispondenze, dove L indica il numero dei livelli ( $L=2^k$  cioè 2, 4, 8, 16, 32...); nella modulazione QAM si hanno due segnali modulanti, ognuno a L livelli ed  $M=L^2$  (4, 16, 64, 256, 1024, ...)

Analogica	Digitale
AM, o DSB	
Prodotto o DSB-SC	ASK, L-ASK
QAM	M-QAM, QPSK
PM	PSK, BPSK, 2-PSK, L-PSK
FM	L-CP-FSK, (MSK) FSK, L-FSK

Per alcune modulazioni esistono più nomi (ASK=2-ASK), (PSK=2-PSK=BPSK), (QPSK=4PSK). Si hanno inoltre alcune equivalenze (2-ASK=2-PSK), (4-PSK=4-QAM).



Il segnale FSK deriva dalla modulazione FM:

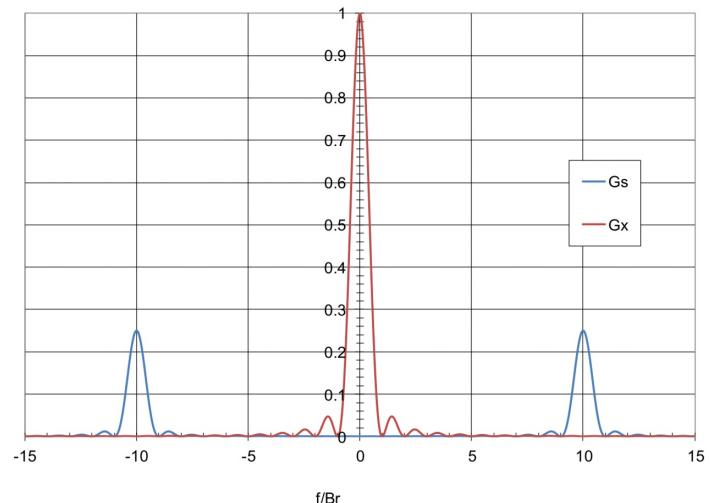
$$\begin{aligned}
 & \text{FM} \rightarrow L - \text{CP-FSK} \\
 & V_0 \cos \left[ \omega_0 t + k \int_{-\infty}^{+\infty} x(\tau) d\tau - \varphi_0 \right] \xrightarrow{\alpha(t)} \text{FSK} \\
 & f_1 \quad 0 \quad f_2 \quad 0 \quad \xrightarrow{\text{oscillazione}}
 \end{aligned}$$

Il segnale modulato  $s(t)$  è a potenza finita e per le modulazioni ASK, PSK, M-QAM il suo spettro di potenza  $G_{s,bil}(\omega)$  si ottiene dallo spettro di potenza  $G_{x,bil}(\omega)$  del segnale modulante (un PAM multilivello) in modo analogo a quanto visto per la trasformata di Fourier di una modulazione a prodotto, con l'unica differenza della costante 1/4 anziché 1/2.

Per le modulazioni di frequenza (FSK, MSK, GMSK) il calcolo dello spettro è molto più complesso.

$$G_{s,bil}(\omega) = \frac{1}{4} G_{x,bil}(\omega - \omega_0) + \frac{1}{4} G_{x,bil}(\omega + \omega_0)$$

Il segnale modulato è di norma filtrato in banda base fra  $f_s/2$  a  $f_s$ . La banda del modulato (raddoppia) è fra  $f_s$  e  $2f_s$ .



## MODULAZIONI E CODIFICHE IEEE 802.11 n (WiFi)

MCS index	Spatial streams	Modulation type	Coding rate	Data rate (Mbit/s)			
				20 MHz channel		40 MHz channel	
				800 ns GI	400 ns GI	800 ns GI	400 ns GI
0	1	BPSK	1/2	6.5	7.2	13.5	15
1	1	QPSK	1/2	13	14.4	27	30
2	1	QPSK	3/4	19.5	21.7	40.5	45
3	1	16-QAM	1/2	26	28.9	54	60
4	1	16-QAM	3/4	39	43.3	81	90
5	1	64-QAM	2/3	52	57.8	108	120
6	1	64-QAM	3/4	58.5	65	121.5	135
7	1	64-QAM	5/6	65	72.2	135	150
8	2	BPSK	1/2	13	14.4	27	30
9	2	QPSK	1/2	26	28.9	54	60
10	2	QPSK	3/4	39	43.3	81	90
11	2	16-QAM	1/2	52	57.8	108	120
12	2	16-QAM	3/4	78	86.7	162	180
13	2	64-QAM	2/3	104	115.6	216	240
14	2	64-QAM	3/4	117	130	243	270
15	2	64-QAM	5/6	130	144.4	270	300
16	3	BPSK	1/2	19.5	21.7	40.5	45
17	3	QPSK	1/2	39	43.3	81	90
18	3	QPSK	3/4	58.5	65	121.5	135
19	3	16-QAM	1/2	78	86.7	162	180
20	3	16-QAM	3/4	117	130	243	270
21	3	64-QAM	2/3	156	173.3	324	360
22	3	64-QAM	3/4	175.5	195	364.5	405
23	3	64-QAM	5/6	195	216.7	405	450
24	4	BPSK	1/2	26	28.8	54	60
25	4	QPSK	1/2	52	57.6	108	120
26	4	QPSK	3/4	78	86.8	162	180
27	4	16-QAM	1/2	104	115.6	216	240
28	4	16-QAM	3/4	156	173.2	324	360
29	4	64-QAM	2/3	208	231.2	432	480
30	4	64-QAM	3/4	234	260	486	540
31	4	64-QAM	5/6	260	288.8	540	600

MCS index	Spatial streams	Modulation type	Coding rate	Data rate (Mbit/s)			
				20 MHz channel		40 MHz channel	
				800 ns GI	400 ns GI	800 ns GI	400 ns GI
0	1	BPSK	1/2	6.5	7.2	13.5	15
1	1	QPSK	1/2	13	14.4	27	30
2	1	QPSK	3/4	19.5	21.7	40.5	45
3	1	16-QAM	1/2	26	28.9	54	60
4	1	16-QAM	3/4	39	43.3	81	90
5	1	64-QAM	2/3	52	57.8	108	120
6	1	64-QAM	3/4	58.5	65	121.5	135
7	1	64-QAM	5/6	65	72.2	135	150

Si esamina 20MHz, 800ns GI. Ho 20 MHz di banda, assumiamo temporaneamente di avere un'unica modulazione con  $f_s = 13$  Msimboli/s

- $m = \text{bit/simbolo}$  (BPSK=1, QPSK=2, 16-QAM=4, 64-QAM=6)

- $f_s = R_s * f_s * m$

All'aumentare di  $R_c$ , diminuisce la ridondanza per bit e la resistenza al rumore.

All'aumentare di  $m$  migliora l'efficienza spettrale ma diminuisce la resistenza al rumore.

Fissata una BER, il sistema sceglie dinamicamente l'indice MCS più alto che la gara

Come le marce di un cambio automatico al variare della pendenza (MCS=7, max velocità).

OFDM (ORTHOGONAL FREQUENCY DIVISION MULTIPLEXING) (FACOLTATIVO)

L'OFDM è una tecnica di multiplazione di frequenza per segnali digitali, basata sull'utilizzo combinato di un insieme di sottoportanti fra loro ortogonali. E' al momento utilizzatissima in tutti gli ambiti delle telecomunicazioni (WiFi, ADSL, DVB, 4G...).

In sintesi, nell'OFDM anziché inviare un unico flusso dati, con un tempo di simbolo brevissimo, si preferisce avere N flussi dati in parallelo, con un tempo di simbolo aumentato di un fattore N. Ognuno di questi flussi modulerà una diversa sottoportante.

Il vantaggio principale consiste nella maggiore robustezza contro l'interferenza intersimbolo, causata nelle applicazioni radio dalla presenza di cammini multipli («echi»).

Se il tempo di simbolo è lungo, è possibile inserire un intervallo di guardia di durata maggiore della risposta impulsiva del canale, in modo da esaurire gli echi di un simbolo prima della trasmissione del simbolo successivo.

Ogni sottoportante è modulata con una modulazione numerica (di solito M-QAM) il cui numero di livelli può cambiare per adattarsi al canale (migliore il rapporto segnale rumore, più alto il numero di bit per simbolo che si possono inviare). In alcuni casi il numero di livelli, in un dato momento, è lo stesso per tutte le sottoportanti (es WiFi). In altri può variare da sottoportante a sottoportante (ADSL, ecc.). In questo caso l'OFDM viene anche chiamata DMT (Discrete Multi Tone).

La modulazione OFDM viene realizzata mediante una FFT inversa, per cui N è potenza di 2.

CAMMINI MULTIPLI E TEMPO DI GUARDIA (FACOLTATIVO)

Se ad esempio uno invia su un canale radio un milione di simboli al secondo usando una modulazione convenzionale ad unica portante, la durata di ogni simbolo è di un microsecondo. Questo rende difficile contrastare l'interferenza intersimbolo causata dai cammini multipli (o «echi»), la cui dispersione temporale è dovuta alla differenza di lunghezza dei vari percorsi (riflessioni, ecc.).

La dispersione temporale è data dalla lunghezza della risposta impulsiva (senza limitazioni di banda si ha una delta per ogni cammino).

Se invece si usano in parallelo 1000 portanti, la durata di ogni simbolo aumenta in linea di principio di un fattore 1000 e diventa di 1ms.

La maggiore durata rende tollerabile l'inserimento di un tempo di guardia: piccolo rispetto al tempo di simbolo (per limitare la perdita di efficienza) ma allo stesso tempo maggiore della durata della risposta impulsiva (per esaurire gli echi prima della ricezione del simbolo successivi).

L'interferenza intersimbolo può essere evitata se la dispersione temporale (intervallo fra ricezione primo e ultimo eco) è minore del tempo di guardia. Prendendo questo ad esempio pari a 1/8 del tempo di simbolo (125micros), si ottiene una differenza massima di 37.5 km fra il percorso più corto e quello più lungo.

La dispersione temporale è molto maggiore all'aperto che all'interno di edifici, e non ha nulla a che fare con le caratteristiche del segnale inviato.

## TEMPO UTILE E TEMPO DI SIMBOLI

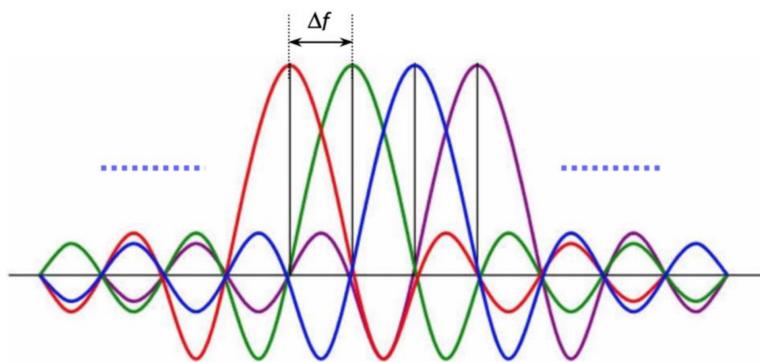
L'OFDM divide una banda  $B$  in  $N$  sottoportanti equidistanziate di  $\Delta f = B/N$ .  $N$  è la dimensione della FFT, ad esempio 64.

Le portanti sono ortogonali su  $T=1/\Delta f$  perché su  $T$  ognuna di esse ha un numero intero di periodi (vedi dimostrazione su Calandrino Chiani, Comunicazioni Elettriche, Pitagora).

Si usa quindi  $T$  come tempo utile (o di integrazione) di simbolo, o  $T_u$ .

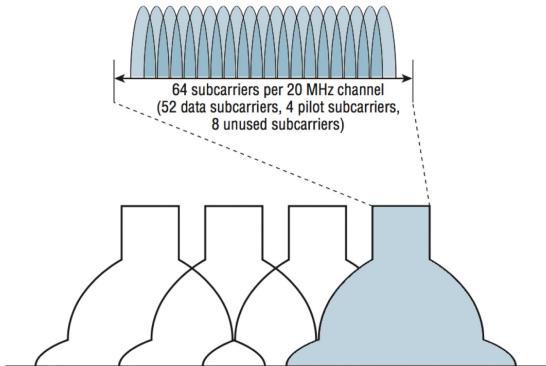
Al tempo utile viene aggiunto un tempo di guardia  $T_g$  per permettere ai cammini multipli di esaurirsi evitando così l'interferenza intersimbolo, ottenendo così il tempo di simbolo OFDM,  $T_{OFDM}$ .

Ogni portante sarà modulata a  $L$  livelli (su due vie nel caso dell' M-QAM) quindi porterà  $m$  bit per ogni tempo di simbolo OFDM. Nel WiFi  $m$  è lo stesso per tutte le portanti, mentre nell'ADSL e in altri casi no. L'inverso del tempo di simbolo OFDM fornisce la frequenza dei simboli OFDM.



### ESEMPIO 802.11 a,g,n,ac : SOTTOPORTANTI (CANALE 20 MHz)

- $B=20 \text{ MHz}$ ,  $N=64$  (dimensione FFT)
  - $\Delta f=B/N=312.5 \text{ kHz}$
  - $T_u (\text{useful}) = 1/\Delta f=3.2 \mu\text{s}$
- 64 sottoportanti
  - WiFi a/g: 48 dati, 4 pilota, 12 nulle
  - WiFi n/ac: 52 dati, 4 pilota, 8 nulle



- Per permettere agli echi di esaurirsi, si aggiunge al tempo utile  $T_u$  un tempo di guardia  $T_g$  circa 4 volte la dispersione temporale prevista (Interni <100ns, esterni maggiore), ottenendo  $T_{OFDM}$ .
- Il suo inverso è la frequenza di simbolo OFDM. Moltiplicando per le sottoportanti si ottiene la frequenza di simbolo dati.
- Per canale a 20 MHz (48 o 52 sottoportanti dati),  $T_u=3.2 \mu\text{s}$ . Posso avere due tempi di guardia:
- $T_g (\text{long})=0.8 \mu\text{s}$ 
  - $T_{OFDM}=T_u+T_g=4 \mu\text{s}$ 
    - $f_{\text{OFDM}}=1/T_{OFDM}=250 \text{ kSymbols/s}$
    - $f_{\text{data}}=f_{\text{OFDM}}*52=13 \text{ Msymbols/s (n,ac)}$
- $T_g (\text{short})=0.4 \mu\text{s}$ 
  - $T_{OFDM}=T_u+T_g=3.6 \mu\text{s}$ 
    - $f_{\text{OFDM}}=1/T_{OFDM}=277.778 \text{ kSymbols/s}$
    - $f_{\text{data}}=f_{\text{OFDM}}*52=14.444 \text{ Msymbols/s (n,ac)}$
- Nel WiFi tutte le sottoportanti hanno la stessa modulazione e la stessa code rate (indice MCS), per cui indicando con  $m$  i bit per simbolo della modulazione:
  - $f_b=R_c * f_{\text{data}} * m$

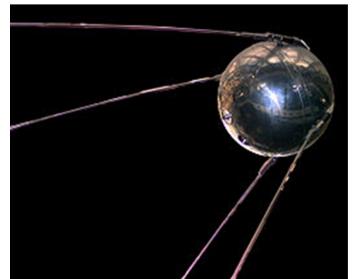
→ FACOLTATIVO

SECONDA PARTE DEL CORSO: INTRODUZIONE E CENNI STORICI SU INTERNET

1957 —

4 ottobre 1957 Viene lanciato il primo satellite artificiale, lo **Sputnik**.

- Non è geostazionario e non è un satellite per telecomunicazioni. Ha però una radio a bordo che emette dei bip, ascoltabili dai radioamatori di tutto il mondo. È lucido per permetterne una più facile osservazione agli astrofili dilettanti.
- È figlio della Guerra Fredda, ed in particolare della ricerca sui missili balistici intercontinentali (ICBM).
- L'impatto propagandistico è enorme. Il predominio tecnologico USA è messo per la prima volta in discussione.
- Inizia la corsa allo spazio.



2 mesi dopo (6 dicembre 1957): Il razzo USA Vanguard (US NAVY) **Kaputnik** con a bordo il satellite VT3 esplode sulla rampa di lancio in diretta TV.

- Alla perdita del primato si aggiunge l'umiliazione del fallimento.
- La stampa USA è feroce.
- Al centro ABMA (US ARMY), dove lavora lo scienziato tedesco Werner Von Braun (progettista delle V1 e V2), prima bloccato per motivi politici, viene chiesto di porre rimedio nel più breve tempo possibile.



1958 —

2 mesi dopo (31 gennaio 1958): il satellite USA **Explorer 1**, realizzato in 84 giorni da Caltech

JPL viene messo in orbita da un razzo Jupiter-C

- Nel febbraio 1958 viene fondata l'ARPA (Advanced Research Projects Agency, poi DARPA) Lo scopo è quello di assicurare la supremazia tecnologica degli Stati Uniti.
- 29 luglio 1958: Fondata la **NASA** (National Aeronautics and Space Administration).



ANNI 60 —

Paul Baran (RAND Co.) iniziò ad interessarsi alla possibilità di realizzare una rete di telecomunicazioni in grado di sopravvivere ad un attacco nucleare nei primi anni 60. **Nascita di Internet**.

Nel sistema telefonico tradizionale, a causa della struttura gerarchica, fra A e B un solo percorso è possibile.

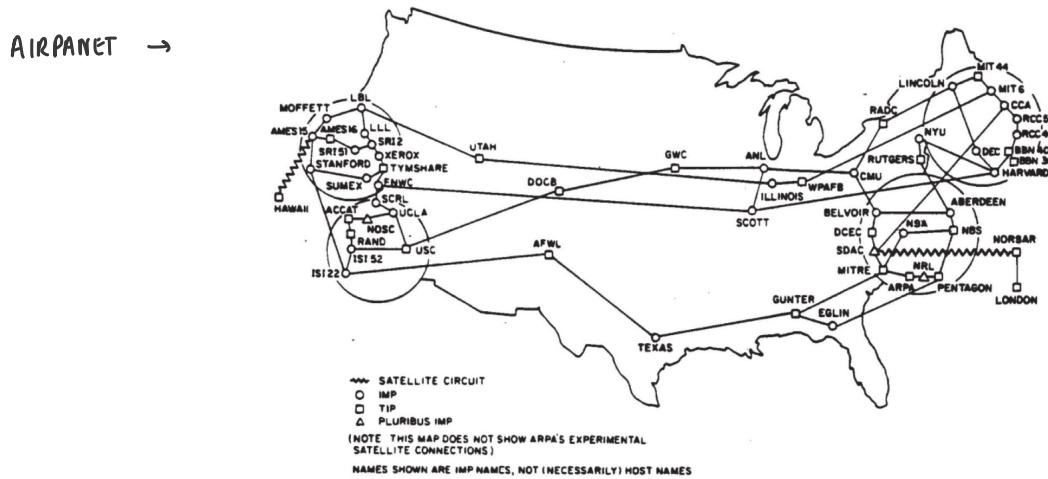
## BASI DI INTERNET

### Elementi essenziali del progetto:

- Architettura distribuita (non un unico punto di fallimento) e ridondante (più percorsi)
- Comutazione di pacchetto (Packet switching) di tipo «connectionless» al posto di commutazione di circuito (Circuit switching)
- La commutazione di pacchetto divide i messaggi in pacchetti di lunghezza arbitraria; se “connectionless” ogni pacchetto («datagram») è instradato autonomamente come una lettera.

### Scopo:

- A patto che esista un percorso continuo fra sorgente e destinazione la comunicazione deve essere possibile.
- Il percorso attraverso i nodi intermedi deve essere trovato automaticamente e deve essere dinamico
- Vedremo che le DTN vanno oltre e rilasciano anche il vincolo del percorso continuo!



## EVOLUZIONE



- 1963: Memorandum for Members and Affiliates of the Intergalactic Computer Network, from J. C. R. Licklider (ARPA)
- A joke by a visionary man (visionary=having or showing clear ideas about what should happen or be done in the future)
- 1969: First man on the Moon on 21 July
- 1969: First message on the ARPANET on October 29 • (“lo” for “login”, but after 2 characters the host crashed)
- 1973: TCP/IP Protocols
- by Vinton Cerf and Bob Kahn
- 1991: World Wide Web birth (first web site)
- by Berners-Lee and Robert Cailliau at CERN, HTML language, HTTP protocol
- 2001: Inter Planetary Internet Architecture studies start (DARPA).

**Internet è basata su un software aperto.**

## MODELLO DI RETE

Ci sono due tipi di tecnologie di trasmissione diffusamente impiegati: i collegamenti **broadcast** e i collegamenti punto a punto (**point to point**).

I collegamenti punto a punto connettono coppie di computer. Brevi messaggi, chiamati in alcuni contesti **pacchetti**, possono dover visitare una o più macchine intermedie per spostarsi dalla sorgente alla destinazione in una rete composta di collegamenti punto a punto. Spesso sono possibili più percorsi di diversa lunghezza; quindi, nelle reti punto a punto è importante trovarne di validi. La trasmissione punto a punto con esattamente un trasmettitore e un ricevitore è a volte chiamata **unicast**.

Al contrario, le reti broadcast hanno un solo canale di comunicazione condiviso da tutte le macchine della rete. I pacchetti inviati da qualunque macchina sono ricevuti da tutte le altre. Un campo indirizzo all'interno del pacchetto individua il destinatario. Alla ricezione del pacchetto ogni macchina controlla il campo indirizzo. Se il pacchetto è indirizzato alla macchina ricevente viene processato; se è indirizzato a un'altra macchina viene semplicemente ignorato.

Una rete wireless è un classico esempio di collegamento broadcast in cui la comunicazione è condivisa in una regione di copertura che dipende dal canale wireless e dalla macchina che trasmette.

I sistemi broadcast danno di solito anche la possibilità d'indirizzare un pacchetto a tutti i destinatari usando un codice speciale nel campo destinazione. Quando è trasmesso un pacchetto con questo codice, esso viene ricevuto e processato da tutte le macchine sulla rete.

Un criterio alternativo per classificare le reti è la loro scala. La distanza è una metrica di classificazione importante, perché reti su scale differenti impiegano tecnologie diverse.

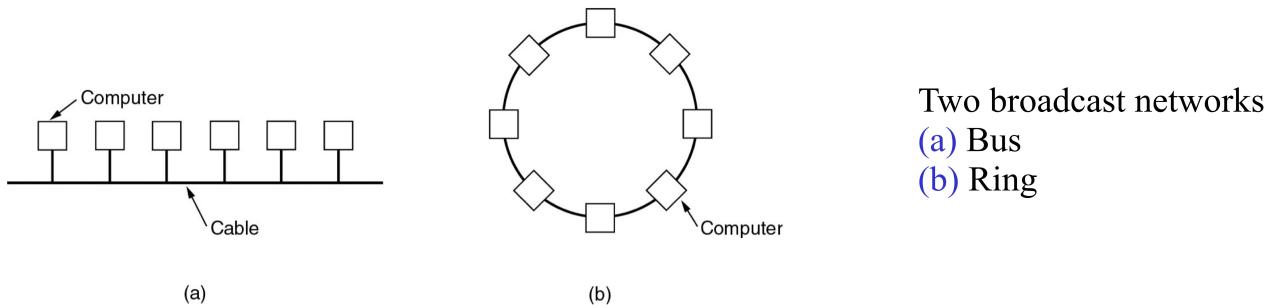
Interprocessor distance	Processors located in same	Example
1 m	Square meter	Personal area network
10 m	Room	Local area network
100 m	Building	
1 km	Campus	
10 km	City	Metropolitan area network
100 km	Country	
1000 km	Continent	
10,000 km	Planet	The Internet

- **PAN** (Personal Area Network): rete molto piccola ristretta a pochi pc
- **LAN** (Local Area Network): rete locale la cui dimensione può essere relativa a un edificio o un campus, quindi di dimensioni geografiche limitate.
- **MAN** (Metropolitan Area Network): Rete che copre un intera città
- **WAN** (Wide Area Network): rete che copre un'intero stato o continente
- **Internet**: copre l'intero pianeta

## LOCAL AREA NETWORKS:

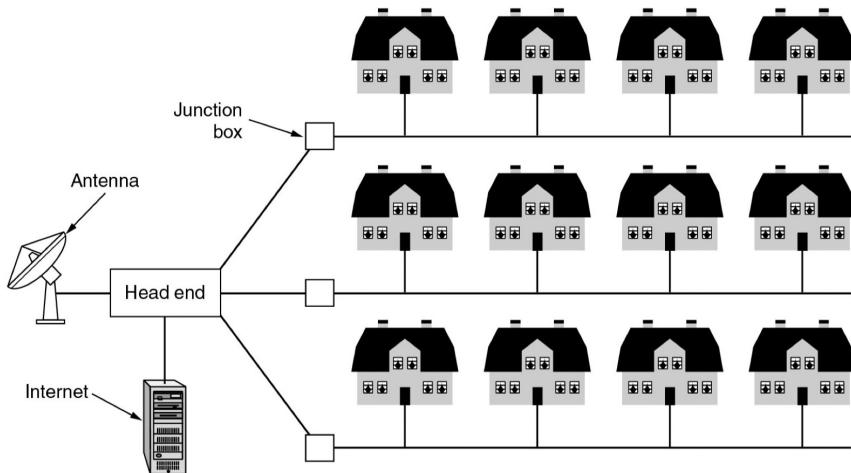
**LAN** (local area network, reti locali). Una rete LAN è una rete privata che opera all'interno o nelle vicinanze di un singolo edificio come un appartamento, un ufficio o una fabbrica. Le reti broadcast sia wireless sia cablate si possono ulteriormente dividere in statiche e dinamiche, a seconda del modo in cui è allocato il canale.

I metodi di allocazione dinamica per un canale condiviso possono essere centralizzati o non centralizzati. Nel metodo di allocazione centralizzato esiste una singola entità, per esempio la stazione base nelle reti cellulari, che stabilisce a chi spetta di volta in volta l'uso del mezzo. Può farlo accettando richieste e prendendo decisioni in base ad algoritmi interni. Nel metodo di allocazione non centralizzato non esiste un'entità centrale; ogni macchina deve decidere in autonomia se trasmettere o meno.



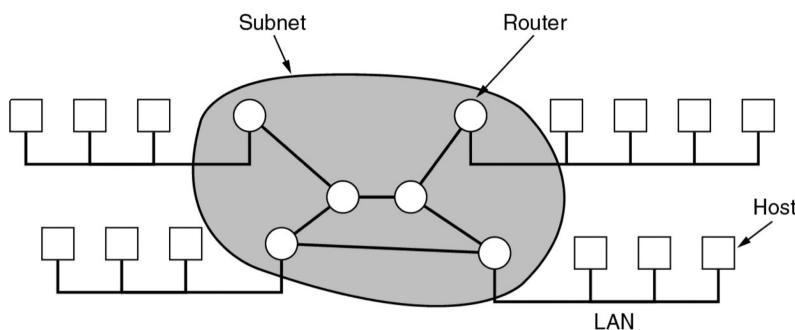
## METROPOLITAN AREA NETWORKS:

Una **MAN** (metropolitan area network, rete metropolitana) copre un'intera città. L'esempio più noto di MAN è la rete di TV via cavo disponibile in molte città degli Stati Uniti. Questo sistema si è sviluppato partendo dagli originari sistemi di antenna comunitari, installati nelle zone con cattiva ricezione della televisione via etere. In questi sistemi primitivi veniva installata una grande antenna sulla cima di una collina nelle vicinanze e da qui il segnale era trasmesso alle abitazioni.



## WIDE AREA NETWORKS:

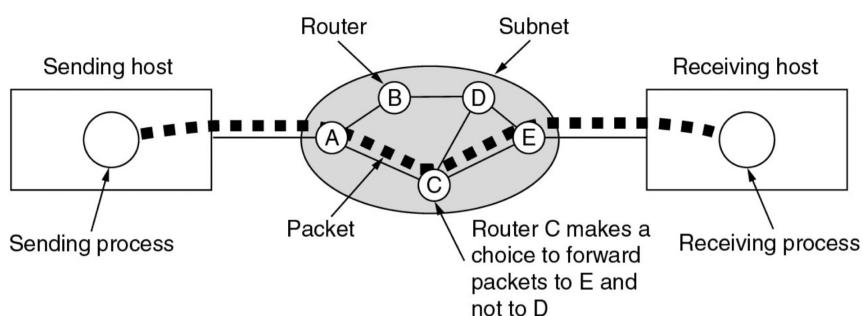
Una **WAN** (wide area network, rete geografica) copre un'area geograficamente estesa, spesso una nazione o un continente. In una WAN cablata, ad esempio un'azienda con uffici sparsi in varie città, ogni ufficio dispone di un computer su cui girano programmi (applicazioni) degli utenti. Chiameremo queste macchine *host* (letteralmente “ospiti”, in quanto ospitano software e utenti). La parte restante della rete che connette questi host è quindi chiamata la sottorete di comunicazione o in breve sottorete (*subnet*). Il compito di una sottorete è quello di trasportare messaggi da un host all'altro, esattamente come il sistema telefonico trasporta le parole (in realtà suoni) da chi parla a chi ascolta.



Le WAN, come le abbiamo descritte, sembrano simili a una LAN cablata su larga scala, ma vi sono importanti differenze che vanno al di là della lunghezza dei cavi. Usualmente in una WAN gli host e la sottorete hanno proprietari e proprietari diversi.

Una seconda differenza consiste nel fatto che i router connettono usualmente reti diverse dal punto di vista tecnologico. Le reti all'interno di uffici usano switch Ethernet, per esempio, mentre le linee di trasmissione su lunga distanza implementano connessioni SONET.

L'ultima differenza concerne ciò che si connette alla sottorete. Potrebbero essere singoli computer, come quando ci si connette a una rete LAN, o potrebbe essere un'intera LAN.



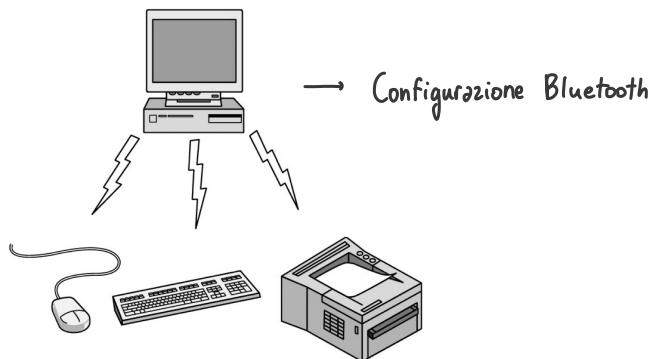
Esistono anche altri due tipi di WAN:

- **VPN** (virtual private network, rete privata virtuale). In confronto a una linea dedicata, una VPN presenta il vantaggio, tipico delle virtualizzazioni, di fornire un riutilizzo
- La seconda variante è che la sottorete può essere utilizzata da diverse aziende. L'operatore della sottorete è chiamato fornitore (o provider) di servizi di rete e gli uffici sono suoi clienti. Un tale operatore di sottorete è chiamato un **ISP** (internet service provider) e la sottorete prende il nome di rete ISP; i clienti che si connettono a un ISP ricevono un servizio di connessione a Internet.

## WIRELESS NETWORKS:

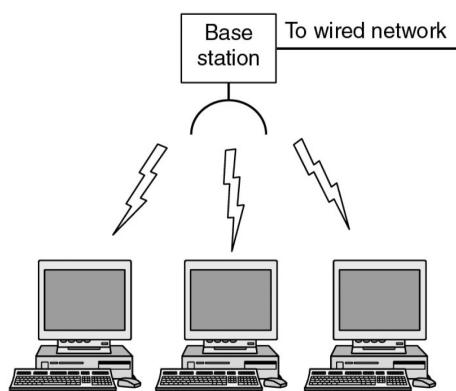
- **Reti interconnesse:**

Un insieme di reti interconnesse si chiama **internetwork** o **internet** (con la i minuscola). Questi termini sono usati nel loro significato generico, in contrasto con la Internet mondiale (che è una specifica internet) che indicheremo sempre con l'iniziale maiuscola. La internet mondiale usa le reti ISP per connettere reti aziendali, reti domestiche e molte altre reti. Tendenzialmente, se differenti parti della rete implementano differenti tecnologie (per esempio broadcast rispetto a punto a punto e cablata rispetto a wireless) probabilmente abbiamo una internetwork.



- **LAN wireless:**

Le reti LAN wireless sono attualmente molto diffuse. In questi sistemi ogni computer ha un ricevitore radio e un'antenna per comunicare con altri computer. Nella maggior parte dei casi ogni computer comunica con un dispositivo posto sul soffitto, chiamato AP (access point), router wireless, o base station. Tuttavia, se gli altri computer sono abbastanza vicini tra loro, possono comunicare direttamente in una configurazione peer-to-peer. Lo standard per le reti LAN wireless è IEEE 802.11, comunemente noto come Wi-Fi e attualmente molto diffuso.



- **WAN wireless:**

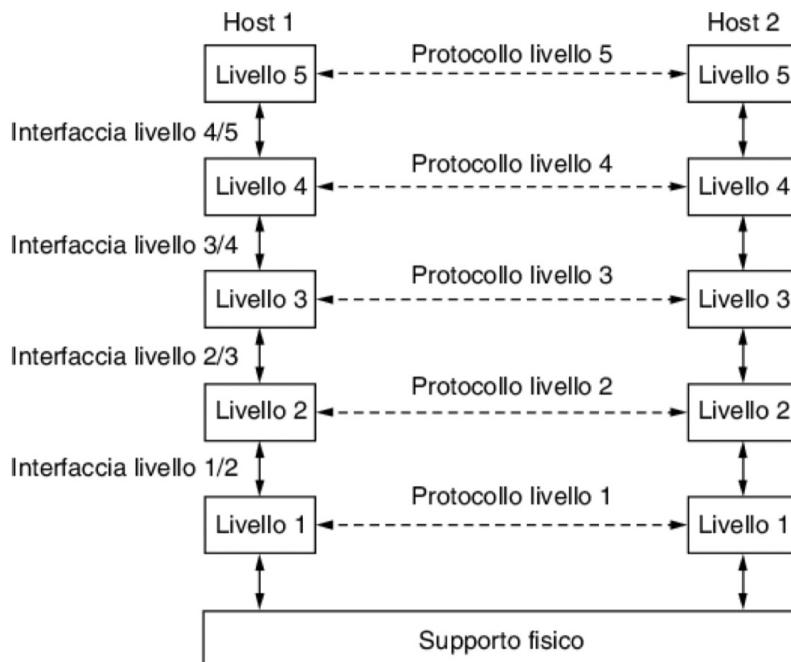
Alcuni tipi di reti WAN fanno uso di tecnologie wireless: nei sistemi satellitari ogni computer a terra ha un'antenna attraverso la quale può trasmettere e ricevere dati da un satellite in orbita. Tutti i computer possono ascoltare messaggi provenienti dal satellite e in alcuni casi anche le trasmissioni trasmesse da terra da altri computer verso il satellite. Le reti satellitari sono intrinsecamente broadcast.

Un altro esempio di WAN wireless è la rete telefonica cellulare.

## SOFTWARE DI RETE - GERARCHIE DEI PROTOCOLLI

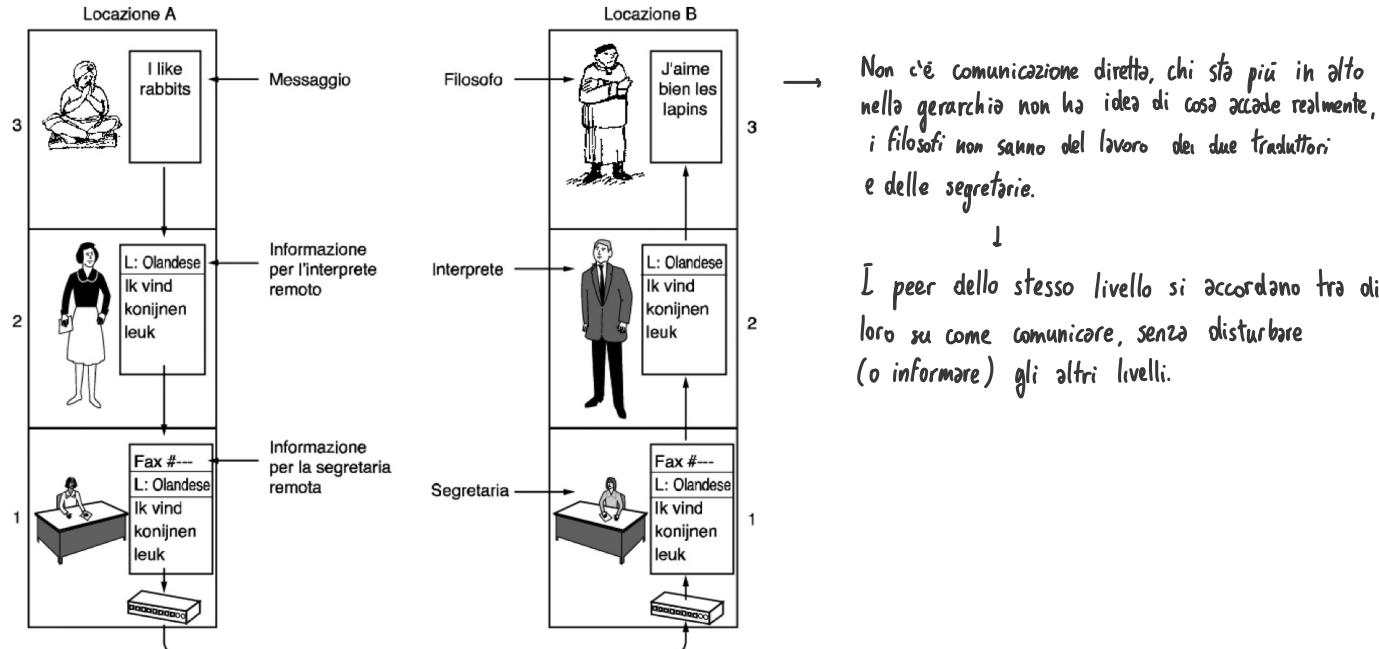
Per diminuire la complessità, la maggior parte delle reti è organizzata come una pila di livelli (*layer*) o *strati*, costruiti uno sull'altro. Numero, nome, contenuto e funzione di ogni livello variano da rete a rete. Lo scopo di ogni livello è quello di offrire determinati servizi ai livelli superiori, schermendola dai dettagli implementativi.

Quando il livello  $n$  all'interno di un computer è in comunicazione con il livello  $n$  di un altro computer, le regole e le convenzioni usate in questa comunicazione sono globalmente note come protocolli del livello  $n$ . Fondamentalmente, un **protocollo** è un accordo tra le parti che comunicano sul modo in cui deve procedere la comunicazione. Le entità che formano i livelli di pari grado su diversi computer sono chiamati **peer** (*pari*). I peer possono essere processi software, dispositivi hardware o esseri umani. Sono i peer a comunicare tra loro usando il protocollo.

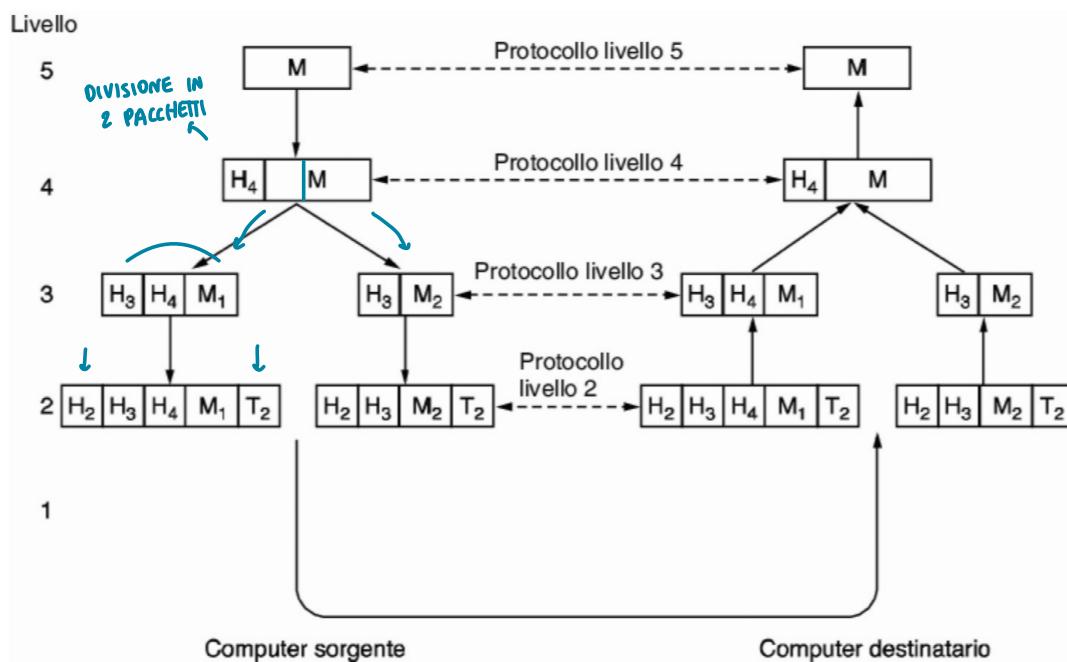


In realtà i dati non sono trasferiti direttamente dal livello  $n$  di un computer al livello  $n$  di un altro. Ogni livello passa i dati a quello sottostante, fino a raggiungere il più basso. Sotto al livello 1 si trova il **supporto fisico** attraverso cui è possibile la comunicazione vera e propria. Tra ciascuna coppia di livelli contigui si trova un'**interfaccia**, che definisce le operazioni elementari e i servizi che il livello inferiore rende disponibili a quello soprastante. L'insieme di livelli e protocolli si chiama **architettura di rete**. L'elenco dei protocolli usati da uno specifico sistema è chiamato **pila di protocolli** (*protocol stack*).

L'idea di una comunicazione multilivello può essere resa efficacemente dalla seguente analogia: immaginiamo due filosofi (peer a livello 3), che non hanno linguaggi in comune; ciascuno assume un interprete (peer a livello 2) che a propria volta contatta una segretaria (peer a livello 1). Il filosofo 1 passa un messaggio (in inglese) attraverso l'interfaccia 2/3 al suo interprete che dice "mi piacciono i conigli". Gli interpreti si sono accordati su un linguaggio neutro noto a entrambi, l'olandese, cosicché il messaggio viene tradotto (la scelta della lingua è il protocollo del livello 2 e dipende dai peer di quel livello). L'interprete, quindi, da il messaggio alla segretaria per la trasmissione, cosicché arrivi all'altra segretaria, che lo passa all'interprete, che lo traduce, e arriva al filosofo 2.



Consideriamo ora un esempio più tecnico: un messaggio M è prodotto da un processo applicativo che lavora al livello 5 e passato al livello 4 per la trasmissione. Il livello 4 mette un **header (intestazione)** davanti al messaggio per identificarlo e passa il risultato al livello 3. L'header include informazioni di controllo, come l'indirizzo, per consentire al livello 4 del computer destinatario di consegnare il messaggio.



In molte reti non c'è un limite alla dimensione dei messaggi trasmessi nel protocollo del livello 4, ma quasi sempre c'è un limite imposto dal protocollo del livello 3. Di conseguenza, il livello 3 deve spezzare i messaggi in arrivo in unità più piccole chiamate *pacchetti*, aggiungendo un header di livello 3 davanti ad ogni pacchetto. In questo esempio M è diviso in due parti, M<sub>1</sub> ed M<sub>2</sub> che vengono trasmessi separatamente.

Il livello 3 decide la linea di uscita da usare e passa i pacchetti al livello 2. Il livello 2 aggiunge a ogni pezzo un **trailer (informazione aggiuntiva in coda)** e lo passa al livello 1 per la trasmissione fisica. Nel computer destinatario il messaggio si muove verso l'alto, passando da livello in livello, e gli header vengono rimossi man mano.

Nessuno degli header per i livelli inferiori a n viene passato al livello n.

L'astrazione dei peer è fondamentale per la progettazione delle reti. Usandola è possibile spezzare in diverse parti più piccole e gestibili il compito impossibile di progettare l'intera rete, poiché basta progettare i singoli livelli.

### PROGETTAZIONE DEI LIVELLI

Alcuni problemi fondamentali nella progettazione delle reti si presentano livello dopo livello:

- 1) Controllo degli errori:** individuazione di errori (*error detection*) e correzione degli errori (*error correction*).
- 2) Routing (instradamento):** consiste nel trovare un percorso valido attraverso la rete, che dovrebbe prendere questa decisione in modo automatico.
- 3) Addressing o naming:** ogni livello richiede un meccanismo per identificare chi trasmette e chi riceve un particolare messaggio.
- 4) Multiplexing:** molti progetti di rete condividono dinamicamente la banda concedendola alle necessità a breve termine degli host, piuttosto che assegnare a ogni host una frazione prefissata di banda che ciascuno può o meno usare. Tale condivisione viene applicata nei livelli bassi a un singolo link o nei livelli alti a una rete e a applicazioni che usano la rete.
- 5) Flow control (controllo di flusso):** è un feedback da parte del ricevente alla sorgente, utilizzato nell'eventualità per impedire che una sorgente veloce inondi di dati un ricevente lento. A volte la rete è intasata perché troppi computer cercano di inviare un traffico eccessivo di dati e non riesce a consegnare i pacchetti. Tale sovraccaricamento della rete è chiamato **congestione**.

### SERVIZI ORIENTATI ALLA CONNESSIONE E SENZA CONNESSIONE

Ogni livello può offrire a quelli sovrastanti due tipi diversi di servizio: orientati alla connessione oppure senza connessione.

Un **servizio orientato alla connessione** assomiglia al sistema telefonico. Per parlare con qualcuno si deve prendere il telefono, comporre il numero, parlare e poi riagganciare. Allo stesso modo, per usare un servizio di rete orientato alla connessione, l'utente deve innanzitutto stabilire una connessione, usarla e quindi rilasciarla. L'aspetto essenziale di una connessione è che funziona come un tubo: il trasmettitore vi spinge oggetti (bit) a una estremità e il ricevitore li prende dall'altra. Nella maggior parte dei casi l'ordine è conservato, quindi i bit arrivano nella sequenza secondo cui sono stati trasmessi.

Al contrario del servizio orientato alla connessione, un **servizio senza connessione** si comporta come il servizio postale. Ogni messaggio (lettera) porta l'indirizzo completo del destinatario ed è instradato attraverso il sistema postale in modo indipendente dai messaggi successivi. I messaggi assumono nomi diversi a seconda del contesto; a livello rete un messaggio prende il nome di pacchetto. Ogni nodo intermedio lungo il percorso deve ricevere completamente n pacchetto prima di inoltrarlo al nodo successivo; questo approccio viene chiamato **store-and-forward**.

Normalmente, quando due messaggi vengono inviati allo stesso destinatario, il primo che viene spedito è anche il primo che arriva. Può tuttavia accadere che il primo che è stato spedito subisca un ritardo e che quindi il secondo arrivi per primo.

Ogni tipo di servizio può ulteriormente essere caratterizzato in base alla sua **affidabilità**. Alcuni servizi sono affidabili, nel senso che non perdono mai dati.

In alcune applicazioni il ritardo di trasferimento introdotto dalle conferme di ricezione è inaccettabile. Una di queste applicazioni è il traffico vocale digitalizzato per il **voice over IP**. Per un utente telefonico è meglio ascoltare un po' di rumore sulla linea di tanto in tanto che percepire un ritardo nell'attesa delle conferme.

Un servizio senza connessione non affidabile (cioè privo di conferma) spesso viene chiamato servizio **datagram**, per analogia con il servizio telegrafico (*telegram service*). che non restituisce una conferma a chi trasmette.

Vi è ancora un altro servizio chiamato **request-reply** (*domanda-risposta*). In questo servizio la sorgente trasmette un singolo datagramma contenente una richiesta; il messaggio ricevuto dal mittente rappresenta la risposta.

	Servizio	Esempio
Orientato alla connessione	Flusso affidabile di messaggi	Sequenza di pagine
	Flusso affidabile di byte	Scaricamento di filmati
	Connessione inaffidabile	Voice over IP
Senza connessione	Datagram inaffidabile	E-mail spazzatura (spam)
	Datagram con conferma	Messaggistica istantanea
	Request-reply	Interrogazione di un database

Comunicazioni affidabili e non affidabili coesistono perché la comunicazione affidabile potrebbe non essere disponibile a un dato livello; per esempio, Ethernet non fornisce una comunicazione affidabile; i pacchetti si possono corrompere di tanto in tanto mentre sono in transito. È compito dei protocolli di livelli superiori prendersi carico del problema: in particolare molti servizi affidabili sono implementati su datagrammi inaffidabili.

### PRIMITIVE DI SERVIZIO

Un servizio è formalmente specificato da un insieme di **primitive** (operazioni) che i processi utenti hanno a disposizione per accedere al servizio. Queste primitive istruiscono il servizio a eseguire alcune azioni o a riferire quelle prese da entità di pari livello.

L'insieme delle primitive disponibili dipende dalla natura del servizio offerto. Le primitive per un servizio orientato alla connessione sono diverse da quelle di un servizio senza connessione.

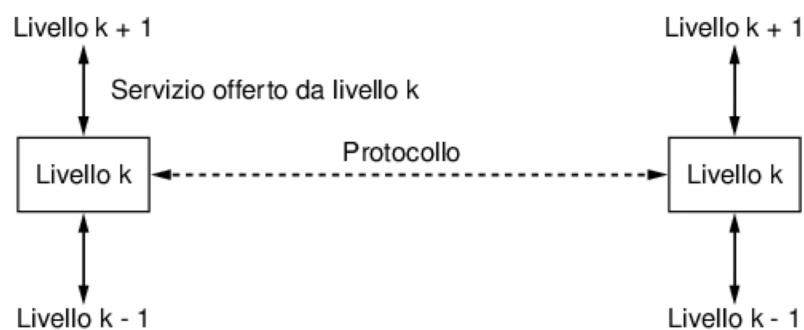
Primitiva	Significato
LISTEN	Attesa bloccante di una connessione in arrivo
CONNECT	Stabilisce una connessione con un peer in attesa
ACCEPT	Accetta una richiesta di connessione da un peer
RECEIVE	Attesa bloccante per un messaggio in arrivo
SEND	Manda un messaggio al peer
DISCONNECT	Termina una connessione



## RELAZIONE TRA SERVIZI E PROTOCOLLI

Servizi e protocolli sono concetti distinti. La distinzione è però così importante che è necessario evidenziarla. Un servizio è un insieme di primitive (operazioni) che un livello offre a quello superiore. Il servizio definisce quali operazioni il livello sia in grado di offrire su richiesta dei suoi utenti, ma non dice nulla di come queste operazioni sono implementate. Un servizio è correlato all'interfaccia tra due livelli, dove quello inferiore è il provider del servizio mentre quello superiore è l'utente.

Un protocollo, invece, è un insieme di regole che controllano il formato e il significato dei pacchetti o messaggi scambiati tra le entità pari all'interno di un livello. Le entità usano i protocolli per implementare le loro definizioni dei servizi. Sono libere di cambiare i loro protocolli, a patto di non cambiare il servizio visibile agli utenti. In questo modo, il servizio e il protocollo sono completamente disaccoppiati.

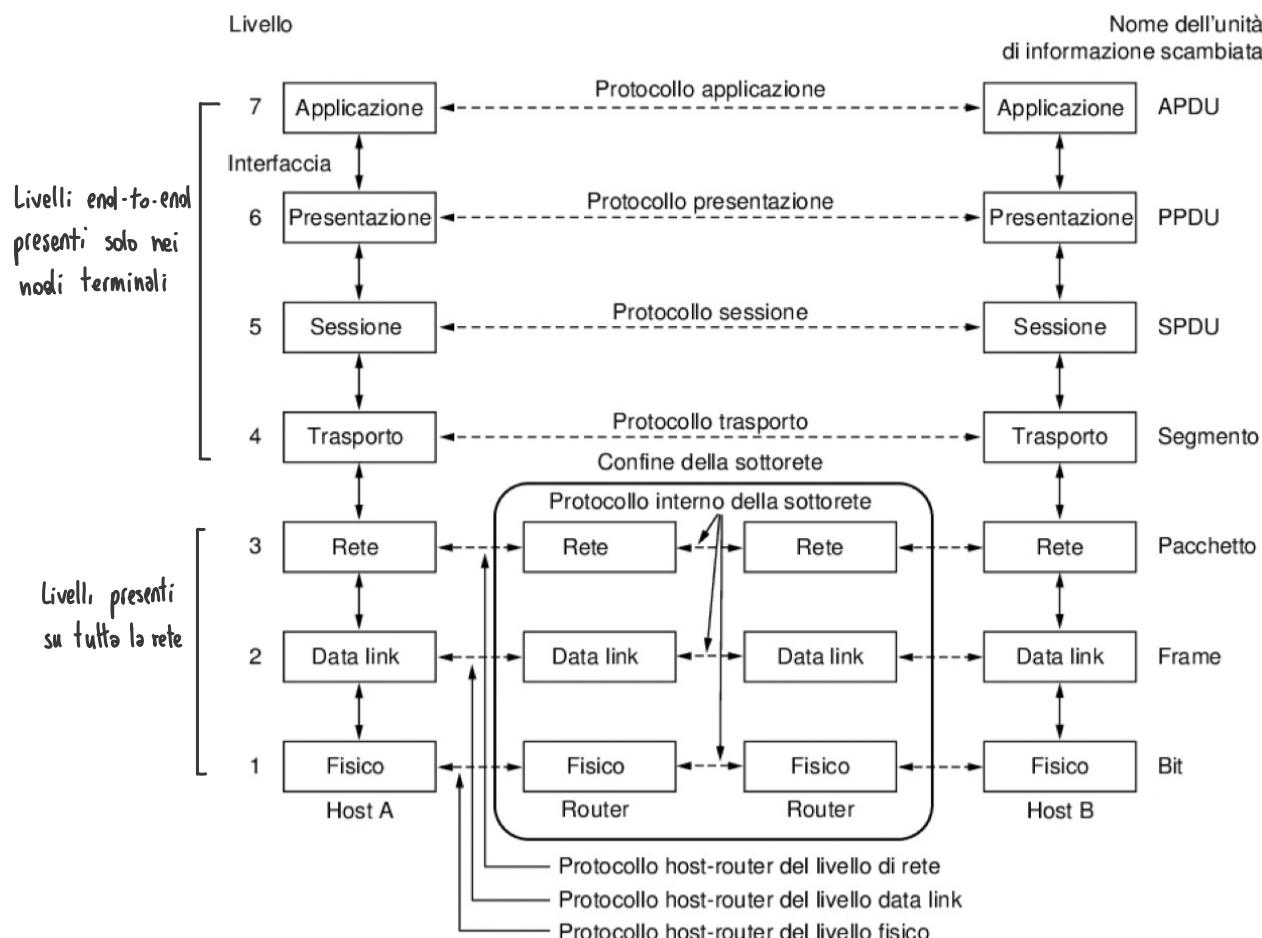


In altri termini, i servizi si riferiscono alle interface tra i livelli; i protocolli, invece, riguardano i pacchetti scambiati tra entità di pari livello che risiedono su computer diversi.

## MODELLO DI RIFERIMENTO

### IL MODELLO DI RIFERIMENTO OSI

Il modello OSI si fonda su una proposta sviluppata dall'**international standards organization (ISO)** come primo passo verso la standardizzazione internazionale dei protocolli impiegati nei diversi livelli. Si chiama modello di riferimento **ISO OSI** (*open system interconnection*) perché riguarda la connessione di sistemi aperti, cioè sistemi che sono “aperti” verso la comunicazione con altri. Il modello OSI ha sette livelli.



### **1) Il livello fisico**

Il livello fisico si occupa della trasmissione di **bit** grezzi sul canale di comunicazione. Le specifiche riguardano per lo più interfacce meccaniche o elettriche e temporizzazioni, oltre al mezzo di trasmissione che si trova sotto al livello fisico.

### **2) Il livello data link**

Il compito principale del livello data link (*collegamento dati*) consiste nel far diventare una trasmissione grezza una linea che appare priva di errori non rilevati. Esegue questo compito mascherando gli errori reali in modo che il livello rete non li veda. L'obiettivo è raggiunto forzando il trasmittente a suddividere i dati d'ingresso in **frame** trasmessi sequenzialmente.

### **3) Il livello di rete**

Il livello di rete controlla il funzionamento della sottorete. Si occupa della modalità con cui i **pacchetti** sono inoltrati dalla sorgente alla destinazione. Quando nella sottorete sono presenti contemporaneamente troppi pacchetti, creano delle congestioni: questo controllo spetta al livello di rete, per consentire la comunicazione tra reti eterogenee (routing).

## 4) Il livello di trasporto

La funzione essenziale del livello di trasporto è quella di accettare dati dal livello superiore, dividerli in unità più piccole quando necessario, passarle al livello di rete e assicurarsi che tutti i **segmenti** arrivino correttamente a destinazione. Il livello di trasporto copre tutto il percorso da sorgente a destinazione; inoltre stabilisce che tipo di servizio offrire al livello sessione e, in definitiva, agli utenti della rete.

## 5) Il livello sessione

Il livello sessione permette a utenti su computer diversi di stabilire tra loro una sessione. Le sessioni offrono diversi servizi, tra cui: controllo del dialogo (tenere traccia di quando è il turno di trasmettere e quando di ricevere), gestione dei token (e vitare che le due parti tentino la stessa operazione critica al medesimo istante) e sincronizzazione (supervisionare una lunga trasmissione per consentire la sua ripresa dal punto in cui si è interrotta a causa di un crash e del seguente recupero).

## 6) Il livello presentazione

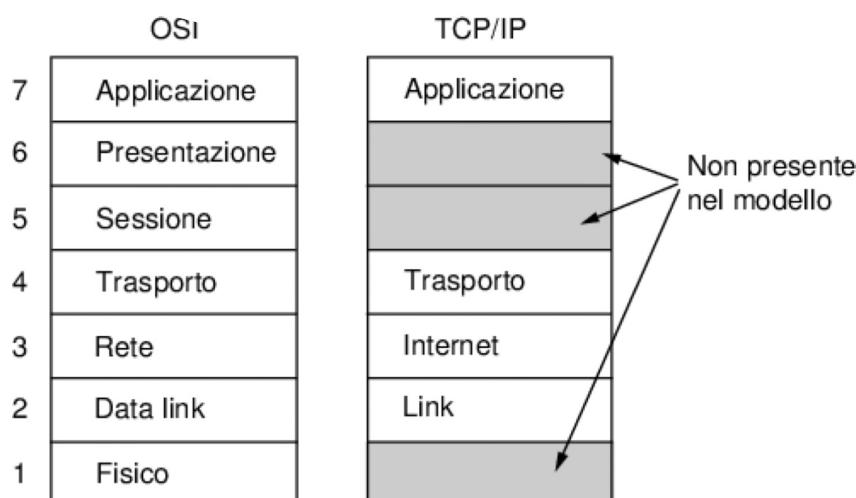
Si occupa della sintassi e della semantica dell'informazione trasmessa. Il livello presentazione gestisce queste strutture dati astratte e consente lo scambio e la definizione di strutture dati di livello superiore (per esempio transazioni bancarie). (**riformattazione**)

## 7) Il livello applicazione

Il livello applicazione comprende una varietà di protocolli comunemente richiesti dagli utenti. Un protocollo applicativo largamente usato è HTTP (*hypertext transfer protocol*), la base del World Wide Web. (**Interfaccia utente e macchina**).

## IL MODELLO DI RIFERIMENTO TCP/IP

Il modello di riferimento TCP/IP è il modello di riferimento del progenitore di tutte le reti di calcolatori geografiche, ARPANET, e il suo successore Internet. Quando furono aggiunte reti satellitari e via radio, i protocolli esistenti incontrarono difficoltà nel collegamento, richiedendo una nuova architettura di riferimento. Per questo motivo la capacità di collegare tra loro più reti in modo semplice è stata sin dall'inizio uno dei principali obiettivi di progetto. Questa architettura è poi diventata nota con il nome di modello di riferimento TCP/IP, dal nome dei suoi due protocolli principali. Poiché erano previste applicazioni con richieste divergenti, che spaziavano dal trasferimento di file alla trasmissione della voce in tempo reale, era inoltre necessaria un'architettura flessibile.



## 1) Il livello link

Describe cosa devono fare i collegamenti (linee seriali e la classica Ethernet) per esaurire le necessità di questo livello internet senza connessione. È un interfaccia tra l'host e il mezzo trasmissivo.

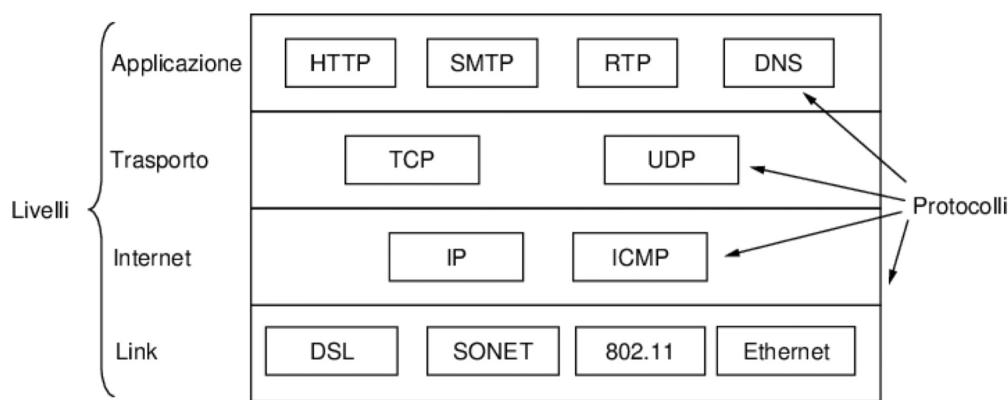
## 2) Il livello internet

È il perno che tiene insieme l'intera architettura: il suo compito è permettere agli host di inviare pacchetti su qualsiasi rete e fare in modo che questi possano viaggiare indipendentemente verso la destinazione (potenzialmente su reti diverse). Potrebbero persino arrivare con un ordine diverso da quello con cui sono stati spediti, e in questo caso è compito dei livelli superiori di riordinarli. Il livello internet definisce un formato ufficiale per i pacchetti e un protocollo chiamato **IP** (*internet control message protocol*). L'*instradamento* dei pacchetti è il problema principale, insieme alla necessità di garantire l'assenza di congestioni.

## 3) Il livello di trasporto

È progettato per consentire la comunicazione tra peer degli host sorgente e destinazione, come nel livello trasporto OSI. In questo livello sono stati definiti due protocolli di trasporto end-to-end: il primo, **TCP** (*transmission control protocol*), è un protocollo affidabile orientato alla connessione che permette a un fusso di byte emessi da un computer di raggiungere senza errori qualsiasi altro computer sulla internet. Suddivide il flusso di byte entrante in messaggi e passa ciascun frammento al livello internet. A destinazione, il processo TCP ricevente ricompone il messaggio ricevuto per formare il fusso di uscita. TCP gestisce anche il controllo di fusso, per garantire che una sorgente veloce non possa congestionare un ricevente lento con una quantità di messaggi superiore a quelli che sa gestire.

Il secondo protocollo di questo livello, **UDP** (*user datagram protocol*), è un protocollo inaffidabile senza connessione per le applicazioni che non vogliono la garanzia di ordinamento e il controllo di flusso di TCP, ma preferiscono gestire queste funzioni in modo autonomo.



## 4) Il livello applicazione

Contiene tutti i protocolli di livello superiore. I primi gestivano un terminale virtuale (TELNET), lo scambio di file (FTP), la posta elettronica (SMTP). Con gli anni ne sono stati aggiunti molti altri, come il Domain Name System (DNS) che fa corrispondere i nomi degli host ai loro indirizzi di rete; HTTP, il protocollo per prelevare pagine sul WWW e RTP, il protocollo per trasmettere contenuti in tempo reale come audio e film.

## CONFRONTO TRA I MODELLI DI RIFERIMENTO OSI E TCP/IP

I modelli di riferimento OSI e TCP/IP hanno molto in comune. Sono entrambi basati sul concetto di pila (stack) di protocolli indipendenti, e la funzione dei livelli è grosso modo simile. Nonostante queste somiglianze fondamentali, i due modelli hanno molte differenze. Nel modello OSI sono presenti tre concetti essenziali: servizi, interfacce, protocolli. Il modello TCP/IP in origine non faceva una netta distinzione tra questi tre concetti. La conseguenza è che nel modello OSI i protocolli sono nascosti meglio che nel modello TCP/IP e si possono sostituire con la relativa facilità all'evolvere della tecnologia. Un'altra differenza consiste nella modalità di comunicazione: il modello OSI supporta nel livello di rete entrambi i tipi di comunicazione, orientata o meno alla connessione, ma nel livello di trasporto (importante perché visibile agli utenti) supporta solo la comunicazione orientata alla connessione. Il modello TCP/IP ha solo una modalità nel livello di rete (senza connessione), ma supporta entrambe in quello di trasporto offrendo così agli utenti una vera scelta.

### CRITICA DEL MODELLO E DEI PROTOCOLLI OSI

1. Scarso tempismo;
2. Tecnologia scadente;
3. Implementazioni scadenti;
4. Incapacità politica.

### CRITICA DEL MODELLO DI RIFERIMENTO TCP/IP

1. Non distingue servizio, interfaccia e protocollo;
2. È poco generale e inadatto per descrivere pile di protocolli diverse (tipo Bluetooth);
3. Il livello link non è un vero e proprio livello, ma un'interfaccia tra livelli di rete e link;
4. Non fa distinzione tra livello fisico e link, totalmente diversi.

## IL MODELLO IBRIDO

5	Application layer
4	Transport layer
3	Network layer
2	Data link layer
1	Physical layer

La forza del modello di riferimento SI risiede nel modello stesso (ad esclusione dei livelli di presentazione e sessione), che si è dimostrato eccezionalmente utile nelle discussioni riguardanti le reti di calcolatori. Invece la forza del modello di riferimento TCP/IP sono i protocolli ampiamente utilizzati per molti anni.

Questo modello ha **cinque livelli** che vanno dal livello fisico attraverso quelli di link, rete, trasporto fino al livello applicazione. Il **livello fisico** specifica come trasmettere bit su differenti tipi di mezzi di trasporto in forma di segnali elettrici (o altri analogici). Il **livello link** è coinvolto quando bisogna spedire messaggi di lunghezza finita direttamente tra computer connessi con un livello prefissato di affidabilità, Ethernet e 802. 11 sono esempi di protocollo a livello link. Il **livello di rete** tratta come combinare link multipli nelle reti e nelle internetwork in modo da poter spedire pacchetti tra computer distanti. Questo include il compito di trovare un percorso attraverso cui mandare i pacchetti (*protocollo IP*). Il **livello di trasporto** si occupa di dare garanzie di consegna al livello di rete aumentandone l'affidabilità e fornisce astrazioni sulle modalità di consegna dei byte, che possono anche essere visti come una sequenza ordinata e affidabile (byte stream). (*TCP*). Infine, il **livello applicazione** contiene programmi che usano la rete.

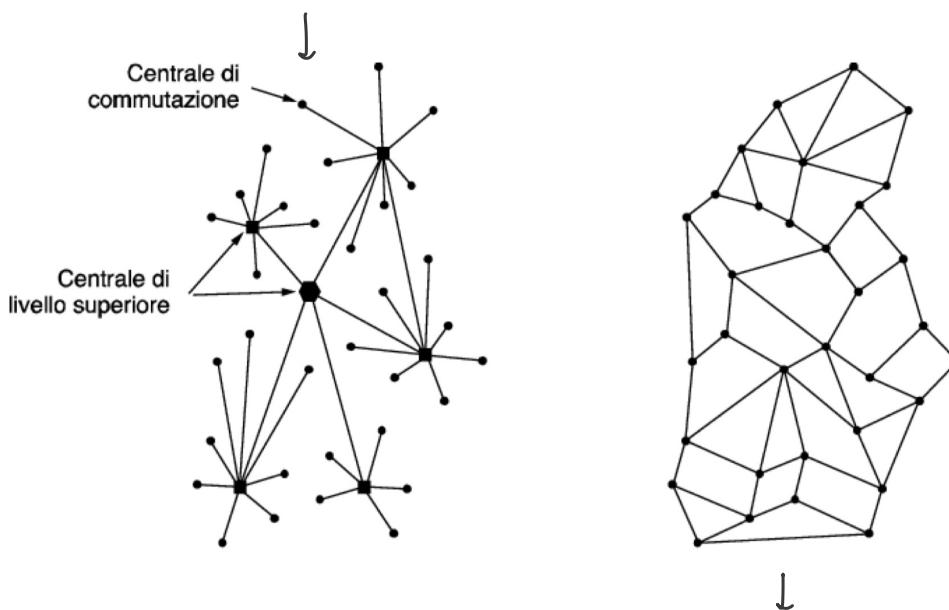
## ESEMPI DI RETI

### 1) Internet

Internet non è una rete, ma una vasta raccolta di reti diverse che usano determinati protocolli e offrono certi servizi comuni. È un sistema inconsueto, che non ha un progettista e non è controllato da nessuno.

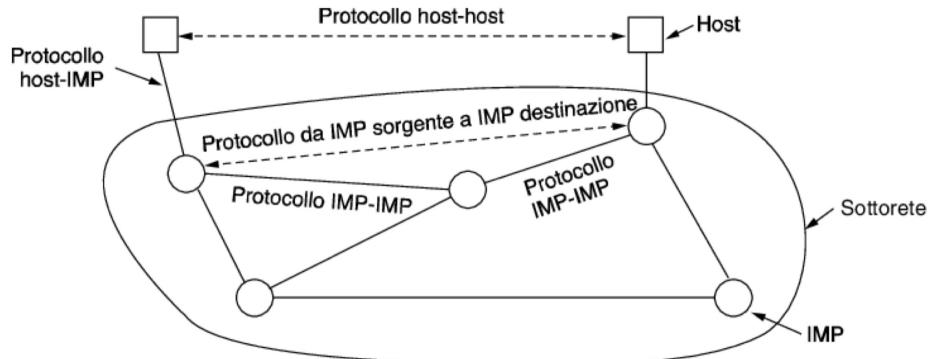
### 2) ARPANET

La storia inizia alla fine degli anni '50, durante la guerra fredda, quando il dipartimento della difesa USA commissiona una rete di controllo che possa sopravvivere a una guerra nucleare. Ai tempi le comunicazioni militari usavano la rete telefonica pubblica, considerata vulnerabile: nella figura i punti neri rappresentano le centrali di commutazione telefonica, ognuna delle quali era connessa a migliaia di telefoni. A loro volta queste centrali erano collegate da centrali di commutazione di livello superiore per formare una gerarchia nazionale. La vulnerabilità del sistema stava nel fatto che la distruzione di poche centrali di alto livello avrebbe frammentato la rete in molte isole separate.

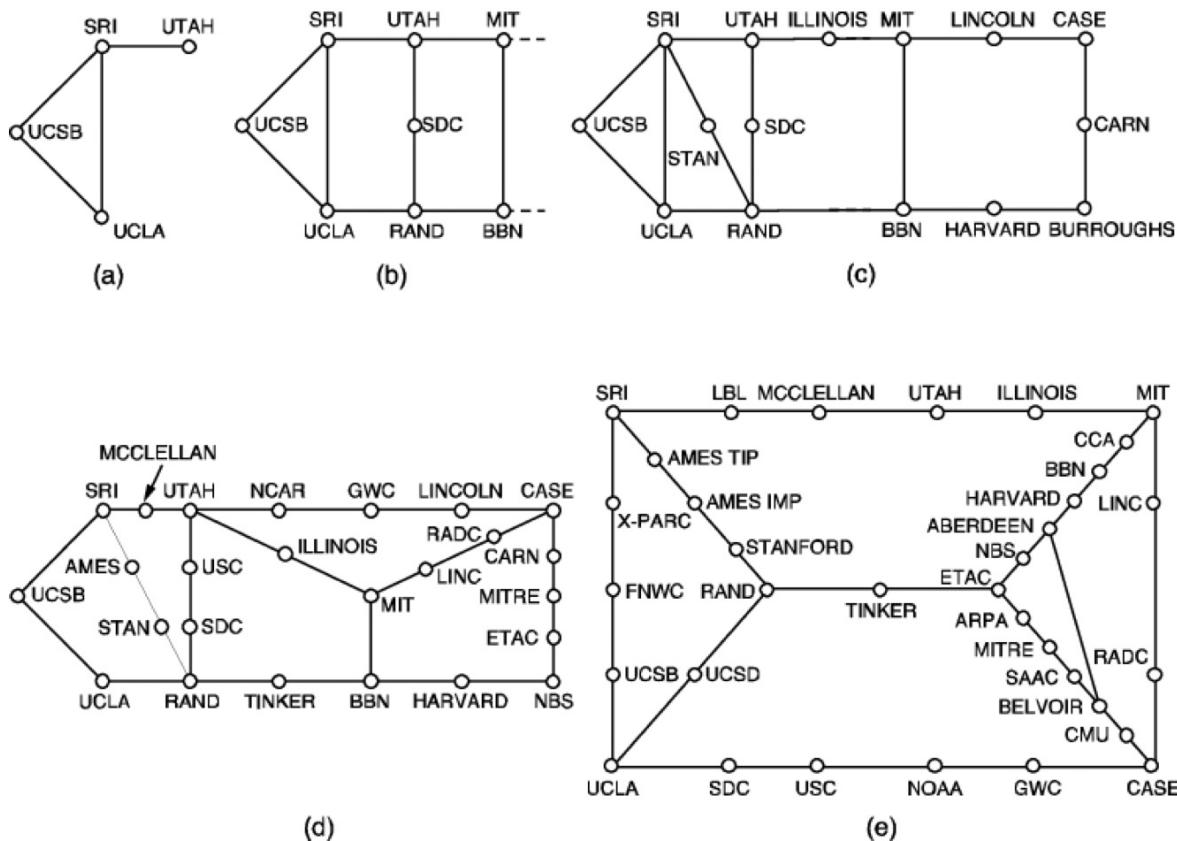


Nel 1960, Paul Baran, un impiegato della RAND Corporation, propose di usare una tecnologia digitale a commutazione di pacchetto. Per capire ciò che successe dopo, torniamo all'ottobre 1957 quando l'unione sovietica lanciò lo Sputnik. A seguito del fallimento, crearono una singola organizzazione per la ricerca, **ARPA** (*advanced research projects agency*). Nel 1967 sposarono l'idea della commutazione a pacchetto avanzata da Baran, intenzionati a costruire ciò che successivamente divenne noto con il nome **ARPANET**. La sottorete sarebbe stata composta da minicomputer chiamati **IMP** (*interface message processors*) collegati da linee di trasmissione. Per consentire un'alta affidabilità ogni IMP doveva essere collegato ad almeno altri due IMP.

Ogni nodo della rete doveva consistere in un IMP e un host, posti nella stessa stanza e collegati da un cavo. Ogni pacchetto sarebbe stato integralmente ricevuto prima dell'inoltro; quindi la sottorete fu la prima rete a commutazione di pacchetto del tipo store-and-forward. Il software della sottorete era composto dalla parte lato IMP della connessione tra IMP e host, dal protocollo IMP-IMP e da un protocollo da IMP sorgente a IMP destinazione progettato per migliorare l'affidabilità.



In qualche modo una rete sperimentale entrò in servizio nel dicembre 1969 con quattro nodi: UCLA, UCSB, SRI e Università dello Utah, scelti perché ciascuno aveva un gran numero di contratti con ARPA e utilizzava computer host diversi e completamente incompatibili. La rete crebbe rapidamente man mano che altri IMP vennero consegnati e installati; presto coprì tutti gli Stati Uniti.



Crescita di ARPANET: (a) dicembre 1969, (b) luglio 1970, (c) marzo 1971, (d) aprile 1972, (e) settembre 1972

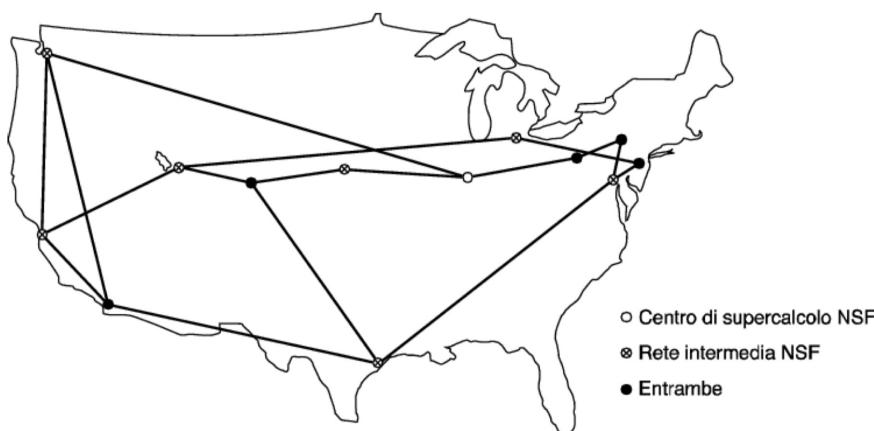
Nel corso degli anni '80 altre reti furono connesse ad ARPANET, soprattutto LAN. Al crescere delle dimensioni della rete rintracciare un host diventò sempre più difficoltoso, quindi fu creato il DNS (domain name system) per organizzare i computer in domini e abbinare i nomi degli host agli indirizzi IP. Da allora DNS si è evoluto in un database distribuito, a uso generale e capace di memorizzare una quantità d' informazioni relative ai nomi.

### 3) NSFNET

Negli ultimi anni '70 l'organismo statunitense NSF (*national science foundation*) si accorse dell'enorme impatto che ARPANET aveva sulla ricerca universitaria permettendo a scienziati di tutto il paese di condividere i dati e collaborare a progetti di ricerca.

Negli ultimi anni '80 NSF progettò un successore di ARPANET che sarebbe stato aperto a tutti i gruppi di ricerca universitari. A ogni supercomputer fu affiancato un fratellino, rappresentato da un microcomputer LSI-11 chiamato **fuzzball**.

L'intera rete, composta da backbone e reti regionali, fu chiamata NSFNET. Il collegamento con ARPANET era realizzato da una connessione tra un IMP e un fuzzball situati nella sala machine della Carnegie Mellon University.



NSFNET ebbe un successo istantaneo e risultò sovraccarico sin dal primo giorno.

### ARCHITETTURA DI INTERNET

Inizialmente le applicazioni principali di Internet (1970-1990) erano le email, le notizie, l'accesso da remoto e il trasferimento di file.

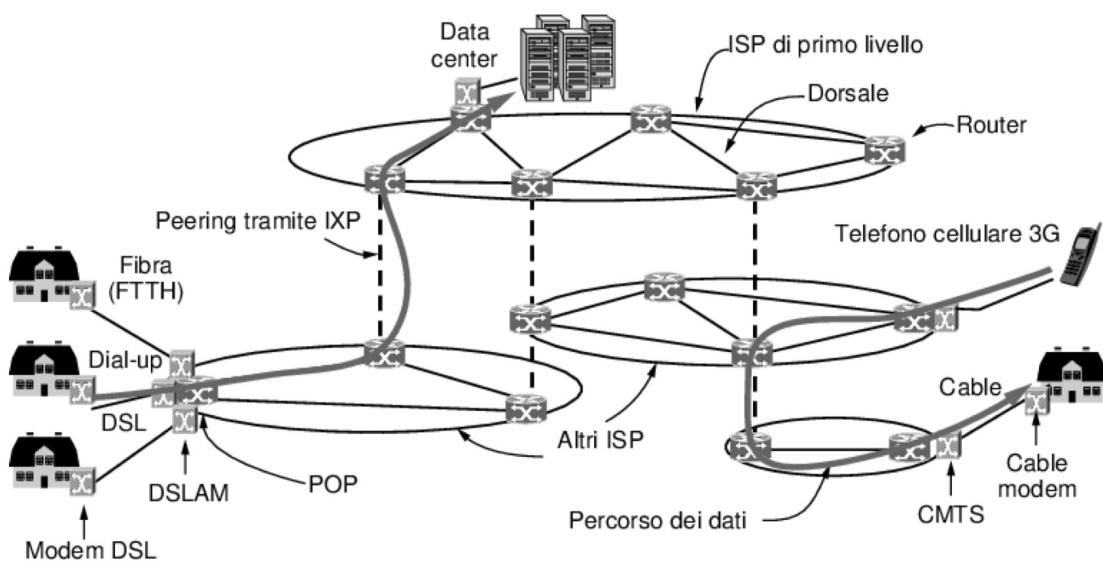
Per collegarsi a Internet, il computer si connette a un **internet service provider**, o più semplicemente **ISP**, che fornisce all'utente l'accesso a Internet o connettività. Questo permette al computer di scambiare pacchetti con tutti gli altri host accessibili in Internet. L'utente potrebbe inviare pacchetti per navigare sul Web o per migliaia di altri motivi, non importa quali. Esistono molti tipi di accesso a Internet e di solito sono classificati in base alla banda che forniscono e al loro costo, anche se la caratteristica più importante è la connettività.

La **DSL**, acronimo di **digital subscriber line**, riutilizza la linea telefonica di casa per la trasmissione di dati digitali. Il computer è connesso a un dispositivo chiamato **modem DSL** che effettua la conversione da pacchetti digitali a segnali analogici, che in questo modo passano sulla linea telefonica. Dall'altro capo, un dispositivo chiamato **DSLAM** (*digital subscriber line access multiplexer*) converte i segnali in pacchetti.

La DSL è un modo di usare la linea telefonica locale che fornisce una banda più larga del dial-up, che consiste nell'inviare bit su una chiamata telefonica tradizionale, al posto di una comunicazione vocale. Il **dial-up** viene effettuato con molti tipi di modem ai due capi. La parola **modem** è l'abbreviazione di modulator demodulator e si riferisce a qualunque dispositivo che converta bit digitali in segnali analogici.

Un altro metodo è quello di inviare segnali sul sistema della TV via cavo (cable).

Il dispositivo collegato al capo situato in casa è chiamato **cable modem**, mentre il dispositivo all'altro capo (**cable headend**) è chiamato **CMTS** (*cable modem termination system*).

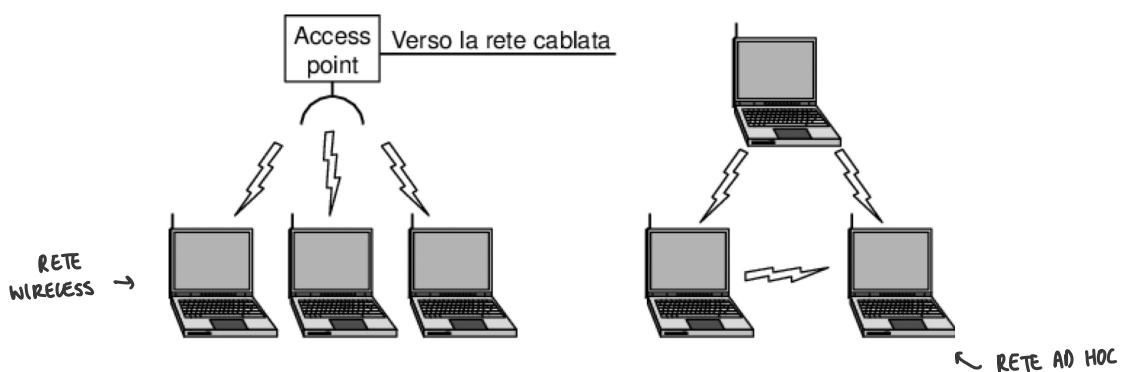


Anche la modalità wireless viene utilizzata per l'accesso a Internet. Siamo ora in grado di mandare pacchetti da casa all'ISP. Chiamiamo il punto in cui i pacchetti entrano nella rete ISP con il nome di **POP** (*point of presence*) dell'ISP. Le reti ISP possono essere regionali, nazionali o internazionali; la loro architettura chiamata **backbone** (*dorsale*) dell'ISP. Se il pacchetto è destinato a un host servito direttamente dall'ISP, il pacchetto viene instradato sulla dorsale e consegnato all'host, altrimenti deve essere passato a un altro ISP. Gli ISP interconnettono le loro reti per scambiarsi traffico tramite gli **IXP** (*Internet eXchange Point*). Si dice che gli ISP fanno **peering** uno con l'altro.

In cima alla "catena alimentare" c'è un pugno di aziende, quali AT&T e Sprint, che dispongono di grandi dorsali internazionali con migliaia di router connessi da fibre ottiche a banda larga. Questi ISP, chiamati **tier 1** (di primo livello), non pagano il transito e formano una dorsale di Internet dato che cinque altro deve collegarsi a loro per poter raggiungere il resto di Internet. Le aziende che forniscono molti contenuti come Google e Yahoo! concentrano i loro calcolatori in **data center** ben connessi al resto di Internet. I data center, essendo progettati per ospitare computer, non persone, possono essere riempiti di armadi di calcolatori chiamati **server farm**. Data center per **colocation** o **hosting** permettono ai clienti di posizionare le loro attrezzature, quali i server, molto vicine ai POP degli ISP, in tal modo le connessioni tra server e dorsali sono corte e veloci.

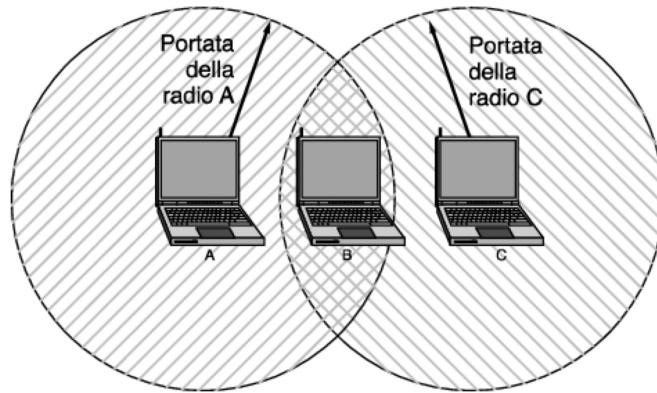
### LAN WIRELESS 802.11

Le reti 802.11 sono formate da client, come computer portatili e cellulari, e da infrastrutture chiamate **AP** (*access point, punto di accesso*) installate negli edifici. Tutti gli access point sono connessi a una rete cablata e tutte le comunicazioni tra i client passano attraverso almeno uno di loro. È anche possibile che due client che siano nel raggio radio uno dell'altro si parlino direttamente, come due computer in un ufficio, senza un access point. Questa configurazione è chiamata **rete ad hoc** (*ad hoc network*).



Si supponga che il computer A stia trasmettendo al computer B, ma che la portata radio del trasmittente A sia troppo corta per raggiungere il computer C. Se C vuole, a sua volta, trasmettere a B, può mettersi in ascolto prima di iniziare la trasmissione, ma fatto che non senta niente non significa che la sua trasmissione avverrà con successo. L'impossibilità che C senta A prima di iniziare la trasmissione provoca collisioni. Dopo una collisione il mittente aspetta un altro, più lungo, tempo di ritardo casuale per ritrasmettere il pacchetto.

Nonostante questo e altri problemi, tale tipo di approccio funziona sufficientemente bene nella pratica.



## STANDARDIZZAZIONE DELLE RETI

Esistono molti costruttori e fornitori di reti, ognuno con le proprie impostazioni. In assenza di coordinamento il caos sarebbe totale, e gli utenti non avrebbero nulla in mano. L'unica strada è raggiungere un accordo su alcuni standard di rete. Gli standard non solo permettono la comunicazione tra computer diversi, ma aumentano la quota di mercato dei prodotti che vi aderiscono.

### • Il Who's Who del mondo delle telecomunicazioni

**ITU-T**, il settore di standardizzazione delle telecomunicazioni, che si occupa di sistemi di telefonia e di scambio dati.

**ITU-R**, il settore delle radiocomunicazioni, si occupa di coordinare l'uso delle frequenze radio da parte di gruppi di interesse in competizione tra di loro ovunque nel mondo. L'altro settore è quello dello sviluppo, l'**ITU-D**, che promuove lo sviluppo di tecnologie di informazione e comunicazione terrestre per ridurre il "digital divide" tra le nazioni che hanno effettivamente accesso alle tecnologie dell'informazione e i paesi con accesso limitato. Il compito di ITU-T è emanare raccomandazioni tecniche su interface per telefonia, telegrafia e trasmissione dati, che spesso diventano standard riconosciuti a livello internazionale anche se, tecnicamente, si tratta di semplici suggerimenti che i governi possono o meno adottare.

### • Il Who's Who del mondo degli standard internazionali

Gli standard internazionali sono definiti e pubblicati da **ISO** (*international standards organization*), una libera organizzazione di volontari fondata nel 1946. ISO definisce standard su una vastissima gamma di argomenti. Per quanto riguarda le telecomunicazioni ISO e ITU-T spesso collaborano (ISO è membro di ITU-T), per evitare di emettere ufficialmente due standard internazionali incompatibili tra loro.

Un altro ente importante nel mondo degli standard è l'**IEEE** (*institute of electrical and electronics engineers*), il principale ente professionale del mondo. Il comitato IEEE 802 ha standardizzato molti tipi di LAN:

Number	Topic
802.1	Overview and architecture of LANs
802.2 ↓	Logical link control
802.3 *	Ethernet
802.4 ↓	Token bus (was briefly used in manufacturing plants)
802.5	Token ring (IBM's entry into the LAN world)
802.6 ↓	Dual queue dual bus (early metropolitan area network)
802.7 ↓	Technical advisory group on broadband technologies
802.8 ↑	Technical advisory group on fiber optic technologies
802.9 ↓	Isochronous LANs (for real-time applications)
802.10 ↓	Virtual LANs and security
802.11 *	Wireless LANs
802.12 ↓	Demand priority (Hewlett-Packard's AnyLAN)
802.13	Unlucky number. Nobody wanted it
802.14 ↓	Cable modems (defunct: an industry consortium got there first)
802.15 *	Personal area networks (Bluetooth)
802.16 *	Broadband wireless
802.17	Resilient packet ring

→ I gruppi di lavoro 802.1  
più importanti sono  
contrassegnati con \*  
quelli marcati da ↓ sono in  
quiescenza. Quelli con +  
hanno rinunciato all'incarico  
o si sono sciolti.

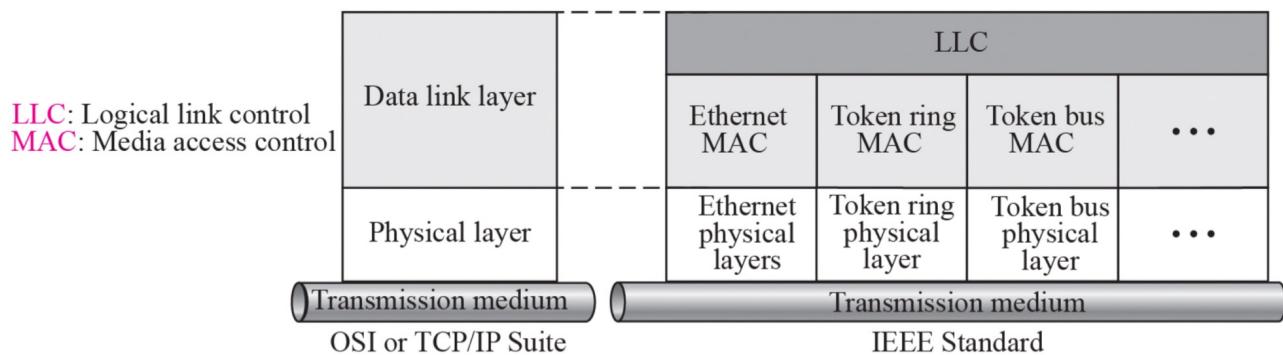
## • Il Who's Who del mondo degli standard di Internet

La Internet mondiale ha un proprio meccanismo di standardizzazione, molto diverso da ITU-T e ISO. È divisa in gruppi di lavoro, che riguardano nuove applicazioni, informazioni degli utenti, integrazione OSI, instradamento e indirizzamento, sicurezza, gestione della rete e standard. Si crearono così tanti gruppi di lavoro (più di 70) che è stato necessario raggruppare per aree; il capo di ogni area partecipa al comitato direttivo. Per quanto riguarda gli standard Web, il **World Wide Web consortium** (W3C) sviluppa protocolli e linee guida che facilitano la crescita a lungo termine del Web.

## LIVELLO 2 : DATA LINK LAYER → IEEE 802.3 (ETHERNET)

Ethernet includes the **physical** and Data Link layer.

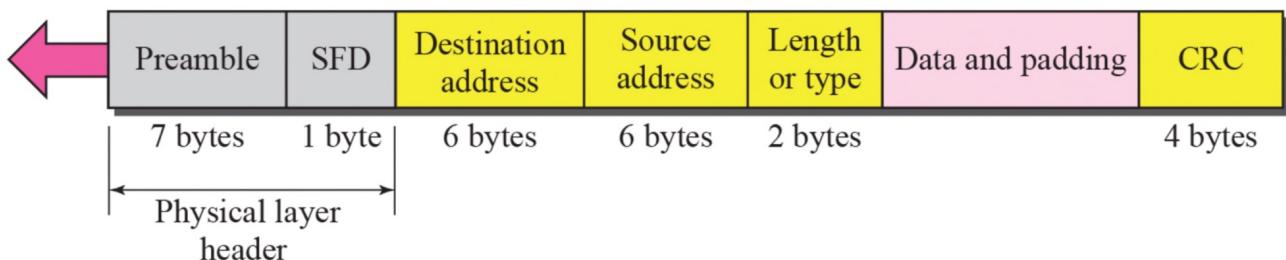
The **Data Link layer** is in turn divided into the MAC and the LLC sub-layers.



In Ethernet c'è un accesso a contesa: nel data link layer sono presenti nodi in cui vengono scambiati i token (**token ring**).

**Preamble:** 56 bits of alternating 1s and 0s.

**SFD:** Start frame delimiter, flag (10101011)



Per prima cosa c'è un campo, **Preamble** (*preambolo*) di 8 byte, ognuno dei quali contiene la sequenza di bit 0101010, con eccezione dell'ultimo byte, in cui gli ultimi due bit sono impostati a 11; questo ultimo byte è chiamato **delimitatore di inizio frame (start of frame delimiter)**.

Di seguito vengono due **indirizzi**: uno per la destinazione e uno per la sorgente. Ognuno è lungo 6 byte; il primo bit trasmesso dell'indirizzo di destinazione è uno 0 per indirizzi ordinari e 1 per un gruppo di indirizzi.

Quando un frame viene inviato a un indirizzo di gruppo, tutte le stazioni di quel gruppo lo ricevono. La trasmissione diretta a un gruppo di stazioni è definita **multicast**. L'indirizzo composto da tutti bit 1 è riservato per le trasmissioni **broadcast**. Un frame che contiene tutti 1 nel campo "Indirizzo di destinazione" è ricevuto da tutte le stazioni della rete. La trasmissione multicast è più selettiva, ma richiede una gestione del gruppo per assicurare che non esistano due stazioni al mondo con lo stesso indirizzo. L'idea è che ogni stazione si possa rivolgere a un'altra solo dando il giusto numero di 48 bit. Per fare ciò, i primi 3 byte del campo d'indirizzo sono usati per un OUI (organizationally unique identifier). I valori per questo campo sono assegnati da IEEE e indicano un produttore; il produttore assegna gli ultimi 3 byte dell'indirizzo e programma l'intero indirizzo all'interno di NIC prima che sia venduto.

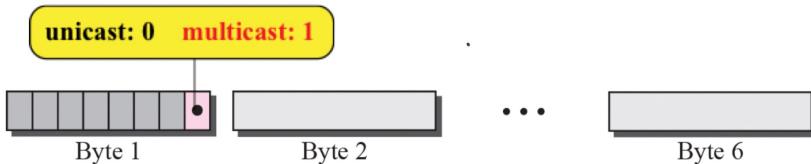
**d:** Hexadecimal digit

**d<sub>1</sub>d<sub>2</sub> : d<sub>3</sub>d<sub>4</sub> : d<sub>5</sub>d<sub>6</sub> : d<sub>7</sub>d<sub>8</sub> : d<sub>9</sub>d<sub>10</sub> : d<sub>11</sub>d<sub>12</sub>**

6 bytes = 12 hexadecimal digits = 48 bits

There are 3 address types:

- Unicast (d<sub>2</sub> even)
- Multicast (d<sub>2</sub> odd)
- Broadcast (all "1", or FF:FF:FF:FF:FF:FF)



Define the type of the following destination addresses:

- a. 4A:30:10:21:10:1A
- b. 47:20:1B:2E:08:EE
- c. FF:FF:FF:FF:FF:FF

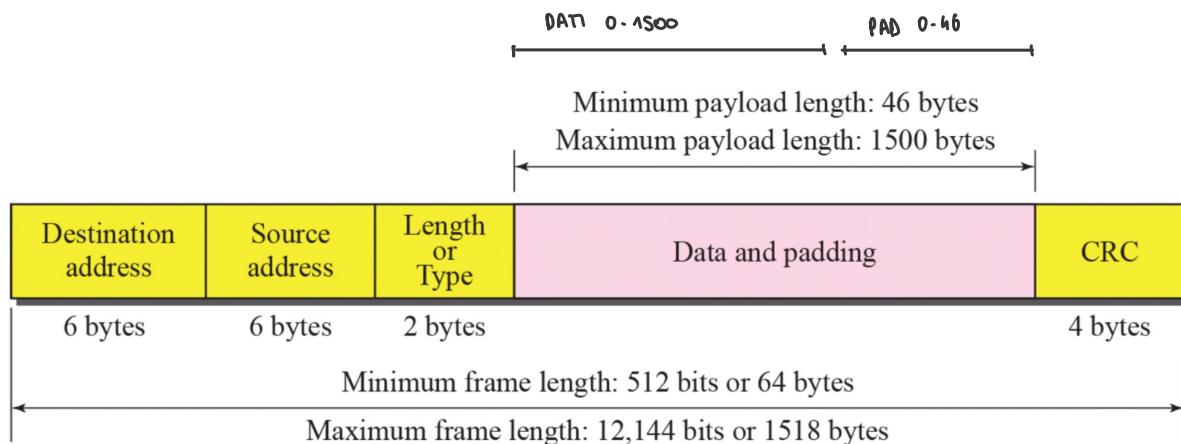
Solution:

- a. This is a unicast address because A in binary is 1010 (d2 pari)
- b. This is a multicast address because 7 in binary is 0111 (d2 dispari)
- c. This is a broadcast address because all digits are F's.

The address is sent left-to-right, byte by byte; for each byte, it is sent right-to-left, bit by bit:

← 11100010 00000100 11011000 01110100 00010000 01110111

Il campo successivo è **Type (tipo)** o **Lenght (lunghezza)**, a seconda che il frame sia Ethernet o IEEE 802.3. Ethernet usa un campo Type per indicare al ricevente cosa fare col frame: ad esempio il codice 0x0800 significa che i dati contengono un pacchetto IPv4.



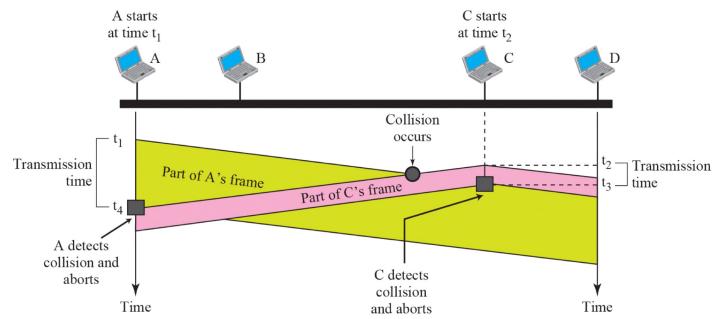
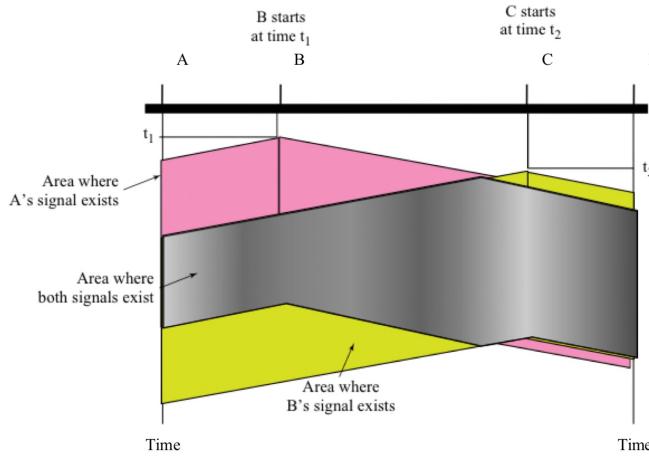
Dopo troviamo il campo **Data (dati)**, lungo fino a 1500 byte. Questo limite, scelto arbitrariamente al tempo della creazione dello standard Ethernet, si basava sul fatto che un transceiver aveva bisogno di una RAM che gli consentisse di conservare l'intero frame: nel 1978 la RAM era costosa.

Il frame non ha solo una lunghezza massima, ma anche una lunghezza minima; anche se qualche volta può tornare utile, un campo Data lungo 0 byte causa un problema.

Quando rileva una **collisione**, un transceiver tronca il frame corrente: ciò significa che sul cavo compaiono continuamente bit sparsi e pezzi di frame. Per aiutare a distinguere i frame validi dalla spazzatura, Ethernet richiede che i frame validi siano lunghi almeno 64 byte dall'indirizzo di destinazione al checksum inclusi.

Se la parte occupata dai dati è lunga meno di 46 byte, il campo **Pad (riempimento)** viene utilizzato per riempire il frame fino alla dimensione minima.

Imporre una lunghezza minima serve anche per un altro (e più importante) motivo: impedire a una stazione di completare la trasmissione di un frame breve prima che il primo bit abbia raggiunto la fine del cavo, dove potrebbe collidere con un altro frame.

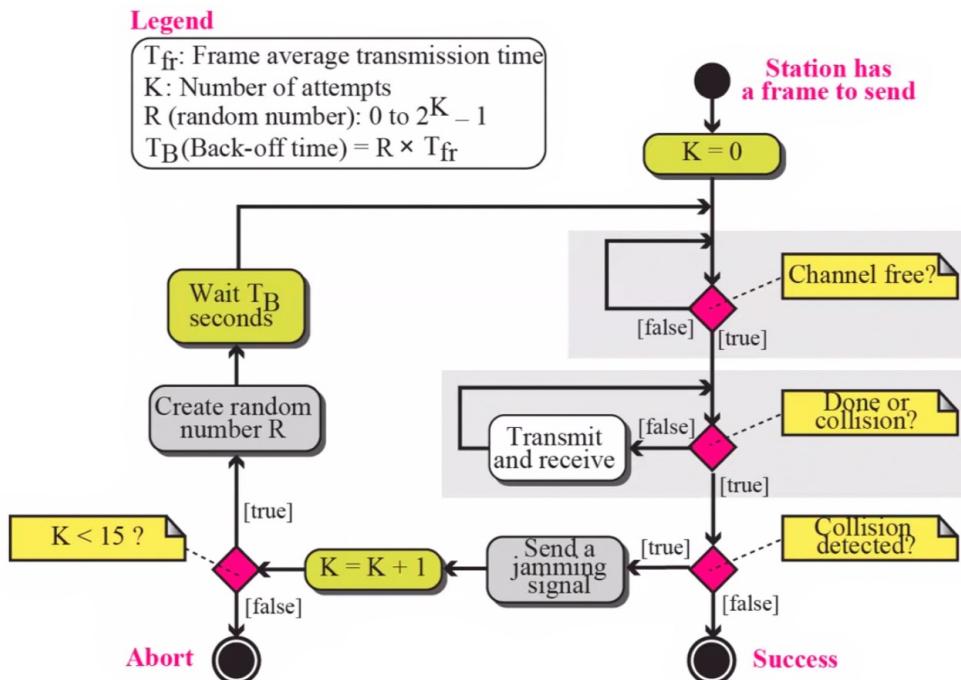


*Collision of the first bit in CSMA/CD*

### Space time model of a collision in CSMA

#### CARRIER SENSING MULTIPLE ACCESS - COLLISION DETECTION

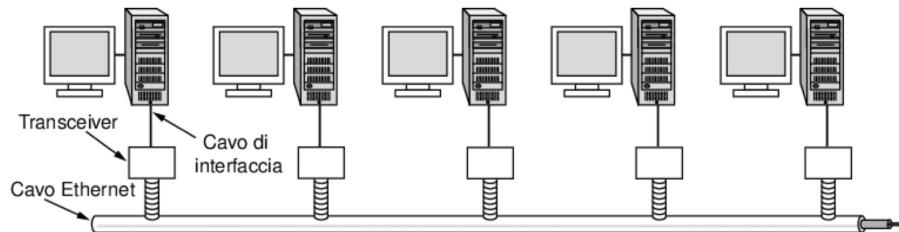
Ethernet classica usa l'**algoritmo CSMA/CD 1-persistente**. Con ciò si intende che le stazioni controllano il canale quando hanno un frame da spedire e lo spediscono non appena il canale risulta libero; monitorano il canale per trovare collisioni mentre spediscono. Se c'è una collisione, interrompono la trasmissione con un breve messaggio caotico per poi ritrasmettere dopo un periodo di tempo casuale.



Il campo finale è il **Checksum**; si tratta di un CRC a 32 bit, un codice di rilevazione di errore che si usa per determinare se i bit del frame siano stati ricevuti correttamente. Esso rivela unicamente gli errori, facendo scartare il frame in caso positivo.

## EVOZIONE DI ETHERNET

Bob Metcalfe e il collega David Boggs progettarono e implementarono la prima LAN (Metcalfe e Boggs, 1976), nella quale veniva usato un singolo cavo coassiale lungo e spesso e andava a 3 Mbps. Chiamarono il sistema Ethernet ispirandosi a luminiferous ether, attraverso il quale un tempo si pensava si propagassero le radiazioni elettromagnetiche. L'Ethernet di Xerox ebbe un successo tale che DEC, Intel e Xerox proposero uno standard nel 1978 per una Ethernet da 10 Mbps, chiamata standard DIX. Con un minimo cambiamento, nel 1983 lo standard DIX divenne lo standard IEEE 802.3.



## EVOZIONE DELLA TOPOLOGIA DI ETHERNET

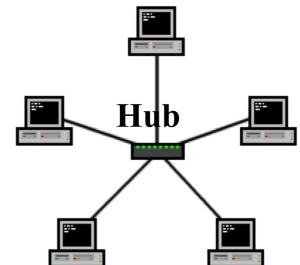
### • Bus:

- Thick coaxial cable with “vampires”;
- Thin coaxial cable with T junction and BNC connectors;
- Segments could be connected by max 4 repeaters;
- One collision domain, CSMA/CD (half-duplex) necessary;
- Reliability problems.



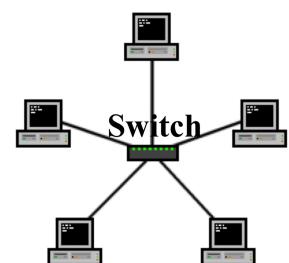
### • Physical star:

- Nodes are connected to a hub (physical repeater), which is the centre of the star; twisted pair cables and RJ45 connectors used;
- The layout is star, but it is logically equivalent to a bus, because all the cables are electrically connected (although we have separate Tx and Rx channels);
- One collision domain, CSMA/CD (half-duplex) still necessary because in case of collision the hub send the jamming signal.



### • Logical star:

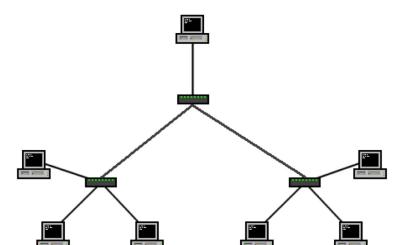
- The centre of the star is a switch; the inputs are buffered and frames are moved from the input to the output socket by means of a high speed internal bus;
- Only one node connected to each socket; each node has its collision domain, i.e. no collisions possible;
- CSMA/CD no more necessary; full-duplex possible;
- Better throughput;
- Importance of flow control to “pause” transmitting nodes before congestion occurs.



### • Hierarchical structures:

- One central switch to serve three building, one building switch to serve three floors;
- One floor switch to serve three areas.

Each standard variant has its own constraints on max cable lengths and max “radius”.

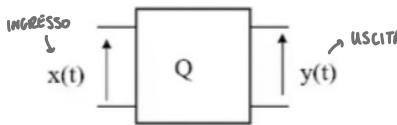


APPENDICE: MISURE IN DECIBEL

**RAPPORTI DI POTENZE:** Le misure in decibel si riferiscono a rapporti di grandezze omogenee, ed in quanto tali sono adimensionali. Sono misure logaritmiche, particolarmente adatte a rappresentare grandezze molto diverse tra loro, come spesso accade nei circuiti elettrici.

Ad esempio per il guadagno di potenza di un doppio bipolo si ha:

$$G_{P,\text{dB}} = 10 \log_{10} \left( \frac{P_y}{P_x} \right) \quad \frac{P_y}{P_x} = 10^{G_{P,\text{dB}}/10}$$



Dalle proprietà dei logaritmi, a prodotti corrispondono somme in dB, a rapporti di differenze:

$$G = G_1 G_2 \quad G_{\text{dB}} = G_{1,\text{dB}} + G_{2,\text{dB}} \quad G = \frac{G_1}{G_2} \quad G_{\text{dB}} = G_{1,\text{dB}} - G_{2,\text{dB}}$$

**Formule di conversione** per i rapporti di potenze:

$$10 \log_{10} (10^n) = n \cdot 10$$

Si aggiunge o sottrae 10 ←

G <sub>dB</sub>	G <sub>P</sub>
30	1000
20	100
10	10
0	1
-10	0.1
-20	0.01
-30	0.001

Un valore di particolare interesse è quello corrispondente a 2

$$\downarrow \quad 10 \log_{10} (2) = 3.0103 \rightarrow 3 \text{ dB}$$

Il valore esatto viene in pratica sempre approssimato a 3, e così per le potenze di due avremo 4=6dB, 8=9dB, e per le potenze negative 1/2=-3dB, 1/4=-6dB ecc...

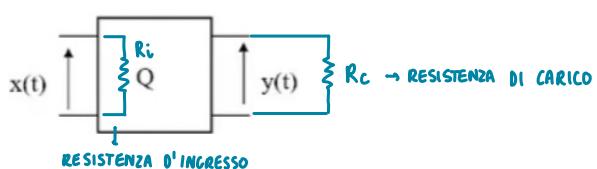
$$10 \log_{10} (2^2) = 3.0103 \rightarrow 6 \text{ dB} \quad 10 \log_{10} (2^3) = 3.0103 \rightarrow 9 \text{ dB}$$

RAPPORTI DI RADICI QUADRATE DI POTENZE:

Per la misura di rapporti di grandezze legati alla radice quadrata di potenze, cioè in pratica a rapporti di **tensioni, correnti, intensità di campo**, viene usata una definizione leggermente diversa.

Considerando ad esempio il guadagno di tensione del doppio bipolo precedente si ha:

$$G_{V,\text{dB}} = 20 \log_{10} \left( \frac{V_y}{V_x} \right) \quad \frac{V_y}{V_x} = 10^{G_{V,\text{dB}}/20}$$



La presenza di due formule può essere all'origine di errori molto rilevanti dal punto di vista numerico. La ragione è legata al desiderio di non dover specificare se il guadagno è in tensione (o corrente) o in potenza.

Partendo dal guadagno di potenza, supponendo resistive le impedenze di ingresso e di carico di un doppio bipolo, e quindi identiche (come spesso in pratica) si ha:

$$G_{P,\text{dB}} = 10 \log_{10} \left( \frac{P_y}{P_x} \right) = 10 \log_{10} \left( \frac{V_y^2 / R_c}{V_x^2 / R_i} \right) \text{ se } \underline{R_i = R_c} = 20 \log_{10} \left( \frac{V_y}{V_x} \right) = G_{V,\text{dB}}$$

## Formule di conversione per i rapporti di radici quadrate di potenze:

$$20 \log_{10} (10^n) = n \cdot 20$$

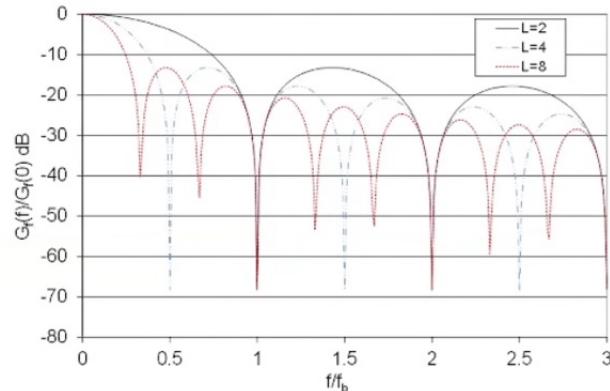
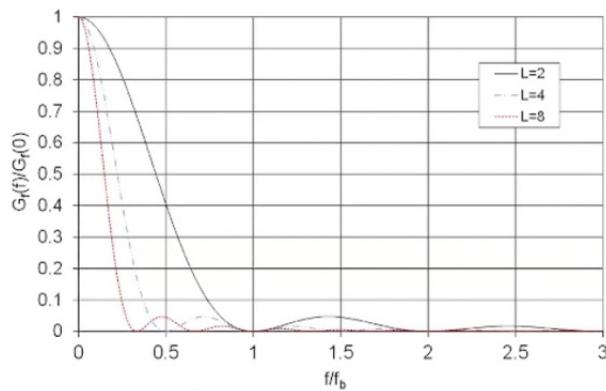
$G_{dB}$	$G_P$	$G_V$
30	1000	31.62
20	100	10
10	10	3.162
0	1	1
-10	0.1	0.3162
-20	0.01	0.1
-30	0.001	0.03162

Ad ogni ordine di grandezza in più corrisponde un incremento di 20 db, e viceversa, come evidenziato nella tabella per la colonna Gv.

Un valore di particolare interesse è quello corrispondente a 2 e sue potenze o radici:

$$20 \log_{10} (2) \rightarrow 6 \text{ dB} \quad 20 \log_{10} (\sqrt{2}) \rightarrow 3 \text{ dB} \rightarrow \sqrt{2} = 1.414 \quad -3 \text{ dB} \rightarrow 1/\sqrt{2} \rightarrow 0.707 \text{ dB}$$

### ESEMPIO: SPETTO PAM, CODIFICA MULTILIVELLO E NRZ



### POTENZE E TENSIONI IN dB

Le misure in decibel sono adimensionali, riferendosi a rapporti di grandezze omogenee. A volte tuttavia è comodo utilizzare i decibel per rappresentare potenze e tensioni, anziché rapporti delle medesime. In questo caso si procede considerando il rapporto fra la grandezza, potenza o tensione, e la sua unità di misura, che ora viene indicata nel pedice. Ad esempio, per una potenza:

$$P_{dBW} = 10 \log_{10} \left( \frac{P}{1W} \right) \quad P_{dBmW} = P_{dBm} = 10 \log_{10} \left( \frac{P}{1mW} \right) \quad P_{dBmW} = 30 + P_{dBW}$$

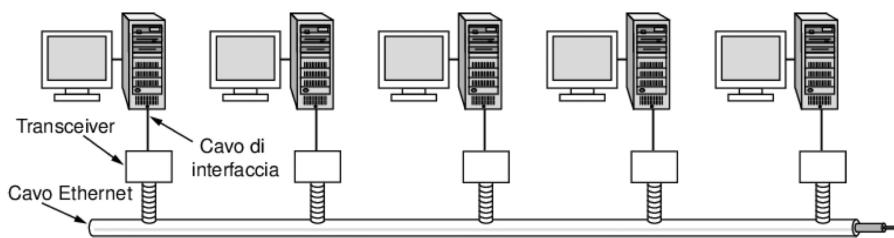
Analogamente per una tensione X (per evitare confusione con V, simbolo di un Volt):

$$X_{dBV} = 20 \log_{10} \left( \frac{X}{1V} \right) \quad \text{da cui} \rightarrow X_{dBmV} = 60 + X_{dBV}$$

### ETHERNET CLASSICA (10 Mbit/s)

Ethernet classica si insinuava attraverso un edificio sotto forma di un singolo lungo cavo al quale tutti i computer erano collegati.

Il primo tipo, comunemente chiamata **thick Ethernet** (*Ethernet spessa*), fu seguita da **thin Ethernet** (*Ethernet sottile*), più flessibile e che utilizzava collettori standard BNC. Thin Ethernet era più economica e più facile da installare, ma poteva estendersi per soli 185 metri per segmento (invece dei 500 metri di thick Ethernet).

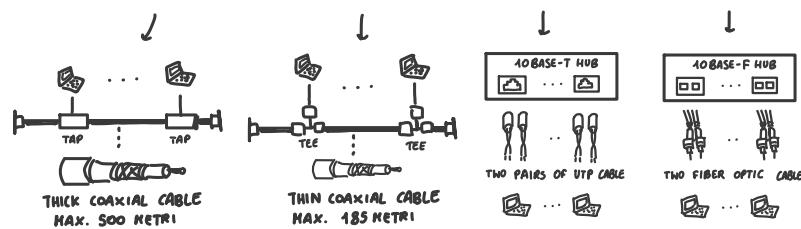


Ogni variante di Ethernet ha una lunghezza massima del cavo per ogni segmento (cioè la lunghezza su cui non si effettua una amplificazione) su cui il segnale si propaga.

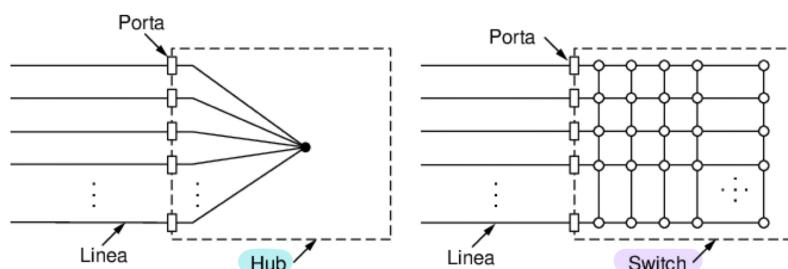
Per permettere di costruire reti più grandi, più cavi possono essere connessi attraverso **repeater** (*ripetitori*). Un repeater è un dispositivo di livello fisico che riceve, amplifica (cioè rigenera) e ritrasmette segnali in entrambe le direzioni. Per quanto riguarda il software, una serie di segmenti connessi da ripetitori non sono diversi da un singolo cavo (eccezione fatta per i piccoli ritardi causati dai ripetitori).

Su ognuno di quei cavi le informazioni sono spedite usando la **codifica Manchester**. Una Ethernet può contenere diversi segmenti di cavo e diversi repeater, ma non ci possono essere due transceiver<sup>2</sup> a più di 2,5 km di distanza e nessun percorso tra due transceiver può attraversare più di quattro ripetitori. Il motivo di queste restrizioni è che il protocollo MAC.

	<b>10Base5</b>	<b>10Base2</b>	<b>10Base-T</b>	<b>10Base-F</b>
<b>Medium</b>	Thick coax	Thin coax	2 UTP (doppino)	2 fibers
<b>Max hop length</b>	500m	185m	100m	2000m

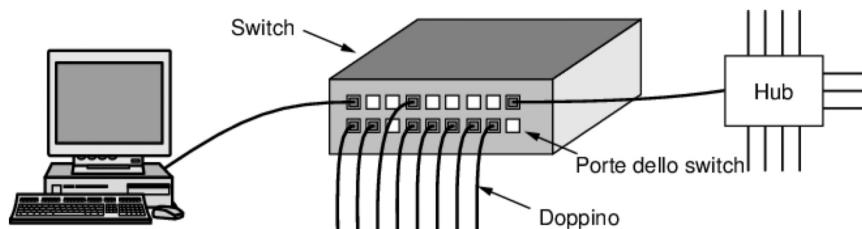


Ethernet si emancipò presto dall'architettura del lungo cavo di Ethernet classica. I problemi associati al reperimento di guasti o connettori lenti la guidarono verso un diverso tipo di cablaggio, nel quale ogni stazione aveva un cavo dedicato che andava verso un **hub** centrale. Un hub connette elettricamente tutti i cavi collegati come se fossero saldati assieme.



Esiste un modo alternativo di gestire il carico crescente: si tratta di Ethernet commutata (*switched Ethernet*). Il cuore di questo sistema è uno **switch** (*commutatore*) che contiene una scheda hardware di collegamento tra le interfacce di rete o **backplane** ad alta velocità che connette tutte le porte. Dall'esterno uno switch assomiglia a un hub. Sono entrambi scatole, tipicamente dotati di un numero di porte da 4 a 48, ognuna fornita di un connettore standard RJ-45 per doppino. Ogni cavo connette lo switch o un hub a un solo computer.

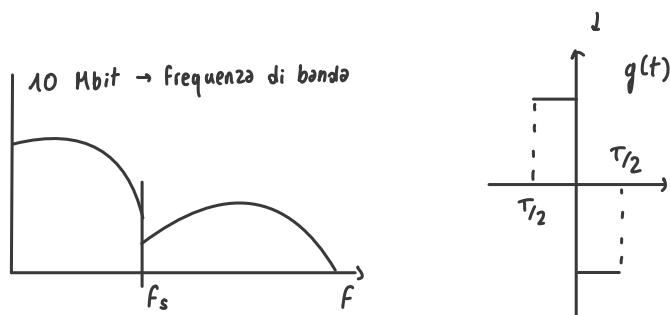
Dentro lo switch, tuttavia, accade qualcosa di molto diverso. Gli switch devono solo far uscire i frame verso le giuste porte di destinazione. Quando una porta dello switch riceve un frame Ethernet da una stazione, lo switch verifica l'indirizzo Ethernet per capire a quale porta sia destinato.



Che cosa succede se più di una stazione o porta vuole spedire un frame nello stesso istante? Anche in questo caso gli switch si differenziano dagli hub. In un hub tutte le stazioni sono nello stesso **dominio di collisione** (*collision domain*). Devono utilizzare l'algoritmo CSMA/CD per pianificare le loro trasmissioni. In uno switch ogni porta ha il proprio dominio di collisione indipendente. Nel caso comune in cui il cavo sia full duplex, sia la stazione sia la porta possono spedire il frame lungo il cavo contemporaneamente, senza preoccuparsi delle altre stazioni e porte. Le collisioni sono ora impossibili e CSMA/CD non è più necessario. Tuttavia, se il cavo è half duplex, la stazione e la porta si contenderanno la trasmissione nel solito modo con CSMA/CD.

### FAST ETHERNET (ETHERNET 100 Mbit/s)

L'idea alla base di fast Ethernet, dato che la velocità di 10 Mbps venne messa sotto pressione, era di mantenere tutti i vecchi formati di frame per calcoli, interfacce e regole procedurali, riducendo semplicemente il tempo bit da 100 ns a 10 ns. Era indifferente utilizzare uno switch o un hub, la velocità poteva essere ridotta pur usando l'algoritmo CSMA/CD; non si poteva più usare però la topologia a bus. Il problema principale era la banda del segnale: con 100 Mbit/s, la banda diventava 10 volte quella di prima, perché questo tipo di impulso ha moltissime transizioni (sincronismo).



Nonostante tutto era necessario fare anche altre scelte: la più importante riguardava i tipi di cavi da supportare. Un contendente era il doppino di categoria 3, con cui sarebbe stato possibile collegare i computer desktop con fast Ethernet senza dover ricablarlo l'edificio. I doppini di categoria 5, però, potevano arrivare facilmente a 100 metri diversamente da quelli di categoria 3, quindi molti edifici furono comunque ricablati con UTP di categoria 5 per Ethernet **100Base-TX**, che iniziò a dominare il mercato. La sua progettazione è più semplice, in quanto i cavi possono gestire una velocità di clock di 125MHz. Sono utilizzati solo due doppini per stazione: uno da e uno verso l'hub. Non sono utilizzate né la codifica binaria diretta (come NRZ) né la codifica Manchester, viene invece utilizzata la codifica 4B/5B: 4 bit di dati sono codificati con 5 bit di segnale e spediti a 125 MHz per ottenere 100 Mbps.

Questo schema è semplice, ma presenta sufficienti transizioni per la sincronizzazione e utilizza la banda del cavo relativamente bene. Il sistema 100Base-TX è full duplex; le stazioni possono trasmettere 100 Mbps su di un doppino e ricevere a 100 Mbps sull'altro nello stesso momento. Fast Ethernet permette interconnessioni attraverso sia hub che switch. Per assicurare che l'algoritmo CSMA/CD continui a funzionare, la relazione tra la dimensione minima del frame e la lunghezza massima del cavo deve essere mantenuta in proporzione alla velocità della rete che va da 10 Mbps a 100 Mbps. Quindi o va aumentata la dimensione minima di frame di 64 byte o dovrà diminuire proporzionalmente la lunghezza massima di cavo di 2500 m. La scelta più facile è stata di abbassare la distanza tra ogni coppia di stazioni di un fattore 10, poiché un hub con un cavo da 100 metri rientra già in questo nuovo limite massimo.

Tutti gli switch fast Ethernet possono gestire un mix di stazioni a 10 e 100 Mbps. Per facilitare l'aggiornamento, lo stesso standard fornisce un meccanismo chiamato **autonegoziazione** (*autonegotiation*) che permette a due stazioni di negoziare automaticamente la velocità ottimale (10 o 100 Mbps) e il tipo di duplex (half o full). Funziona bene per la maggior parte dei casi, ma causa problemi di disallineamento del duplex: un capo del collegamento negozia e l'altro no ed è impostato in modalità full-duplex.

Half-duplex → hub

Full-duplex → switch

## GIGABIT ETHERNET

Lo standard gigabit Ethernet fu ratificato da IEEE nel 1999 con il nome di 802.3ab. Gli obiettivi del comitato per gigabit Ethernet erano gli stessi posti per fast Ethernet: aumentare le prestazioni di dieci volte mantenendo la compatibilità con tutti gli standard Ethernet esistenti, continuare a supportare il CSMA/CD, ecc..

Come fast Ethernet, tutte le configurazioni gigabit Ethernet sono punto a punto. Gigabit Ethernet supporta due modalità operative: full duplex e half duplex. La modalità normale è quella full duplex, che permette al traffico di viaggiare contemporaneamente in entrambe le direzioni. Questa modalità è utilizzata quando c'è uno switch centrale collegato ai computer (o ad altri switch). Poiché non si verifica contesa, non si utilizza il protocollo CSMA/CD, perciò la lunghezza massima del cavo dipende dall'energia del segnale e non dal tempo che un picco di rumore impiega nel peggiore dei casi per propagarsi all'indietro fino al trasmittente.

L'altra modalità operativa, half duplex, è utilizzata quando i computer sono collegati a un hub e non a uno switch. In questa modalità le collisioni sono ancora possibili, perciò è richiesto il protocollo CSMA/CD standard. Poiché un frame di 64 byte (il più piccolo ammissibile) può essere trasmesso 100 volte più velocemente che nella Ethernet classica, la lunghezza massima del cavo risulta 100 volte più corta: 25 metri. Questa restrizione di lunghezza fu così dolorosa da richiedere l'aggiunta di due funzionalità allo standard, al fine di aumentare la lunghezza massima del cavo a 200 metri: La prima funzionalità, chiamata **carrier extension** (*estensione della portante*), indica essenzialmente all'hardware di aggiungere dei dati di riempimento dopo il frame normale, in modo da estendere la dimensione del pacchetto fino a 512 byte. La seconda funzionalità, chiamata **frame bursting**, permette a un trasmittente di inviare una sequenza concatenata di più frame in una singola trasmissione.

Mandare messaggi di segnalazione a 1 Gbps richiede di codificare e spedire un bit per nanosecondo. Questo fu inizialmente ottenuto usando cavi in rame brevi e schermati (la versione 1000Base-CX) e fibre ottiche. Per quanto riguarda quest'ultime, sono permesse due diverse lunghezze d'onda e quindi due versioni diverse: 0,85 micron (corta, per 1000Base-SX) e 1,3 m (lunga, per 1000Base-LX).

È necessario un sistema di segnalazione più complicato per far andare Ethernet a 1000 Mbps su cavi di Categoria 5. Innanzitutto tutti e quattro i doppini del cavo vengono utilizzati e ogni paio è impiegato in entrambe le direzioni contemporaneamente, grazie a un'elaborazione digitale per separare i segnali. Su ogni cavo sono utilizzati cinque livelli di tensione che trasportano 2 bit per ottenere 125 milioni di simboli al secondo. La mappa per produrre i simboli dai bit non è semplice; si tratta di mescolare attraverso transizioni e poi applicare un codice di correzione d'errore per associare quattro valori a cinque livelli di segnale.

Name	Medium	Specified distance
1000BASE-CX	Twinaxial cabling	25 meters
1000BASE-SX	Multi-mode fiber	220 to 550 meters dependent on fiber diameter and bandwidth
1000BASE-LX	Multi-mode fiber	550 meters
1000BASE-LX	Single-mode fiber	5 km[4]
1000BASE-LX10	Single-mode fiber using 1,310 nm wavelength	10 km
1000BASE-ZX	Single-mode fiber at 1,550 nm wavelength	~ 70 km
1000BASE-BX10	Single-mode fiber, over single-strand fiber: 1,490 nm downstream 1,310 nm upstream	10 km
<b>1000BASE-T</b>	Twisted-pair cabling (Cat-5, Cat-5e, Cat-6, or Cat-7)	100 meters
1000BASE-TX	Twisted-pair cabling (Cat-6, Cat-7)	100 meters

## ETHERNET 10 - GIGABIT

Fu ratificata da IEEE per la prima volta nel 2002: 1000 volte più veloce della Ethernet originale: a chi potrebbe servire? La risposta è all'interno dei data center e nelle centrali di commutazione per collegare router di fascia alta, switch e server, così come su collegamenti a lunga distanza e larga banda tra centrali che gestiscono intere MAN basate su Ethernet e fibra. I collegamenti a lunga distanza usano fibra ottica, mentre quelli brevi possono utilizzare rame o fibra. Tutte le versioni di 10-gigabit Ethernet supportano solo la modalità full-duplex.

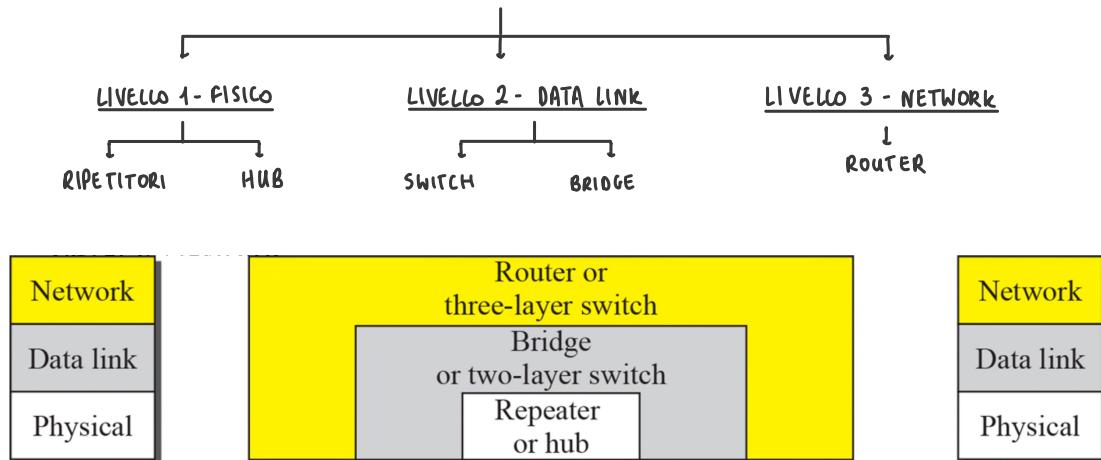
CSMA/CD non fa più parte dell'architettura e gli standard si concentrano sui dettagli dei livelli fisici che possono andare a velocità davvero elevata. La compatibilità è ancora importante, quindi le interface di rete 10-gigabit Ethernet fanno uso di auto negoziazione e, se necessario, ripiegano verso la velocità più elevata supportata da entrambe le estremità della linea. Tutte le versioni di 10GbE inviano un flusso sequenziale di dati prodotto mischiando i bit e poi applicando una codifica **64B/66B**, che ha un overhead minore della codifica **8B/10B**.

→ **GIGABIT ETHERNET**  
codifica

**Ethernet 40/100 Gbit/s**, ratificata nel 2010, raggiunge velocità decisamente più elevate, supportando sempre solo operazioni full duplex, preservando tutte le altre caratteristiche.

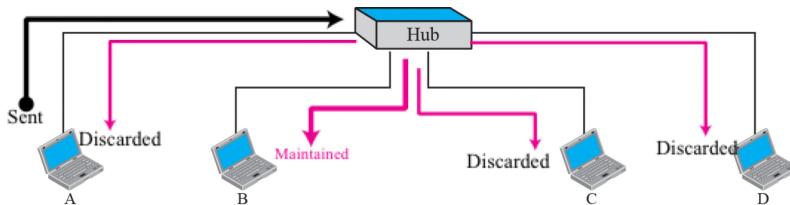
## DISPOSITIVI DI COMMUTAZIONE

I dispositivi di interconnessione si classificano in base al livello in cui operano.



### • Hub

Un hub, o *ripetitore*, che si trova al livello 1 non tratta frame o pacchetti, ma solo bit. Inoltre ogni bit a tutte le porte, e non ha capacità di filtraggio.

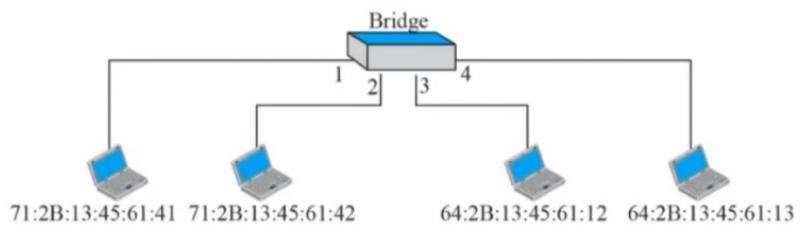


### • Switch (bridge)

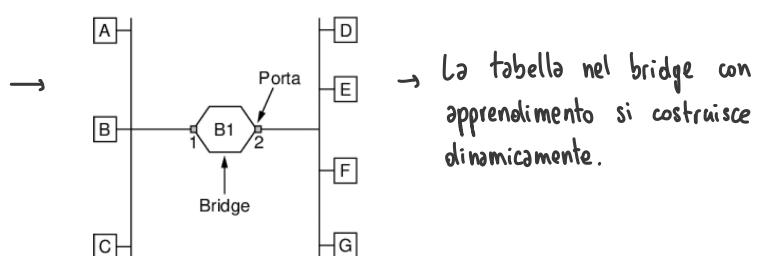
Gli switch Ethernet sono un modo attuale di chiamare i bridge; forniscono funzionalità che vanno oltre quelle di Ethernet classica e gli hub Ethernet per facilitare l'unione di più LAN in una rete più grande. (bridge = switch)

I bridge operano a livello 2, quindi esaminano gli indirizzi del livello data link per inoltrare i frame (non i pacchetti). Con un bridge si possono unire due LAN separate, raggiungendo il doppio della capacità di una singola LAN pur mantenendo questa capacità. Il bridge possiede una tabella di switch e usa gli indirizzi MAC per trovare la porta a cui inoltrare il frame di input, con capacità di filtraggio.

Bridge table	
Address	Port
71:2B:13:45:61:41	1
71:2B:13:45:61:42	2
64:2B:13:45:61:12	3
64:2B:13:45:61:13	4



Nel **learning bridge** si parte da una tabella vuota, che si riempie automaticamente man mano che “conosce” gli indirizzi delle varie porte a cui invia/riceve frame.

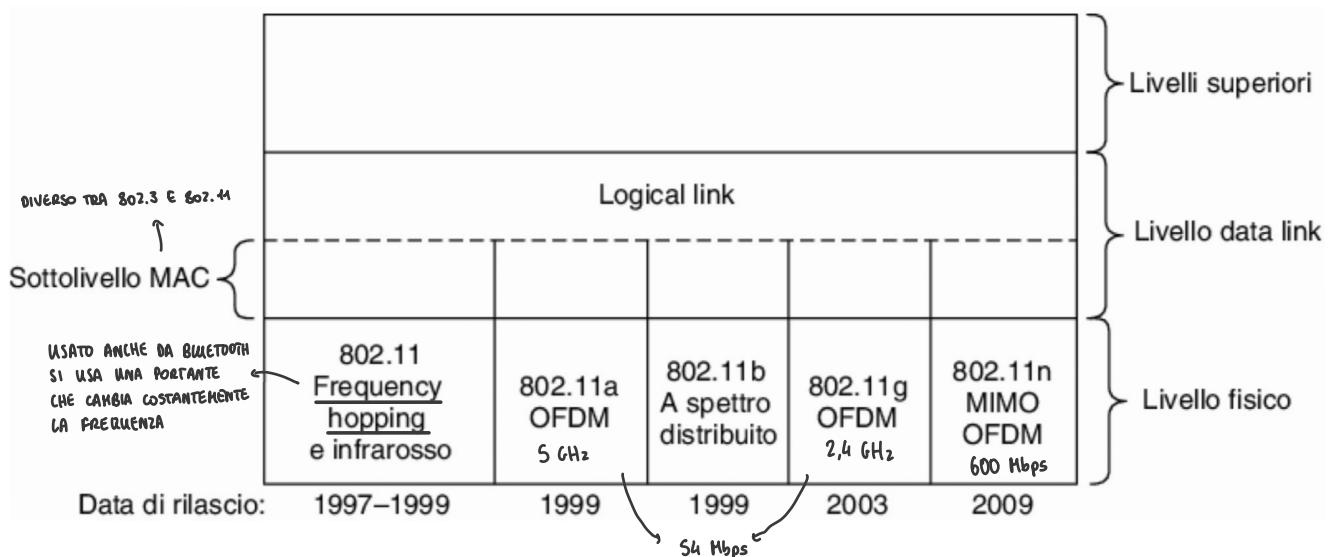


## IEEE 802.11 (WiFi)

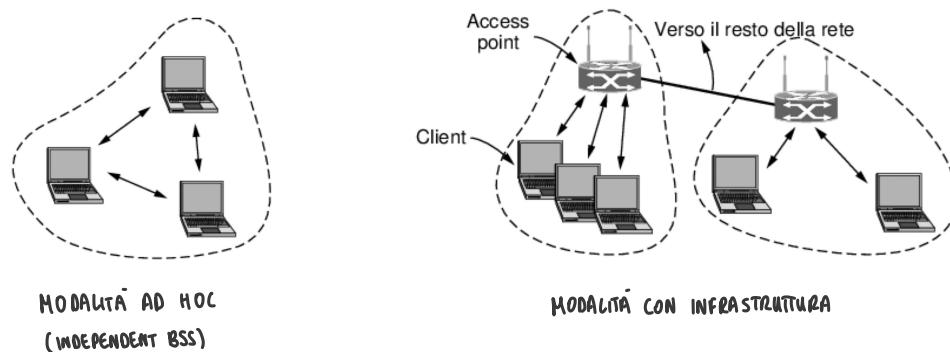
Il principale standard per le **LAN Wireless** è IEEE 802.11, meglio conosciuto come **WiFi** (*Wireless Fidelity*).

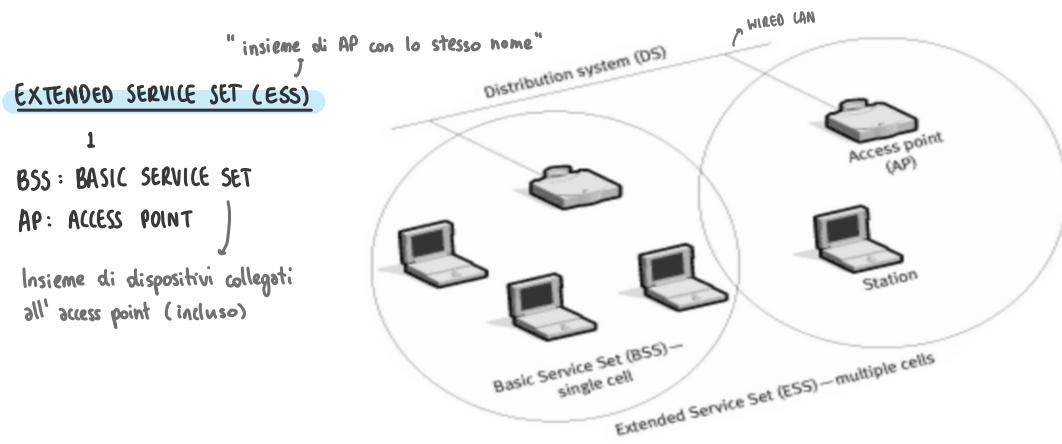
Tutti i protocolli 802, inclusi 802.11 e Ethernet, presentano una certa somiglianza nella struttura. Il livello fisico corrisponde abbastanza bene al livello fisico OSI, ma il livello data link nei protocolli 802 è diviso in due o più sottolivelli.

In 802.11 il sottolivello MAC determina com'è allocato il canale, cioè a chi tocca trasmettere. Sopra c'è il sottolivello **LLC** (*logical link control*) il cui compito è nascondere le differenze tra i differenti 802 e renderli indistinguibili al livello di rete; oggi LLC è un livello di raccordo che identifica il protocollo (ad esempio IP) trasportato all'interno di un frame 802.11.

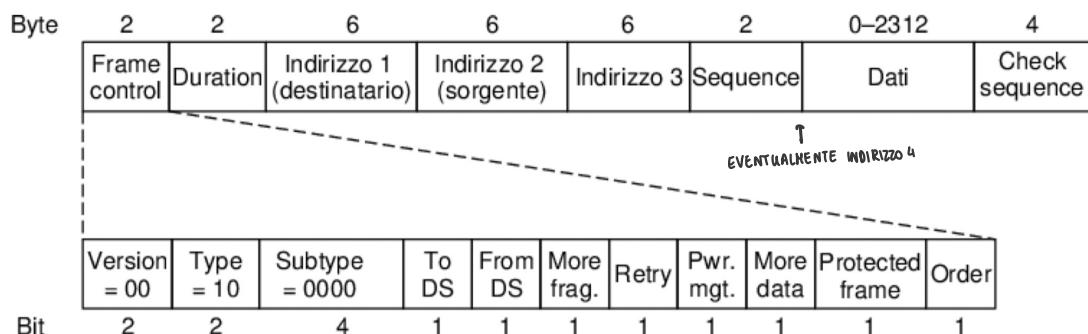


Le reti 802.11 possono essere utilizzate in due modalità: nella **modalità con infrastruttura** (*infrastructure BSS*) (**BSS = Basic Service Set**) ogni client è associato a un **AP** (access point) connesso a sua volta all'altra rete. Il client invia e riceve i propri pacchetti attraverso l'AP. Alcuni access point possono essere connessi insieme, in genere attraverso una rete cablata che prende il nome di **sistema di distribuzione** (*distribution system*). L'altra modalità prende il nome di **rete ad hoc**. Questa modalità consiste in una collezione di computer associati tra loro che possono inviare direttamente i frame. Non esiste alcun access point. Dal momento che l'accesso a Internet è la killer application per le reti wireless, le reti ad hoc non sono molto popolari.



STRUTTURA DEL FRAME DI 802.11

Lo standard 802.11 definisce tre diverse classi di frame: dati, controllo e gestione. Ognuna ha un'intestazione composta da una varietà di campi utilizzati all'interno del sottolivello MAC. Nel formato del **frame di dati** prima compare il campo *Frame control*, costituito da 11 sottocampi. Il primo di questi è *Protocol version*, impostato a 00. È lì per permettere alle future versioni di 802.11 di operare contemporaneamente nella stessa cella. Quindi ci sono i campi *Type* (tipo) (con possibili valori dati, controllo e gestione) e *Subtype* (sottotipo) (ad esempio RTS o CTS); per un frame di dati regolare (senza qualità di servizio) sono fissati a 10 e 0000 (in binario). I bit *To DS* (verso DS) e *From DS* (da DS) sono fissati a indicare se il frame stia andando o venendo dalla rete connessa all'AP, chiamata sistema di distribuzione. Il bit *More fragments* (ulteriori frammenti) significa che seguiranno più frammenti. Il bit *Retry* (riprova) indica la ritrasmissione di un frame spedito in precedenza. Il bit *Power management* (gestione energia) indica che il mittente sta andando in modalità power-save. Il bit *More data* (ulteriori dati) indica che il mittente ha altri dati per il ricevente. Il bit *Protected Frame* (frame protetto) indica che il corpo del frame è stato criptato per sicurezza. Infine, il bit *Order* (in ordine) indica al ricevente che il livello superiore si aspetta che la sequenza di frame arrivi rigorosamente in ordine.



Il secondo campo del frame di dati, *Duration* (durata), indica per quanto tempo il frame e il suo acknowledgment occuperanno il canale, misurato in microsecondi. È presente in tutti i tipi di frame, inclusi i frame di controllo, ed è ciò che le stazioni utilizzano per gestire il meccanismo del NAV (channel booking duration).

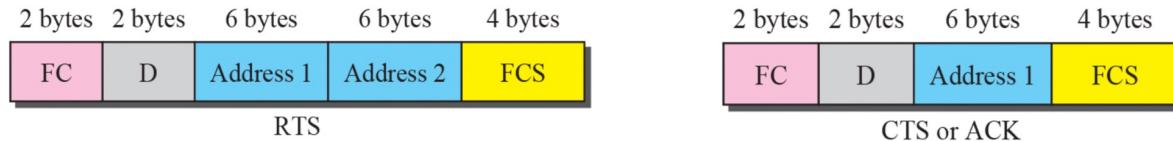
Poi ci sono gli indirizzi. I frame di dati spediti da o per un AP hanno tre indirizzi, tutti in formato standard IEEE 802. Il primo indirizzo è il ricevente e il secondo è il mittente. A che cosa serve il terzo? Un AP è semplicemente un punto di staffetta per i frame quando viaggiano tra un client e un altro punto della rete. Il terzo indirizzo corrisponde a questo distante punto finale.

Il campo **Sequence** (sequenza) numera i frame in modo da poter individuare duplicati. Il campo **Dati** contiene il payload, fino a 2312 byte. I primi byte di questo campo sono in formato conosciuto come **LLC** (*logical link control*). Infine c'è la *Frame check sequence*, che è lo stesso CRC a 32 bit di 802.3.

INVIAI OGNI 100 ms  
PER AVVISARE L'ESS

I **frame di gestione** hanno lo stesso formato dei frame di dati, più un formato per la porzione dati che varia a seconda di Subtype (ad esempio, parametri nei frame di beacon).

I **frame di controllo** sono brevi. Come tutti i frame hanno i campi Frame control, Duration, e Frame check sequence. Tuttavia, potrebbero anche avere un solo indirizzo e nessuno spazio per i dati. La maggior parte delle informazioni chiave viene trasmessa nel campo Subtype (ACK, RTS e CTS). → NON HANNO IL PAYLOAD



Ricapitolando: frame di dati (type 10) i più diffusi, frame di gestione (type 00) e frame di controllo (type 01)

**Canali condivisi:** quando un canale viene condiviso, sono possibili due approcci (entrambi presenti in IEEE 802.11):

- **Polling (centralized):** l'AP “chiede” alle stazioni se hanno frame da inviare.
- **Contention (distributed):** le stazioni devono competere per usare il medium, si utilizza l'algoritmo CSMA-CA (*CSMA Collision Avoidance*).

Polling (PCF) e contention (DCF) possono lavorare simultaneamente nelle reti 802.11 ad infrastruttura, mentre lavorano solo in DCF nelle reti ad hoc.

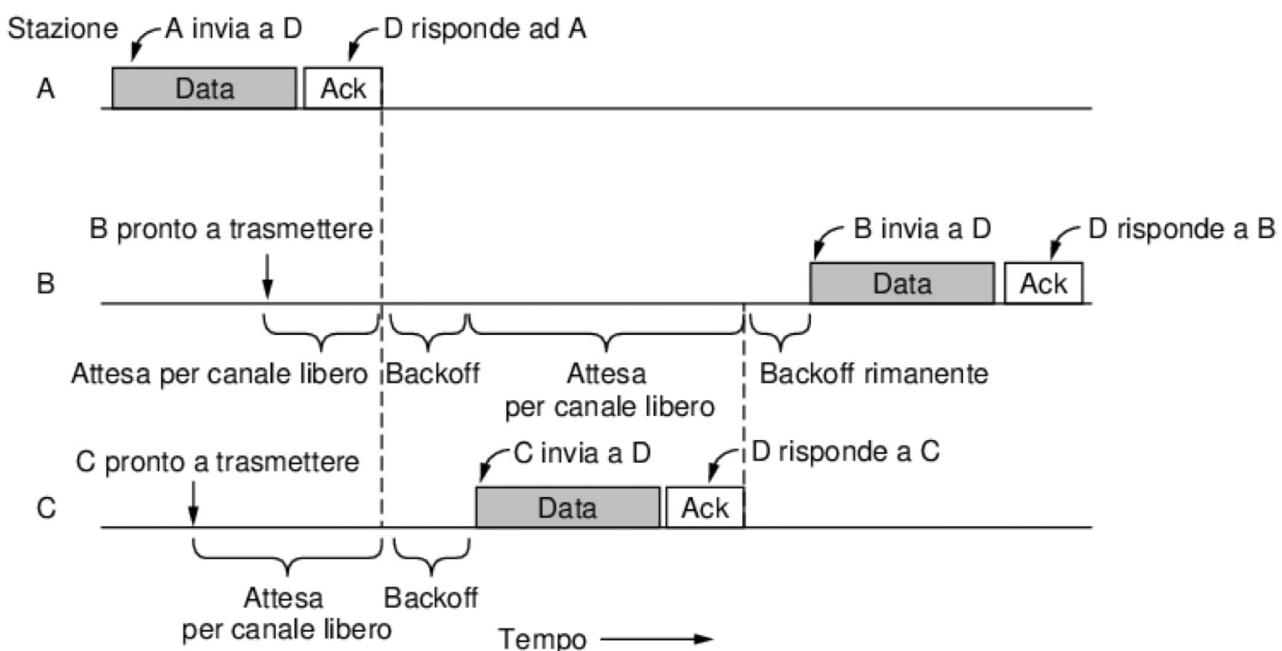
Nelle recenti QoS aware versions (802.11e) è stato creato l'Hybrid Coordination Function (HCF), rispettivamente con:

- Enhanced Distributed Channel Access (EDCA, evoluzione del DCF);
- HCF Controlled Channel Access (HCCA, evoluzione del PCF).

## PROTOCOLLO DEL SOTTOLIVELLO MAC DI 802.11

Con Ethernet, una stazione attende finché il mezzo trasmisivo non diventa libero e inizia a trasmettere. Se non riceve alcun picco di rumore mentre spedisce i primi 64 byte, il frame è quasi certamente stato consegnato correttamente. Con il wireless, questo meccanismo di rilevamento delle collisioni non funziona. Invece, 802.11 prova a evitare le collisioni con un protocollo chiamato **CSMA/CA** (*CSMA with collision avoidance*).

Questo protocollo è concettualmente simile a CSMA/CD di Ethernet, con il rilevamento della portante prima dell'invio e il backoff esponenziale dopo le collisioni. Tuttavia, una stazione con un frame da spedire inizia con un backoff casuale (a meno che non abbia utilizzato il canale di recente e il canale sia libero) e non aspetta che avvenga una collisione. Il numero di slot di backoff è scelto nell'intervallo tra 0 e 15 nel caso del livello fisico OFDM. La stazione attende finché il canale è libero, rilevando che non c'è segnale per un certo periodo di tempo e conta alla rovescia gli slot inattivi, facendo delle pause quando vengono inviati dei frame.



La stazione A è la prima a spedire un frame. Mentre A spedisce, le stazioni B e C diventano pronte per spedire. Vedono che il canale è occupato e attendono che si liberi. Poco dopo che A riceve un acknowledgement, il canale diventa libero. Tuttavia, invece di spedire subito un frame e farlo collidere, entrambe B e C eseguono un random backoff. C sceglie un backoff breve, perciò spedisce per primo. B mette in pausa il conto alla rovescia quando rileva che C sta utilizzando il canale e ricomincia quando C ha ricevuto un acknowledgement. B completa presto il backoff e spedisce il suo frame.

Dal confronto con Ethernet emergono due differenze principali. Primo, far partire presto i backoff aiuta a evitare le collisioni. Secondo, poiché non possono essere rilevate, gli acknowledgement sono usati per dedurre le collisioni.

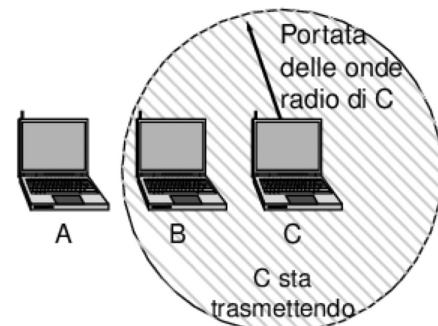
Questo modo di operare è chiamato **DCF** (*distributed coordination function*) perché ogni stazione agisce indipendentemente, senza alcun controllo centrale. Lo standard include anche un modo di operare opzionale chiamato **PCF** (*point coordination function*) in cui l'AP controlla tutta l'attività nella sua cella.

Il secondo problema è che i raggi di trasmissione di stazioni diverse possono essere differenti. Con un cavo, il sistema è progettato in maniera tale che tutte le stazioni possano sentirsi a vicenda.

Di conseguenza, possono sorgere situazioni come il **problema del terminale nascosto**.

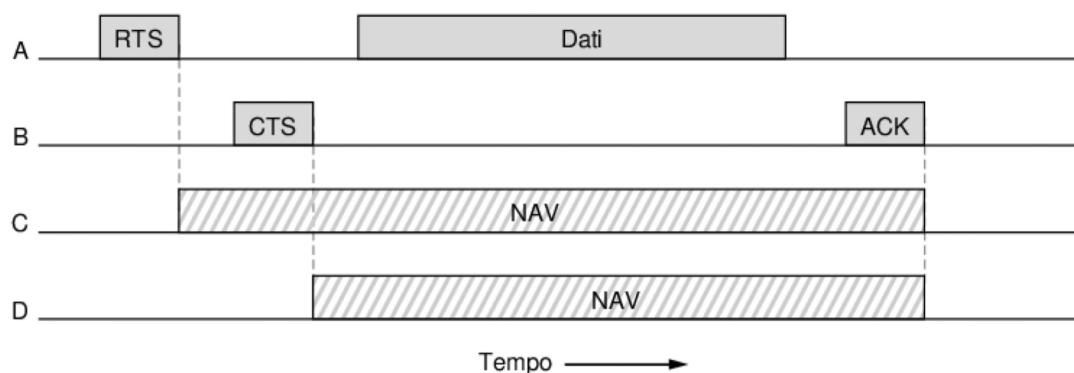
Poiché non tutte le stazioni sono a reciproca portata radio, le trasmissioni che avvengono in una parte di una cella potrebbero non essere ricevute in qualche altro punto nella stessa cella. In questo esempio, la stazione C sta trasmettendo alla stazione B. Se A rileva la portante sul canale, non sentirà nulla e dedurrà erroneamente di poter iniziare a trasmettere a B. Questa decisione genera una collisione.

A vuole trasmettere dati a B  
ma non è in grado di scoprire  
se B è occupato



Per ridurre ambiguità su quale stazione stia inviando dati, 802.11 definisce la rilevazione del canale come composta sia dalla rilevazione fisica sia da quella virtuale. La rilevazione fisica semplicemente controlla il mezzo trasmissivo per capire se c'è un segnale valido. Con la rilevazione virtuale ogni stazione registra quando il canale è occupato tenendo traccia dei **NAV** (*network allocation vector, vettore di allocazione della rete*). Ogni trame possiede un campo NAV che indica quanto tempo sia necessario per completare la sequenza di cui il frame fa parte. Le stazioni che vedono il frame sapranno che il canale resterà occupato per il periodo di tempo indicato dal NAV, indipendentemente dal fatto che sia rilevato un segnale fisico.

Un meccanismo opzionale di RTS/CTS utilizza il NAV per impedire che i terminali spediscono frame contemporaneamente a terminali nascosti. In questo esempio, A vuole spedire dati a B. C è una stazione nel raggio di copertura di A. D è una stazione nel raggio di copertura di B, ma non in quello di A. Il protocollo inizia quando A decide di voler spedire dati a B. A inizia spedendo un frame RTS a B per richiedere il permesso di spedirgli un frame. Se B riceve la richiesta, risponde con un frame CTS per indicare che il canale è libero per la spedizione. Una volta ricevuto il CTS, A spedisce il frame e fa partire un timer per l'acknowledgement (ACK timer). Dopo aver ricevuto correttamente il frame di dati, B risponde con un acknowledgement, completando lo scambio. Se l'ACK timer di A scade prima dell'arrivo dell'acknowledgement viene ipotizzata una collisione e l'intero protocollo ricomincerà di nuovo dopo un backoff.

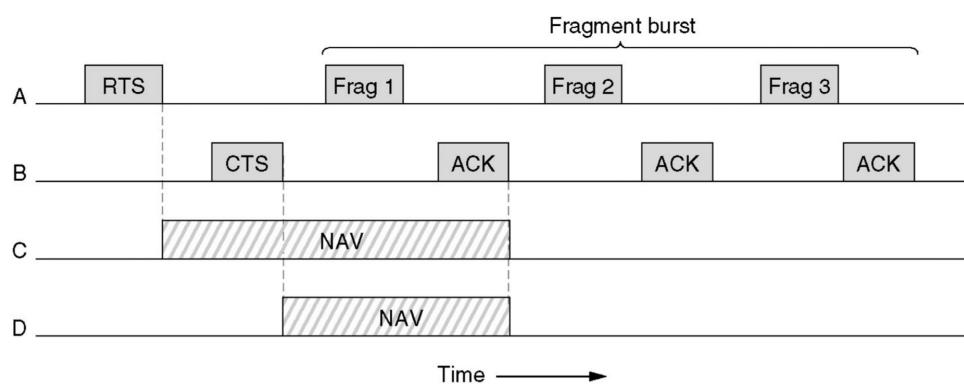


Consideriamo ora questo scambio dal punto di vista di C e D. C'è nel raggio di copertura di A, quindi potrebbe ricevere il frame RTS. Se lo riceve, capisce che qualcuno presto spedirà dati. Dalle informazioni fornite dalla richiesta RTS potrà fare una stima del tempo necessario alla sequenza, incluso l'acknowledgement finale. Quindi, per il bene di tutti, desiste dallo spedire qualsiasi cosa finché non sarà terminato lo scambio. Lo fa aggiornando le sue informazioni interne sui NAV per indicare che il canale è occupato. D non vede il frame RTS, ma vede quello CTS, quindi anche lui aggiorna il suo stato interno sui NAV. Si noti che i segnali NAV non sono trasmessi; sono solo note interne per rimanere inattivi per un certo periodo di tempo.

Tuttavia, mentre RTS/CTS sembra in teoria valida, non è efficace con i frame corti (che vengono spediti al posto degli RTS) e con gli AP (che chiunque può raggiungere, per definizione). Nelle altre situazioni, rallenta solo le operazioni.

CSMA/CA con rilevazione fisica e virtuale è il cuore del protocollo 802.11. Tuttavia, ci sono diversi altri meccanismi sviluppati per funzionare con esso.

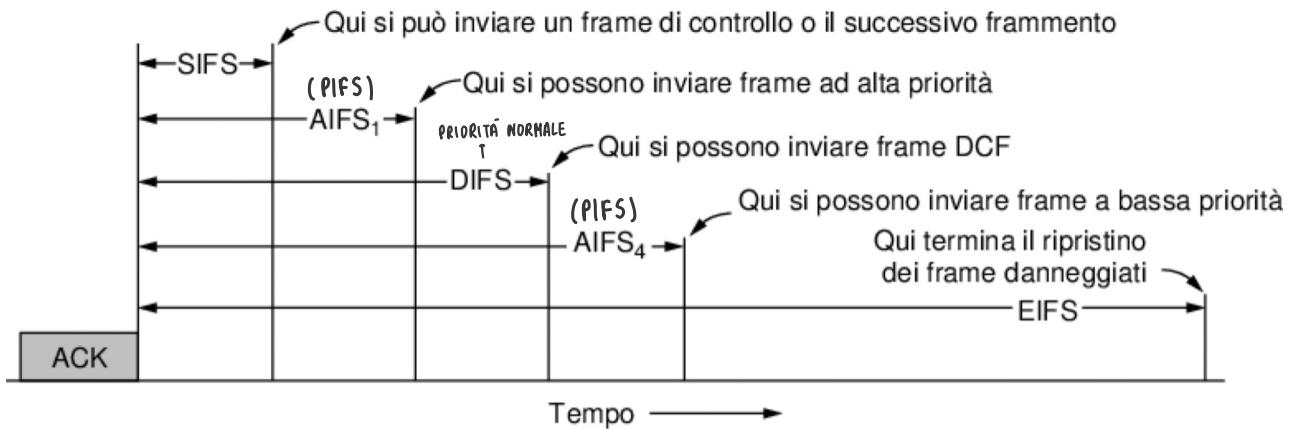
Frame più corti possono essere implementati riducendo la dimensione massima del messaggio accettato dal livello di rete. In alternativa, 802.11 permette di divider i frame in pezzi più piccoli, chiamati **frammenti**, ognuno col proprio checksum. La dimensione di un frammento non è indicata dallo standard, ma è un parametro che può essere aggiustato dall'AP. I frammenti sono numerati individualmente e soggetti ad acknowledgement tramite un protocollo *stop and wait*. Una volta che il controllo del canale è stato acquisito, **molteplici frammenti vengono spediti in una rapida sequenza**. Partono uno dopo l'altro con in mezzo un acknowledgement (e forse ritrasmissioni), finché l'intero frame non sia stato spedito con successo o sia stato raggiunto il tempo massimo di trasmissione permesso. Il meccanismo NAV rende le altre stazioni inattive solo fino al successivo acknowledgement, ma un altro meccanismo è utilizzato per consentire la spedizione di una raffica di frammenti senza che le altre stazioni spediscano un frame nel bel mezzo.



I frammenti offrono maggior efficienza (specialmente nei canali radio molto trafficati) perché:  
più è grande il payload, maggiore è l'efficienza  
efficienza =  $\frac{\text{payload}}{\text{payload} + \text{overhead}}$   
più è grande un pacchetto (block) più è probabile  
che ci sia almeno un bit sbagliato nel pacchetto (con)  
 $n$  = bit di un blocco  
 $P_{e,bit}$  = probabilità d'errore  $\rightarrow P_{e,bn} = 1 - (1 - P_{e,bit})^n$

IEEE 802.11 ha un meccanismo intelligente per fornire questo tipo di **qualità di servizio (QoS, quality of service)**. Funziona estendendo CSMA/CA con intervalli scelti con cura tra i frame. Una volta che il frame è stato spedito è richiesto un certo intervallo di tempo libero prima che una stazione possa spedire un frame per verificare che il canale non sia più utilizzato. Il trucco sta nel definire diversi intervalli di tempo per diversi tipi di frame. Vediamo illustrati cinque intervalli. L'intervallo tra frame di dati è chiamato **DIFS (CF interframe spacing)**. Ogni stazione può cercare di acquisire il controllo del canale per spedire un nuovo frame dopo che il mezzo trasmissivo è rimasto libero per DIFS. Valgono le normali regole di contesa e in caso di collisione potrebbe essere necessario il backoff esponenziale binario. L'intervallo più breve è **SIFS (short interframe spacing)**; viene usato per permettere a coloro che partecipano a uno scambio di messaggi di poter partire per primi.

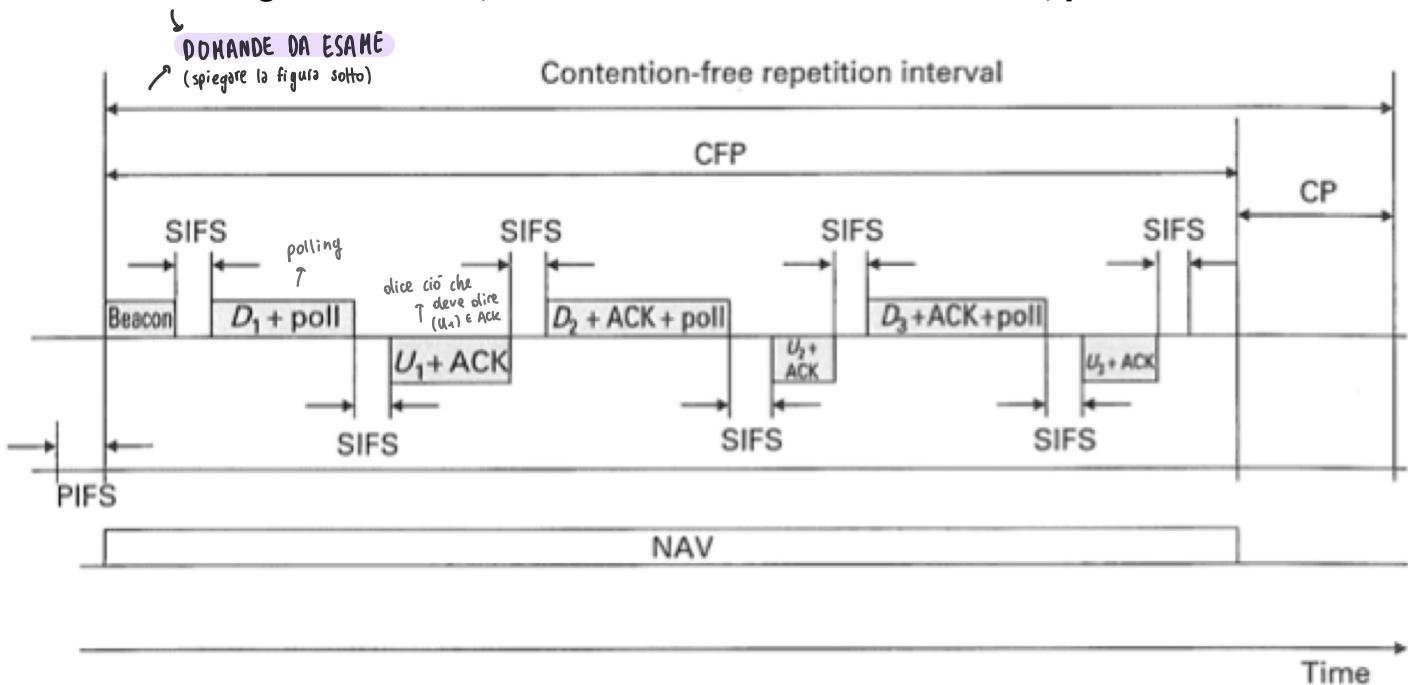
Alcuni esempi includono il permettere al ricevente l'invio di acknowledgement, altre sequenze di frame di controllo come RTS e CTS o il consentire al mittente di trasmettere una raffica di frammenti. Spedire il frammento successivo dopo aver aspettato un solo SIFS impedisce che un'altra stazione possa intromettersi con un frame nel bel mezzo di uno scambio.



I due intervalli **AIFS** (*arbitration interframe space*) mostrano esempi di due differenti livelli di priorità. L'intervallo breve, AIFS<sub>1</sub>, è minore di DIFS, ma più lungo di SIFS. Può essere utilizzato dall'AP per spostare voce o altro traffico ad alta priorità all'inizio della coda. L'AP attenderà un periodo più breve prima di spedire il traffico voce e quindi lo spedirà prima del normale traffico. Il lungo intervallo, AIFS<sub>4</sub>, è maggiore di DIFS ed è utilizzato per il traffico di sottofondo (traffico di background) che può essere differito finché non sia stato smaltito il traffico regolare. L'AP aspetterà per un intervallo più lungo prima di spedire tale traffico, dandogli la possibilità di essere trasmesso prima. L'implementazione completa della qualità di servizio definisce quattro diversi livelli di priorità con diversi parametri di backoff, come pure diversi parametri di inattività.

L'ultimo intervallo di tempo, **EIFS** (*extended interframe spacing*), è utilizzato solo dalle stazioni che hanno ricevuto un frame difettoso o sconosciuto per segnalare il problema.

**Se una stazione deve trasmettere 10 pacchetti e ne ha trasmesso 1, ha la priorità sulla trasmissione degli altri 9? NO, se sono frammenti è un discorso, pacchetti è un altro.**

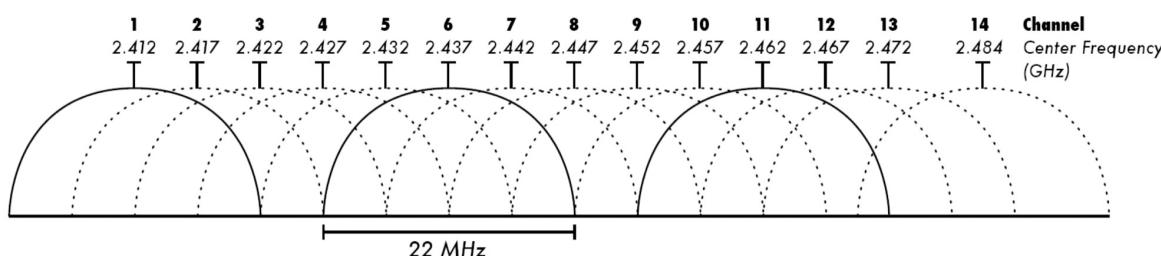


**Problema del terminale esposto:** B vuole inviare dati a C, quindi ascolta il canale; quando sente una trasmissione, conclude erroneamente di non poterlo fare anche se A potrebbe in effetti trasmettere a D (non visibile). Questa decisione spreca un'opportunità di trasmissione.

IL LIVELLO FISICO DI 802.11 → OFDM → già visto, molto importante!

CANALI (22 Mhz, 2.4 GHz band, b and g)

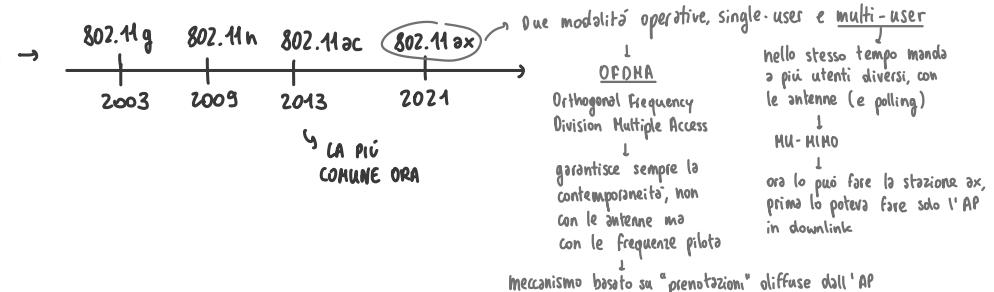
14 CANALI (MASSIMO) OGNIUNO A 22 Mhz (nelle versioni "b" e "g") → Overlapping, interferenze fra canali adiacenti  
1, 6 e 11 non si sovrappongono  
In a, b, g il minimo è 20 KHz e non 22



Il metodo di trasmissione 802.11a è basato su OFDM, dato che questo utilizza lo spettro in maniera efficiente e resiste alla degradazione del segnale wireless. I bit sono inviati su 52 sottocanali in parallelo; 48 per trasportare i dati e 4 per la sincronizzazione. Ogni simbolo dura 4ms e invia 1, 2, 4 o 6 bit. I bit sono innanzitutto codificati per la correzione degli errori con un codice di convoluzione binario, quindi solo 1/2, 2/3 o 3/4 dei bit non sono ridondanti. Con diverse combinazioni, 802.11a può raggiungere otto diverse velocità, che vanno da 6 a 54 Mbps. Questi tassi di trasmissione sono significativamente più veloci di quelli di 802.11b e ci sono anche meno interferenze nella banda a 5 GHz. Tuttavia, 802.11b ha un raggio di copertura sette volte più grande di quello di 802.11a, il che è più importante in molte situazioni.

Il comitato IEEE iniziò a lavorare su un livello fisico ad alta capacità chiamato 802.11n, ratificato nel 2009. L'obiettivo per 802.11n era un throughput di almeno 100 Mbps dopo la rimozione di tutti gli overhead. Questo obiettivo richiede un aumento dell'intera velocità di un fattore quattro. Perché ciò accadesse, il comitato raddoppiò i canali da 20 MHz a 40 MHz e ridusse gli overhead dei frame, permettendo a un gruppo di frame di essere spedito congiuntamente. Più significativamente, 802.11n utilizza fino a quattro antenne per trasmettere fino a quattro flussi d'informazione contemporaneamente. I segnali di questi flussi interferiscono col ricevitore, ma possono essere separati utilizzando le tecniche di comunicazione **MIMO** (*multiple input multiple output*). L'utilizzo di più antenne dà una grande spinta alla velocità oppure migliora raggio di copertura e affidabilità.

SOMMARIO DELL'EVOLUZIONE DI 802.11

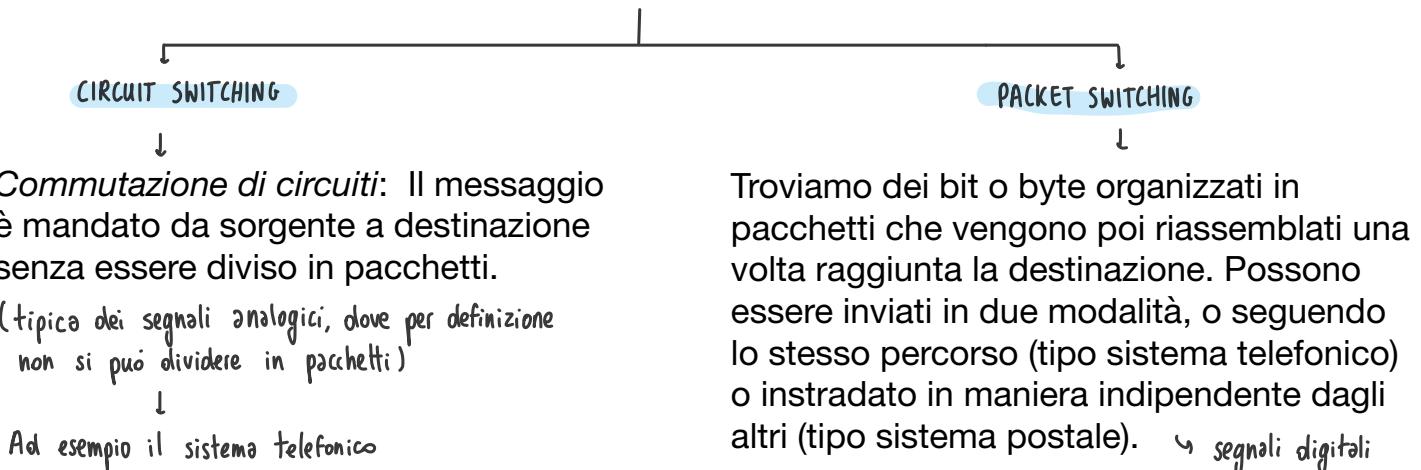


LIVELLO 3 - NETWORK LAYER

Il livello rete si occupa del trasporto dei pacchetti lungo tutto il percorso dall'origine alla destinazione finale, che per essere raggiunta può richiedere molti salti attraverso router intermedi lungo il percorso. Questa funzione è chiaramente distinta da quella del livello data link, che ha il solo obiettivo di spostare i frame da un capo all'altro di un cavo. Per raggiungere i propri obiettivi, il livello di rete deve conoscere la topologia della rete di comunicazione (ossia l'insieme dei router e dei collegamenti) e scegliere i percorsi appropriati attraverso di essa. Nel compiere questa scelta deve anche evitare di sovraccaricare alcune linee di comunicazione lasciando altre completamente libere. Deve infine occuparsi dei nuovi problemi che nascono quando la sorgente e la destinazione si trovano in reti diverse.

Per capire come fa, è necessario rivedere alcuni concetti già visti in precedenza:

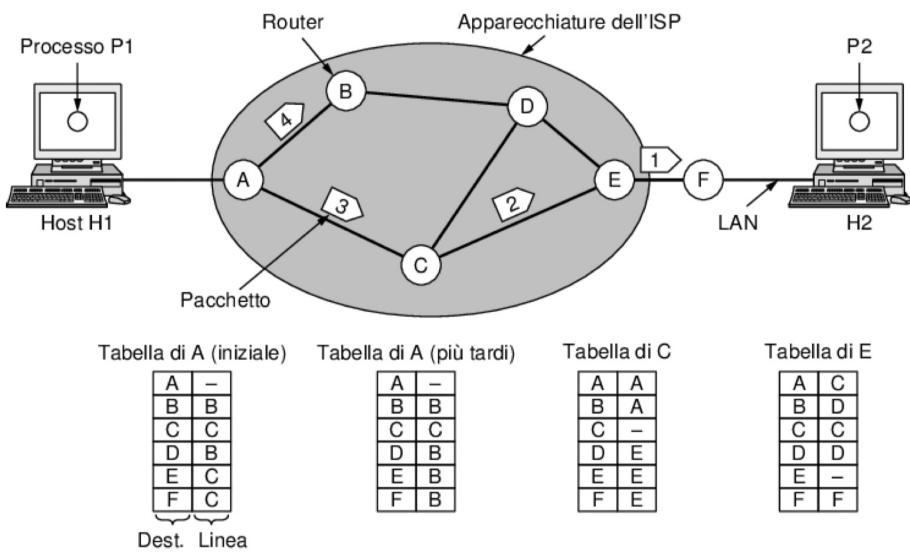
**Switching:** è chiaro che il passaggio di un messaggio da sorgente a destinazione implica molte decisioni. Quando un messaggio raggiunge un dispositivo di connessione, bisogna decidere da quale porta di output il pacchetto deve essere inviato. In poche parole, il dispositivo si comporta come uno switch che connette una porta ad un'altra porta.



Sono possibili due diverse organizzazioni, che dipendono dal tipo di servizio offerto. Se il servizio è **senza connessione**, i pacchetti sono inoltrati nella rete individualmente e instradati indipendentemente l'uno dall'altro; non occorre configurazione anticipata. In questo contesto i pacchetti sono chiamati spesso datagrammi o **datagram** e la rete è chiamata rete datagram. Se il servizio è **orientato alla connessione**, prima di inviare i pacchetti si deve stabilire un percorso che colleghi il router sorgente al router destinazione. Questa connessione è chiamata **VC** (*virtual circuit, circuito virtuale*), in analogia con i circuiti fisici configurati dal sistema telefonico e la rete è chiamata rete a circuito virtuale.

IMPLEMENTAZIONE DEL SERVIZIO SENZA CONNESSIONE (connection-less)

Si supponga che il processo P1 abbia un lungo messaggio per P2. Il processo sorgente porge il messaggio al livello di trasporto dando istruzioni di inoltrarlo al processo P2 che si trova sull'host H2. Il livello di rete suddivide il messaggio in quattro pacchetti, 1, 2, 3, e 4, e invia un pacchetto dopo l'altro al router. Ogni router ha una tabella interna che indica dove devono essere inviati i pacchetti diretti a ogni possibile destinazione. Ogni voce della tabella è costituita da una coppia di valori, che rappresentano una destinazione e la linea di trasmissione da utilizzare per raggiungerla. Possono essere utilizzate solo linee collegate direttamente.



Non appena raggiungono A, i pacchetti 1, 2 e 3 sono archiviati temporaneamente (per verificarne il checksum) e poi ogni pacchetto viene inoltrato verso C secondo la tabella di A all'interno di un nuovo frame. Il pacchetto 1 viene quindi trasmesso a E e poi a F. Quando arriva a F, il pacchetto è incapsulato in un frame e inviato ad H2 attraverso la LAN; i pacchetti 2 e 3 seguono lo stesso percorso.

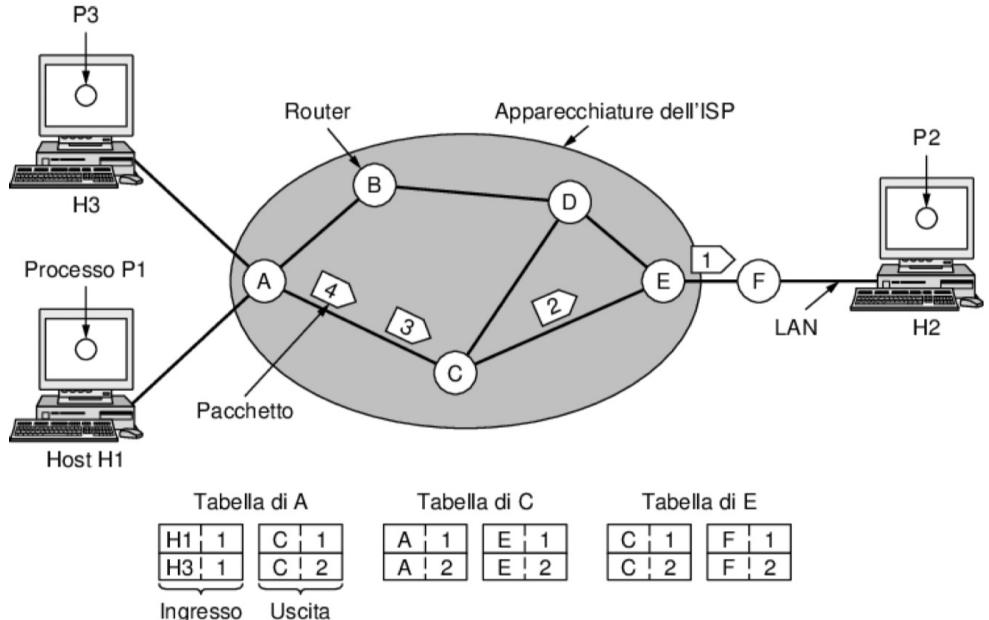
Al pacchetto 4, invece, accade qualcosa di diverso. Quando raggiunge A viene inviato al router B, anche se la destinazione finale rimane F. A, probabilmente, dopo avere rilevato un ingorgo di traffico da qualche parte nel percorso A-C-E, ha aggiornato la sua tabella di routing (tabella di A "più tardi"). L'algoritmo che gestisce la tabella e prende le decisioni di instradamento è chiamato **algoritmo di routing**.

↓  
Non è che normalmente questo succede, è un caso d'esempio, ma può succedere → **NON È GARANTITA LA CONSEGNA ORDINATA**

### IMPLEMENTAZIONE DEL SERVIZIO ORIENTATO ALLA CONNESSIONE

L'idea alla base dei circuiti virtuali è quella di evitare di dover scegliere una nuova strada per ogni pacchetto inviato. Invece, quando è stabilita una connessione, il percorso dal computer sorgente a quello di destinazione viene scelto durante l'impostazione della connessione e archiviato nelle tabelle dei router. Quel percorso è utilizzato per tutto il traffico che scorre attraverso la connessione, con funzionamento identico a quello del sistema telefonico. Quando la connessione viene rilasciata, anche il circuito virtuale viene terminato. Con un servizio orientato alla connessione, ogni pacchetto contiene un identificatore che indica il circuito virtuale di appartenenza. L'host H1 ha stabilito la connessione 1 con l'host H2. La connessione è memorizzata nella prima voce delle tabelle di routing. La prima linea della tabella di A indica che, se da H1 arriva un pacchetto contenente l'identificatore di connessione numero 1, allora i dati devono essere inviati al router C assegnando ancora l'identificatore 1. In modo analogo la prima voce della tabella di C instrada il pacchetto a E, ancora con l'identificatore di connessione numero 1. Che cosa accade se anche H3 cerca di stabilire una connessione con H2? L'host sceglie l'identificatore di connessione numero 1 (perché sta iniziando la connessione e questa è la sua unica connessione) e richiede alla rete di stabilire un circuito virtuale. Ciò conduce alla seconda riga della tabella: in questo caso si verifica un conflitto, perché sebbene A possa facilmente distinguere i pacchetti della connessione 1 provenienti da H1 dai pacchetti della connessione 1 provenienti da H3, C non è in grado di farlo. Per questo motivo A assegna un diverso identificatore di connessione al traffico in uscita relativo alla seconda connessione.

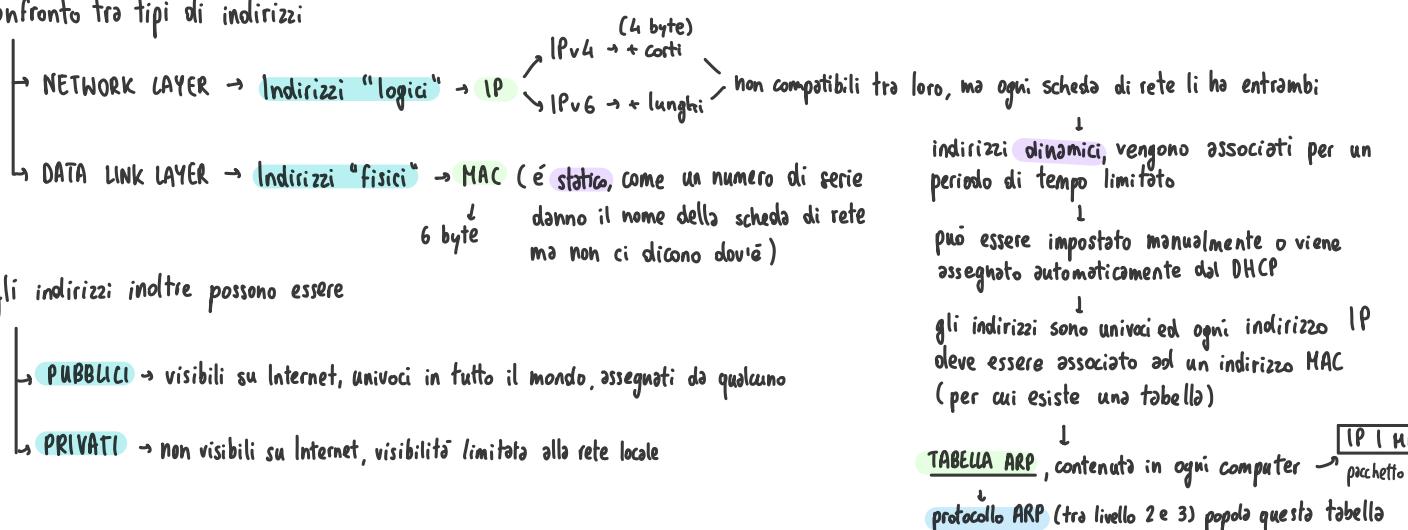
Ai router è stata data la capacità di sostituire gli identificatori di connessione nei pacchetti in uscita proprio per evitare conflitti di questo tipo. In alcuni contesti questa capacità è chiamata **commutazione di etichetta (label switching)**.



Un esempio di servizio orientato alla connessione è **MPLS (multiprotocol label switching)**, usato in Internet all'interno delle reti dei ISP; con MPLS i pacchetti IP sono incapsulati in un'intestazione MPLS avente un identificatore di connessione a 20 bit.

Problema	Rete datagram	Rete a circuito virtuale
Impostazione del circuito	Non necessaria	Necessaria
Indirizzamento	Ogni pacchetto contiene gli indirizzi completi di sorgente e destinazione	Ogni pacchetto contiene un breve identificatore di circuito virtuale
Informazioni di stato	I router non mantengono informazioni di stato sulle connessioni	Ogni circuito virtuale richiede spazio nella tabella del router per la connessione
Routing	Ogni pacchetto è inoltrato in maniera indipendente	Il percorso viene scelto durante l'impostazione del circuito e tutti i pacchetti lo seguono
Effetto del malfunzionamento di un router	Nessuno, tranne i pacchetti persi perché in transito durante il malfunzionamento	Tutti i circuiti virtuali che passano dal router vengono chiusi
Qualità di servizio	Difficile	Facile se abbastanza risorse possono essere allocate in anticipo a ogni circuito virtuale
Controllo della congestione	Difficile	Facile se abbastanza risorse possono essere allocate in anticipo a ogni circuito virtuale

### Confronto tra tipi di indirizzi:

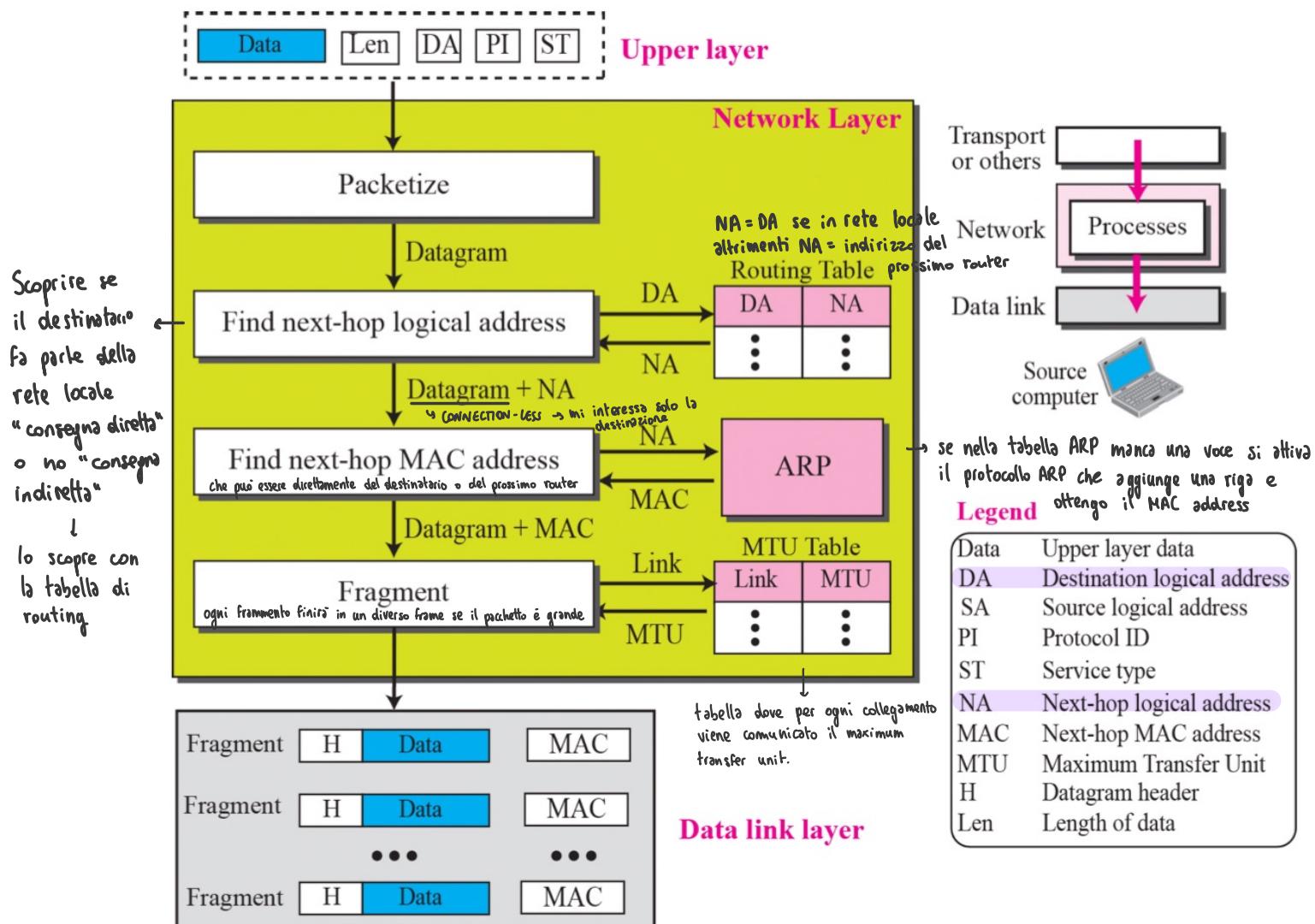


La **tavella ARP** presente su ogni computer deve mantenere la corrispondenza tra indirizzo IP e indirizzo MAC di ogni scheda di rete (può essere anche impostata in maniera statica, anche se non è molto comune). Viene popolata tramite un protocollo ARP (Address Resolution Protocol) che si colloca sopra il livello 2 ma non usa ancora il protocollo IP, un protocollo ausiliario che diciamo "sta a metà", si appoggia direttamente ai frame.

Protocollo ARP → se non trova gli indirizzi → richiesta: lo manda in broadcast sulla rete locale → manda ARP reply con tabella di 4 campi (IP/MAC sorgente e destinatario) riempita.  
 ↓  
 risolvere IP: trovare indirizzo MAC corrispondente

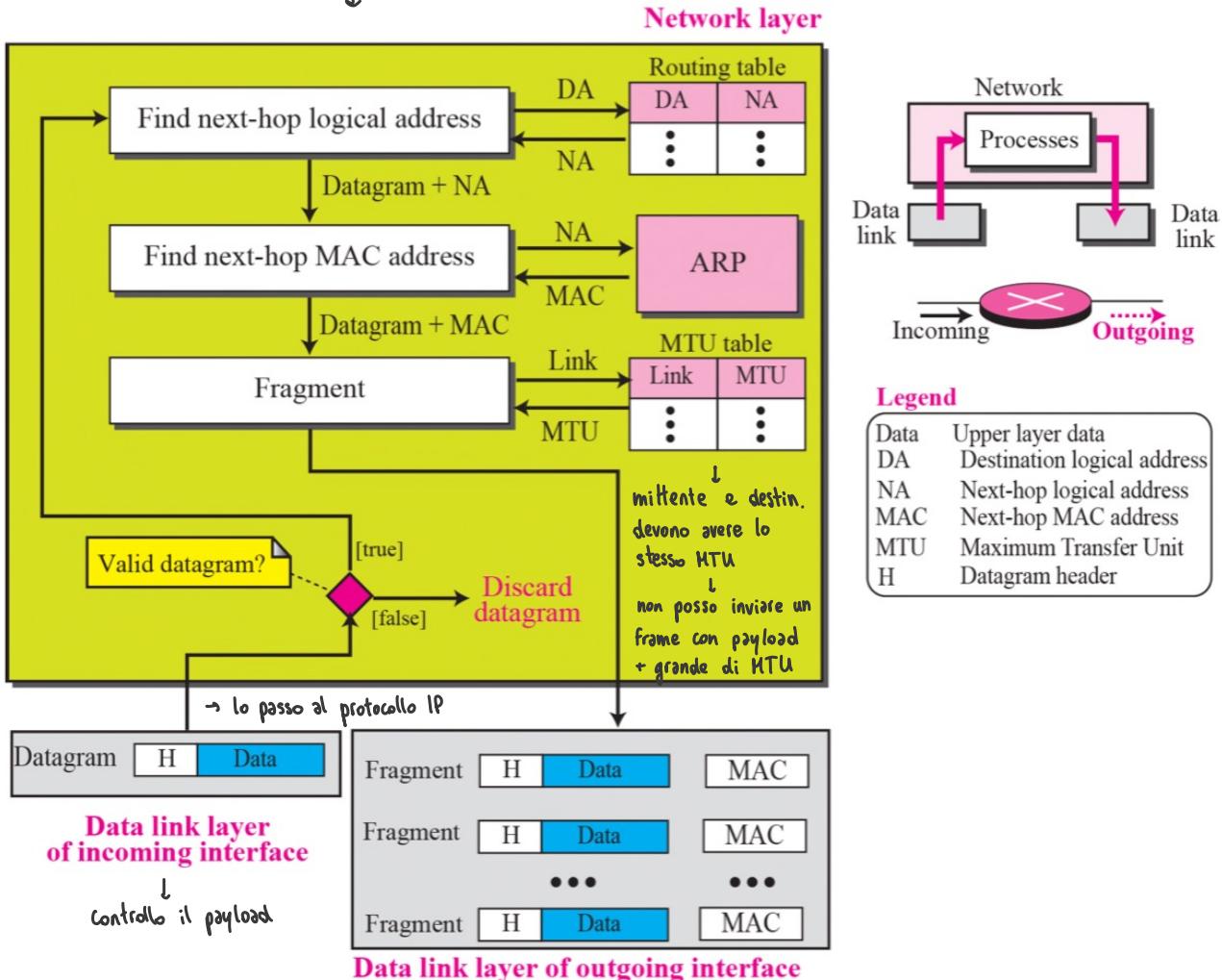
**Tabelle di routing:** (livello 3) sono tabelle che ci permettono di sapere qual è il prossimo router. Se voglio mandare un pacchetto ad un dispositivo non collegato alla rete locale, devo mandare l'indirizzo IP ad un router locale che lo manderà ad un altro router ecc.. Ogni router "chiede" quali sono i router vicini, riempie una tabella e la manda in broadcast. Ciò si ripete per ogni router, così che il router stesso sia in grado di ricreare la topologia della rete. Fa ciò con un algoritmo chiamato DISTRA, con cui trova il percorso a lunghezza minima.

### 1) FUNZIONAMENTO ILLUSTRATO DEI PROTOCOLLI A LIVELLO 3 → CUORE DI INTERNET → DOMANDA DA ESAME

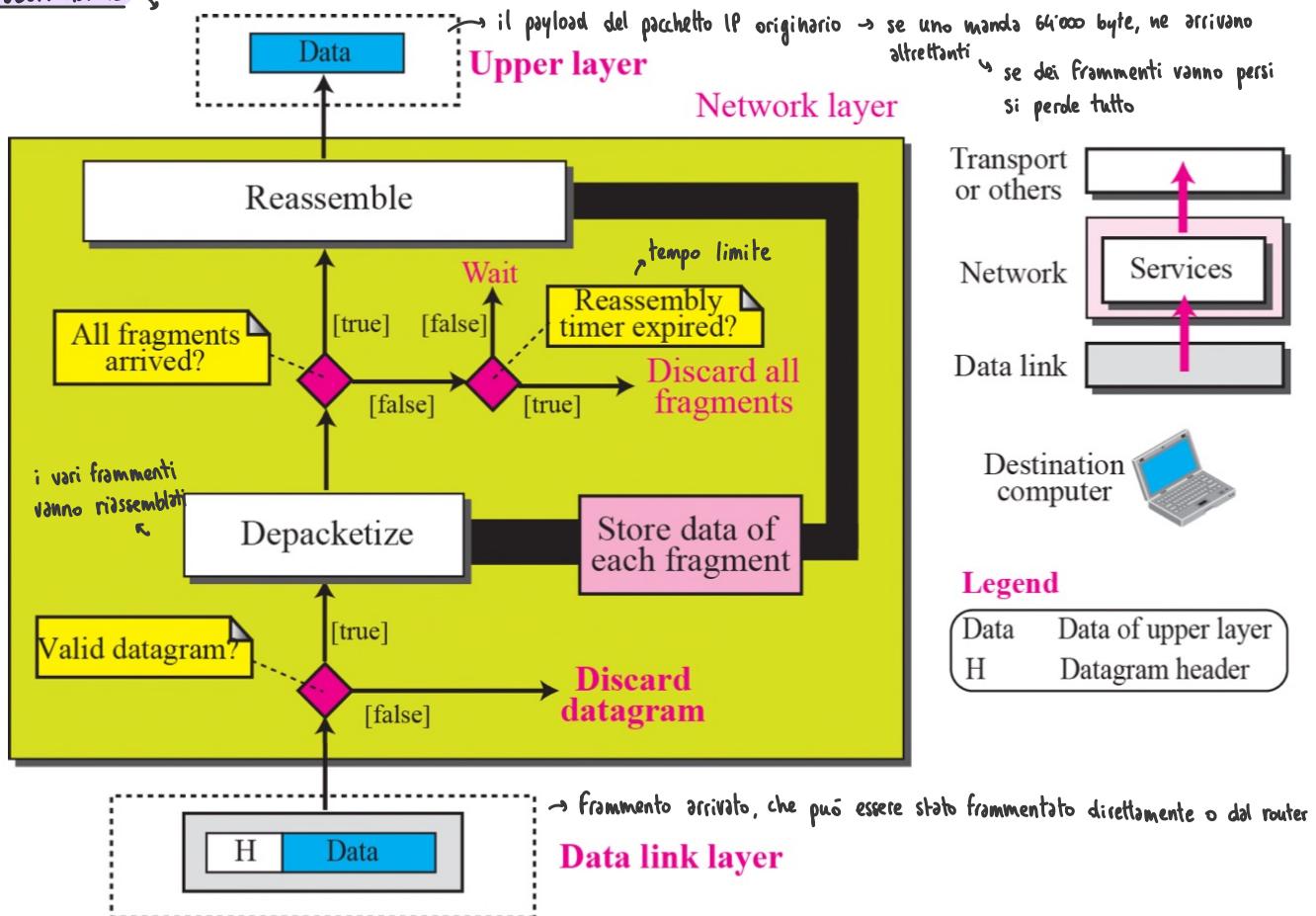


## 2) FUNZIONAMENTO DEI PROCESSI AD OGNI ROUTER

→ Funziona esattamente allo stesso modo del grafico di prima



## 3) COMPUTER DI DESTINAZIONE

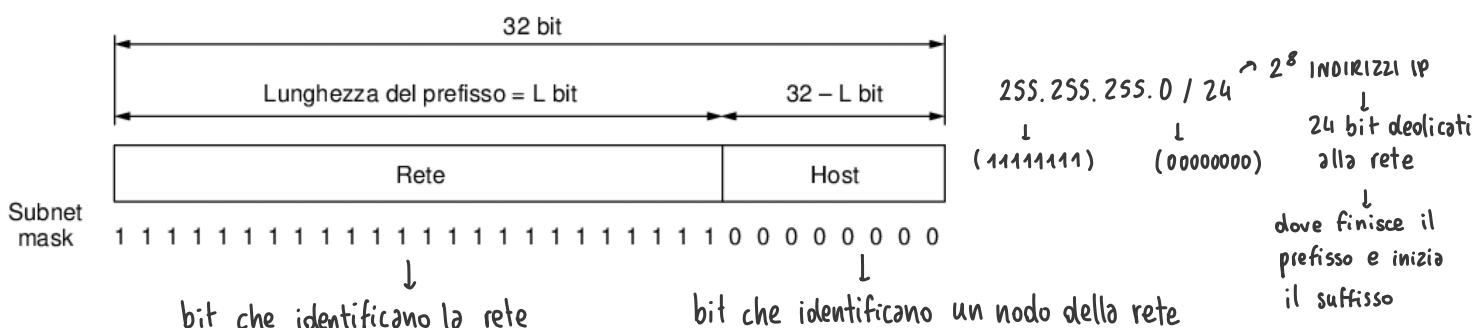


PROTOCOLLO IPv4

L'identificatore usato nel livello IP del protocollo TCP/IP per identificare ogni dispositivo connesso ad Internet si chiama indirizzo IP. Un indirizzo IPv4 è un indirizzo a 32-bit che definisce universalmente e unicamente la connessione di un host o router ad Internet. È importante ricordare che l'indirizzo IP non è parte del computer, ma si riferisce alla connessione che si instaura, infatti un dispositivo può avere tanti indirizzi IP quante connessioni (solitamente un host ne ha uno, ma ad esempio un router ne ha tanti). Gli indirizzi IP hanno validità topologica, dicono dove si trova il dispositivo o router connesso.

Ogni indirizzo a 32 bit è composto di una parte di rete di lunghezza variabile nei primi bit e di una parte per l'host negli ultimi. La parte di rete ha lo stesso valore per tutti gli host di una singola rete, come in una LAN Ethernet. Questo significa che una rete corrisponde a un blocco contiguo di indirizzi IP; tale blocco è chiamato **prefisso**. Gli indirizzi di rete sono scritti in **notazione decimale puntata**. In questo formato ognuno dei 4 byte è rappresentato in forma decimale con un numero che varia tra 0 e 255. Per esempio l'indirizzo esadecimale di 32 bit 80D00297 è scritto come 128.208.2.151. I prefissi sono scritti dando l'indirizzo IP più basso del blocco e la grandezza del blocco. La dimensione è determinata dal numero di bit nella porzione di rete; i bit rimanenti nella parte dedicata all'host possono variare. Questo significa che la dimensione deve essere una **potenza di due**. Per convenzione viene scritta dopo il prefisso dell'indirizzo IP come una barra obliqua / (**slash**) seguito dalla lunghezza in bit della parte dedicata alla rete.

A volte i prefissi sono descritti semplicemente dalla loro lunghezza, come in "/16". La lunghezza del prefisso corrisponde a una maschera binaria di 1 nella parte di rete. Quando viene scritta in questo modo è chiamata **subnet mask** (*maschera di sottorete*).

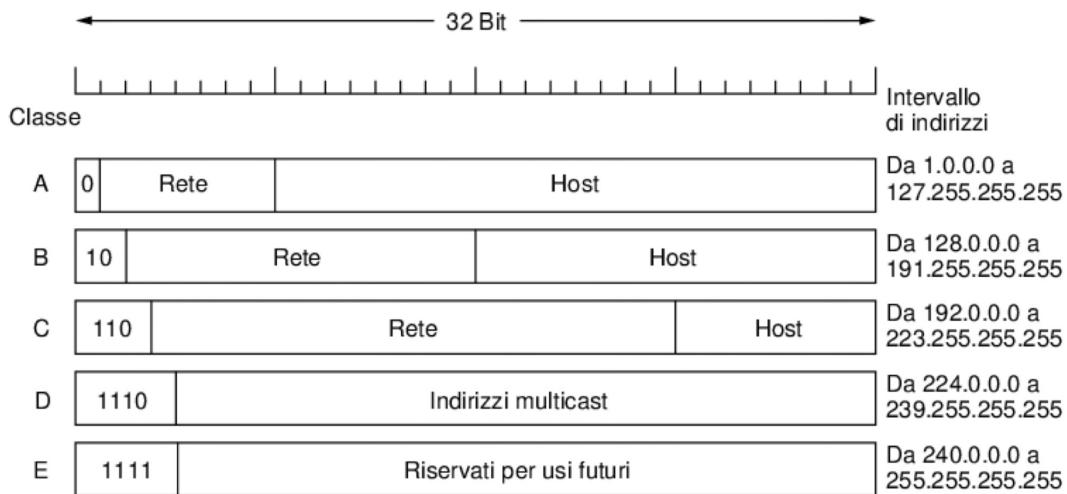


$137.204.143.18 /25 \rightarrow$  si complica quando non abbiamo multipli di 8  
 $\underline{11111111} \underline{11111111} \underline{11111111} \underline{10000000}$   
 255 255 255 128  
 $10001001 \ 11001100 \ 10001111 \ 01100010 \rightarrow$  conversione binaria dell'indirizzo IP, se faccio l'AND bit a bit con la maschera ottengo i bit del prefisso

ESEMPIO  $\rightarrow 17.63.110.114 /24 \rightarrow$  qual è il numero di indirizzi?  $2^{32} - 2^4 = 256 \rightarrow$  essendo la maschera 255.255.255.0 (potenza di 2) il primo indirizzo del blocco sarà 17.63.110.0

$\downarrow$   
 $17.63.110.114$   
 $255.255.255.0$   
 $\downarrow \downarrow \downarrow \downarrow$   
 $17.63.110.0$

Prima del 1993 gli indirizzi IP erano suddivisi in cinque categorie, secondo una classificazione chiamata **classfull addressing (indirizzamento per classi)**.



La dimensione dei blocchi di indirizzi era fissa, poi vennero introdotte le subnet per assegnarla in modo flessibile. Attualmente i bit che indicano se un indirizzo appartiene a una classe non vengono più usati, ora gli indirizzi IP possono avere prefissi di qualsiasi lunghezza.

Ci sono ulteriori **indirizzi speciali**: l'indirizzo IP **0.0.0.0** è utilizzato dagli host al momento del boot e significa "questa rete" o "questo host". Gli indirizzi IP che hanno lo 0 come numero di rete si riferiscono alla rete corrente, e grazie a loro i computer possono utilizzare la rete locale senza conoscerne il numero (devono però conoscere la maschera di rete, per sapere quanti 0 includere). L'indirizzo composto da **tutti 1**, o 255.255.255.255, viene utilizzato per indicare tutti gli host di una rete. Permette la trasmissione broadcast sulla rete locale. Infine tutti gli indirizzi espressi nella **forma 127** sono riservati per effettuare controlli all'interno dell' host stesso (*loopback*). I pacchetti diretti a quell'indirizzo non sono immessi nel cavo, ma elaborati localmente e trattati come pacchetti in arrivo, spesso utilizzati per effettuare dei test.

0 0	Questo host
0 0      ...      0 0	Host
1 1	Trasmissione broadcast sulla rete locale
rete      1 1 1 1      ...      1 1 1 1	Trasmissione broadcast su una rete remota
127	Loopback
(Qualsiasi cosa)	

Indirizzi privati → nessun pacchetto contenente questi indirizzi può apparire su Internet → tre intervalli riservati

10.0.0.0 – 10.255.255.255/8 (16.777.216 host)

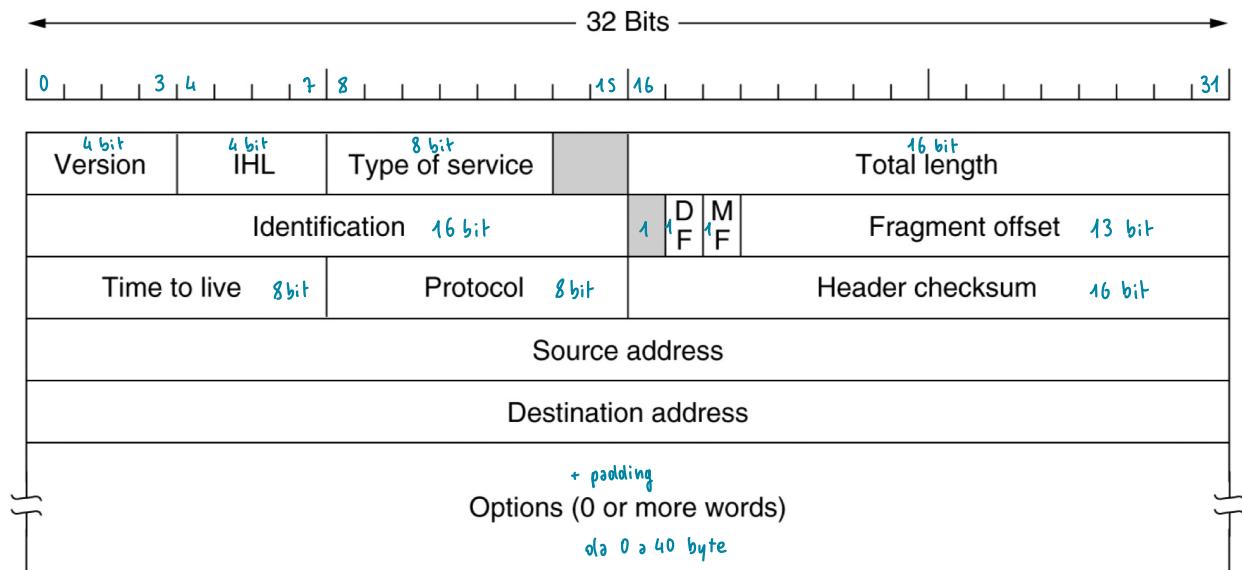
172.16.0.0 – 172.31.255.255/12 (1.048.576 host)

192.168.0.0 – 192.168.255.255/16 (65.536 host)

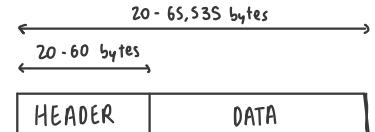
L'**Internet Protocol (IP)** è il meccanismo di trasmissione utilizzato dai protocolli TCP/IP nel network layer.

TCP/IP Layers		TCP/IP Protocols				
Application Layer		HTTP	FTP	Telnet	SMTP	DNS
Transport Layer		TCP		UDP		
Network Layer		IP		ARP	ICMP	IGMP
Network Interface Layer		Ethernet	Token Ring		Other Link-Layer Protocols	

I pacchetti nel network layer sono chiamati datagrammi. Un datagramma IPv4 è costituito da una parte di intestazione e da un corpo (il payload). L'**intestazione** ha una parte fissa di 20 byte e una parte opzionale di lunghezza variabile. I bit vengono trasmessi in ordine da sinistra a destra e dall'alto verso il basso, a partire dal bit di ordine più elevato del campo *Version*, che tiene traccia della versione del protocollo usato per il datagramma.



Poiché la lunghezza dell'intestazione è variabile, il campo *IHL* indica la sua lunghezza: il valore minimo è 5, quando non sono presenti altre opzioni; la dimensione massima è 60 byte e il campo Options non può occupare più di 40 byte.

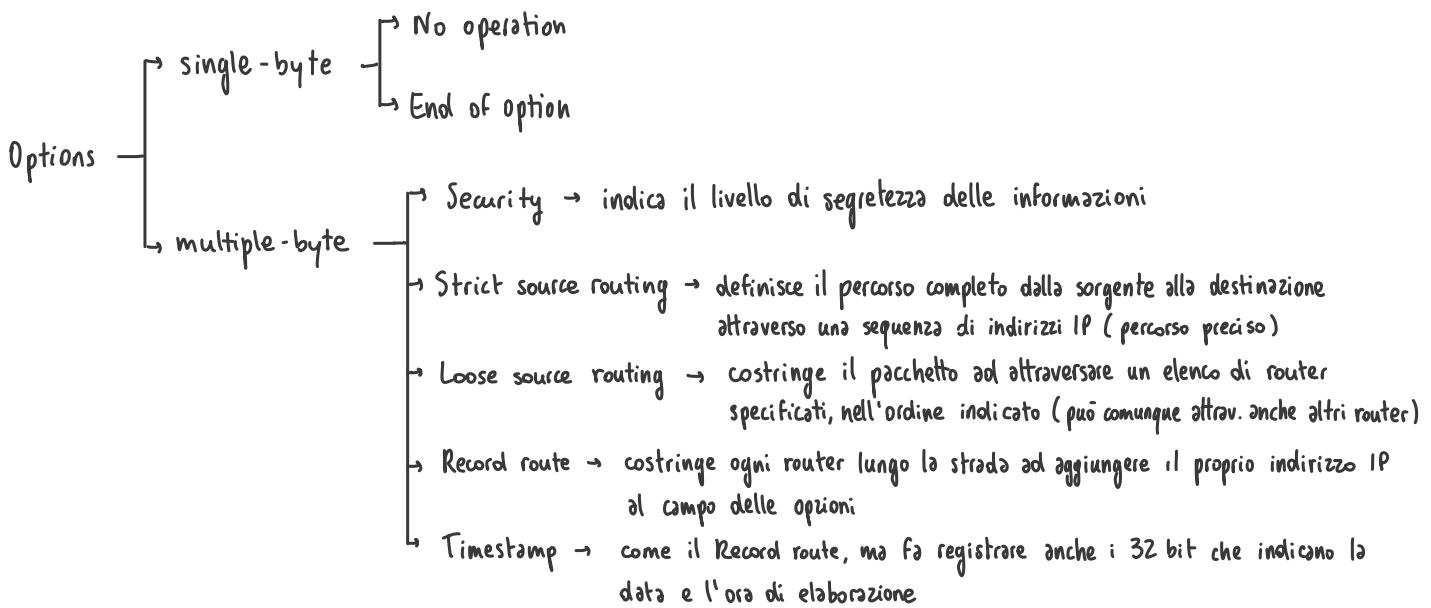


Il campo *Type of service* indica la classe di servizio di appartenenza del pacchetto. *Total lenght (lunghezza totale)* tiene conto di tutto il contenuto del datagramma, intestazione e dati; la lunghezza massima è 65.535 byte. *Identification (identificazione)* serve all'host di destinazione per determinare a quale datagramma appartiene il frammento appena arrivato. Dopo c'è un bit inutilizzato, che può essere utile per rilevare il traffico dannoso. Seguono poi due campi di 1 bit per la frammentazione: *DF “don’t fragment”* rappresenta un ordine che impone ai router di non dividere in frammenti il datagramma; *MF “more fragments”* è impostato a 1 per tutti i frammenti tranne l'ultimo, per comunicare quando sono arrivati tutti i frammenti. *Fragment offset* indica la posizione del frammento nel datagramma corrente. Tutti i frammenti tranne l'ultimo di un datagramma devono essere multipli di 8 byte. Il campo *Time to live* è un contatore utilizzato per limitare la vita di un pacchetto. Il campo *Protocol* indica quale processo di trasporto è in attesa di quei dati. Dato che l'intestazione trasporta informazioni vitali come gli indirizzi, *Header Checksum* lo protegge.

FRAMMENTAZIONE → un pacchetto è arrivato con un M bit a 0 → significa che non ci sono più frammenti, è l'ultimo  
 ↓  
 può un frammento essere riconfigurato? ↓  
 ↓  
 si M bit = 1 significa che c'è almeno un altro frammento  
 ↓  
 M bit a 1 e fragmentation offset a 0 significa che è il primo frammento  
 ↓  
 tuttavia non sappiamo se il pacchetto originale è stato frammentato o no  
 ↓  
 un pacchetto non frammentato è considerato l'ultimo frammento

Il campo **Options (opzioni)** contiene opzioni di lunghezza variabile. Ognuna inizia con un codice di 1 byte che la identifica; alcune sono seguite da un campo **Option lenght** di 1 byte che precede uno o più byte di dati. Il campo Opzioni è riempito con multipli di quattro byte. In origine erano definite cinque opzioni:

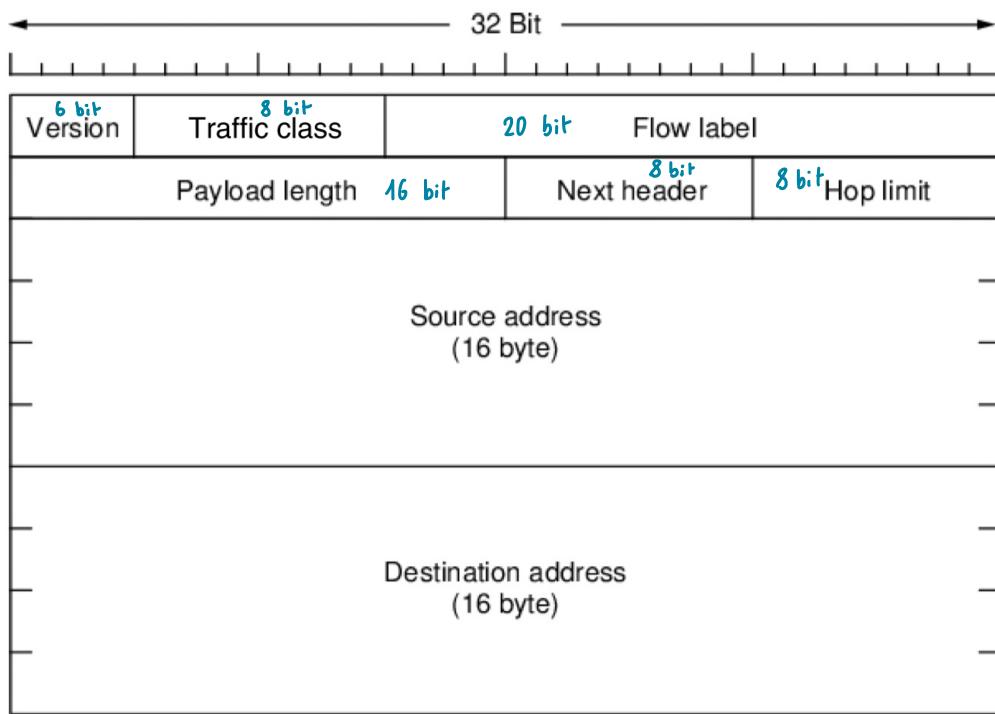
Opzione	Descrizione
Security	Specifica il livello di segretezza del datagramma
Strict source routing	Definisce il percorso completo da seguire
Loose source routing	Elenco dei router che devono fare parte del percorso
Record route	Fa sì che ogni router aggiunga il proprio indirizzo IP
Timestamp	Fa sì che ogni router aggiunga indirizzo e ora



## PROTOCOLLO IPv6

Molto presto gli indirizzi a disposizione di IPv4 iniziarono a scarseggiare, fino a terminare. Per risolvere il problema, l'unica soluzione a lungo termine fu quella di adottare indirizzi più grandi; **IPv6** è un progetto che usa indirizzi a 128 bit. È un protocollo a livello di rete differente, che non interagisce con IPv4 nonostante le similarità.

L'**intestazione** di IPv6 differisce innanzitutto da IPv4 per il campo *Version*: è sempre di 6 bit per IPv6 (e 4 per IPv4). I router sono in grado di esaminare questo campo per determinare il tipo di pacchetto.



Il campo *Traffic class* è usato per distinguere le classi di servizio dei pacchetti con differenti richieste di consegna in tempo reale.

Il campo *Flow label* (*etichetta di flusso*) consente a una sorgente e a una destinazione di marcare un gruppo di pacchetti che, avendo gli stessi requisiti, devono essere trattati allo stesso modo, formando una pseudoconnessione. Per esempio un flusso di pacchetti potrebbe avere rigorosi requisiti di ritardo e quindi aver bisogno di una banda riservata; il flusso può essere impostato in anticipo e gli si può assegnare un identificatore: quando appare un pacchetto con *Flow label* diverso da zero, tutti i router cercano l'etichetta nelle loro tabelle interne per scoprire il tipo di trattamento speciale richiesto dal pacchetto. Ai fini della qualità del servizio, ogni flusso è rappresentato dall'indirizzo sorgente, dall'indirizzo di destinazione e dal numero di flusso, perciò tra una data coppia di indirizzi IP possono essere attivi anche  $2^{20}$  flussi.

Il campo *Payload lenght* (*lunghezza del payload*) indica il numero di byte che segue l'intestazione di 40 byte. È stato abbandonato il nome del campo *Total lenght* di IPv4 perché il significato è leggermente cambiato: i 40 byte di intestazione non sono più conteggiati nel calcolo della lunghezza. Questo cambiamento implica che il payload possa essere ora di 65.535 byte invece che di 65.515 byte.

Il motivo per cui è stato possibile semplificare l'intestazione è che ci possono essere altre intestazioni estese (opzionali). Il campo *Next header (intestazione successiva)* indica quale delle sei (per il momento) intestazioni estese, se presente, segue l'intestazione corrente. Il campo *Hop limit (limite di hop)* è utilizzato per impedire ai pacchetti di vivere per sempre. In pratica funziona come il campo *Time to live* di IPv4, vale a dire che è decrementato a ogni salto del pacchetto. Gli ultimi campi rappresentano l'indirizzo sorgente e di destinazione, a 16 byte di lunghezza fissa.

È istruttivo confrontare l'intestazione IPv4 con l'intestazione IPv6 per vedere che cosa è stato eliminato in IPv6. Il campo *IHL* è stato tolto, perché l'intestazione IPv6 ha una lunghezza fissa. Il campo *Protocol* è stato tolto, perché il campo *Next header* indica che cosa c'è dopo l'ultima intestazione IP (per esempio un segmento UP o TCP).

Tutti i campi che riguardano la frammentazione sono stati rimossi, perché IPv6 gestisce la frammentazione in modo diverso. Quando un host invia un pacchetto IPv6 che è troppo grande, il router che non è in grado di inoltrarlo restituisce un messaggio di errore invece di frammentarlo. Questo messaggio indica all'host di suddividere tutti pacchetti futuri diretti a quella destinazione. Fare in modo che l'host invii pacchetti della dimensione corretta è in definitiva più efficiente che costringere i router a frammentare i pacchetti al volo.

Infine il campo *Checksum* è stato eliminato, perché l'elaborazione di questa informazione riduce enormemente le prestazioni.

Per scrivere gli **indirizzi a 16 byte** è stata scelta una nuova notazione. Gli indirizzi sono scritti come otto gruppi di quattro cifre esadecimale separate da due punti:

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

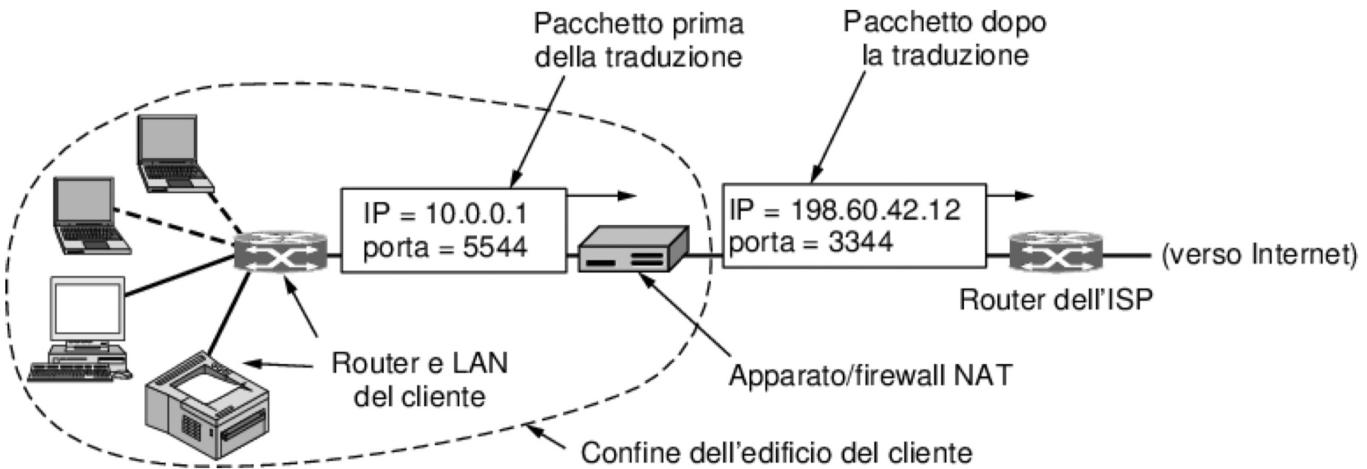
per accontentare i fruttori del  
connection-oriented  
↑

Importante da ricordare: servivano indirizzi più grandi (e più veloci) quindi niente Checksum, frammentazione diversa + Flow label.

### NAT - NETWORK ADDRESS TRANSLATION

La questione dell'esaurimento degli indirizzi IP non è un problema teorico che potrebbe presentarsi in un lontano futuro: sta accadendo proprio qui e ora. La soluzione a lungo termine è che tutta Internet passi a IPv6. Questa transizione sta procedendo lentamente e ci vorranno anni prima che si completi. Di conseguenza era necessario trovare una soluzione rapida attuabile in tempi brevi. La soluzione adottata si chiama **NAT (network address translation)**. L'idea di base di NAT è assegnare a ogni azienda o casa un singolo indirizzo IP (o, al massimo, un piccolo numero di indirizzi) per il traffico di Internet. Dentro la rete del cliente, ogni computer riceve un indirizzo IP unico, utilizzato per instradare il traffico interno alla rete locale. Tuttavia quando un pacchetto sta per lasciare la rete locale per dirigersi verso l'ISP viene eseguita una traduzione di indirizzo dall'unico indirizzo IP interno a quello pubblico condiviso.

(utilizza i 3 intervalli di indirizzi privati)



Dentro i locali dell'azienda, ogni macchina ha un unico indirizzo espresso nella forma 10.x.y.z, ma quando un pacchetto lascia la rete locale, passa attraverso un apparato NAT che converte l'indirizzo IP interno (10.0.0.1 nella figura) nel vero indirizzo IP assegnato all'azienda (198.60.42.12 in questo esempio). L'apparato NAT è spesso abbinato a un firewall, inserito all'interno di un singolo apparecchio che protegge la rete locale, controllando attentamente tutti i dati che entrano e che escono dalla LAN.

### In che modo l'apparato NAT sceglie l'indirizzo di destinazione interno?

I progettisti di NAT hanno osservato che la maggior parte dei pacchetti IP trasporta al suo interno TCP o UDP; entrambi i protocolli hanno intestazioni che contengono una porta sorgente e una porta di destinazione. Le **porte** sono numeri interi lunghi 16 bit che indicano dove inizia e finisce la connessione TCP e offrono il campo che permette di far funzionare NAT. Quando un processo desidera stabilire una connessione TCP con un processo remoto, si lega a una porta TCP inutilizzata sulla sua macchina; questa porta è chiamata **porta sorgente** e indica al codice di TCP dove devono essere inviati i pacchetti in arrivo appartenenti alla connessione. Il processo fornisce anche una **porta di destinazione** che indica chi deve ricevere i pacchetti sulla parte remota. Le porte da 0 a 1023 sono riservate a servizi noti; per esempio, la porta 80 è utilizzata dai server Web, e ciò consente a client remoti di individuare il servizio.

Ogni volta che un pacchetto diretto verso l'esterno raggiunge l'apparato NAT, l'indirizzo sorgente 10.x.y.z è sostituito dall'indirizzo IP dell'azienda. Inoltre, il campo *Source port* è sostituito da un indice che punta alla tabella di traduzione da 65.536 voci dell'apparato NAT. Questa voce della tabella contiene l'indirizzo IP originale e la porta sorgente originale. Infine i codici di controllo delle intestazioni IP e TCP sono ricalcolati e inseriti nel pacchetto. È necessario sostituire il campo *Source port* perché, per esempio, due connessioni distinte dalle macchine 10.0.0.1 e 10.0.0.2 possono entrambe utilizzare la porta 5000; perciò la porta sorgente non basta, da sola, a identificare il processo che sta trasmettendo.

Quando un pacchetto trasmesso dall'ISP raggiunge l'apparato NAT, la *Source port* nell'intestazione TCP viene estratta e utilizzata come indice nella tabella di associazione dell'apparato. Dalla voce individuata, l'apparato estrapola l'indirizzo IP interno e la *Source port* TCP originale, quindi inserisce entrambe le informazioni nel pacchetto; poi ricalcola e inserisce nel pacchetto sia il checksum IP sia quello TCP. Alla fine il pacchetto viene passato al router interno per il normale inoltro basato sull'indirizzo 10.x.y.z.

NAT TABLE →  
(ESEMPIO)

Original Source	Original Port	Translated Source	Translated Source Port	Destination	Destination Port
		IP of NAT device	It works as a label to identify the original pair (IP, port)	Unaltered	Unaltered
192.168.0.1 (PRIVATO)	1400 ←	137.204.57.12 LA RISPOSTA ARRIVA QUI	2405 "ETICHETTA"	173.194.40.81 (PUBBLICO)	80 (PUBBLICO)
192.168.0.2	1653	137.204.57.12	2406 RIESCE A INSTRADARE ANCHE CON STESSO INDIRIZZO	173.194.40.81	80 PORTA DEL SERVER WEB
192.168.0.5	1653	137.204.57.12	2407 ↓ PORTE ESSENZIALI!	173.194.40.81	80

NAT è molto usato nella pratica, specialmente per reti domestiche o in piccole aziende, in quanto è l'unica tecnica che riesca a gestire il problema della mancanza di indirizzi IP. È diventato una cosa unica con firewall e privacy, perché di norma blocca i pacchetti in arrivo non richiesti. Per questa ragione è inverosimile che sparisca, anche qualora IPv6 fosse ampiamente adottato.

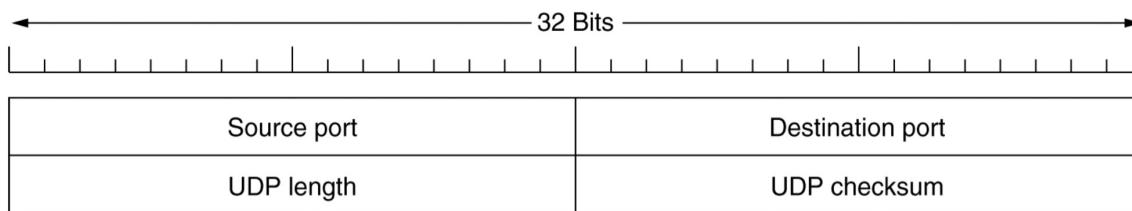
LIVELLO 4 - TRANSPORT LAYER

Il livello di rete provvede alla consegna di pacchetti end-to-end usando datagrammi o circuiti virtuali. Il **livello di trasporto** si basa sul livello di rete per fornire il trasporto dei dati da un processo su una macchina sorgente a un processo su una macchina di destinazione con un livello di affidabilità desiderato e indipendente dalle reti fisiche attualmente in uso.

Il livello di trasporto Internet possiede due protocolli principali: un protocollo non orientato alla connessione e uno orientato alla connessione. I protocolli si completano l'un l'altro. Il protocollo non orientato alla connessione è **UDP**; non fa praticamente altro che spedire pacchetti tra le applicazioni, permettendo loro di costruirsi sopra i propri protocolli come necessario. Il protocollo orientato alla connessione è **TCP**. Fa praticamente tutto: crea connessioni e aumenta l'affidabilità con le ritrasmissioni, implementa anche il controllo di fusso e il controllo di congestione, tutto per conto delle applicazioni che lo usano.

UDP (USER DATAGRAM PROTOCOL)

UDP trasmette **segmenti** costituiti da un'intestazione di 8 byte seguita dal payload. L'**intestazione** è composta da due **porte** che servono per identificare gli endpoint all'interno dei computer di sorgente e destinazione. Quando arriva un pacchetto UDP, il suo payload è consegnato al processo associato alla porta di destinazione. I campi delle porte sorgente e destinazione sono il vantaggio principale dell'utilizzo di UDP rispetto a IP, perché consentono la consegna del segmento all'interno del frame alla corretta applicazione.



La porta sorgente serve principalmente quando si deve inviare una risposta al mittente. Copiando dal segmento in ingresso il campo *Source port* nel campo *Destination port* del segmento in uscita, il processo che invia la risposta può specificare il processo sulla macchina sorgente che deve riceverla. Il campo *UDP length* (lunghezza UDP) include l'intestazione di 8 byte e tutti i dati (lunghezza massima 65.515 byte).

È fornito anche un campo *Checksum* opzionale per migliorare l'affidabilità. Verifica la somma dell'intestazione, dei dati e di una pseudointestazione concettuale IP . Quando viene fatta questa somma, il campo Checksum è impostato a zero e il campo dati viene prolungato con un ulteriore byte a zero se il numero di byte della sua lunghezza è dispari. L'algoritmo per il calcolo del checksum prevede semplicemente di sommare tutti i gruppi di 16 bit in complemento a uno e prenderne il risultato. Di conseguenza, quando un destinatario fail calcolo sull'intero segmento includendo il campo Checksum, il risultato dovrebbe essere 0. Se il checksum non è calcolato, è salvato come uno 0. Tuttavia non calcolarlo è stupido, a meno che la qualità dei dati sia poco importante (come per la voce digitalizzata).

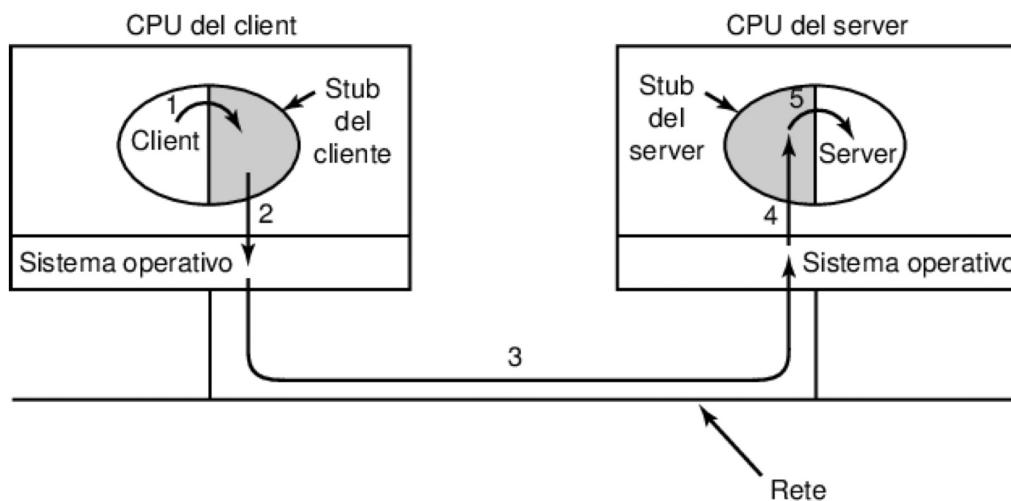
La **pseudointestazione per il caso di IPv4** contiene l'indirizzo IPv4 di 32 bit delle macchine sorgente e destinazione, il numero di protocollo per UDP (17) e il numero di byte del segmento UDP (inclusa l'intestazione). È diverso ma analogo a IPv6. Anche TCP usa la stessa pseudointestazione per il suo checksum.



Il protocollo UDP non si occupa del controllo di flusso, del controllo di congestione o della ritrasmissione dopo la ricezione di un segmento errato; questi compiti sono lasciati ai processi utente. UDP si occupa invece di fornire un'interfaccia al protocollo IP, con la caratteristica aggiunta del demultiplexing di più processi utilizzando le porte e opzionalmente del rilevamento di errori end-to-end; questo è tutto ciò che fa.

### RPC (REMOTE PROCEDURE CALL)

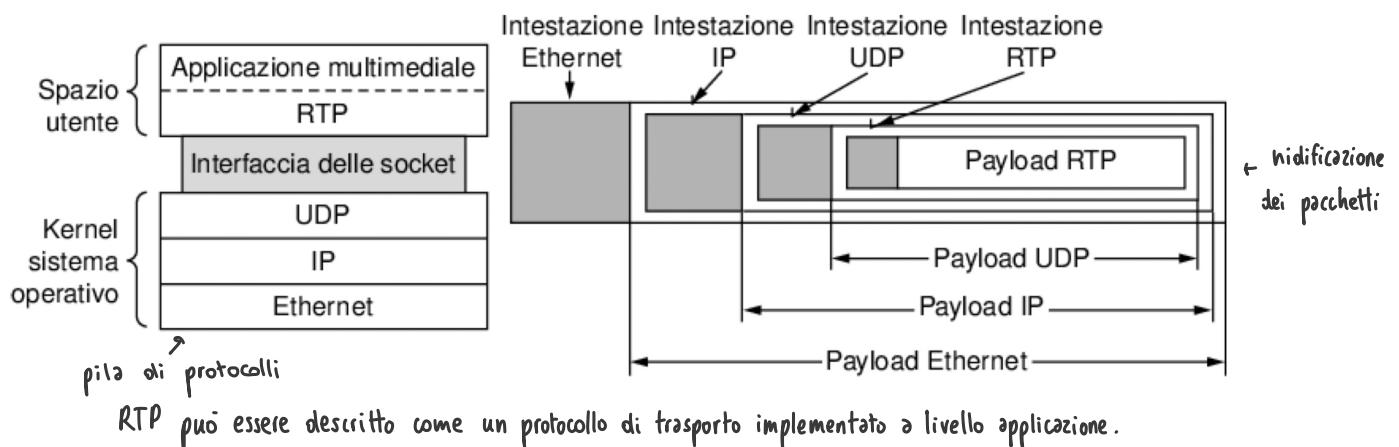
Quando un processo sulla macchina 1 chiama una procedura sulla macchina 2, il processo chiamante su 1 viene sospeso e l'esecuzione della procedura chiamata avviene su 2. Le informazioni possono essere trasportate dal chiamante al chiamato nei parametri e possono tornare indietro nel risultato della procedura. Il passaggio dei messaggi non è visibile al programmatore. Questa tecnica è nota come **RPC** (*remote procedure call, chiamata a procedura remota*) ed è diventata la base per molte applicazioni di rete. Tradizionalmente la procedura chiamante è nota come **client**, mentre la procedura chiamata è nota come **server**; L'idea alla base di RPC è eseguire una chiamata a procedura remota in maniera più simile possibile a una locale. Nella forma più semplice, per chiamare una procedura remota, il programma client deve essere associato a una piccola procedura di libreria, chiamata **client stub**, che rappresenta la procedura del server nello spazio di indirizzamento del client. Allo stesso modo il server è associato a una procedura chiamata **server stub**. Queste procedure nascondono il fatto che la chiamata a procedura dal client al server non sia locale.



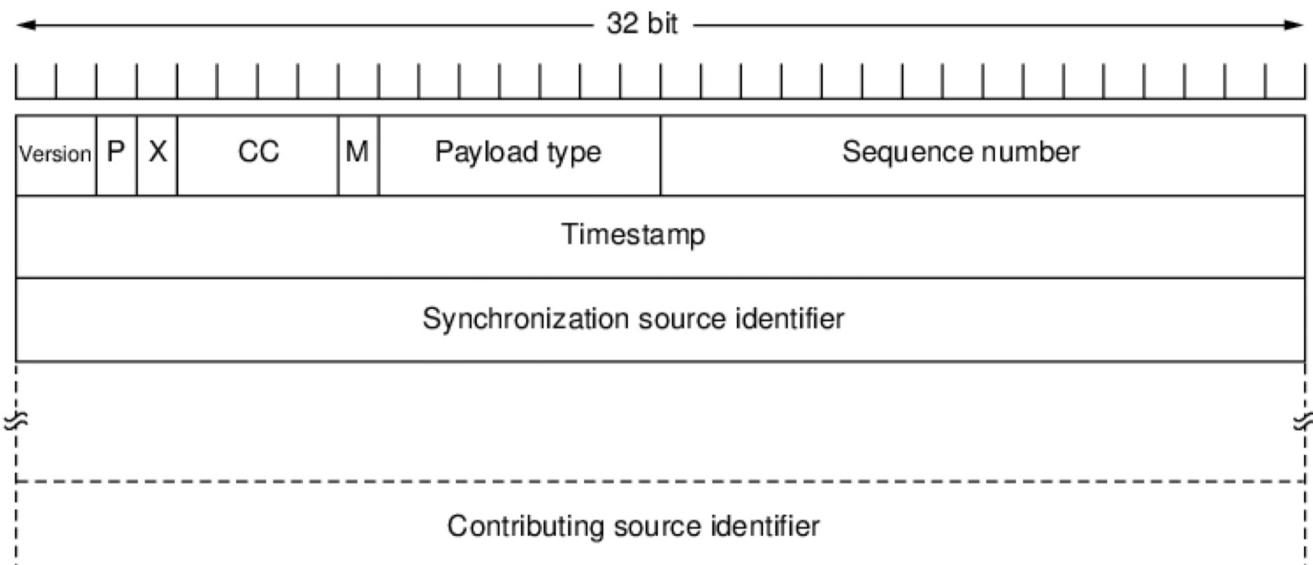
Il passaggio 1 rappresenta il client che chiama il suo stub. Questa è una chiamata a procedura locale, con i parametri inseriti nello stack secondo il metodo usuale. Il passaggio 2 rappresenta il client stub che inserisce i parametri in un messaggio ed effettua una chiamata di sistema per inviare il messaggio. L'inserimento dei parametri nel pacchetto prende il nome di marshaling. Il passaggio 3 mostra il kernel che invia il messaggio dalla macchina client alla macchina server. Il passaggio 4 rappresenta il kernel che passa il pacchetto in ingresso al server stub. Per finire, il passaggio 5 mostra il server stub che chiama la procedura del server con i parametri decodificati. La risposta segue lo stesso percorso nell'altra direzione.

### RTP (REAL TIME TRANSPORT PROTOCOL)

RTP viene normalmente eseguito nello spazio utente appoggiandosi su UDP (nel sistema operativo). Un'applicazione multimediale è composta di più flussi audio, video, di testo e forse di altro tipo. Questi flussi vengono passati alla libreria RTP che si trova nello spazio utente insieme all'applicazione; la libreria segue il multiplexing dei fusti e li codifica in pacchetti RTP, che vengono poi inseriti in una socket. Nella parte di kernel che si occupa della socket vengono generati pacchetti UDP che incapsulano quelli RTP, che vengono poi passati al livello IP per essere trasmessi. Il processo inverso ha luogo al ricevente e l'applicazione multimediale riceve i dati dalla libreria RTP e si occupa di rappresentarli.



L'intestazione RTP consiste di tre gruppi di 32 bit e di alcune estensioni facoltative.



Il bit P indica che il pacchetto è stato riempito fino a ottenere un multiplo di 4 byte. Il bit X indica che è presente un'intestazione estesa. Il campo CC indica il numero di sorgenti attive presenti, da 0 a 15. Il bit M è un bit di contrassegno specifico dell'applicazione. Il campo *Payload type* indica quale algoritmo di codifica è stato utilizzato (per esempio audio a 8 bit non compresso, MP3 e così via); dal momento che ogni pacchetto contiene questo campo, la codifica può cambiare durante la trasmissione. *Sequence number* è un contatore incrementato a ogni invio di un pacchetto RTP; serve a rilevare i pacchetti persi. *Timestamp* è prodotto da chi genera il flusso per indicare quando è stato creato il primo campione nel pacchetto.

Il campo *Synchronization source identifier* specifica a quale flusso appartiene il pacchetto e rappresenta l'informazione utilizzata per il multiplexing e il demultiplexing di più flussi dati in un singolo fusso di pacchetti UDP. Per finire, i campi facoltativi *Contributing source identifier* sono utilizzati da apparecchiature di studio quali i mixer ad esempio.

### RTP (REAL TIME TRANSPORT CONTROL PROTOCOL)

RTP si accompagna a un protocollo chiamato **RTCP**, che gestisce le retroazioni verso la sorgente, la sincronizzazione e l'interfaccia utente. Non trasporta campioni multimediali. Tra le diverse funzioni, viene utilizzato per fornire alla sorgente di flusso un feedback su ritardi, jitter, larghezza di banda, congestione e altre proprietà di rete. Fornendo un feedback continuo, gli algoritmi di codifica si possono adattare costantemente per fornire la migliore qualità. RTCP gestisce anche la sincronizzazione tra flussi, e fornisce un metodo per denominare le sorgenti (per esempio con testo ASCII).

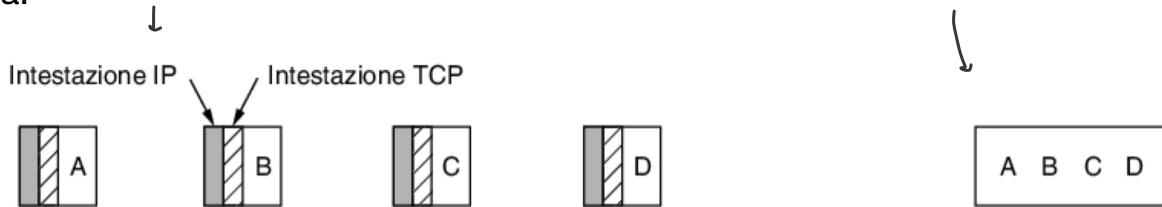
### TCP (TRANSMISSION CONTROL PROTOCOL)

TCP è stato progettato appositamente per fornire un flusso di byte affidabile end-to-end su una internetwork affidabile. È molto più complesso dell'UDP in quanto deve offrire un servizio orientato alla connessione affidabile, a partire dal servizio datagram offerto da IP. I dati sono visti come un flusso di byte, anche se sono organizzati in pacchetti chiamati **segmenti**.

Il servizio TCP è ottenuto con la creazione di punti terminali di un sistema di comunicazione da parte di mittente e ricevente, chiamati **socket**. Ogni socket possiede un numero (indirizzo) composto dall'indirizzo IP dell'host e da un numero di 16 bit locale all'host, chiamato **porta**. Per ottenere il servizio TCP si deve stabilire esplicitamente una connessione tra una socket su una macchina e una socket su un'altra macchina. Una socket può essere usata per più connessioni contemporaneamente. I numeri di porta minori di 1024 sono riservati per servizi standard, e prendono il nome di **well-known port**.

Porta	Protocollo	Utilizzo
20, 21	FTP	Trasferimento di file
22	SSH	Login remoto, rimpiazzamento di Telnet
25	SMTP	Posta elettronica
80	HTTP	World Wide Web
110	POP-3	Accesso remoto alla posta elettronica
143	IMAP	Accesso remoto alla posta elettronica
443	HTTPS	Web sicuro (HTTP su SSL/TLS)
543	RTSP	Controllo di riproduttori multimediali
631	IPP	Condivisione di stampanti

Tutte le connessioni TCP sono di tipo full-duplex punto a punto. Full-duplex indica che il traffico può procedere in entrambe le direzioni contemporaneamente. Punto a punto significa che ogni connessione ha esattamente due punti terminali. TCP non supporta il multicast o il broadcast. Una connessione TCP è un **flusso di byte**, non un flusso di messaggi. I confini dei messaggi non vengono conservati da un'estremità all'altra della connessione. Per esempio, se il processo mittente segue quattro invii da 512 byte su un flusso TCP, questi dati possono essere consegnati al processo ricevente come quattro blocchi di 512 byte, due blocchi di 1024 byte, un solo blocco di 2048 byte o in altri modi ancora.



TCP → consegna ordinata dei dati, ritrasmissione di quelli persi

controllo di flusso → non vengono spediti più dati di quanti ne può ricevere la sorgente in un dato momento

controllo di congestione → si cerca di non spedire più dati di quanti ne può smaltire la rete → DECENTRALIZZATO a retroazione

ogni istanza del TCP decide per conto suo

Nel TCP i dati inviati (in byte), organizzati in segmenti, devono essere confermati da un **ACK (acknowledgment)**: il TCP è bidirezionale, e gli ACK possono sfruttare i segmenti dati della connessione di ritorno. Funzionamento: ogni segmento inviato è contraddistinto da un numero di sequenza che indica il primo byte del payload; ogni ACK indica quale è il primo byte che il ricevitore si aspetta di ricevere e implicitamente conferma tutti i byte precedenti (ACK cumulativo); se un segmento è perso il ricevitore all'arrivo dei segmenti successivi invia degli ACK duplicati (DupACK). Ogni segmento ricevuto genera l'invio di un ACK (esclusa delayed ACK per i quali si ha un ACK ogni due); il tempo che intercorre fra l'invio di un segmento e l'arrivo dell'ACK da lui causato si chiama *RTT (Round Trip Time)*. È il tempo dell'anello di retroazione del TCP. Più è grande, peggio funziona il TCP.

IP → connection-less → unreliable

TCP → connection-oriented → affidabile → MECCANISMO A RETROAZIONE

segmento → destinatario

←→ acknowledgment → CUMULATIVO → può succedere che ne mandi 1 ogni 2 → delayed ACK  
20 byte

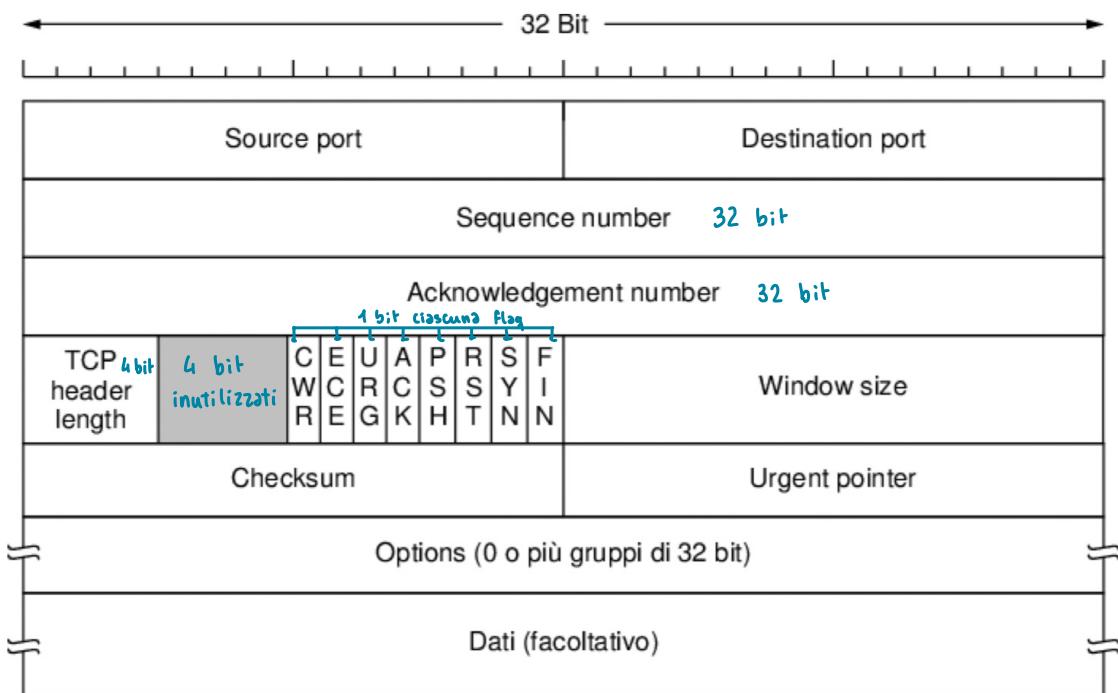
## IL PROTOCOLLO TCP

Una funzionalità vitale di TCP, che guida la struttura del protocollo, consiste nel fatto che ogni byte in una connessione TCP ha un proprio numero di sequenza a 32 bit. Le entità TCP di invio e ricezione scambiano i dati sotto forma di segmenti. Un **segmento TCP** consiste di un'intestazione fissa di 20 byte (più una parte facoltativa) seguita da zero o più byte di dati. Il software TCP decide la dimensione dei segmenti e può accumulare in un segmento i dati provenienti da più invii oppure divider i dati di un invio in più segmenti.

Il protocollo di base utilizzato dalle entità TCP è il protocollo a finestra scorrevole con una dimensione dinamica della finestra. Quando un mittente trasmette un segmento, avvia anche un timer. Quando il segmento arriva a destinazione, l'entità TCP ricevente invia un segmento (con i dati, se esistono, oppure senza) contrassegnato da un numero di acknowledgement uguale al numero di sequenza successivo che prevede di ricevere la dimensione della finestra disponibile. Se il timer del mittente scade prima della ricezione dell'acknowledgement, questi ritrasmette il segmento.

### INTESTAZIONE DEL SEGMENTO TCP

Ogni segmento inizia con un'intestazione di 20 byte con formato fisso; l'intestazione fissa può essere seguita da alcune opzioni dell'intestazione. Dopo le opzioni seguono fino a 65.495 byte di dati, dove i primi 20 fanno riferimento all'intestazione IP e i secondi a quella TCP. Sono ammessi segmenti senza dati, ampiamente usati per acknowledgment e messaggi di controllo.



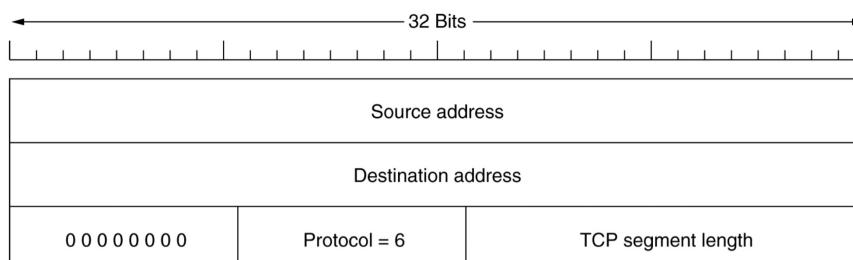
I campi *Source port* e *Destination port* identificano gli estremi locali della connessione. Una porta TCP più l'indirizzo IP del suo host formano un unico punto terminale a 48 bit. I punti terminali di sorgente e destinazione insieme identificano la connessione. I campi *Sequence number* e *Acknowledgement number* svolgono le loro solite funzioni. Occorre notare che il secondo specifica il successivo byte previsto, non l'ultimo byte ricevuto correttamente; si tratta di un **acknowledgement cumulativo**, perché riassume i dati ricevuti con un solo numero e non va al di là dei dati persi.

Il campo *TCP header length* indica quanti gruppi di 32 bit sono contenuti nell'intestazione TCP. L'informazione è necessaria perché il campo *Options* ha una lunghezza variabile e, di conseguenza, anche l'intestazione. Segue un campo di 4 bit inutilizzati.

Seguono otto flag di un bit:

- **CWR** e **ECE**: usati per segnalare la congestione quando viene utilizzata **ECN** (*explicit congestion notification*); ECE è regolata per mandare un ECN-Echo a un mittente TCP per indicargli di rallentare quando il destinatario TCP riceve un'indicazione di congestione dalla rete. CWR è usato per segnalare una condizione di congestion window reduced dal mittente TCP al destinatario TCP così che esso sappia che il mittente ha rallentato e possa smettere di spedire degli ECN-Echo.
- **URG**: è impostato a 1 quando si usa Urgent pointer (puntatore urgente), che indica lo spiazzamento in byte (partendo dal numero di sequenza corrente) in cui si trovano i "dati urgenti."
- **ACK**: è impostato a 1 per indicare che Acknowledgement number è valido e questo è il caso della quasi totalità dei pacchetti. Se ACK è 0, il segmento non contiene un acknowledgement, pertanto il campo Acknowledgement number viene ignorato.
- **PSH**: segnala la presenza di dati PUSH. Al ricevente viene gentilmente chiesto di consegnare i dati all'applicazione all'arrivo e di non archiviarli nel buffer per poi trasmetterle un buffer completo.
- **RST**: viene utilizzato per reimpostare una connessione che è diventata confusa a causa di un malfunzionamento dell'host o per altre ragioni.
- **SYN**: viene utilizzato per stabilire le connessioni. La richiesta di connessione presenta SYN = 1 e ACK = 0 per indicare che il campo Acknowledgement non è utilizzato. La risposta alla richiesta di connessione porta un acknowledgement, pertanto possiede SYN = 1 e ACK = 1. In pratica il bit SYN è utilizzato per segnalare entrambi CONNECTION REQUEST e CONNECTION ACCEPTED, mentre il bit ACK distingue tra le due possibilità
- **FIN**: viene utilizzato per rilasciare una connessione. Specifica che il mittente non ha altri dati da trasmettere. Tuttavia, dopo avere chiuso una connessione, il processo di chiusura potrebbe continuare a ricevere dati all'infinito. Entrambi i segmenti SYNe FIN possiedono numeri di sequenza ed è quindi garantito che verranno elaborati nell'ordine corretto.

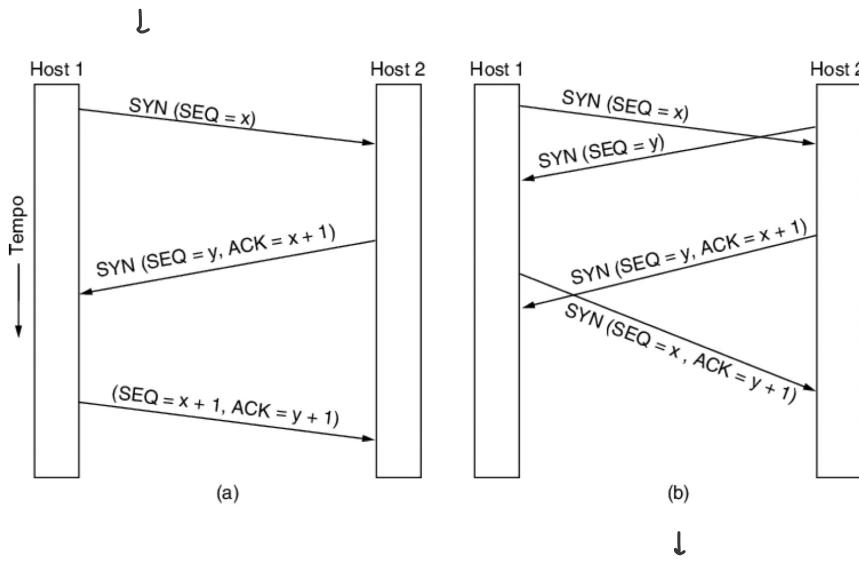
Il controllo di flusso in TCP è gestito con una finestra scorrevole a dimensione variabile. Il campo *Window size* indica quanti byte possono essere inviati a partire da quello che ha ricevuto acknowledgement. In TCP gli acknowledgement e le autorizzazioni per inviare dati aggiuntivi sono completamente separati. Questa divisione (di fatto una finestra a dimensione variabile) offre una flessibilità aggiuntiva. Per aumentare l'affidabilità viene anche fornito un *Checksum*; questo fa un controllo dell'intestazione, dei dati e di una pseudointestazione concettuale esattamente come UDP, tranne che la pseudointestazione ha un numero di protocollo per TCP (6) e il checksum è obbligatorio. Il campo *Options* fornisce un modo per aggiungere funzionalità aggiuntive non previste dall'intestazione standard.



→ PSEUDO INTESAZIONE inclusa nel Checksum  
funziona esattamente come quella di UDP  
ma con Protocol = 6

### INSTAUARZIONE DELLA CONNESSIONE TCP

Per stabilire una connessione, un lato (per esempio il server) attende in modo passivo una connessione in ingresso eseguendo nell'ordine le primitive LISTEN e ACCEPT che possono indicare una sorgente specifica oppure nessuna in particolare. L'altro lato (diciamo il client) segue una primitiva CONNECT, specificando l'indirizzo IP e la porta a cui vole connettersi, la dimensione massima del segmento TCP che è intenzionato ad accettare e, facoltativamente, alcuni dati utente. La primitiva CONNECT invia un segmento TCP con il bit SYN a 1 e il bit ACK a 0, poi attende una risposta. Quando questo segmento arriva a destinazione, l'entità TCP controlla se esiste un processo che ha eseguito una LISTEN sulla porta indicata nel campo Destination port e in caso negativo invia una risposta con il bit RST a 1 per rifiutare la connessione. Se un processo è in ascolto sulla porta, gli viene dato il segmento TCP in ingresso; quindi può accettare o rifiutare la connessione. Se accetta, viene restituito al mittente un segmento di acknowledgement. Si noti che un segmento SYN consuma un byte nello spazio delle sequenze, così da poter ricevere un acknowledgement non ambiguo.



Nel caso che due host tentino contemporaneamente di stabilire una connessione tra le stesse due socket, il risultato di questi eventi è la costituzione di una sola connessione, non due, perché le connessioni sono identificate dai loro punti terminali. Se sia la prima attivazione che la seconda generano una connessione identificata da  $(x, y)$ , viene comunque creata una sola voce nella tabella per  $(x, y)$ . Si ricordi che il numero di sequenza iniziale, scelto da ciascun host, dovrebbe variare lentamente invece di essere una costante, come 0. Questa regola serve a proteggere dai pacchetti in ritardo e duplicati.

### MODELLO DI GESTIONE DELLA CONNESSIONE TCP

I passi richiesti per stabilire e rilasciare le connessioni possono essere rappresentati in una macchina a stati finiti con gli 11 stati elencati:

Stato	Descrizione
CLOSED	Nessuna connessione è attiva o in sospeso
LISTEN	Il server è in attesa di una chiamata in ingresso
SYN RCV	È arrivata una richiesta di connessione; in attesa di ACK
SYN SENT	L'applicazione ha iniziato ad aprire una connessione
ESTABLISHED	Il normale stato di trasferimento dei dati
FIN WAIT 1	L'applicazione ha detto di aver terminato
FIN WAIT 2	L'altro lato ha accettato il rilascio
TIME WAIT	Attende la scadenza di tutti i pacchetti
CLOSING	Entrambi i lati hanno cercato di chiudere contemporaneamente
CLOSE WAIT	L'altro lato ha iniziato il rilascio
LAST ACK	Attende la scadenza di tutti i pacchetti

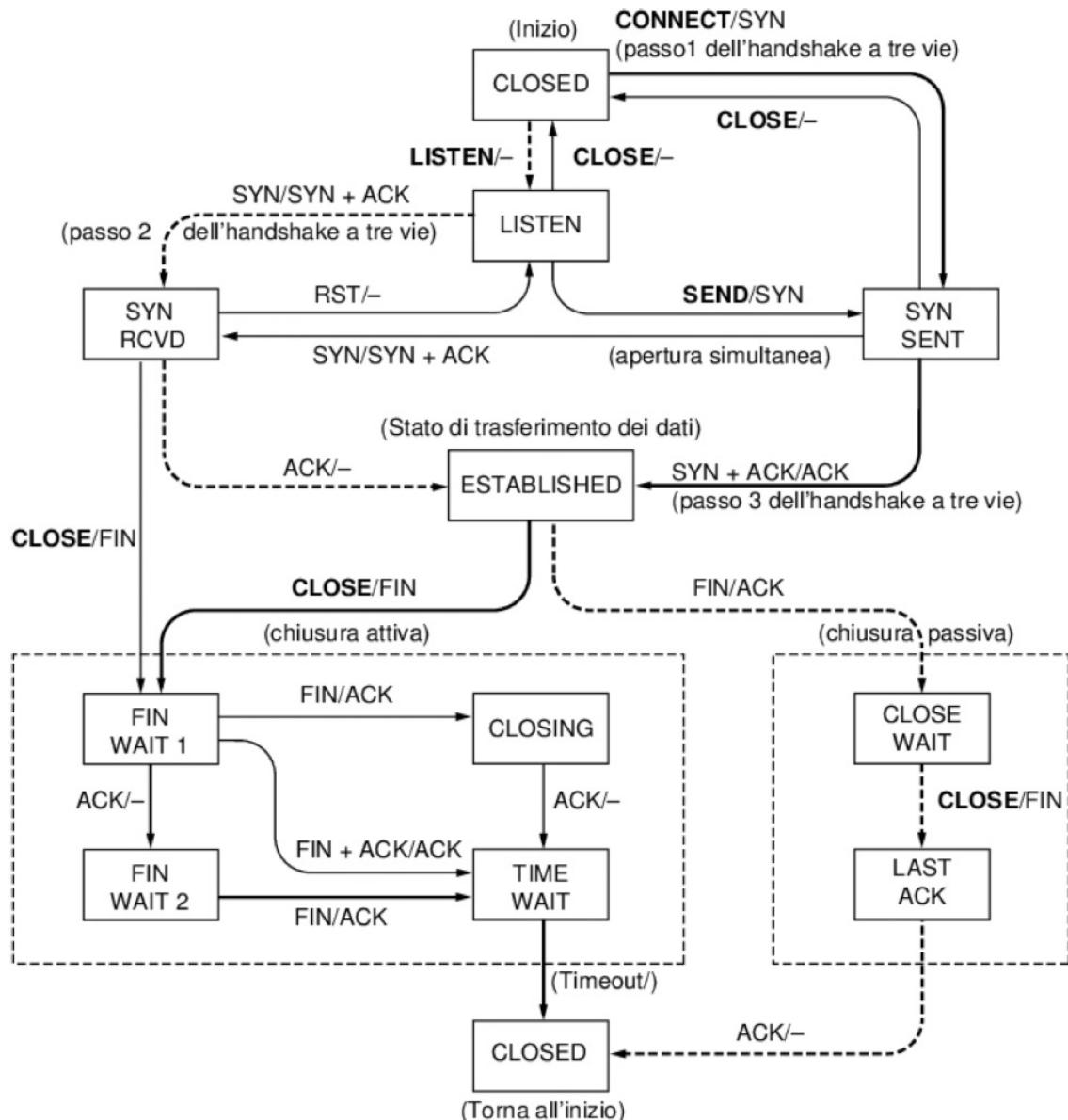
A cosa serve il timer? → TIME-WAIT → fa fronte alla possibile perdita dell'ultimo ACK

↓

Tutti i segmenti nel TCP devono ricevere la conferma, tranne l'acknowledgment (altrimenti loop infinito). Gli ACK non sono ritrasmissibili, quindi se non ricevo l'ACK provo a rimandare un FIN.

Ogni connessione inizia nello stato CLOSED. Lo stato viene lasciato quando la connessione esegue un'apertura passiva (LISTEN) o un'apertura attiva (CONNECT). Se l'altro lato esegue l'operazione opposta, viene stabilita una connessione e lo stato diventa ESTABLISHED. Il rilascio della connessione può essere iniziato da una qualunque delle parti. Una volta completato, lo stato torna a essere CLOSED. Il caso tipico di un client connesso in modo attivo a un server passivo è mostrato con linee spesse (continue per il client, tratteggiate per il server). Le linee sottili rappresentano sequenze di eventi insolite. Ogni linea nella figura è contrassegnata da una coppia evento/azione. L'evento può essere una chiamata di sistema da parte dell'utente (CONNECT, LISTEN, SEND o CLOSE), l'arrivo di un segmento (SYN, FIN, ACK o RST) oppure un timeout pari al doppio del massimo tempo di vita del pacchetto. L'azione è l'invio di un segmento di controllo (SYN, FIN o RST) oppure nulla (indicato con -).

Potrebbe essere più facile comprendere il diagramma seguendo prima il percorso di un client (la linea continua spessa) e poi il percorso di un server (la linea tratteggiata spessa). Quando un programma applicativo sulla macchina client genera una richiesta CONNECT, l'entità TCP locale crea un record per la connessione, la contrassegna nello stato SYN SENT e invia un segmento SYN. Si noti che molte connessioni potrebbero essere aperte (o venire aperte) contemporaneamente da più applicazioni, pertanto lo stato è relativo alla singola connessione ed è registrato nel relativo record. Quando arriva il SYN + ACK, TCP invia l'ACK finale dell'handshake a tre vie e passa nello stato ESTABLISHED. Ora è possibile inviare e ricevere i dati.



Quando un'applicazione ha finito, segue una primitiva CLOSE che provoca l'invio da parte dell'entità TCP locale di un segmento FIN e l'attesa dell'ACK corrispondente (il riquadro tratteggiato indicato come "chiusura attiva"). Quando arriva l'ACK, viene eseguita una transizione allo stato FIN WAIT 2 e una direzione della connessione si chiude. Quando anche l'altro lato esegue la chiusura, viene ricevuto un FIN che richiede l'invio di un acknowledgement. Ora entrambi i lati sono chiusi, ma TCP attende un tempo uguale al doppio del tempo di vita massimo del pacchetto per garantire che tutti i pacchetti della connessione siano scaduti (nel caso in cui l'acknowledgement fosse andato perso). Alla cadenza del timer, TCP elimina il record della connessione.

Esaminiamo ora la gestione della connessione dal punto di vista del server. Il server segue una LISTEN e attende per vedere che cosa succede. Quando al server arriva un SYN, questo manda un acknowledgement e passa allo stato SYN RCV'D; nel momento in cui anche il SYN del server riceve un acknowledgement, l'handshake a tre vie è completo e il server passa allo stato ESTABLISHED. Da questo momento in poi può avvenire il trasferimento dei dati. Quando il client ha terminato di trasmettere i propri dati effettua una CLOSE, che provoca l'invio da parte del client di un FIN al server (il riquadro tratteggiato indicato come "chiusura passiva"). Il server, ricevuto il segnale FIN, segue a sua volta una CLOSE; operazione che prevede l'invio di un FIN al client. All'arrivo dell'acknowledgement relativo a quest'ultimo FIN da parte del client, il server rilascia la connessione e ne elimina il relativo record.

## PROTOCOLLO TCP: CONTROLLO DI FLUSSO

Per il **controllo di flusso** il protocollo TCP utilizza un **protocollo a finestre scorrevoli**. Ciò che il mittente può spedire dipende non solo dagli ack ricevuti ma anche dallo spazio disponibile nel buffer del ricevente. Lo spazio disponibile nel buffer è indicato dal campo *window size* che la stazione di destinazione manda al mittente in ogni segmento in modo che il mittente possa regolare la trasmissione dei segmenti; la dimensione della finestra indica il numero di byte che il destinatario è in grado di ricevere.

Quando la dimensione della finestra è 0 il mittente non può spedire dati; può comunque sempre inviare un segmento con *dati urgenti*, per esempio per far terminare il processo sull'altra stazione; inoltre può spedire un segmento di un byte per fare in modo che l'altra stazione ripeta qual è il prossimo byte atteso e segnali la dimensione della finestra; questo segmento sonda (che prende il nome di **window probe**) viene mandato quando scade un apposito timer per evitare problemi nel caso fosse stata segnalata una finestra di dimensione 0 e poi fosse andato perduto il segmento con l'aggiornamento della dimensione della finestra.

Quando trasmette un segmento il mittente fa partire un timer; se entro lo scadere del timer non riceve la conferma del destinatario ritrasmette il segmento.

La durata del timer è importante perché se è troppo grande ci saranno ritardi ogni volta che viene perso un pacchetto e se è troppo piccola ci saranno molte ritrasmissioni non necessarie. Il timer dovrebbe essere uguale al tempo di andata e ritorno dalla destinazione, ma questo tempo è molto variabile e dipende anche dalle congestioni della rete.

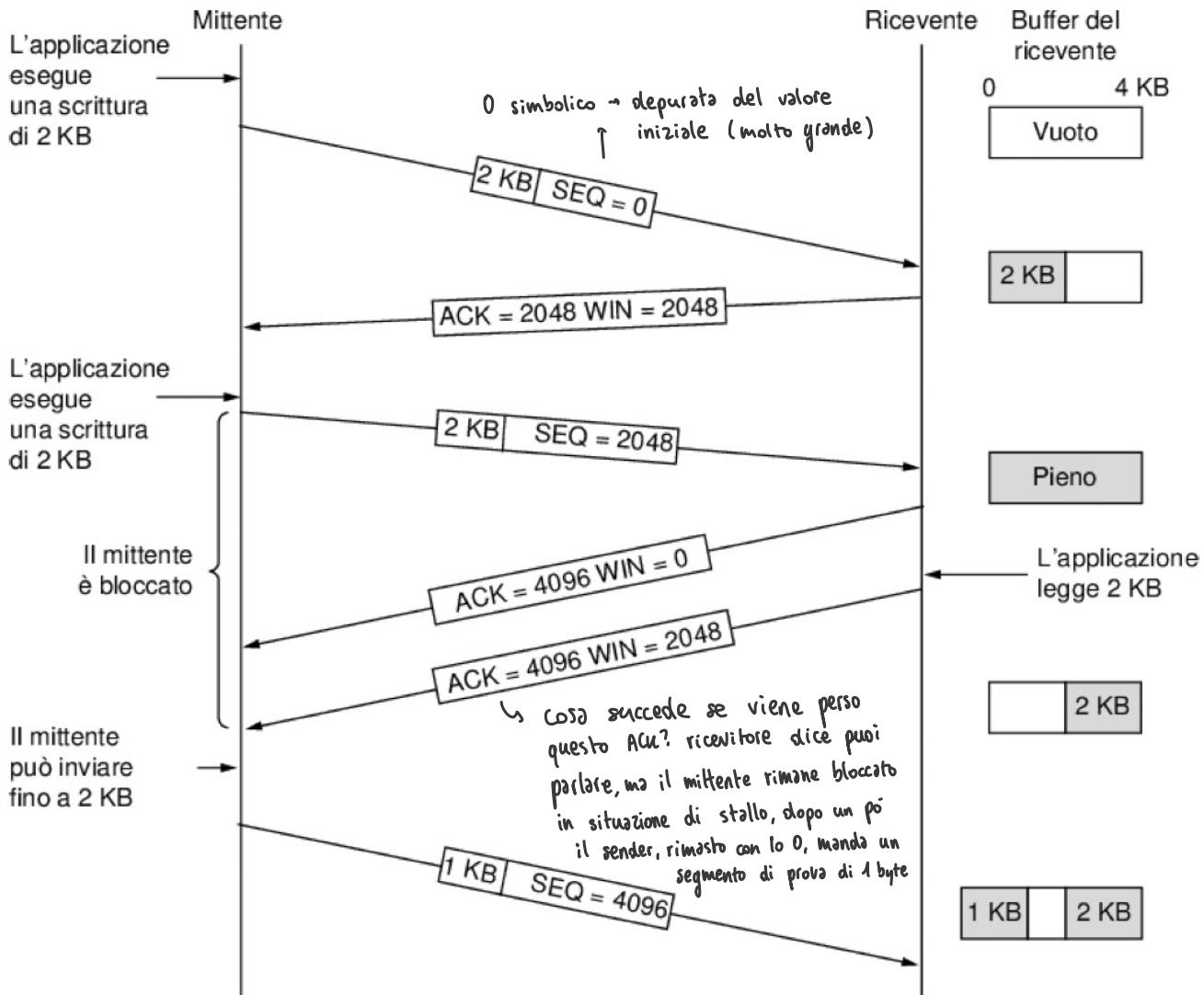
Il timer viene impostato dinamicamente in base a un algoritmo (Jacobson 1988) che misura continuamente le prestazioni della rete; ma se un segmento viene ritrasmesso perché il timer è scaduto la durata del timer viene raddoppiata.

Non è necessario che il mittente trasmetta i dati non appena arrivano dall'applicazione e non è nemmeno richiesto che il ricevente spedisca immediatamente la conferma dei dati ricevuti. Vengono utilizzati dei metodi per rendere più efficiente la trasmissione.

Il destinatario, invece di inviare la conferma per ogni segmento ricevuto, può aspettare di ricevere alcuni segmenti e confermare solo l'ultimo byte ricevuto; per evitare che ciò faccia scadere il timer del mittente anche il destinatario usa un timer, allo scadere del quale deve inviare una conferma al mittente.

Si supponga per esempio che il ricevente abbia un buffer di 4096 byte. Se il mittente trasmette un segmento di 2048 byte che viene correttamente ricevuto, il ricevente darà un acknowledgment al segmento. Tuttavia, dal momento che ora possiede solo 2048 byte di spazio nel buffer (fino a quando l'applicazione non rimuove dei dati), "pubblicizza" una finestra di 2048 byte partendo dal successivo byte atteso. Ora il mittente trasmette altri 2048 byte che ricevono acknowledgment; la finestra pubblicizzata ha però dimensione 0. Il mittente deve fermarsi finché il processo applicativo sull'host ricevente non abbia rimosso dati dal buffer; in quel momento TCP potrà pubblicizzare una finestra più grande e potranno essere spediti altri dati.

Receiver window (rwnd; o advertised window, o anche window)



TCP utilizza più timer (almeno concettualmente) per svolgere il proprio lavoro. Il più importante è quello che fa scattare il **RTO** (retransmission timeout). Quando viene inviato un segmento, si avvia un timer di ritrasmissione. Se il segmento riceve un acknowledgement prima della scadenza del timer, TCP lo ferma; se invece il timer scade prima dell'arrivo dell'acknowledgement, il segmento viene ritrasmesso (e il timer riavviato).

### FINESTRE E VELOCITÀ DI TRASMISSIONE

↓  
W: numero massimo di segmenti che possono essere spediti dopo l'ultimo confermato

↓  
Se voglio  $T_x = \text{banda disponibile}$ , la finestra dovrà essere uguale al prodotto banda ritardo (BDP)  $\rightarrow W = B \cdot RTT$

$$W = \min(cwnd, rwnd) \rightarrow T_x = \min\left(\frac{cwnd}{RTT}, \frac{rwnd}{RTT}\right)$$

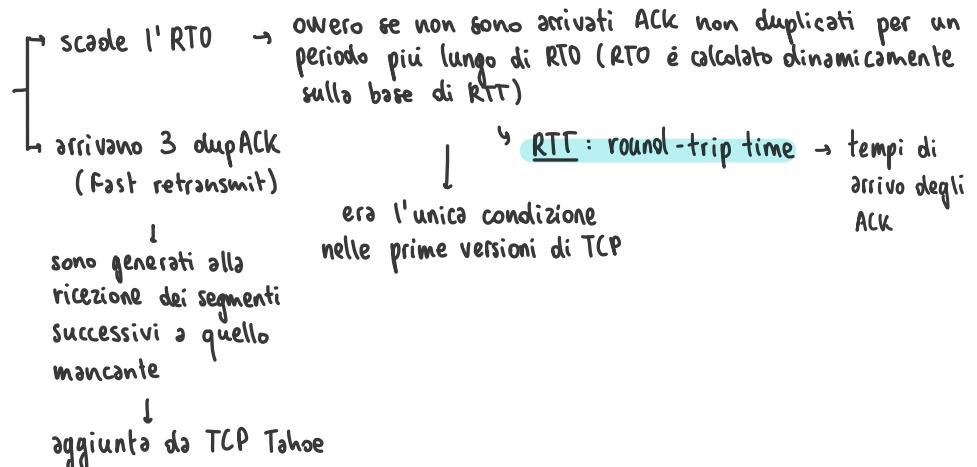
↓  
massima Tx ammessa dal controllo di congestione

↓  
massima Tx ammessa dal controllo di flusso

], La velocità di trasmissione può quindi essere limitata dal controllo di flusso o da quello di congestione

↓  
Sia cwnd che rwnd cambiano nel tempo

PERDITE → si ritiene sia andato perso un segmento se



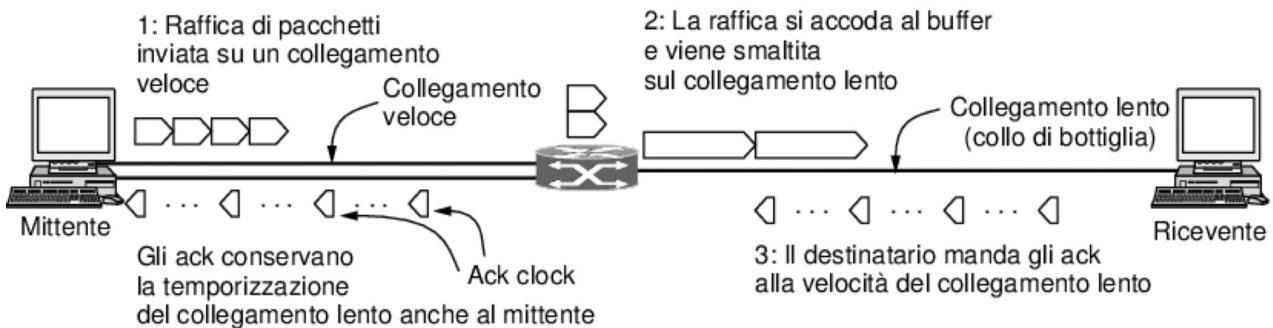
### PROTOCOLLO TCP: CONTROLLO DI CONGESTIONE

Quando il carico applicato a qualsiasi rete è più di quello che questa riesce a gestire, si crea una congestione, e Internet non fa eccezione. Il livello di rete rileva la congestione quando le code sui router aumentano e cerca di gestirla scartando dei pacchetti. È compito del livello di trasporto ricevere i segnali di congestione dal livello di rete e rallentare il tasso del traffico inviato sulla rete.

Il controllo della congestione di TCP utilizza una finestra e interpreta la perdita di pacchetti come segnali binari di congestione. Per farlo, TCP mantiene una finestra di congestione la cui dimensione è pari al numero di byte che il mittente può avere sulla rete in qualsiasi momento. Il tasso corrispondente è la dimensione della finestra diviso per il round-trip time della connessione. TCP aggiusta la dimensione della finestra. Si ricordi che la finestra di congestione viene mantenuta in aggiunta alla finestra di controllo del flusso che specifica il numero di byte che il destinatario può inserire nel suo buffer. Entrambe le finestre vengono seguite in parallelo e il numero di byte che possono essere spediti è il più piccolo delle due finestre. TCP smetterà di spedire dati se la finestra di controllo di congestione o quella di controllo di flusso sono temporaneamente piene.

Tutti gli algoritmi di TCP usati su Internet assumono che i pacchetti persi siano causati da una congestione; si affidano ai timeout analogamente ai minatori con i canarini. È necessario un buon timer di ritrasmissione a rilevare i segnali di perdita di pacchetti in modo preciso e tempestivo. Dato un buon timeout di ritrasmissione, il mittente TCP può tracciare il numero di byte in sospeso che occupa la rete, cercando semplicemente la differenza tra le sequenze numeriche trasmesse e quelle che hanno ricevuto acknowledgement.

Analizziamo per esempio cosa accade quando un mittente su una rete veloce (la linea a 1 Gbps) spedisce un gruppo di quattro pacchetti molto ravvicinati tra loro a un destinatario su una rete lenta (la linea a 1 Mbps) che è il collo di bottiglia, o la parte più lenta, sul percorso. Inizialmente i quattro pacchetti viaggiano sul collegamento alla velocità a cui possono essere spediti dal mittente. Nel router vengono accodati durante l'inoltro perché ci vuole più tempo per spedire un pacchetto su un collegamento lento rispetto che per ricevere il pacchetto successivo su un collegamento veloce. Ma la coda non è lunga, perché solo un piccolo numero di pacchetti è stato spedito in una volta. Da notare la lunghezza maggiore dei pacchetti sul collegamento lento: lo stesso pacchetto di 1 KB ora è più lungo perché si impiega più tempo per spedirlo.

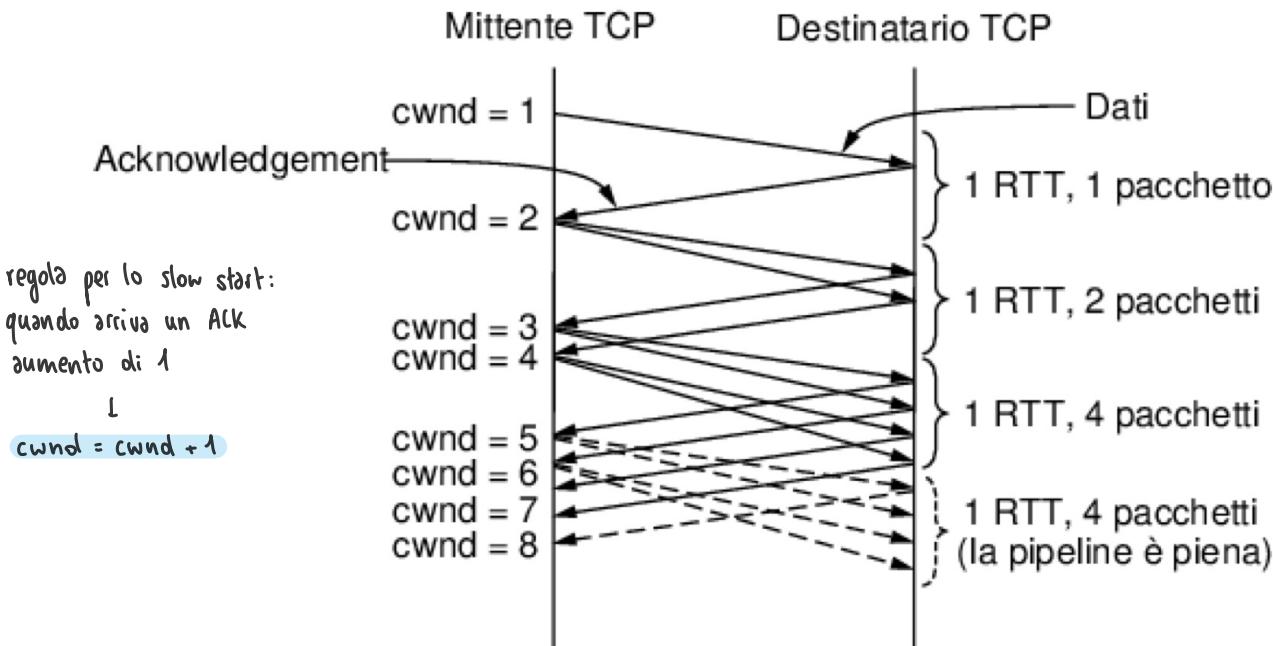


Alla fine i pacchetti arrivano al destinatario, dove verrà generato un acknowledgement. Le tempistiche degli acknowledgement riflettono i tempi impiegati da pacchetti per arrivare al destinatario dopo aver attraversato il collegamento più lento e risultano sparpagliati se confrontati ai pacchetti originali sul collegamento veloce. Nell'attraversare la rete per tornare al mittente gli acknowledgement mantengono inalterate queste tempistiche. Gli acknowledgement tornano al mittente a circa lo stesso tasso a cui i pacchetti possono essere spediti sul collegamento più lento lungo il percorso; questo è esattamente il tasso che il mittente desidera usare. Se inserisce nuovi pacchetti nella rete con questo tasso, essi saranno spediti alla velocità consentita dal collegamento più lento, ma non saranno messi in coda e non congestioneranno alcun router lungo il percorso. Questa temporizzazione è conosciuta come **ack clock** ed è una parte essenziale di TCP. Usando un ack clock, TCP rende più regolare il traffico ed evita code inutili a router.

### ALGORITMO SLOW START

Quando viene stabilita una connessione, il mittente inizializza la finestra di congestione a un valore iniziale basso, di al massimo quattro segmenti; i dettagli sono descritti nell'RFC 3390, l'uso di quattro segmenti è un incremento, sulla base dell'esperienza, da un primo valore iniziale di un solo segmento. Il mittente, quindi, spedisce la finestra iniziale. I pacchetti impiegheranno un round-trip time per ottenere i relativi acknowledgement. Per ogni segmento che ha ricevuto 'acknowledgement prima dello scadere del timer di ritrasmissione il mittente aggiunge alla finestra di congestione il valore in byte di un segmento; in più, quando quel segmento ha ricevuto un acknowledgement ci sarà un segmento in meno in rete. Il risultato è che ogni acknowledgement, permette a due segmenti di essere inviati. La finestra di congestione raddoppia a ogni round-trip time. Questo algoritmo è chiamato **slow start** (partenza lenta): nel primo round-trip il mittente immette un pacchetto nella rete (e il destinatario riceve un pacchetto), due pacchetti sono spediti nel successivo round-trip time e quindi quattro nel terzo round-trip time.

Lo slow start funziona bene su una gamma di velocità e round-trip time e usa un ack clock per allineare il tasso delle trasmissioni del mittente al percorso di rete. Si guardi il modo in cui gli acknowledgement ritornano dal mittente al destinatario nella figura sotto. Quando un mittente riceve un acknowledgement aumenta la finestra di congestione di un'unità e immediatamente spedisce due pacchetti sulla rete: un pacchetto è per l'incremento, l'altro è un rimpiazzo per il pacchetto il cui acknowledgement è stato ricevuto; in ogni momento il numero di pacchetti senza acknowledgement è dato dalla finestra di congestione. Tuttavia questi due pacchetti non arriveranno necessariamente a destinazione così ravvicinati come quando sono stati spediti; Se il percorso nella rete è lento, gli acknowledgement arrivano lentamente (dopo un ritardo di un RTT). Se il percorso di rete è veloce, gli acknowledgement arrivano velocemente (ancora, dopo un RTT). Tutto ciò che il mittente deve fare è seguire la tempistica dell'ack clock per inviare nuovi pacchetti, che è ciò che fa lo slow start.



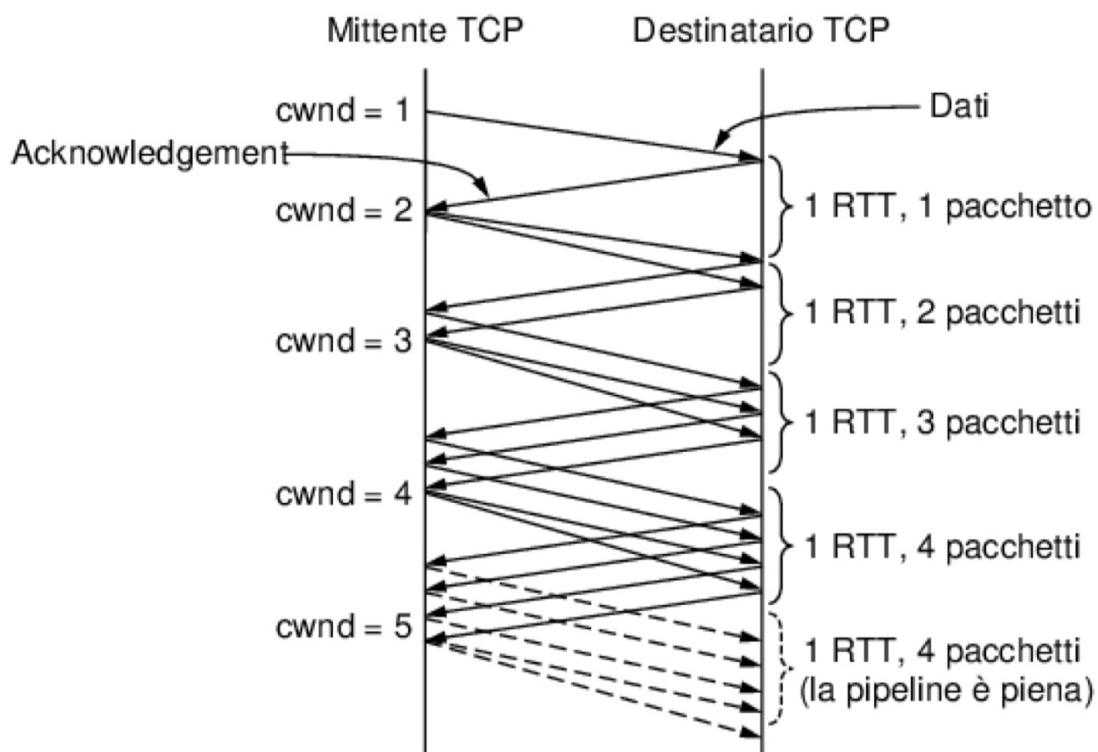
Poiché lo slow start causa una crescita esponenziale, prima o poi (più prima che poi) spedirà in rete troppi pacchetti e troppo velocemente. Quando ciò avviene, si creano code nella rete e quando le code saranno piene uno o più pacchetti saranno persi. Dopo che questo accade, il mittente TCP andrà in timeout quando un acknowledgement non riuscirà ad arrivare in tempo. Nella figura viene data evidenza del fatto che lo slow start cresce troppo velocemente; dopo due RTT, ci sono quattro pacchetti in rete. Questi quattro pacchetti impiegano un intero RTT per arrivare al destinatario, quindi una finestra di quattro pacchetti ha la giusta dimensione per questa connessione. Tuttavia, non appena vengono ricevuti gli acknowledgement di questi pacchetti, lo slow start continua a far crescere la finestra di congestione, raggiungendo gli otto pacchetti in un altro RTT. Solo quattro di questi pacchetti possono raggiungere il destinatario in un RTT, indipendentemente da quanti ne vengono spediti. In altre parole, la pipeline di rete è piena. Ulteriori pacchetti spediti in rete dal mittente si accumuleranno nelle code sui router, in quanto non potranno essere consegnati al destinatario abbastanza velocemente. Congestione e perdita di pacchetti sopraggiungeranno presto.

Per tenere sotto controllo lo slow start, il mittente mantiene una soglia per il collegamento chiamata **slow start threshold** (soglia di slow start). Inizialmente questo valore è fissato, in modo arbitrariamente elevato, alla dimensione della finestra di controllo di flusso, in maniera tale da non limitare la connessione. TCP continua a far crescere la finestra di congestione nello slow start finché non si verifica un timeout o la finestra di congestione supera la soglia (o si è riempita la finestra del destinatario).

Ogni volta che viene rilevata la perdita di un pacchetto, per esempio attraverso un timeout, la soglia di slow start è fissata per essere la metà della finestra di congestione e l'intero processo ricomincia. L'idea è che la finestra corrente sia troppo grande perché ha già causato una congestione che solo ora viene rilevata da un timeout. Metà della finestra, usata con successo in un momento precedente, è probabilmente una stima migliore per una finestra di congestione vicina alla capacità del percorso, ma che non causerà perdite. Nel nostro esempio la crescita della finestra di congestione a otto pacchetti può causare perdite, mentre la finestra di congestione di quattro pacchetti nel RTT precedente era il valore corretto.

Ogni volta che viene oltrepassata la soglia dello slow start, TCP passa dallo slow start all'**incremento additivo**; in questa modalità, la finestra di congestione viene aumentata di un segmento per ogni round-trip time. Questo, però, viene solitamente implementato con un aumento (più piccolo) per ogni segmento, per cui si riceve un acknowledgement invece di crescere a ogni RTT, come nel caso dello slow start. Indichiamo la finestra di congestione con  $cwnd$  e la grandezza massima di segmento con  $MSS$ . Un'approssimazione di uso comune consiste nell'aumentare  $cwnd$  di  $(MSS \times MSS)/cwnd$  per ognuno dei  $cwnd/MSS$  pacchetti che potrebbero ricevere un acknowledgement.

Alla fine di ogni RTT la finestra di congestione del mittente è cresciuta abbastanza per poter introdurre un pacchetto aggiuntivo in rete. Rispetto allo slow start, il tasso lineare di crescita è molto più lento. Non fa molta differenza per piccole finestre di congestione, come in questo caso, ma ne fa molta sul tempo impiegato per far crescere la finestra di congestione a 100 segmenti, ad esempio.



Dopo che viene perso un pacchetto il destinatario non può mandare acknowledgement per quelli successivi, quindi il numero di acknowledgement rimarrà fisso e il mittente non sarà in grado di spedire alcun nuovo pacchetto sulla rete, perché la sua finestra di congestione rimarrà piena. Questa condizione può continuare per un periodo relativamente lungo finché scatta il timer e il pacchetto perso viene ritrasmesso. A questo punto TCP parte nuovamente con lo slow start.

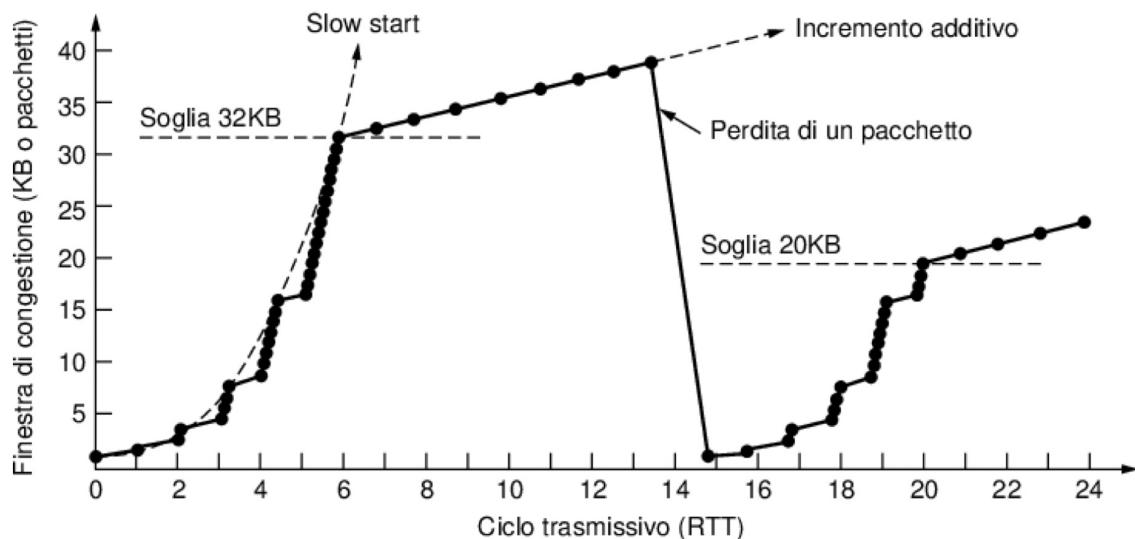
Il mittente ha un modo veloce per riconoscere che uno dei suoi pacchetti è stato perso. Non appena i pacchetti successivi al pacchetto perduto arrivano a destinazione, innescano degli acknowledgement che ritornano al mittente. Questi acknowledgement portano lo stesso numero e sono chiamati **duplicate acknowledgement** (dupACK). Ogni volta che il mittente riceve un acknowledgement duplicato è probabile che un altro pacchetto sia arrivato al destinatario e il pacchetto perso non si sia ancora fatto vivo.

### FAST RETRANSMISSION E INCREMENTO ADDITIVO → TCP Tahoe (1988)

Poiché i pacchetti possono prendere diversi percorsi attraverso rete, possono arrivare fuori ordine generando acknowledgement duplicati anche se nessun pacchetto è stato perso. Tuttavia, nella maggior parte dei casi, questo non è comune su Internet; anche quando c'è un riordino dei pacchetti a causa di percorsi diversi, i pacchetti ricevuti non sono troppo fuori ordine. Quindi TCP assume in modo abbastanza arbitrario che tre acknowledgement duplicati implichino che un pacchetto sia stato perso e l'identità del pacchetto perso si possa facilmente dedurre dal numero negli acknowledgement: è il prossimo pacchetto nella sequenza. Questo pacchetto può essere ritrasmesso immediatamente, prima che scatti il timeout di ritrasmessione.

Tale meccanismo prende il nome di **fast retransmission**. Quando entra in funzione, la soglia di slow start viene di nuovo portata a metà della corrente finestra di congestione, proprio come con un timeout. Lo slow start può essere fatto ripartire portando la finestra di congestione a un pacchetto. Con questa dimensione della finestra, dopo il round-trip time necessario per mandare un acknowledgement relativo al pacchetto ritrasmesso, verrà spedito un nuovo pacchetto insieme a tutti i dati spediti prima del rilevamento della perdita.

La dimensione massima di un segmento nell'esempio è di 1 KB. All'inizio la finestra di congestione era 64 KB, ma si era verificato un timeout, quindi la soglia fu impostata a 32 KB e la finestra di congestione a 1 KB per la trasmissione 0. La finestra di congestione cresce esponenzialmente finché raggiunge la soglia (32 KB). Invece che in maniera continua, la finestra è incrementata ogni volta che arriva un acknowledgement, il che porta a un profilo con una serie di gradini. Superata la soglia la finestra aumenta linearmente e viene incrementata di un segmento ogni RTT.



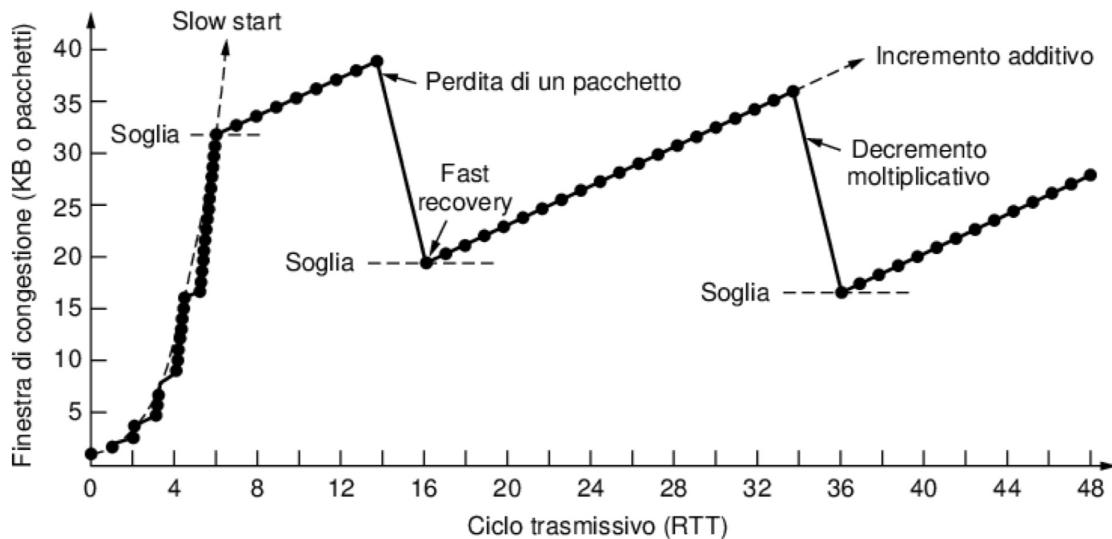
Al tredicesimo ciclo una trasmissione si perde nella rete. Questo fatto viene rilevato quando arrivano tre acknowledgement duplicati. In quel momento il pacchetto perso viene ritrasmesso, la soglia è portata a metà della finestra corrente (finora 40 KB, quindi la metà è 20 KB) e lo slow start comincia di nuovo, da capo.

**TCP Tahoe** (che include buoni timer di ritrasmessione) fornisce un algoritmo di controllo della congestione funzionante che risolve il problema del collasso per congestione.

## FAST RECOVERY → TCP Reno (1990) e newReno

Il **fast recovery** è una modalità temporanea che cerca di mantenere l'ack clock in funzione con una finestra di congestione grande quanto la nuova soglia o nel momento della ritrasmissione veloce quanto metà del valore della finestra di congestione. Si rimane in modalità temporanea finché non ci si accorge che il numero di pacchetti in rete è sceso al di sotto della nuova soglia contando gli acknowledgement duplicati (inclusi i tre che hanno fatto scattare la ritrasmissione veloce); questa procedura impiega circa metà del round-trip time. Da quel momento in poi, un nuovo pacchetto può essere spedito per ogni acknowledgement duplicato ricevuto. Trascorso un round-trip time dopo la ritrasmissione veloce il pacchetto perso avrà ricevuto un acknowledgement. A quel punto il flusso di acknowledgement duplicati cesserà e la modalità di fast recovery sarà abbandonata. La finestra di congestione sarà impostata alla nuova soglia di slow start e aumenterà con crescita lineare.

I risultato di questo comportamento è che TCP evita lo slow start, tranne all'inizio della connessione e quando si verifica un timeout. Quest'ultimo può ancora verificarsi quando va perso più di un pacchetto e la ritrasmissione veloce non recupera adeguatamente. Invece di slow start multipli, la finestra di congestione della connessione in atto segue un profilo a dente di sega per gli incrementi additivi (di un segmento ogni RTT) e i decrementi moltiplicativi (di metà in un RTT).



Perché si comporta in maniera diversa dopo la perdita di un segmento?

Ci sono molti segmenti in volo, un conto è perderne uno, un altro è una congestione grave con molti segmenti persi

TCP Reno è essenzialmente TCP Tahoe con fast recovery. Dopo uno slow start iniziale, la finestra di congestione aumenta linearmente finché viene rilevata la perdita di un pacchetto per via di acknowledgement duplicati. Il pacchetto perso è ritrasmesso e il fast recovery è usato per mantenere attivo ack clock finché la ritrasmissione riceve un acknowledgement. A quel punto la finestra di congestione è riavviata dalla nuova soglia di slow start, invece che da 1. Questo comportamento continua all'infinito e la connessione passa la maggior parte del tempo con la sua finestra di congestione vicina al valore ottimale del prodotto banda-ritardo.

La versione *TCP NewReno* (1996) opera un avanzamento parziale del numero di acknowledgement dopo una ritrasmissione per trovare e rimediare a un'altra perdita.

16/05/21

## ESEMPIO. COM'È FATTA UNA TABELLA ARP?

arp&PING(ICMP).dump

File Modifica Visualizza Vai Cattura Analizza Statistiche Telefonia Wireless Strumenti Aiuto

Aplica un filtro di visualizzazione ... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	00:aa:bc:59:c6:2b	Broadcast	ARP	42	Who has 10.0.1.2? Tell 10.0.1.1 → lui manda in broadcast la richiesta
2	0.021324	00:aa:f6:fd:9c:a3	00:aa:bc:59:c6:2b	ARP	60	10.0.1.2 is at 00:aa:f6:fd:9c:a3 → risposta
3	0.021355	10.0.1.1	10.0.1.2	ICMP	98	Echo (ping) request id=0x01f2, seq=1/256, ttl=64 (reply in 4)
4	0.043686	10.0.1.2	10.0.1.1	ICMP	98	Echo (ping) reply id=0x01f2, seq=1/256, ttl=64 (request in 3)
5	1.001554	10.0.1.1	10.0.1.2	ICMP	98	Echo (ping) request id=0x01f2, seq=2/512, ttl=64 (reply in 6)
6	1.022876	10.0.1.2	10.0.1.1	ICMP	98	Echo (ping) reply id=0x01f2, seq=2/512, ttl=64 (request in 5)
7	2.002841	10.0.1.1	10.0.1.2	ICMP	98	Echo (ping) request id=0x01f2, seq=3/768, ttl=64 (reply in 8)
8	2.024182	10.0.1.2	10.0.1.1	ICMP	98	Echo (ping) reply id=0x01f2, seq=3/768, ttl=64 (request in 7)
9	5.241304	00:aa:f6:fd:9c:a3	00:aa:bc:59:c6:2b	ARP	60	Who has 10.0.1.1? Tell 10.0.1.2 → chiede l'indirizzo Ethernet del 10.0.1.1
10	5.241336	00:aa:bc:59:c6:2b	00:aa:f6:fd:9c:a3	ARP	42	10.0.1.1 is at 00:aa:bc:59:c6:2b

→ scambio di pacchetti

↳ il protocollo ARP qui probabilmente deve completare la tabella

> Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)  
> Ethernet II, Src: 00:aa:bc:59:c6:2b (00:aa:bc:59:c6:2b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
> Address Resolution Protocol (request)

0000 ff ff ff ff ff ff 00 aa bc 59 c6 2b 08 06 00 01 ..... Y+...  
0010 00 00 06 04 00 01 00 aa bc 59 c6 2b 0a 00 01 01 ..... Y+...  
0020 00 00 00 00 00 00 0a 00 01 02 ..... .

arp&PING(ICMP).dump Pacchetti: 10 - visualizzati: 10 (100.0%)

> Frame 1: 42 bytes on wire (336 bits), 42 bytes captured (336 bits)  
> Ethernet II, Src: 00:aa:bc:59:c6:2b (00:aa:bc:59:c6:2b), Dst: Broadcast (ff:ff:ff:ff:ff:ff)  
v Address Resolution Protocol (request)  
  Hardware type: Ethernet (1)  
  Protocol type: IPv4 (0x0800)  
  Hardware size: 6  
  Protocol size: 4  
  Opcode: request (1)  
  Sender MAC address: 00:aa:bc:59:c6:2b (00:aa:bc:59:c6:2b)  
  Sender IP address: 10.0.1.1  
  Target MAC address: 00:00:00\_00:00:00 (00:00:00:00:00:00) → Non lo conosco, quindi tutti 0  
  Target IP address: 10.0.1.2

0000 ff ff ff ff ff ff 00 aa bc 59 c6 2b 08 06 00 01 ..... Y+...  
0010 00 00 06 04 00 01 00 aa bc 59 c6 2b 0a 00 01 01 ..... Y+...  
0020 00 00 00 00 00 00 0a 00 01 02 ..... .

↳ copie di esadecimale e byte

Protocol type (arp.proto.type), 2byte

→ questo pacchetto ARP è di livello "3-" perché ancora non conosce l'IP destinatario ed interno

↳ quando li conosce diventa "3+"

> Frame 3: 98 bytes on wire (784 bits), 98 bytes captured (784 bits)  
> Ethernet II, Src: 00:aa:bc:59:c6:2b (00:aa:bc:59:c6:2b), Dst: 00:aa:f6:fd:9c:a3 (00:aa:f6:fd:9c:a3)  
v Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2  
  0100 .... = Version: 4  
  .... 0101 = Header Length: 20 bytes (5)  
  > Differentiated Services Field: 0x00 (DSCP: CS0, ECN: Not-ECT)  
  Total Length: 84  
  Identification: 0x691f (26911)  
  > Flags: 0x40, Don't fragment  
  ... 0 0000 0000 0000 = Fragment Offset: 0  
  Time to Live: 64  
  Protocol: ICMP (1)  
  Header Checksum: 0xbb87 [validation disabled]  
  [Header checksum status: Unverified]

0000 00 aa f6 fd 9c a3 00 aa bc 59 c6 2b 08 00 45 00 ..... Y+... E.  
0010 00 54 69 1f 40 00 40 01 bb 87 0a 00 01 01 0a 00 .Ti:@:@. ....  
0020 01 02 08 00 22 66 01 f2 00 01 e1 02 bd 5e 00 00 ....f.....^..  
0030 00 00 69 72 0d 00 00 00 00 00 10 11 12 13 14 15 ..i.....  
0040 16 17 18 19 1a 1b 1c 1d 1e 1f 20 21 22 23 24 25 ..... !#\$%  
0050 26 27 28 29 2a 2b 2c 2d 2e 2f 30 31 32 33 34 35 8'()"+,- ./012345

Differentiated Services Field (ip.dsfield), 1byte

→ qui si possono vedere tutti i campi dell'header di IPv4

16/05/22

## ESEMPIO: UDP

udp.dump

File Modifica Visualizza Vai Cattura Analizza Statistiche Telefonia Wireless Strumenti Aiuto

Applica un filtro di visualizzazione... <Ctrl-/>

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
2	0.011515	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
3	0.022743	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
4	0.034083	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
5	0.045216	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
6	0.056457	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
7	0.067582	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
8	0.078759	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
9	0.090016	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
10	0.101206	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
11	0.112452	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
12	0.123847	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
13	0.134960	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
14	0.146277	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470
15	0.157414	10.0.1.1	10.0.1.2	UDP	1512	58146 → 5001 Len=1470

> Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2  
 User Datagram Protocol, Src Port: 58146, Dst Port: 5001  
 Source Port: 58146  
 Destination Port: 5001  
 Length: 1478  
 Checksum: 0x1bda [unverified]  
 [Checksum Status: Unverified]  
 [Stream index: 0]  
 > [Timestamps]  
 UDP payload (1470 bytes)  
 Data (1470 bytes)  
 Data: 000000005ebd06530003fdb7323334350000000003013233343536373839303132333435...  
 [Length: 1470]

presente nei file esempio wireshark → non per l'esame, ma potrebbe fare una domanda

DHCP : è un protocollo a livello applicazione che serve a trovare gli indirizzi (richiesta in broadcast)

include l'indirizzo Ethernet del mittente nel discover, l'informazione ritorna in unicast con acknowledgment (file DHCP2)

se l'indirizzo IP destinatario è già stato conosciuto non molto tempo prima, DHCP può procedere a verificare direttamente se è disponibile (file DHCP1)

## ESEMPIO: TCP (E TELNET) → Telnet non ci interessa, serve a gestire macchine da remoto, obsoleto, ora si usa SSH

telnet.dump

File Modifica Visualizza Vai Cattura Analizza Statistiche Telefonia Wireless Strumenti Aiuto

(ip.addr eq 10.0.1.1 and ip.addr eq 10.0.1.2) and (tcp.port eq 54930 and tcp.port eq 23)

No.	Time	Source	Destination	Protocol	Length	Info
1	0.000000	10.0.1.1	10.0.1.2	TCP	74	54930 → 23 [SYN] Seq=0 Win=29200 Len=0 MSS=1460 SACK_PERM=1 TSval=1510800519 TSecr=0 WS=64
2	0.020991	10.0.1.2	10.0.1.1	TCP	74	23 → 54930 [SYN, ACK] Seq=0 Ack=1 Win=28960 Len=0 MSS=1460 SACK_PERM=1 TSval=1706591392 TSecr=1510800519 WS=64
3	0.021069	10.0.1.1	10.0.1.2	TCP	66	54930 → 23 [ACK] Seq=1 Ack=1 Win=0 TSval=1510800548 TSecr=1706591392
4	0.022667	10.0.1.1	10.0.1.2	TELNET	90	Telnet Data ...
5	0.049082	10.0.1.2	10.0.1.1	TCP	66	23 → 54930 [ACK] Seq=1 Ack=25 Win=0 TSval=1706591420 TSecr=1510800542
6	0.052300	10.0.1.2	10.0.1.1	TELNET	78	Telnet Data ...
7	0.052325	10.0.1.1	10.0.1.2	TCP	66	54930 → 23 [ACK] Seq=25 Ack=13 Win=29248 Len=0 TSval=1510800571 TSecr=1706591424
8	0.052574	10.0.1.1	10.0.1.2	TELNET	69	Telnet Data ...
9	0.073364	10.0.1.2	10.0.1.1	TELNET	81	Telnet Data ...
10	0.114760	10.0.1.1	10.0.1.2	TCP	66	54930 → 23 [ACK] Seq=28 Ack=28 Win=29248 Len=0 TSval=1510800634 TSecr=1706591445
11	0.115316	10.0.1.2	10.0.1.1	TCP	66	23 → 54930 [ACK] Seq=28 Ack=28 Win=28992 Len=0 TSval=1706591485 TSecr=1510800572
12	0.115357	10.0.1.1	10.0.1.2	TELNET	75	Telnet Data ...
13	0.134870	10.0.1.2	10.0.1.1	TELNET	84	Telnet Data ...
14	0.134908	10.0.1.1	10.0.1.2	TCP	66	54930 → 23 [ACK] Seq=37 Ack=46 Win=29248 Len=0 TSval=1510800654 TSecr=1706591507
15	0.137586	10.0.1.2	10.0.1.1	TCP	66	23 → 54930 [ACK] Seq=46 Ack=37 Win=28992 Len=0 TSval=1706591508 TSecr=1510800654

> Frame 4: 90 bytes on wire (720 bits), 90 bytes captured (720 bits)  
 > Ethernet II, Src: 00:aa:bc:59:c6:2b (00:aa:bc:59:c6:2b), Dst: 00:aa:f6:fd:9c:a3 (00:aa:f6:fd:9c:a3)  
 > Internet Protocol Version 4, Src: 10.0.1.1, Dst: 10.0.1.2  
 > Transmission Control Protocol, Src Port: 54930, Dst Port: 23, Seq: 1, Ack: 1, Len: 24  
 > Telnet

0000 00 aa f6 fd 9c a3 00 aa bc 59 c6 2b 08 00 45 10 . . . . . . Y + . E .  
 0010 00 4c 00 bf 40 00 40 06 1b db 0a 00 01 01 0a 00 . L @ @ . . . . .  
 0020 01 02 d6 92 00 17 f9 07 54 b6 e8 aa 01 82 80 18 . . . . . T . . . . .  
 0030 01 c9 16 41 00 00 01 01 08 0a 5a 0c fc 9e 65 b8 . A . . . Z - e .  
 0040 84 a0 ff fd 03 ff fb 18 ff fb if ff fb 20 ff fb . . . . . . . . .  
 0050 21 ff fb 22 ff fb 27 ff fd 05 ! . . . . . . . . .

in questo caso non ci sono tutti i campi flag (nemmeno dell'ACK) tranne il Syn

- > TCP Option - SACK permitted
- > TCP Option - Timestamps: TSval 1510800519, TSecr 0
  - Kind: Time Stamp Option (8)
  - Length: 10
  - Timestamp value: 1510800519
  - Timestamp echo reply: 0
- > TCP Option - No-Operation (NOP)
- > TCP Option - Window scale: 6 (multiply by 64)
  - Kind: Window Scale (3)
  - Length: 3
  - Shift count: 6
  - [Multiplier: 64]
- > [Timestamps]

Acknowledgment number (raw): 0  
 1010 .... = Header Length: 40 bytes (10)

Flags: 0x002 (SYN)

- 000. .... .... = Reserved: Not set
- ...0 .... .... = Nonce: Not set
- .... 0.... .... = Congestion Window Reduced (CWR): Not set
- .... .0.... .... = ECN-Echo: Not set
- .... ..0.... .... = Urgent: Not set
- .... ...0.... .... = Acknowledgment: Not set
- .... ...0.... .... = Push: Not set
- .... .... .0... .... = Reset: Not set
- > .... .... ..1. .... = Syn: Set
- .... .... ...0 .... = Fin: Not set

[TCP Flags: .....S.]

→ opzioni (non le abbiamo ancora viste nel dettaglio)  
 (l'opzione timestamp è sempre quella)

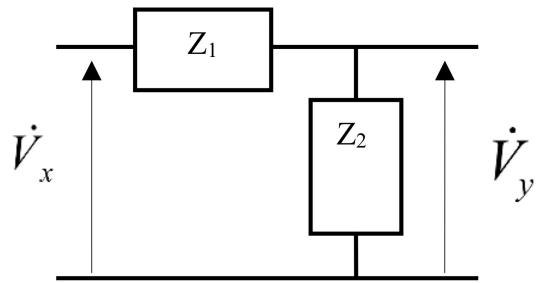
### CALCOLO DELLE FUNZIONI DI TRASFERIMENTO DI PARTITORI NON RESISTIVI

La funzione di trasferimento dipende in generale dal carico ai morsetti di uscita. Se non specificato altrimenti si intende calcolata a circuito aperto.

Dalla legge di Ohm generalizzata,  $V=ZI$ , vista in Elettrotecnica, tenendo conto che la corrente nei due rami è necessariamente la stessa.

$$\dot{V}_x = i(Z_1 + Z_2) \quad \dot{V}_y = iZ_2$$

$$H(\omega) = \frac{\dot{V}_y}{\dot{V}_x} = \frac{Z_2}{Z_1 + Z_2}$$



Le caratteristiche di ampiezza e fase si ottengono velocemente ricordando che il modulo di un rapporto è il rapporto dei moduli e che l'argomento di un rapporto è la differenza fra l'argomento del numeratore e quello del denominatore:

$$\left| \frac{a}{b} \right| = \frac{|a|}{|b|} \quad \arg \left\{ \frac{a}{b} \right\} = \arg \{a\} - \arg \{b\}$$

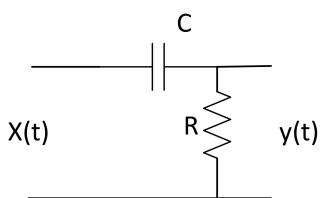
Punti notevoli, da calcolare, sono l'origine, infinito, i punti di massimo e minimo della caratteristica di ampiezza ed eventualmente la pulsazione di taglio a 3dB,  $\omega_t$ , data dalla pulsazione alla quale

$$\frac{T(\omega_t)}{\max [T(\omega)]} = \frac{1}{\sqrt{2}} \approx 0,7$$

IN SERIE → SOMMA DUE IMPEDENZE

IN PARALLELO → SOMMA DEGLI INVERSI DUE IMPEDENZE

### FUNZIONE DI TRASFERIMENTO RETE CR

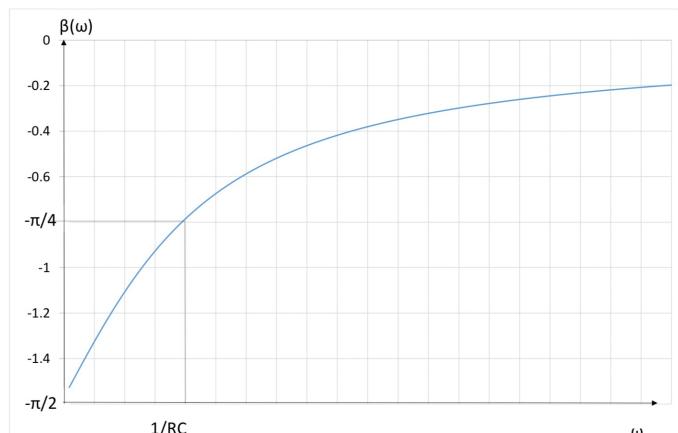
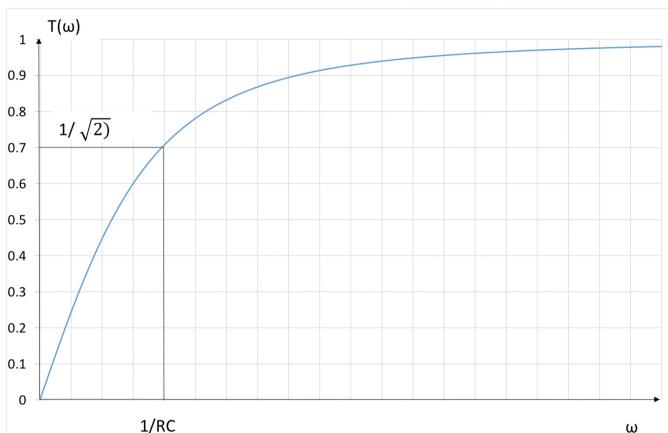


$$H(\omega) = \frac{Z_2}{Z_1 + Z_2} = \frac{R}{R + \frac{1}{j\omega C}} = \frac{j\omega CR}{1 + j\omega CR}$$

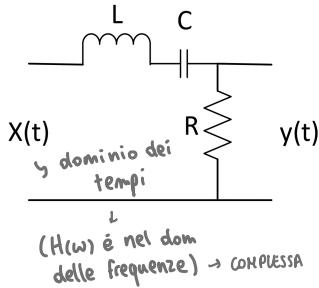
$$\text{Ampiezza: } T(\omega) = |H(\omega)| = \frac{|j\omega CR|}{|1 + j\omega CR|} = \frac{\omega CR}{\sqrt{1 + (\omega CR)^2}}$$

$$\text{Fase: } \beta(\omega) = -\arg \{H(\omega)\} = \arg \{1 + j\omega CR\} - \arg \{j\omega CR\} = \arctg(\omega CR) - \frac{\pi}{2}$$

La rete si comporta come un filtro passa-alto non ideale. La pulsazione di taglio è:  $\omega_t = \frac{1}{RC}$



## FUNZIONE DI TRASFERIMENTO RETE LCR (SERIE)



$\rightarrow$  impedenza sul primo ramo  $Z_1 = j\omega L + \frac{1}{j\omega C}$

L: INDUCTORE      C: CONDENSATORE

$$H(w) = \frac{Z_2}{Z_1 + Z_2} = \frac{R}{R + j\omega L + \frac{1}{j\omega C}} = \frac{j\omega CR}{1 - \omega^2 LC + j\omega CR}$$

$\downarrow$  RESISTENZA

FUNZIONE DI TRASFERIMENTO = TRASFORMATA DI FOURIER DELLA RISPOSTA IMPULSIVA

$$H(w) = F[h(t)]$$

$$\text{Ampiezza: } T(w) = |H(w)| = \frac{|j\omega CR|}{\sqrt{(1 - \omega^2 LC)^2 + (\omega CR)^2}}$$

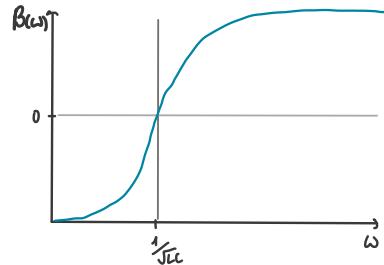
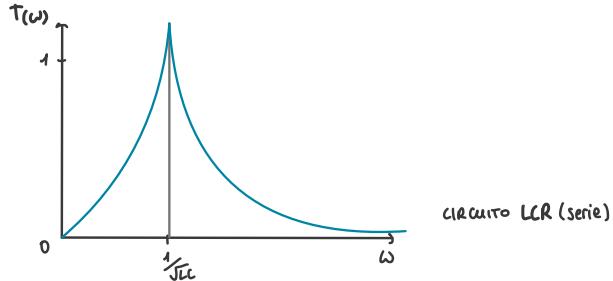
$$\leftarrow \text{Fase: } \beta(w) = -\arg\{H(w)\} = \arg\{1 - \omega^2 LC + j\omega CR\} - \arg\{j\omega CR\} = \arg\{1 - \omega^2 LC + j\omega CR\} - \frac{\pi}{2}$$

il denominatore della funzione di trasferimento

ha parte reale positiva per  $\omega \leq \omega_0 \rightarrow \omega_0 = \frac{1}{\sqrt{LC}}$   $\rightarrow$  PULSAZIONE DI RISONANZA

LA RETE SI COMPORTA COME UN FILTRO PASSA-BANDA REALE

$$\arg\{1 - \omega^2 LC + j\omega CR\} = \begin{cases} \arctg\left(\frac{\omega CR}{1 - \omega^2 LC}\right) & \omega \leq \omega_0 \\ \arctg\left(\frac{\omega CR}{1 - \omega^2 LC}\right) + \pi & \omega > \omega_0 \end{cases} \rightarrow \beta(w) = \begin{cases} \arctg\left(\frac{\omega CR}{1 - \omega^2 LC}\right) & \omega \leq \omega_0 \\ \arctg\left(\frac{\omega CR}{1 - \omega^2 LC}\right) + \pi & \omega > \omega_0 \end{cases}$$



RIPASSO: FUNZIONI DI TRASFERIMENTO E SISTEMI LINEARI → RISPOSTA A SEGNAI SVILUPPABILI IN SERIE DI FOURIER



$$X(\omega)$$

$$Y(\omega) = X(\omega)H(\omega)$$

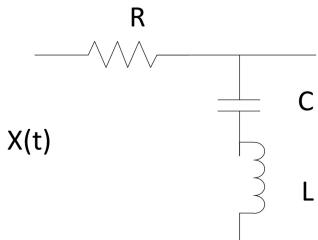
$$x(t) = \sum_{n=-\infty}^{\infty} c_n e^{jn\omega_0 t}$$

$$y(t) = \sum_{n=-\infty}^{\infty} c_n H(n\omega_0) e^{jn\omega_0 t}$$

$$x(t) = A_o + \sum_{n=1}^{\infty} A_n \cos(n\omega_0 t - \varphi_n) \quad y(t) = A_o H(0) + \sum_{n=1}^{\infty} A_n T(n\omega_0) \cos[n\omega_0 t - \varphi_n - \beta(n\omega_0)]$$

$$x(t) = \int_0^{+\infty} V(\omega) \cos[\omega t - \varphi(\omega)] d\omega \quad y(t) = \int_0^{+\infty} V(\omega) T(\omega) \cos[\omega t - \varphi(\omega) - \beta(\omega)] d\omega$$

### FUNZIONE DI TRASFERIMENTO RETE RCL (SERIE)



$$\begin{aligned} Z_1 &= R \\ Z_2 &= j\omega L + \frac{1}{j\omega C} \end{aligned} \rightarrow H(\omega) = \frac{Z_2}{Z_1 + Z_2} = \frac{j\omega L + \frac{1}{j\omega C}}{R + j\omega L + \frac{1}{j\omega C}} = \frac{1 - \omega^2 CR}{1 - \omega^2 LC + j\omega CR}$$

Le caratteristiche di ampiezza e fase sono date da:

$$T(\omega) = |H(\omega)| = \frac{|1 - \omega^2 LC|}{\sqrt{(1 - \omega^2 LC)^2 + (\omega CR)^2}}$$

$$\beta(\omega) = -\arg\{H(\omega)\} = \arg\{1 - \omega^2 LC + j\omega CR\} - \arg\{1 - \omega^2 LC\}$$

Prima di procedere conviene porre  $\omega_0 = \frac{1}{\sqrt{LC}}$  (pulsazione di risonanza) e notare che sia il numeratore che il denominatore della funzione di trasferimento hanno parte reale positiva per  $\omega \leq \omega_0$  negativa altrimenti. Gli argomenti assumono quindi due espressioni:

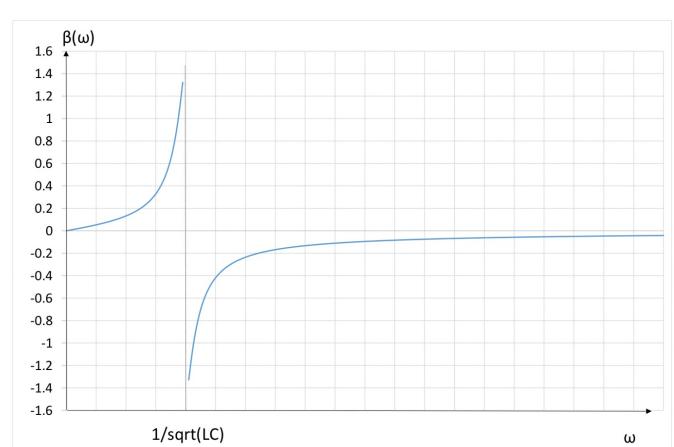
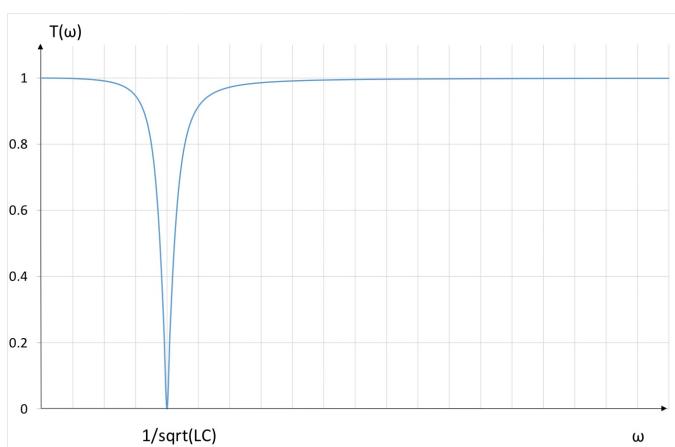
$$\arg\{1 - \omega^2 LC\} = \begin{cases} 0 & \omega \leq \omega_0 \\ +\pi & \omega > \omega_0 \end{cases}$$

$$\arg\{1 - \omega^2 LC + j\omega CR\} = \begin{cases} \arctg\left(\frac{\omega CR}{1 - \omega^2 LC}\right) & \omega \leq \omega_0 \\ \arctg\left(\frac{\omega CR}{1 - \omega^2 LC}\right) + \pi & \omega > \omega_0 \end{cases}$$

L'espressione finale della caratteristica di fase è quindi:  $\beta(\omega) = \arctg\left(\frac{\omega CR}{1 - \omega^2 LC}\right)$

Si noti che la discontinuità della fase avviene alla pulsazione di risonanza, ovvero nel punto di nullo della caratteristica di ampiezza, nel quale la fase è ovviamente non definita.

La rete si comporta come un filtro elimina-banda reale. → NOTCH FILTER



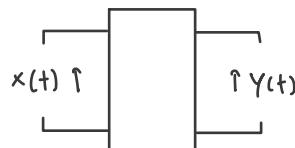
Se diamo in ingresso ad una rete lineare  $\cos^2 \omega_0 t \rightarrow$  qual è la risposta?

$\cos^2 =$  somma di una sinusoida

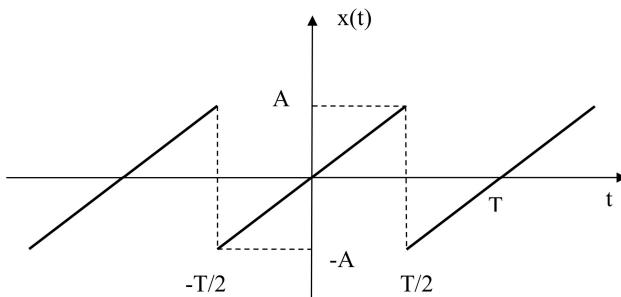
a frequenza doppia

$$X(t) = \cos^2 \omega_0 t = \frac{1 + \cos 2\omega_0 t}{2} = \frac{1}{2} + \frac{1}{2} \cos 2\omega_0 t$$

$$Y(t) = \frac{1}{2} H(0) + \frac{1}{2} T(2\omega_0) \cos[2\omega_0 t - \beta(2\omega_0)]$$



### Sviluppo in serie di un'onda a dente di sega bilanciata (pag. 37)



Si osservi che si tratta di una funzione reale continua nei valori, tempo-continua, dispari. Il segnale  $x(t)$  all'interno di un periodo è dato da un tronco di retta passante per l'origine:

$$x(t) = \frac{2A}{T} t \quad |t| \leq \frac{T}{2}$$

Per determinare gli spettri monolateri, cioè i coefficienti  $A_n$  e  $\varphi_n$  si può procedere in due modi: nel primo passando dal calcolo dei coefficienti  $c_n$  della forma esponenziale, nel secondo, dal calcolo dei coefficienti  $b_n$  della serie dei seni della terza forma (essendo  $x(t)$  dispari, i coefficienti  $a_n$  della serie dei coseni sono nulli).

Soluzione tramite calcolo  $c_n$ :

$$\begin{aligned} c_n &= \frac{1}{T} \int_{-T/2}^{+T/2} x(t) e^{-j n \omega_0 t} dt = \frac{2A}{T^2} \int_{-T/2}^{+T/2} t e^{-j n \omega_0 t} dt = \\ &= \frac{2A}{T^2} \left[ \frac{te^{-jn\omega_0 t}}{-jn\omega_0} \right]_{-T/2}^{+T/2} - \frac{2A}{T^2} \int_{-T/2}^{+T/2} \frac{e^{-jn\omega_0 t}}{-jn\omega_0} dt \\ &= \frac{A}{T} \frac{e^{-jn\omega_0 T/2} + e^{jn\omega_0 T/2}}{-jn\omega_0} + \frac{2A}{jn\omega_0 T^2} \int_{-T/2}^{+T/2} e^{-jn\omega_0 t} dt = \\ &= \frac{A}{T} \frac{2 \cos n\omega_0 T / 2}{-jn\omega_0} = \frac{A}{T} \frac{2 \cos n \frac{2\pi}{T} / 2}{-jn \frac{2\pi}{T}} = \\ &= j \frac{A}{n\pi} \cos n\pi = j \frac{A}{n\pi} (-1)^n \quad n \neq 0 \end{aligned}$$

integrale nullo perché la funzione integranda è somma di un coseno e di un seno con periodo  $\frac{T}{n}$ , e l'integrale di un coseno o di un seno su un numero intero di periodi è nullo.

Si noti che i coefficienti  $c_n$  sono in questo caso puramente immaginari anziché complessi. Questo è un risultato prevedibile perché la funzione  $x(t)$  è reale dispari. Ricordando le relazioni che legano fra loro i coefficienti, gli spettri di ampiezza sono espressi da:

$$A_0 = 0 \quad A_n = |2c_n| = \frac{2A}{n\pi} \quad n \geq 1$$

E quelli di fase da:  $\varphi_n = -\arg \{c_n\} = \begin{cases} \frac{\pi}{2} & n \text{ dispari} \\ -\frac{\pi}{2} & n \text{ pari} \end{cases}$

Soluzione tramite calcolo  $b_n$ :

Poiché la funzione  $x(t)$  è reale dispari, sono diversi da zero solo i coefficienti  $b_n$ .

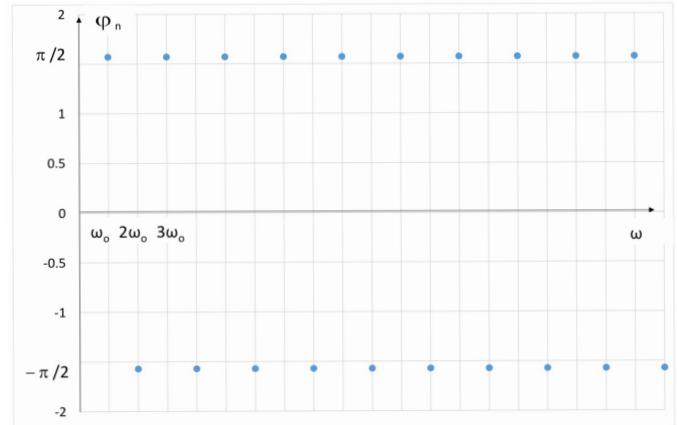
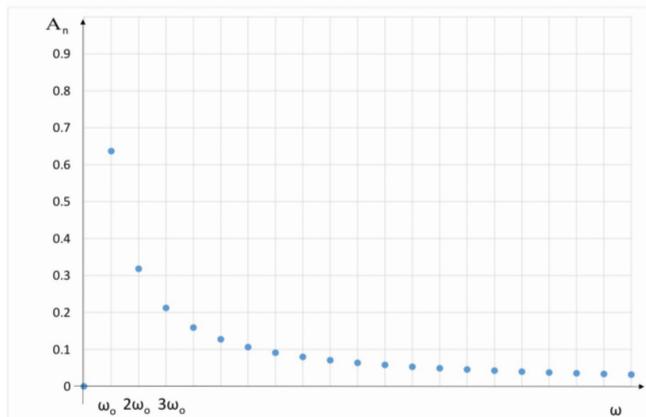
$$\begin{aligned}
 b_n &= \frac{2}{T} \int_{-T/2}^{+T/2} x(t) \sin n\omega_0 t dt = \frac{4A}{T^2} \int_{-T/2}^{+T/2} t \sin n\omega_0 t dt = \\
 &= \frac{4A}{T^2} \left[ \frac{-t \cos n\omega_0 t}{n\omega_0} \right]_{-T/2}^{+T/2} - \frac{4A}{T^2} \int_{-T/2}^{+T/2} \frac{-\cos n\omega_0 t}{n\omega_0} dt \quad \rightarrow \\
 &= \frac{4A}{T^2} \left[ -\frac{T}{2} \frac{2 \cos n \frac{2\pi}{T} \frac{T}{2}}{n \frac{2\pi}{T}} \right] = \\
 &= -\frac{2A}{n\pi} \cos n\pi \quad n \neq 0
 \end{aligned}$$

integrale nullo perché la funzione integranda è un coseno con periodo  $\frac{T}{n}$   
ed è quindi integrata su  $n$  periodi

Ricordando le relazioni che legano fra loro i coefficienti, gli spettri di ampiezza e fase sono espressi da:

$$A_0 = \frac{\theta_0}{2} = 0 \quad A_n = \sqrt{a_n^2 + b_n^2} = |b_n| = \frac{2A}{n\pi} \quad n \geq 1$$

$$\varphi_n = -\arg \{c_n\} = \begin{cases} \frac{\pi}{2} & b_n > 0, n \text{ dispari} \\ -\frac{\pi}{2} & b_n < 0, n \text{ pari} \end{cases}$$



$$x'(t) = x(t) + A \rightarrow c'_n = ? \rightarrow \begin{cases} c_n & n \neq 0 \\ A & n = 0 \end{cases}$$

COMUNICAZIONE VIA SATELLITE (NON PRESENTE ALL'ESAME, A SCOPO INFORMATIVO)**Geostationary Earth Orbit (GEO)**

Un'orbita circolare 35,786 chilometri sopra l'equatore terrestre che segue la direzione di rotazione della terra. Un oggetto in quest'orbita ha un periodo orbitale uguale al periodo di rotazione della terra e per questo sembra immobile, in una posizione fissata nel cielo, da chi osserva da terra. Ai satelliti di comunicazione o meteorologici sono spesso date orbite geostazionarie, così che le antenne riescano a comunicare senza doversi muovere per rintracciarli, ma rimanendo puntate in modo permanente alla posizione in cielo dove sono collocati.

$$\text{Derivazione dell'altitudine geostazionaria} \rightarrow F_{\text{newton}} = \frac{G m_{\text{sat}} m_{\text{terra}}}{r^2} \rightarrow F_{\text{centripetale}} = m_{\text{sat}} \omega^2 r \rightarrow F_{\text{newton}} = F_{\text{centripetale}}$$

$$\frac{G m_{\text{terra}}}{r^2} = \omega^2 r \rightarrow \frac{\mu}{r^3} = \omega^2 \quad \mu = G m_{\text{terra}} \rightarrow r = \sqrt[3]{\frac{\mu}{\omega^2}}$$

**Vantaggi:** non c'è bisogno di tracking; 3 satelliti a 120° nella GEO possono garantire quasi la piena copertura della terra

**Svantaggi:** molta distanza quindi grande attenuazione dello spazio libero; alto delay di propagazione (125ms da Terra a satellite, RTT>500ms); l'inclinazione dell'antenna sulla terra si abbassa con la latitudine; mancanza di copertura delle regioni polari.

(il TCP funziona molto male sui geostazionari)

**Low Earth Orbit (LEO)**

**Vantaggi:** bassa attenuazione; delay molto breve (ed RTT); la poca distanza consente immagini ad alta risoluzione della terra.

**Svantaggi:** si muovono velocemente nel cielo; per coprire la superficie della terra servono costellazioni di decine di satelliti.

Se l'orbita è polare tutte le regioni della Terra possono essere coperte da un solo satellite (non simultaneamente): buono per l'osservazione della terra; periodo tipico d'orbita: 100m; LOS (line of sight) Window: pochi minuti (tipo 8).

GEO → Inmarsat ; Thuraya  
1 GEO

LEO → Iridium ; Globalstar  
66 LEOs

Fallito 12000 LEO, di SpaceX  
LEO-MEO → OneWeb; Starlink  
(planned mega constellations)

## DTN (DELAY/DISRUPTION TOLERANT NETWORKING)

TCP e molti altri protocolli di trasporto sono basati sull'ipotesi che mittente e destinatario siano continuamente connessi da qualche percorso funzionante, altrimenti il protocollo fallisce e i dati non possono essere recapitati. In alcune reti spesso un percorso end-to-end non esiste. Un esempio è una rete spaziale come i satelliti LEO (low-earth orbit) che entrano ed escono dal raggio di azione delle stazioni a terra.

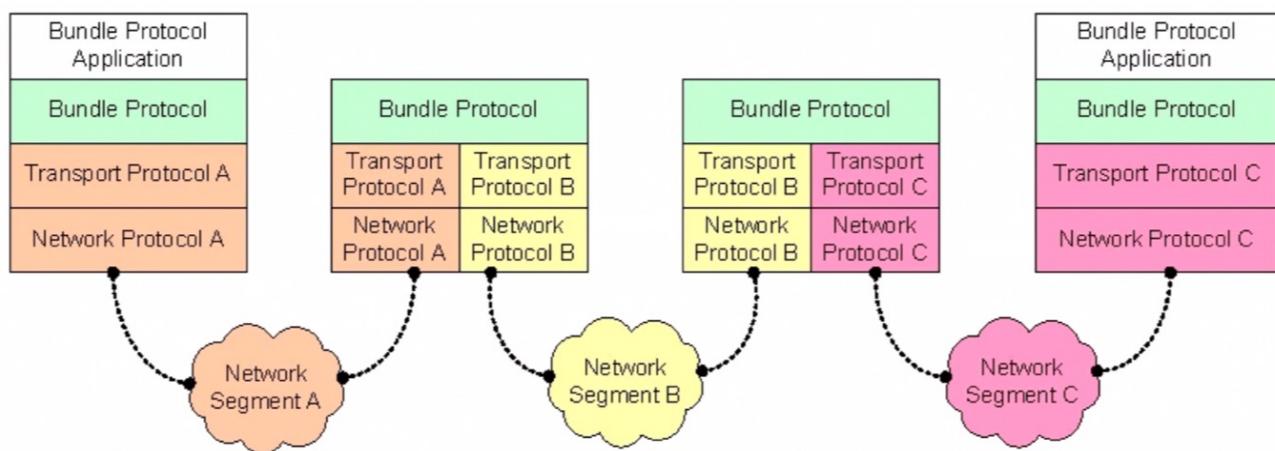
In queste reti occasionalmente connesse i dati possono ancora essere comunicati immagazzinandoli nei nodi e inoltrandoli più tardi quando c'è un collegamento funzionante: alla fine i dati saranno inviati alla destinazione. Una rete la cui architettura è basata su questo approccio è chiamata **DTN** (delay-tolerant network o disruption-tolerant network, rete con tolleranza di ritardo o rete con tolleranza alle rotture).

Il lavoro sulle DTN iniziò nel 2002 quando IETF organizzò un gruppo di ricerca sull'argomento. L'ispirazione per le DTN proveniva dagli sforzi per inviare pacchetti nello spazio: le reti spaziali devono avere a che fare con comunicazioni intermittenti e ritardi molto lunghi. Kevin Fall osservò che le idee per queste Internet Interplanetaria potevano essere applicate alle reti sulla Terra in cui la connettività intermittente era la norma.

Alcuni esempi di situazioni in cui conveniva introdurre la DTN sono: navicelle spaziali, reti militari, una specie di "risposta ai disastri" (terremoti ecc), reti subacquee, sensori, reti nei paesi in via di sviluppo

Ovviamente non si colloca all'interno del modello ISO/OSI, è una nuova architettura, un livello "aggiunto".

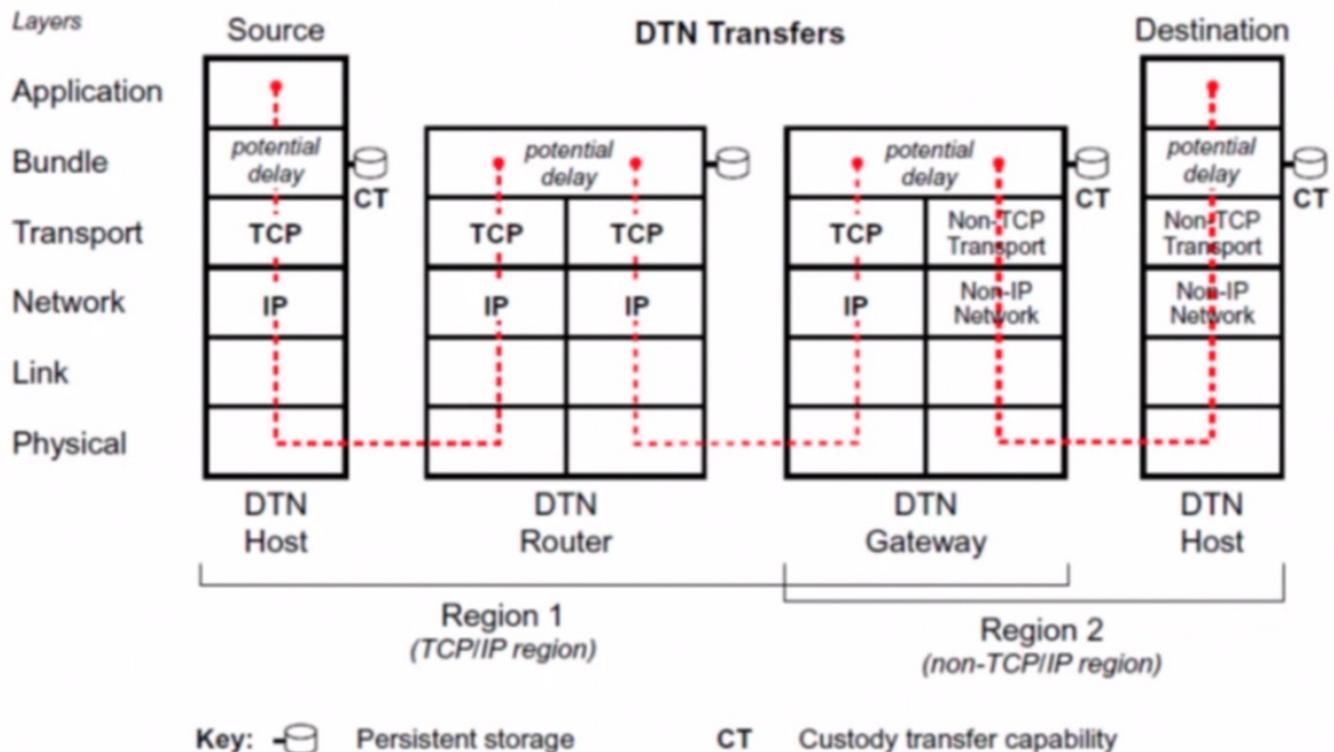
L'**architettura DTN** si basa sull'introduzione del livello **Bundle** tra il livello trasporto e applicazione. I bundle sono grandi pacchetti di dati in questo livello, che adopera un'architettura **store-and-forward** (un bundle viene ricevuto, conservato e poi trasmesso quando è possibile).



I nodi che presentano il bundle protocol sono chiamati nodi DTN → non è necessario che tutti i nodi siano DTN, ma dove occorre il bundle protocol non è end-to-end, ma hop-by-hop

- I protocolli di trasporto in questi segmenti sono specializzati nell'occuparsi dei ritardi per cui sono progettati queste reti;
- I dati devono essere immagazzinati nei nodi intermedi (applicazioni "data mule");

Immagazzinare i dati può essere utile per gestire le mancanze di un path ent-to-end: ciò può avvenire per connessioni con interruzioni casuali (terra) o per connessioni con interruzioni conosciute (spazio); si riesce ad avere una ripresa anche con lunghi RTT.



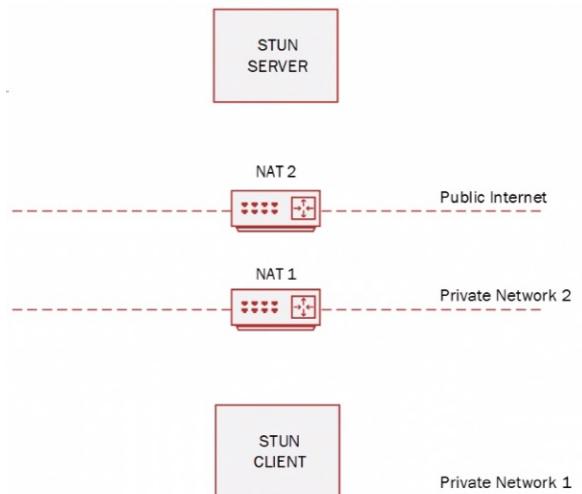
L'immagazzinamento e l'inoltro dei bundle da parte dei nodi DTN suona simile all'accodamento (nei buffer) e l'inoltro dei pacchetti nei router, ma ci sono differenze qualitative. Nei router in Internet gli accodamenti avvengono per millisecondi o al massimo secondi, mentre nei nodi DTN i bundle possono essere memorizzati per ore, finché un autobus non arriva in città, mentre un aeroplano completa il volo, finché un sensore non raccoglie abbastanza energia solare per funzionare, finché un computer non si sveglia e così via. Questi esempi indicano una seconda differenza, cioè che i nodi potrebbero muoversi (con l'autobus o l'aereo) mentre tengono memorizzati dei dati, e questo movimento può anche essere una parte importante nella consegna dei dati. I router in Internet non possono muoversi.

### PROBLEMA DELL'INDIRIZZAMENTO IN PRESENZA DI NAT (FACOLTATIVO) → Esempio basato su Microsoft Teams

Il NAT risparmia indirizzi IP, ma blocca i tentativi di connessione dall'esterno. Attraverso Wireshark sono state individuate più soluzioni: la connettività dei flussi multimediali in presenza di NAT viene implementata usando le procedure STUN, TURN ed ICE.

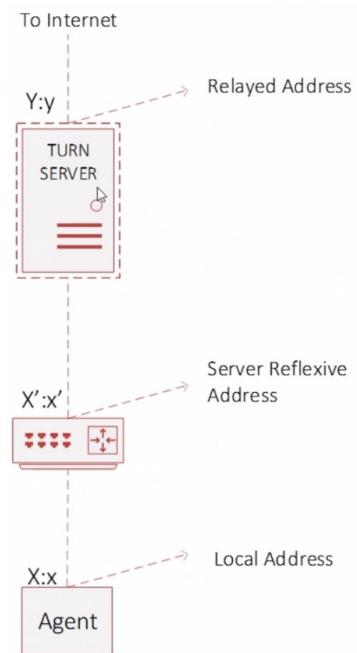
#### • **STUN:**

“Session Traversal Utilities for NAT” permette di scoprire la presenza di NAT, scoprire la coppia indirizzo IP e porta che il NAT ha associato al client, mantenere le associazioni attive;



#### • **TURN:**

“Traversal Using Relays around NAT: Relay Extensions to STUN” è una estensione di STUN; il client usa il server TURN come “relay” per scambiare pacchetti con gli altri utenti, il server TURN alloca un indirizzo di trasporto “relayed” sul server associato all’indirizzo di trasporto del client che i peers possono utilizzare per comunicare con il client



#### • **ICE:**

“Interactive Connectivity

Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal” si occupa di eseguire controlli di connettività tra due nodi, allo scopo di consentire la comunicazione fra di essi anche quando non sono necessariamente dotati di indirizzi IP pubblici; prova sistematicamente e ordinatamente tutte le possibili coppie di indirizzi finché non ne trova una o più funzionanti per stabilire una sessione di scambio dati (UDP, TCP...)

