Reti Wireless e Mobili

Indice

1. Lezione 01 [16/01]

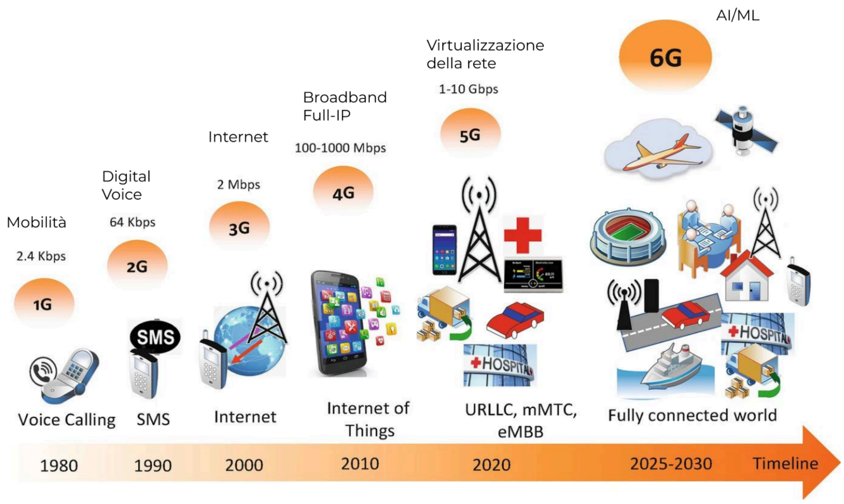
1.1. Introduzione

In questo corso tratteremo dei **dispositivi** che comunicano **wireless** (quindi senza **isolamento**) utilizzando almeno un **hop** ad un punto di accesso alla rete, detto **access point**.

Noi vedremo tre grandi tipologie di dispositivi:

1. **tecnologie a corto raggio**, come il **bluetooth**, che accettano un basso numero di dispositivi e sono indicati per l'uso estremamente locale;
2. **tecnologie wireless ma non mobili**, come il **Wi-fi**, che cerca di dare un alto data rate ma ha una bassa proprietà di **mobilità**;
3. **tecnologie wireless e mobili**, come la **rete cellulare**, che permette il roaming tra access point e la presenza di un alto numero di utenti senza che il servizio e la qualità del servizio cadano, a discapito però di un data rate più basso.

Grazie agli access point siamo in grado di collegarci ai **servizi cloud** ed **edge**, che nient'altro sono che servizi cloud spostati vicino all'utente.



Nella foto precedente vediamo l'evoluzione della rete cellulare di decennio in decennio. In particolare possiamo dire che:

1. in **1G** sono state introdotte le chiamate analogiche in mobilità;
2. in **2G** nasce uno standard globale e si implementa la voce digitale;
3. in **3G** nasce effettivamente il mondo internet;
4. in **4G** viene usata una banda larga in mobilità;
5. in **5G** si ha la virtualizzazione della rete;
6. in **6G** vedremo un uso massiccio di AI e ML.

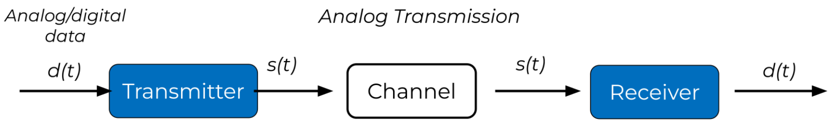
Nel tempo sono cambiati anche i **servizi**. Più andiamo avanti e più la rete si **specializza**: non esiste una soluzione che va bene per tutto, ma ogni tecnologia ha un particolare nel quale spicca.

Le reti wireless e mobili hanno tantissime applicazioni:

1. **IoT e smart city**, che tramite un elevatissimo numero di sensori ed attuatori, gestiti dalla rete, permettono di controllare il traffico e altri aspetti delle città;
2. **guida assistita e autonoma**, che usa reti Wi-fi o mobili molto rapide per permettere una response action quasi istantanea, non potendo cablare le macchine lol;
3. **smart factories**, o Industry 4.0, che hanno un loop simile al precedente.

1.2. Principi di teoria della trasmissione

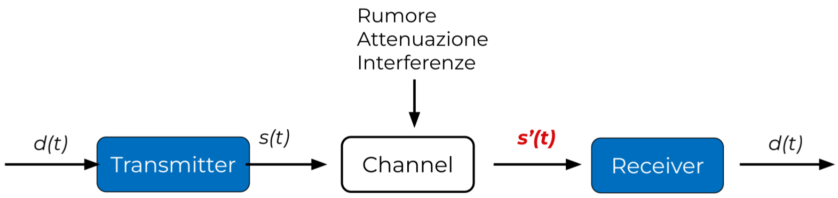
Il modello generale di una **trasmissione dati** è riassunto nel seguente schema.



Nel nostro caso, la trasmissione sarà **analogica**, ovvero tramite **onde elettromagnetiche**.

Come vediamo, un **trasmettitore** codifica i **dati digitali** in ingresso $d(t)$ in **dati analogici** $s(t)$, che vengono poi spediti su un **canale** per raggiungere un **ricevitore**, che deve decodificare il segnale per ottenere nuovamente i dati $d(t)$.

Questo purtroppo è un **mondo ideale**: la situazione reale è quella dell'immagine successiva.



Nella realtà infatti sono presenti **fenomeni** di:

1. **attenuazione**, ovvero un **abbassamento della potenza del segnale** per via di una propagazione prolungata nel tempo;
2. **rumore**, ovvero la presenza di **altri segnali** che si sovrappongono al nostro segnale, come il **rumore termico** o **gaussiano**;
3. **interferenza**, ovvero la **condivisione** dello stesso spettro di frequenze.

Quello che si ottiene è quindi un segnale analogico $s'(t)$, che è ovviamente diverso dal segnale reale che è uscito dall'antenna del trasmettitore. Il nostro compito è capire il segnale $s'(t)$ nel tempo per poterlo ricostruire alla perfezione.

Nelle architetture moderne su cavo il **livello data link affidabile** è in disuso perché si ha una altissima affidabilità su **cavo**, quindi la proprietà di affidabilità si è spostata al **livello di trasporto**.

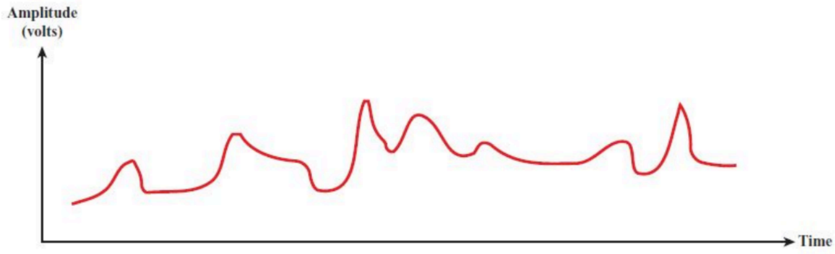
In ambito **wireless** il livello data link fornisce **spesso** la funzionalità di **affidabilità** perché il **canale** è **altamente inaffidabile**:

1. non si ha **protezione**;
2. il mezzo è **totalmente broadcast**;
3. non si possono creare **canali virtuali**.

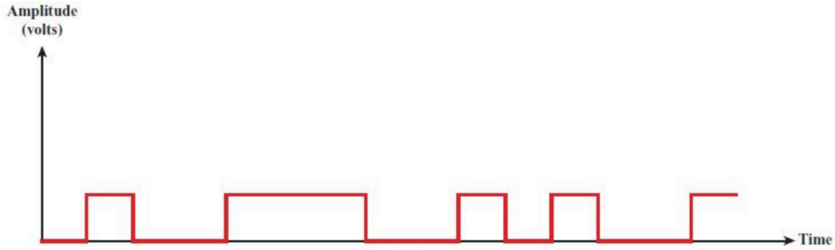
Vedremo quindi tecniche di **ridondanza** e **NACK**, a discapito di un **minore data rate**: infatti, a parità di banda e capacità del canale, il **throughput** via cavo è maggiore di quello wireless.

1.2.1. Dominio del tempo

Possiamo **rappresentare un segnale**, analogico o digitale, utilizzando il **dominio del tempo**: questi grafici mostrano l'ampiezza di questi segnali, in Volt, al variare del tempo.



(a) Analog



(b) Digital

In questi due grafici notiamo come:

1. il **segnale analogico** ha una variazione continua della sua intensità, senza interruzioni e discontinuità;
2. il **segnale digitale** mantiene un livello costante per un determinato periodo di tempo, e poi ha un cambio di livello quasi istantaneo.

I **segnali** che abbiamo quindi sono esattamente il segnale $s(t)$ dello schema precedente.

Un segnale elettromagnetico è un **segnale analogico periodico**, spesso definito come una **sinusoidale**, ovvero

$$s(t) = A\sin(2\pi ft + \varphi) \quad | \quad s(t+T) = s(t) \text{ con } T \text{ periodo} .$$

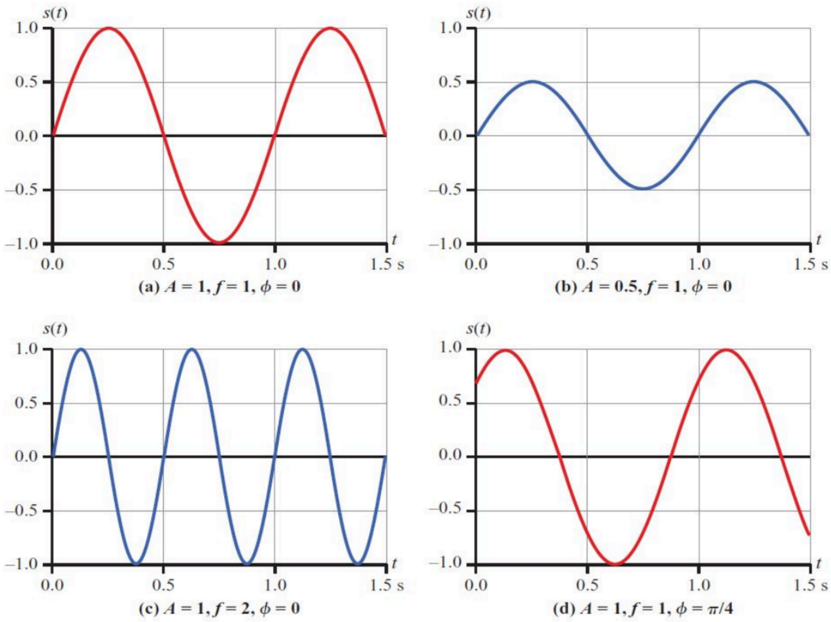
In questa definizione notiamo **tre parametri fondamentali**:

1. **ampiezza** A, ovvero il massimo livello o **forza** del segnale nel tempo, definito in Volt;
2. **frequenza** f, ovvero il numero di cicli al secondo, definito in Hertz;
3. **fase** φ, ovvero la posizione relativa all'interno del periodo.

Dati questi tre parametri possiamo definire due **valori derivati**:

1. **periodo** T, ovvero il tempo impiegato per un ciclo, definito in secondi e calcolato come l'inverso della frequenza;
2. **lunghezza d'onda** λ, ovvero la distanza occupata da un singolo ciclo, definito come $\lambda = Tc$.

Vediamo un esempio di alcune sinusoidali con vari parametri.



Noi dobbiamo giocare con questi parametri per metterci dentro i bit da trasmettere.

1.2.2. Dominio delle frequenze

Il dominio del tempo ci piace, ma si può utilizzare anche il **dominio delle frequenze**.

Ogni segnale periodico può essere **scomposto** in una serie di segnali periodici (onde seno e coseno) con ampiezze, frequenze e fasi differenti. Questo può essere fatto con la **Serie di Fourier**

$$s(t) = \frac{1}{2}c + \sum_{n=1}^{\infty} a_n \sin(2\pi n f t) + \sum_{n=1}^{\infty} b_n \cos(2\pi n f t)$$

dove:

1. f è la **frequenza fondamentale**, definita come inverso del periodo;
2. a_n e b_n sono le ampiezze delle **armoniche** (con $n > 1$);
3. c è la costante che rappresenta il **valore medio** del segnale.

Con questa formula noi possiamo decomporre ogni segnale periodico.

La **trasformata di Fourier** è la funzione

$$\mathcal{F}\{f(t)\}(\omega) = \int_{-\infty}^{\infty} f(t) e^{-i\omega t} \, \mathrm{d}t$$

che permette di ottenere l'ampiezza della frequenze del segnale.

L'**antitrasformata di Fourier** invece è l'inversa della trasformata, definita come

$$f(t) = \frac{1}{2\pi} \int_{-\infty}^{\infty} \mathcal{F}(\omega) e^{i\omega t} \, \mathrm{d}\omega$$

che permette invece di ricavare il segnale dato lo spettro delle frequenze.

Questa funzione è fondamentale: permette di passare dal dominio del tempo al dominio delle frequenze (trasformata) e viceversa (antitrasformata).

Anche se fondamentale, questa funzione presenta alcuni **problemi**:

1. non avendo la nozione di infinito possiamo sì calcolare questa funzione ma dobbiamo introdurre degli **errori**, che però riusciamo a mantenere sotto una certa soglia;
2. la presenza del **rumore** in una **FFT** genera alcune frequenze non richieste con ampiezza non nulla;
3. può presentarsi l'**effetto doppler**, ovvero la frequenze shifta attorno alla frequenza reale.

1.2.3. Campionamento

Per determinare le ampiezze di delle componenti di un segnale abbiamo a disposizione la trasformata di Fourier, mentre per ricavare il segnale usiamo l'antitrasformata.

Un ricevitore deve conoscere questo segnale, quindi deve **campionare** l'antenna. Questo campionamento deve essere fatto in maniera **discreta**, ma il tempo nel quale viviamo è continuo, quindi dobbiamo capire **quando** e **quante volte** campionare. Questo valore determina quanto veloce l'apparato fisico deve lavorare: più va veloce e più serve hardware specializzato.

Abbiamo un risultato utile al nostro problema.

> **Teorema 1.2.3.1** (Teorema di campionamento di Shannon): La frequenza di campionamento deve essere almeno il **doppio** della frequenza massima del segnale in ingresso, campionando a intervalli regolari.

In un segnale periodico, costruito come somma di segnali periodici singoli, il **periodo** è il periodo della frequenza fondamentale f.

Lo **spettro del segnale** (spectrum) è il range di frequenze che lo contiene. La **banda del segnale** (absolute bandwidth) è invece l'ampiezza dello spettro.

1.2.4. Data rate

Vogliamo trasmettere un segnale digitale usando una combinazione di onde sinusoidali. In ogni periodo vogliamo trasmettere α bit. Quello che otteniamo è quindi un **data rate** di αf bit al secondo: infatti, noi mandiamo α bit ad ogni ciclo, ma il numero di cicli è esattamente la frequenza f.

Dovendo approssimare un'**onda quadra** dobbiamo comporre delle armoniche, che però hanno energia più bassa mano a mano che aumentiamo il numero n nella sommatoria. Nonostante ciò, un valore di n alto permette una migliore approssimazione.

La formula che si utilizza per approssimare un'onda quadra è

$$s(t) = \frac{4A}{\pi} \sum_{k=1 \land k \text{ dispari}} \frac{\sin(2\pi k f t)}{k}.$$

1.2.5. Capacità del canale

Definiamo la **capacità del canale** come il **massimo bit rate** al quale è possibile trasmettere dati su un canale di comunicazione in determinate condizioni.

Il **rumore** è un segnale **NON VOLUTO** che si combina al segnale trasmesso, alternandolo e distorcendolo.

L'**error rate** è il **tasso di errore** sui bit. Infatti, a questo livello noi intendiamo il **bit error rate**.

Un **impulso rettangolare** lo si ottiene con una banda infinita, che ovviamente non possiamo avere. Usiamo quindi una **banda finita** molto grande, ma questo porta alcuni problemi:

1. presenza di **rumore** e **distorsione** maggiore in tutta la banda;
2. una banda maggiore non porta per forza un data rate maggiore;
3. **costi economici** elevati;
4. **limitazioni** fisiche e regolamentari del dispositivo.

Il primo risultato che abbiamo per la capacità del canale è la **banda di Nyquist**.

Definizione 1.2.5.1 (Banda di Nyquist): Dato un canale **noise-free**, la banda limita il data rate.

In un **segnale binario** (2 segnali di voltaggio) la capacità vale

$$C = 2B \text{ bit } /s.$$

In un **segnale multi-livello** (M segnali di voltaggio) la capacità vale

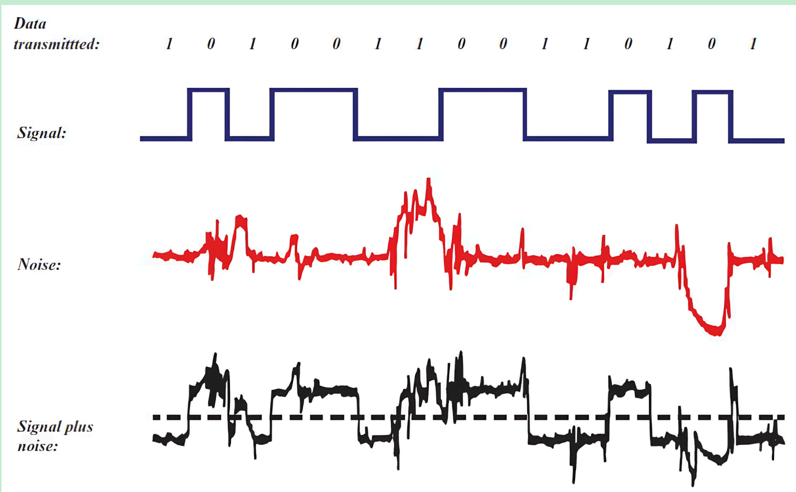
$$C = 2B \log_2(M) \text{ bit } /s.$$

1.2.6. Rumore

Se abbiamo più livelli di voltaggio il **rumore** li può alterare, addirittura cambiandoli. Abbiamo diverse tipologie di rumore:

1. **termico**, che è rumore bianco di fondo, sempre presente;
2. **inter-modulare**, ottenuto durante la modulazione del segnale;
3. **cross talk**, presente quando ci sono dei cavi vicini;
4. **impulso**, rumore esterno rappresentato da una serie di impulsi.

Esempio 1.2.6.1: Vediamo un esempio di come il rumore modifica il segnale.



In questo caso un rumore molto aggressivo compromette due bit del segnale.

Usando invece M livelli di voltaggio questo effetto di distorsione può essere ben peggiore.

1.2.7. Potenza di un segnale

Per misurare il **rapporto tra due potenze** in scala logaritmica usiamo il **Decibel**, definito come

$$\left(\frac{P_1}{P_2}\right)_{\mathrm{dB}} = 10\log_{10}\left(\frac{P_1}{P_2}\right).$$

Quando invece fissiamo il denominatore a 1 mW otteniamo il **Decibel-milliWatt**, che l'unità di misura del rapporto tra una potenza arbitraria e 1 mW.

Usiamo il decibel per descrivere il **rapporto segnale rumore**, o **SNR** (Signal to Noise Ratio). Nyquist non tiene conto del rumore, ma noi ne abbiamo e anche **parecchio**.

Vogliamo sapere quanto il nostro segnale è buono rispetto alla rumorosità del canale, e questo lo possiamo definire come

$$(\mathrm{SNR})_{\mathrm{dB}} = 10\log_{10}\left(\frac{\text{potenza del segnale}}{\text{potenza del rumore}}\right).$$

1.2.8. Ancora capacità del canale

Vediamo un altro risultato teorico sulla capacità del canale.

Definizione 1.2.8.1 (Capacità del canale secondo Shannon): La capacità del canale è la massima capacità teorica di un canale, in bit al secondo, in funzione del SNR, e vale

$$C = B\log_2(1 + \mathrm{SNR}).$$

Questo risultato è **puramente teorico** e considera solo il rumore termico, ma comunque ci dà un limite superiore al data rate che può essere trasmesso senza errori.

Possiamo aumentare il data rate in due modi:

1. **aumentiamo la banda** B, ma il rumore termico è **bianco**, quindi maggiore è la banda e maggiore è il rumore del sistema;
2. **aumentiamo la potenza**, aumentando quindi il SNR, ma la potenza è limitata e si ha la presenza di rumore inter-modulare e cross talk.

1.2.9. Applicazione di Nyquist e Shannon

L'unione delle due formule appena viste ci permette di calcolare i **livelli di voltaggio** che dobbiamo usare per raggiungere un certo data rate.

Esempio 1.2.9.1: Supponiamo di avere a disposizione uno spettro tra 3MHz e 4MHz con un rapporto segnale rumore $\mathrm{SNR}_{\mathrm{dB}} = 24$ dB.

Noi vogliamo sapere quanti **livelli di voltaggio** utilizzare.

$$\begin{aligned} B &= 4 - 3 = 1\mathrm{MHz} \\ \mathrm{SNR}_{\mathrm{dB}} &= 10\log_{10}(\mathrm{SNR}) = 24 \text{ dB} \longrightarrow \mathrm{SNR} = 251 \\ C &= 10^6 \log_2(1 + 251) \approx 8\mathrm{Mbps} \\ C &= 2B\log_2(M) \longrightarrow M = 16. \end{aligned}$$

Esempio 1.2.9.2: Supponiamo di avere a disposizione uno spettro tra 3MHz e 4MHz con un rapporto segnale rumore $\text{SNR}_{\text{dB}} = 12$ dB.

Noi vogliamo sapere ancora quanti **livelli di voltaggio** utilizzare.

$$\begin{aligned} B &= 4 - 3 = 1\text{MHz} \\ \text{SNR}_{\text{dB}} &= 10 \log_{10}(\text{SNR}) = 12 \text{ dB} \longrightarrow \text{SNR} = 10^{1.2} \approx 16 \\ C &= 10^6 \log_2(1 + 16) \approx 4\text{Mbps} \\ C &= 2B \log_2(M) \longrightarrow M = 4. \end{aligned}$$

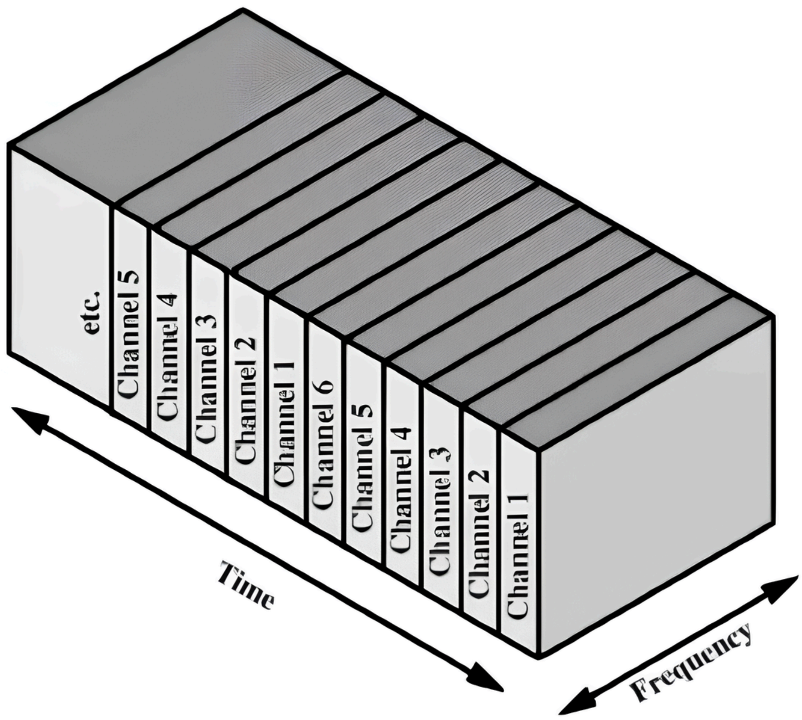
1.2.10. Multiplexing

Finiamo questa lezione con il **multiplexing**.

Molto spesso la capacità del mezzo di trasmissione è molto più grande della capacità della singola comunicazione: per questo vogliamo ottenere più **sotto-canali**, quindi più segnali nello stesso mezzo. Questo ci permette di avere un **maggiore data rate** e un **minore costo** dei bit per secondo.

1.2.10.1. TDM

Il primo multiplexing che si può avere è il **time-division multiplexing**.



Questa divisione sfrutta il fatto che il data rate del mezzo di trasmissione **eccede** il data rate richiesto da un singolo segnale.

Come notiamo, un istante di tempo viene diviso in più canali, nel quale ogni segnale può essere inviato. Lasciamo quindi tutta la banda ma ogni canale ha un tempo limitato per parlare.

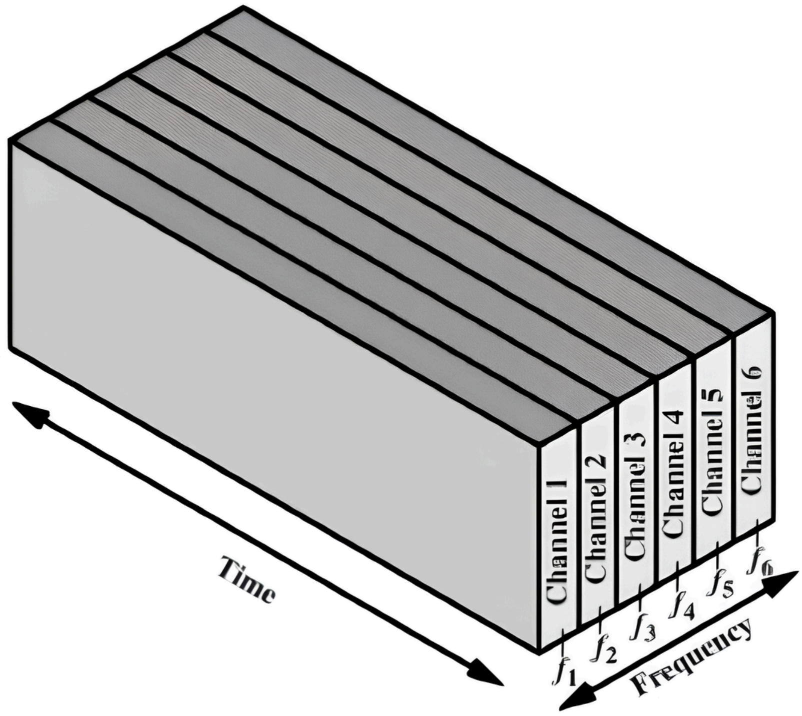
Sinceramente mi sembra solo una divisione più raffinata dell'unità di tempo: se prima davo il canale ad un segnale nell'unità di tempo e mi accorgevo che era troppo, ora do una frazione del tempo al segnale così da sfruttare i tempi morti. Mi sembra quindi una modulazione più raffinata dell'unità di tempo.

In questo multiplexing **non si ha interferenza**, ma è richiesta una **sincronizzazione** e si ha un **uso meno efficiente** della banda.

Per permettere la sincronizzazione si usa spesso una **finestra di delay**.

1.2.10.2. FDM

Il secondo multiplexing che si può avere è il **frequency-division multiplexing**.



In questo caso le «fette» sono opposte: l'unità di tempo è usata per intero, ma la comunicazione avviene su **sotto-bande** diverse.

Si sfrutta infatti il fatto che la banda disponibile sul mezzo di trasmissione **eccede** la banda del singolo segnale per avere il suo data rate.

In questo multiplexing **non serve una sincronizzazione temporale** e si usa la banda in maniera **efficiente**, ma si è suscettibili ad **interferenze** tra canali vicini.

Per evitare fenomeno di interferenza si usa spesso una **banda di guardia**.

2. Lezione 02 [19/01]

2.1. Esercizio

Riprendiamo il secondo esercizio della lezione precedente.

Esempio 2.1.1: Lo spettro di trasmissione va dai 3MHz ai 4MHz, mentre il rapporto segnale-rumore vale $\mathrm{SNR}_{\mathrm{dB}} = 12$ dB.

Le domande che vengono fatte sono tipicamente due:
1. trovare la **capacità del canale**;
2. trovare i **livelli di voltaggio** per avere quella capacità.

Per Shannon e Nyquist la capacità massima è

$$C = B \log_2(1 + \mathrm{SNR})$$
$$C = 2B \log_2(M).$$

Dobbiamo trasformare il SNR senza decibel, quindi

$$\mathrm{SNR} = 10^{\frac{12}{10}} \approx 16.$$

Ma allora per Shannon vale

$$C = 10^6 \log_2(1 + 16) \approx 4\mathrm{Mbps}\ .$$

Ora con Nyquist possiamo calcolare i livelli di voltaggio come

$$M = 2^{\frac{4 \cdot 10^6}{2 \cdot 10^6}} = 4.$$

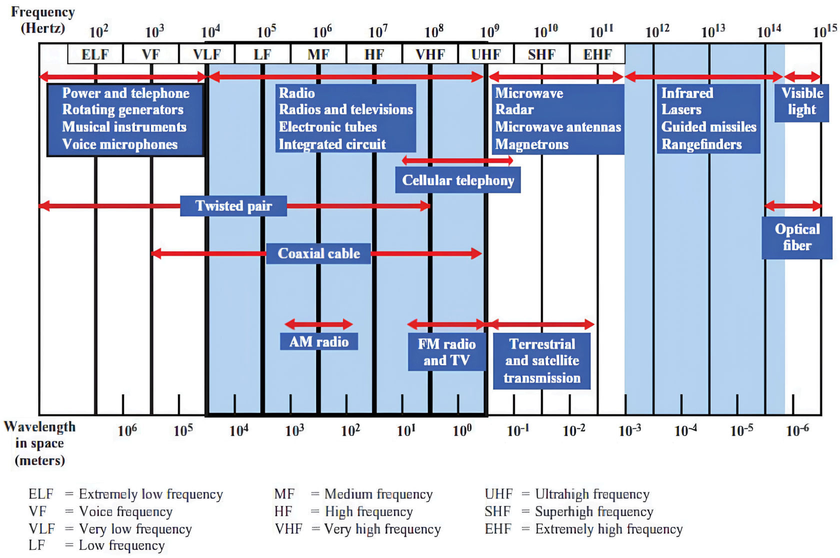
2.2. Comunicazione wireless

2.2.1. Banda base

Le trasmissioni su cavo avvengono in **banda base**, ovvero lo spettro utilizzato per la trasmissione va da 0Hz alla banda massima B.

Esempio 2.2.1.1 (Spettro sonoro): Lo **spettro sonoro** che siamo in grado di sentire va dagli 0Hz (in realtà poco più) fino ai 22MHz, quindi questa è in banda base.

Via cavo questo va benissimo, perché non dobbiamo **sintonizzarci** su un range di frequenze. Sul lato wireless ci sono invece molti **problemi**:
1. se tutti i dispositivi radio usano lo stesso spettro $[0, B]$ tutte le comunicazioni **interferiscono**;
2. più è bassa la frequenza è più l'antenna deve essere **grande**. Si stima che la grandezza deve essere circa la metà della lunghezza d'onda λ per antenne dipole;
3. ogni range di frequenze possiede diverse **proprietà** di propagazione e attenuazione.



In questa immagine vediamo come viene diviso lo **spettro elettromagnetico**.

2.2.2. Banda traslata

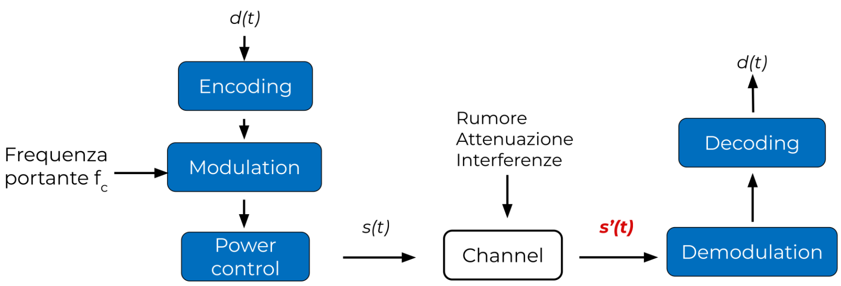
Quello che viene fatto per evitare questa sovrapposizione nelle trasmissioni è la trasmissione in **banda traslata**, o **banda passante**.

Viene scelta una **frequenza carrier**, o **frequenza portante**, e lo spettro da $[0, B]$ viene traslato in

$$\left[f_c - \frac{B}{2}, f_c + \frac{B}{2}\right]$$

dove f_c rappresenta la frequenza carrier.

Come vediamo, la **bandwidth** è mantenuta, avendo effettuato una traslazione delle frequenze massima e minima. Inoltre, manteniamo lo **stesso data rate** di partenza.



Come vediamo, dopo l'**encoding** (che vedremo dopo) prima avviene una **modulazione** con la frequenza portante, modificando i **tre parametri** base di una sinusoide, e poi un'**amplificazione**.

La modulazione è di tre tipi:

1. **amplitude modulation**, che modifica l'ampiezza (come nelle radio AM);
2. **frequency modulation**, che modifica la frequenza (come nelle radio FM);
3. **phase modulation**, che modifica la fase.

Lato ricevente dobbiamo invece fare una **demodulazione**, togliendo la frequenza portante dal segnale, e una volta tornati in **banda base** possiamo eseguire la **decodifica**.

2.2.3. Simboli

Un **simbolo** è una forma d'onda, uno stato (livello di voltaggio) o una condizione significativa del canale di comunicazione che persiste per un intervallo di tempo fissato. Non è rumore, è un qualcosa che ha **significato**.

Il **symbol rate** è il numero di simboli trasmessi al secondo dal livello fisico, misurato in **baud**.

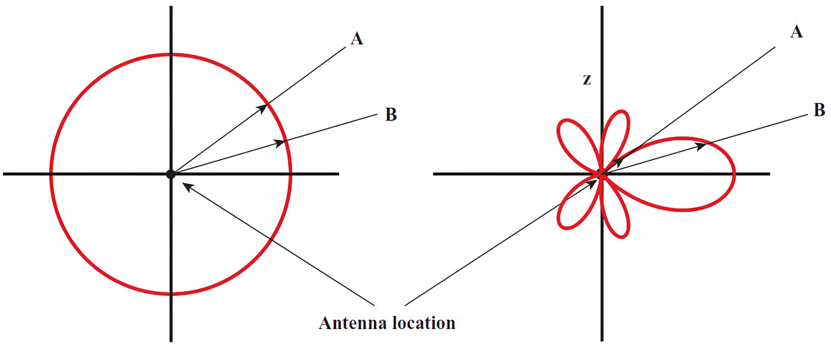
In generale un simbolo può contenere più bit, quindi il symbol rate è **diverso** dal bit rate. Vedremo in particolare che il bit rate **non è peggiore** del symbol rate, e questi due valori sono uguali quando il livello fisico può produrre solo due segnali.

Una data **bandwidth** può supportare diversi data rate, a seconda dell'abilità del ricevente di distinguere 0 e 1 in presenza di rumore. Infatti, un simbolo può **codificare più bit** alla stessa frequenza.

2.2.4. Propagazione delle onde radio

Dato per assodato che la **terra è tonda**, le onde radio si **propagano** in tre modi diversi:
1. sotto i 2MHz il segnale viaggia seguendo la **curvatura terrestre**, anche se i due ricevitori non si vedono;
2. nel range $[2, 30]$MHz il segnale viene **riflesso** dalla ionosfera;
3. sopra i 30MHz, dove ci muoveremo noi, necessitiamo della **Line of Sight** (LoS), ovvero i due ricevitori si devono vedere per poter parlare.

Un altro aspetto da controllare è l'**antenna**.



A sinistra abbiamo un'**antenna omnidirezionale**, ovvero un'**antenna ideale**, che però non è sempre voluta. A destra invece abbiamo un'**antenna direzionale**, che ha un grande **lobo** che punta in **una direzione** e altri piccoli lobi per coprire le altre direzioni.

Tendenzialmente le antenne direzionali sono quelle usate per le **comunicazioni** perché concentrano l'energia in una certa direzione, ovvero la direzione LoS.

2.2.5. Trasmissione LoS

La **trasmissione radio LoS** presenta molti problemi:
1. **free space loss** e **path loss**, che abbiamo anche su cavo (il path loss ovviamente), ed è una **attenuazione del segnale** dovuta alla distanza e all'ambiente in cui il segnale si propaga;
2. **rumore**, al quale siamo sempre sensibili visto che non abbiamo protezione;
3. **multipath**, che grazie a fenomeni di **riflessione**, **diffrazione** e **scattering** causa la ricezione di più onde dello stesso segnale in tempi diversi;
4. **effetto doppler**, ovvero si ha una variazione del segnale a causa del **movimento** di TX, RX e ostacoli; una velocità ampia porta una differenza ampia.

2.2.5.1. Path loss

Il **path loss** è l'**attenuazione del segnale radio** in funzione della **distanza** tra RX e TX, ed è definito come

$$\frac{P_t}{P_r} = \left(\frac{4\pi}{\lambda}\right)^2 d^n = \left(\frac{4\pi f}{c}\right)^2 d^n.$$

Questo rapporto tra la **potenza del segnale trasmesso** e la **potenza del segnale ricevuto** va ad indicare quanta potenza è stata persa. Ovviamente, il valore P_t è di solito maggiore di P_r.

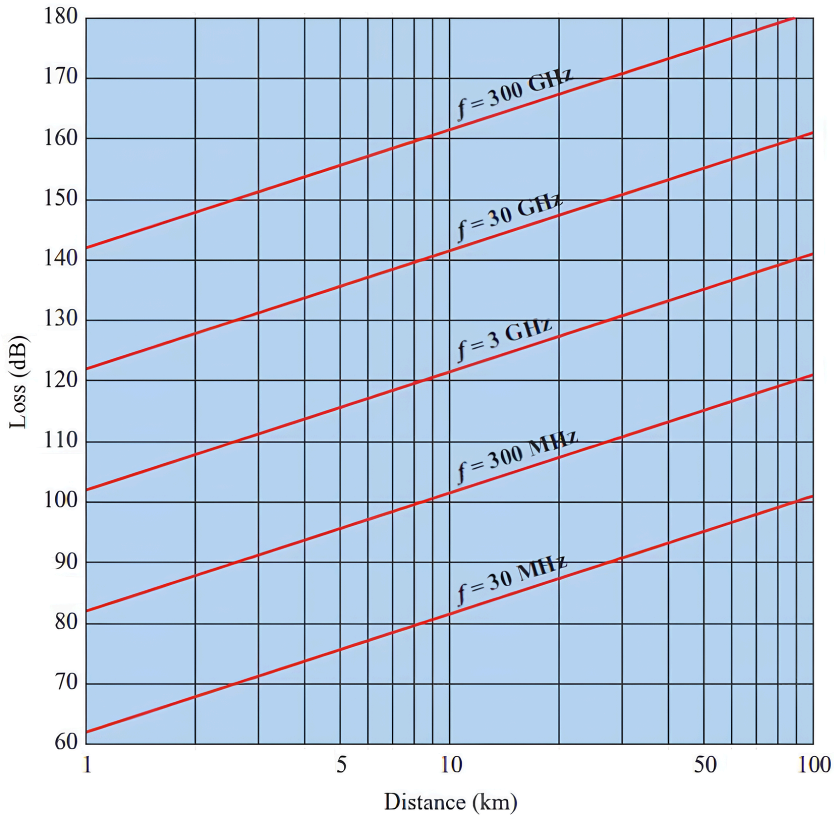
Notiamo che questo rapporto:

1. è **direttamente proporzionale** al quadrato della frequenza;
2. è **direttamente proporzionale** ad una potenza della distanza, e questa dipende dall'ambiente.

Facciamo qualche confronto:

1. a parità di **potenza**, abbiamo **maggiore copertura** se abbiamo delle **frequenze più basse**, visto che abbiamo meno effetto di path loss;
2. a parità di **distanza**, una frequenza più alta comporta un path loss più alto.

Spesso è comodo definire il path loss L in **decibel**.



In questa immagine vediamo quanti decibel perdiamo, indicati sull'asse y, data una certa distanza tra TX e RX, indicata sull'asse x.

Il **gain**, o **guadagno**, di un'antenna è definito come il **rapporto** tra l'intensità della radiazione elettromagnetica in una data direzione e l'intensità che si avrebbe se si usasse un'antenna isotropica.

Le **antenne isotropiche** sono le antenne ideali (e allora chiamale ideali).

Il gain si indica con G ed è misurato in **decibel isotropici**, indicati con dBi.

Questa quantità ci aiuta con il path loss: avendo delle antenne direzionali noi stiamo concentrando l'energia in una certa direzione, quindi dal path loss dobbiamo **togliere** alcune quantità.

Il nuovo path loss, non misurato in decibel, diventa

$$\frac{P_t}{P_r} = \frac{(4\pi f)^2}{G_{\text{tx}} G_{\text{rx}} c^2} d^n$$

che poi trasformati in **decibel** ci dà una perdita pari a

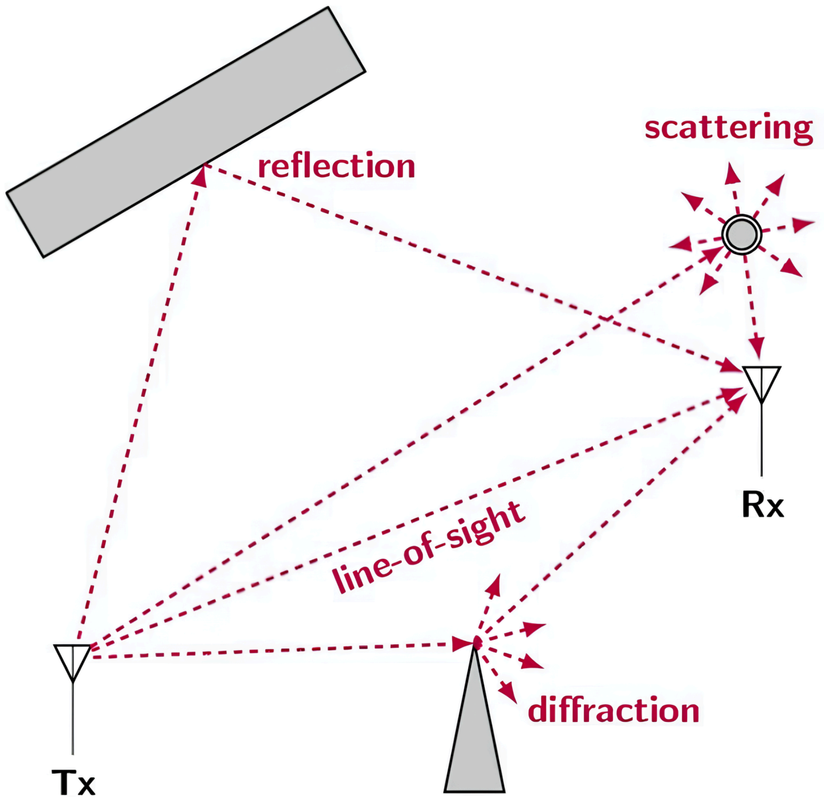
$$L_{\text{dB}} = 10 \log_{10}\left(\frac{P_t}{P_r}\right) = 20\left(\log_{10}\left(\frac{4\pi f d}{c}\right) - \underbrace{\log_{10}(G_{\text{tx}})}_{\text{gain tx}} - \underbrace{\log_{10}(G_{\text{rx}})}_{\text{gain rx}}\right).$$

Questi conti sono ovviamente a parità di distanza e ambiente free space: con le antenne direzionali abbiamo un **path loss minore**.

2.2.5.2. Multipath

Il **multipath** si presenta quando l'ambiente è **complesso** e possono presentarsi effetti di:

1. **riflessione** del segnale;
2. **scattering**, che spara il segnale in tutte le direzioni perché la lunghezza d'onda del segnale è simile a quella dell'oggetto;
3. **diffrazione**, che è come lo scattering ma avviene sui bordi perché la lunghezza d'onda del segnale è molto più piccola di quella dell'oggetto.



Come vediamo, per fare da TX a RX abbiamo il segnale **LoS** ma anche molti altri percorsi, dovuti agli effetti appena presentati.

Il multipath può provocare due **effetti fastidiosi**: il fading e l'interferenza inter-simbolo.

2.2.5.2.1. Fading

Il **fading**, o evanescenza, avviene quando si ha **interferenza distruttiva** tra più onde elettromagnetiche. Abbiamo due tipi di interferenza: **costruttiva**, che va ancora ancora bene, e **distruttiva**, che è fastidiosa perché abbassa i picchi e crea un'onda che non c'entra niente con quella di partenza.

Per risolvere questo problema dobbiamo garantire un **tempo di coerenza**, ovvero una scala temporale in cui possiamo considerare «costanti» le caratteristiche del segnale. Questo tempo di coerenza è tale che

$$T_c = \frac{1}{f_D}.$$

Questo valore dipende dalla **frequenza doppler**, basata sulla velocità di movimento (di chi non lo so) e sulla frequenza, ed è tale che

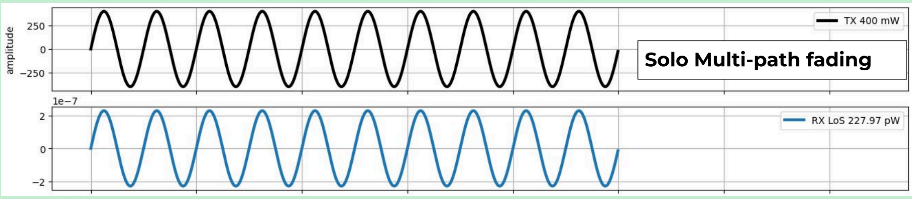
$$f_D = \left(\frac{v}{c}\right) f_c.$$

Se abbiamo alta velocità e alta frequenza allora abbiamo un periodo molto basso, e quindi dobbiamo **campionare** più spesso il segnale.

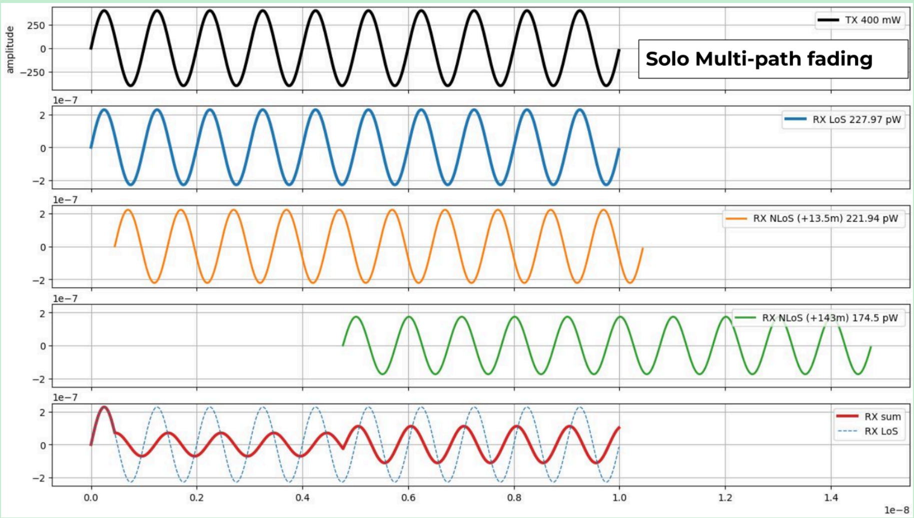
Nel prossimo esempio vediamo una serie di segnali che vanno incontro al problema del fading.

Esempio 2.2.5.2.1.1 (Fading): Allunghiamo per bene questi appunti.

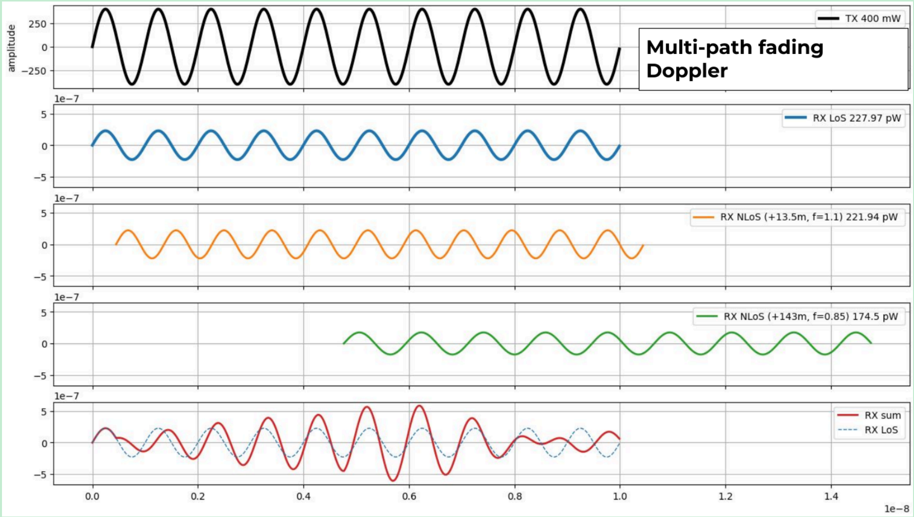
Nella prima immagine vediamo l'effetto del path loss, che rende più debole il segnale.



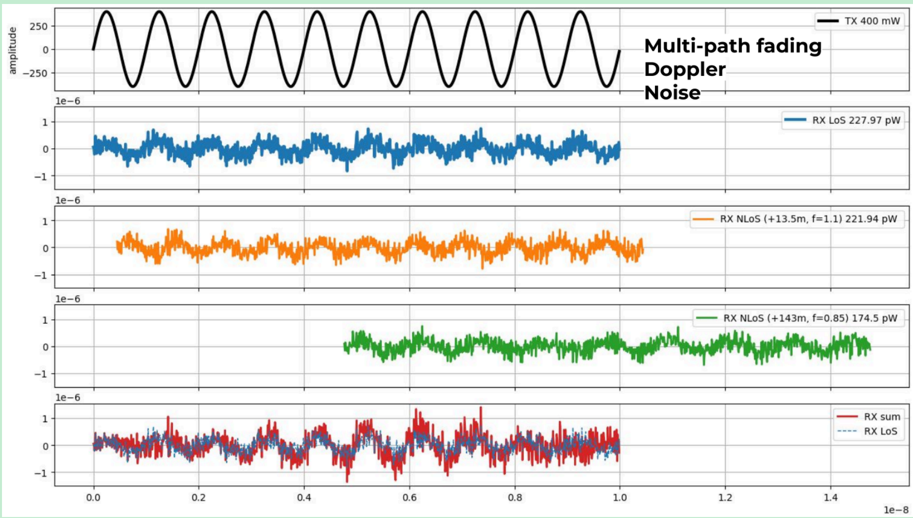
Nella seconda immagine abbiamo invece l'effetto del fading con 2 path che non sono LoS.



Nella terza immagine aggiungiamo l'effetto doppler ai segnali precedenti.



Infine, nell'ultima immagine aggiungiamo anche il rumore.



2.2.5.2.2. Interferenza inter-simbolo

L'**interferenza inter-simbolo** (ISI) avviene molto spesso in ambito mobile.

Questo fenomeno si presenta come una **ricezione sovrapposta** di simboli adiacenti a causa del ritardo di ricezione delle onde del primo simbolo. Arabo vero? In realtà no.

Se l'intervallo di tempo tra un simbolo e l'altro è molto breve può succedere che se le onde non LoS del primo simbolo arrivino al RX nello stesso momento in cui arrivano le onde LoS del secondo simbolo.

`image`

Siamo così sfigati che interferiamo con **noi stessi**.

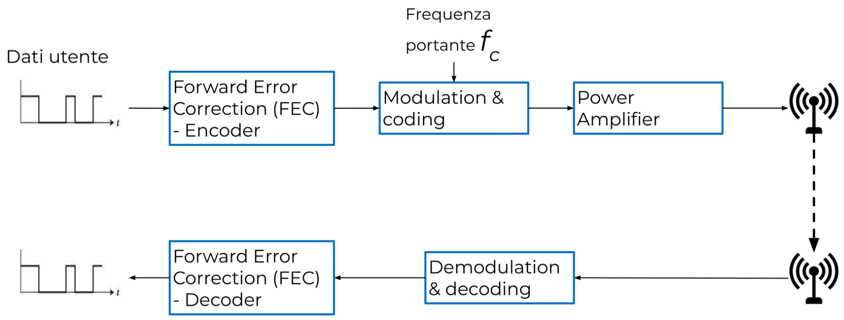
Se siamo **distanti** questo effetto è molto presente, quindi per risolverlo bisogna **aumentare la distanza tra i simboli**, a discapito di un minor data rate. Se invece siamo **vicini** l'effetto è poco presente, quindi possiamo tenere i simboli più vicini e quindi **aumentare** il data rate.

2.2.6. Sistemi multi-antenna

I **MIMO** sono dei sistemi multi-antenna, fine, non serve sapere altro.

2.2.7. Codifica e trasmissione dei dati

Nella seguente immagine vediamo lo **schema della trasmissione radio**.



Il nostro segnale digitale prima passa nel blocco **FEC** con l'**encoder**, poi viene **modulato** sulla frequenza portante e infine viene **amplificato**. Una volta che questo viene spedito sul canale il ricevitore deve **demodulare** il segnale e farlo passare ancora in un blocco FEC con una **decodifica**.

Come vedremo, i blocchi FED e di modulazione sono **dinamici**, ovvero dipendono dal canale.

Ricordiamo che il segnale è una sinusoide

$$s(t) = A \sin(2\pi f_c t + \varphi).$$

2.2.7.1. Codifiche semplici

Esistono diverse tecniche per **codificare** i dati digitali in segnali analogici:

1. **Amplitude-Shift Keying** (ASK): usiamo diversi livelli di ampiezza A per diversi bit;
2. **Frequency-Shift Keying** (FSK): usiamo diverse frequenze f per diversi bit;
3. **Phase-Shift Keying** (PSK): usiamo diverse fasi φ per diversi bit.

Esempio 2.2.7.1.1: Vogliamo codificare 1 per ogni simbolo.

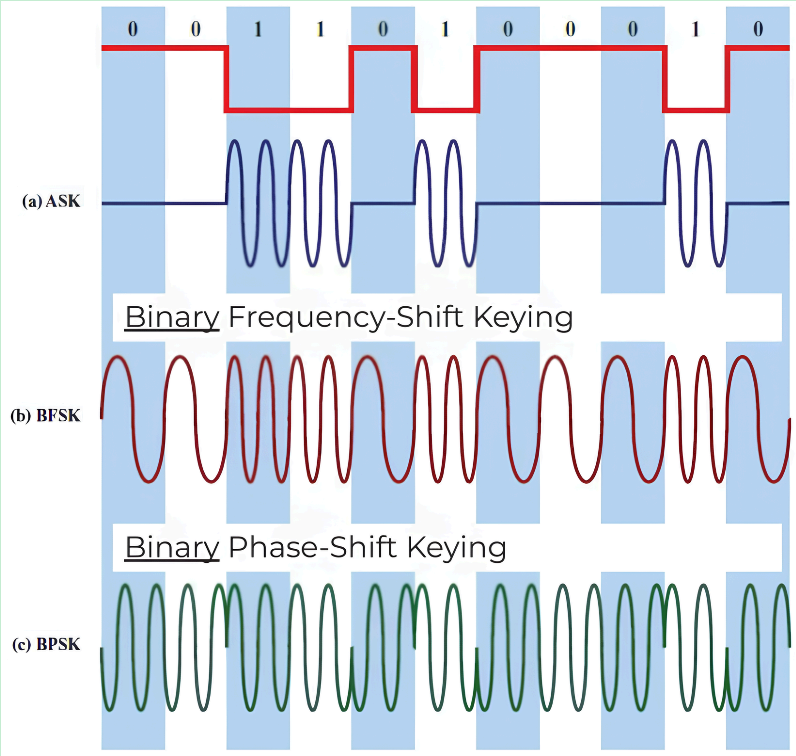
Vediamo i tre segnali che potremmo usare per questo scopo.

$$s(t) = \begin{cases} A \sin(2\pi f_c t) & \text{se } 1 \\ 0 & \text{se } 0 \end{cases}$$

$$s(t) = \begin{cases} A \sin(2\pi f_1 t) & \text{se } 1 \\ A \sin(2\pi f_2 t) & \text{se } 0 \end{cases}$$

$$s(t) = \begin{cases} A \sin(2\pi f_c t) & \text{se } 1 \\ A \sin(2\pi f_c t + \pi) & \text{se } 0 \end{cases}.$$

Vediamo un grafico di come sono fatti questi segnali.



Come vediamo, i primi due segnali sono ok, ma «sentire» un cambiamento in queste onde è abbastanza difficile se non si ha un segnale di ottima qualità. Il terzo segnale invece è pieno di **interruzioni di fase**, che sono molto semplici da vedere e sentire.

Una versione alternativa del robusto phase-shift è il **Differential Phase-Shift Keying** (DPSK), che non ha una codifica fissa ma **variabile**.

Ogni volta che leggo uno 0 **mantengo la fase**, mentre ogni volta che leggo un 1 la fase viene **shiftata** di 180°. Questa tecnica è molto comoda perché non richiede un allineamento preciso e si **identifica** facilmente.

2.2.7.2. Codifiche sofisticate

Esistono però delle codifiche più **sofisticate**, che permettono di trasmettere più di un bit per simbolo:

1. **MFSK** o Multilevel Frequency-Shift Keying, nel quale la **M** del nome indica il numero di livelli di voltaggio, con i quali codifichiamo $L = \log_2(M)$ bit;
2. **QPSK** o Quadrature Phase-Shift Keying, che codifica 2 bit per simbolo;
3. **X-QAM** o Quadrature Amplitude Modulation, nel quale la **X** del nome permette di ricavare il numero di bit codificati come $L = \log_2(X)$.

2.2.7.2.1. QPSK

Con **QPSK** riusciamo a mandare 2 bit per ciascun simbolo usando un segnale

$$s(t) = \begin{cases} A\cos\left(2\pi f_c t + \frac{\pi}{4}\right) & \text{se } 11 \\ A\cos\left(2\pi f_c t + \frac{3\pi}{4}\right) & \text{se } 01 \\ A\cos\left(2\pi f_c t - \frac{3\pi}{4}\right) & \text{se } 00 \\ A\cos\left(2\pi f_c t - \frac{\pi}{4}\right) & \text{se } 10 \end{cases}$$

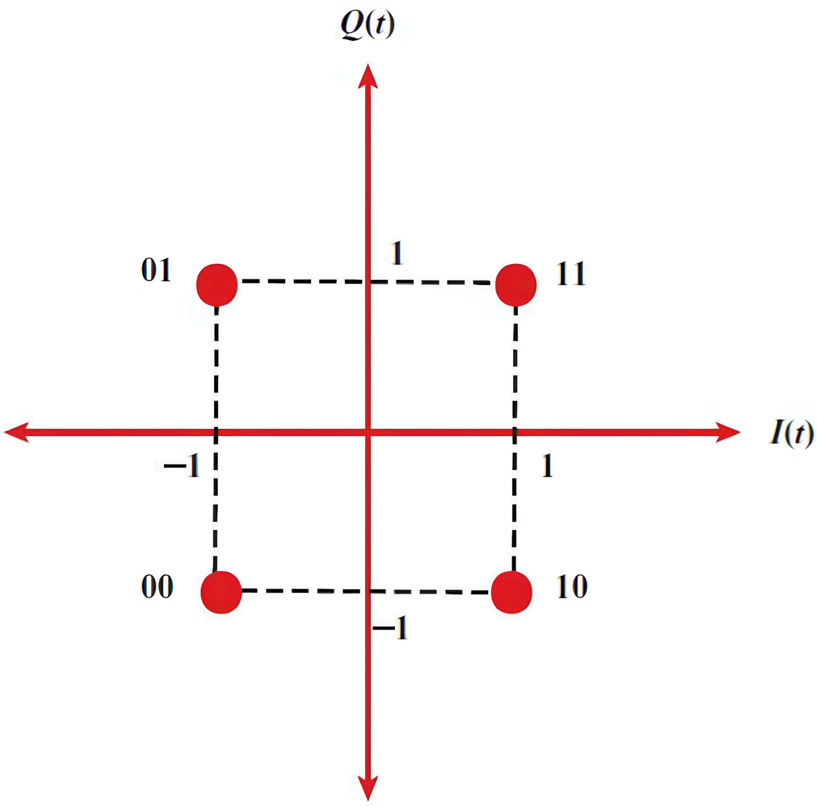
formato da 4 fasi diverse distanziate di 90° e usando una codifica Gray per i punti adiacenti.

Questo segnale può essere «compresso» in una formula unica

$$s(t) = \frac{1}{\sqrt{2}} I(t) \cos(2\pi f_c t) - \frac{1}{\sqrt{2}} Q(t) \sin(2\pi f_c t)$$

che dipende dai valori $I(t)$ e $Q(t)$.

Questi valori si ricavano dal **digramma della costellazione**.



Quando vogliamo trasmettere un valore **AB** dobbiamo ricavare i valori di $I(t)$ e $Q(t)$ dalla costellazione, usando A per il valore $I(t)$ e B per il valore $Q(t)$.

In fase di ricezione facciamo l'operazione inversa, ovvero riceviamo un punto del piano, che va **mappato** nel punto più vicino della costellazione, usato per ricavare i due bit trasmessi.

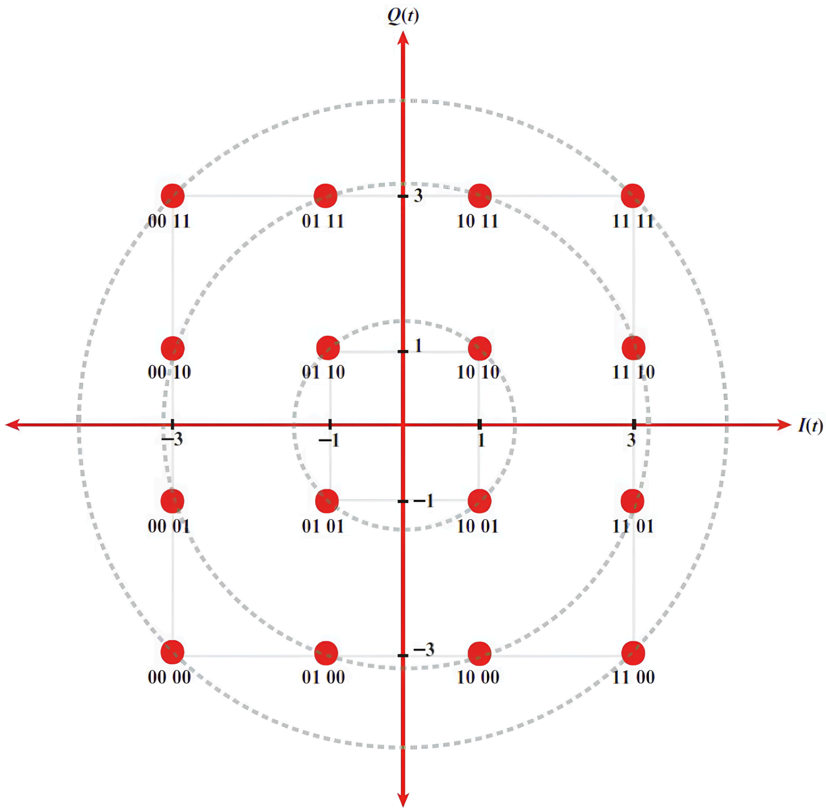
Si chiama **quadratura** perché le due sinusoidi sono shiftate di 90°.

2.2.7.2.2. X-QAM

In maniera simile possiamo definire il segnale di **X-QAM** come

$$s(t) = I(t) \cos(2\pi f_c t) - Q(t) \sin(2\pi f_c t).$$

In questo caso però stiamo combinando **variazioni di ampiezza** e **fase**. Per ogni punto noi dobbiamo capire su quale **circonferenza** ci troviamo (ampiezza) e, successivamente, in che **punto** siamo (fase).



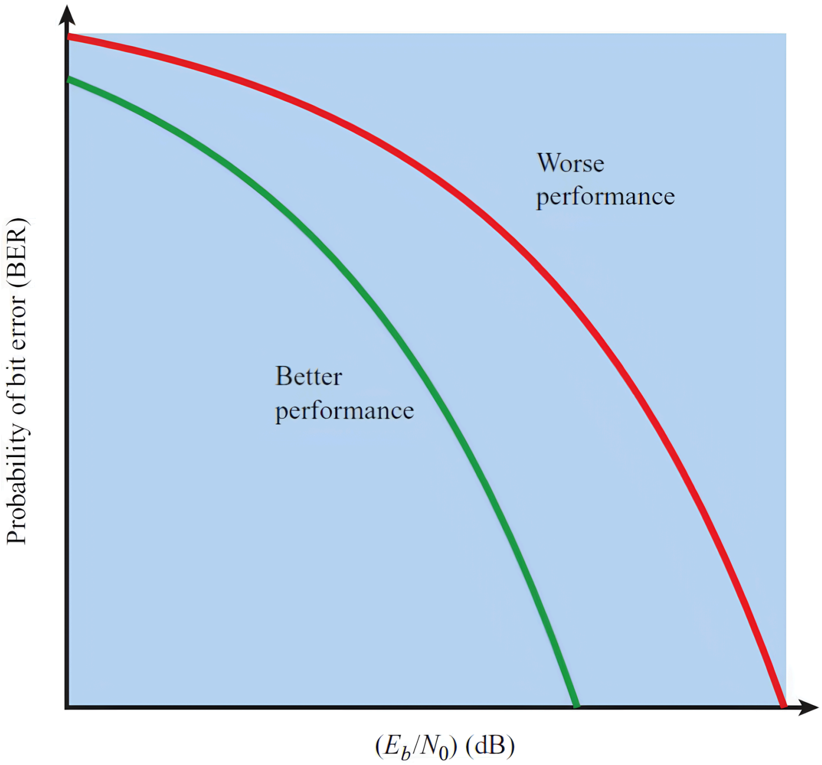
Come vediamo, la costellazione (questa è di 16-QAM) ora è molto più **densa** di prima. Come abbiamo fatto con QPSK, la prima metà dei bit è usata per $I(t)$ mentre la seconda metà dei bit è usata per $Q(t)$.

2.2.7.2.3. Confronto tra codifiche

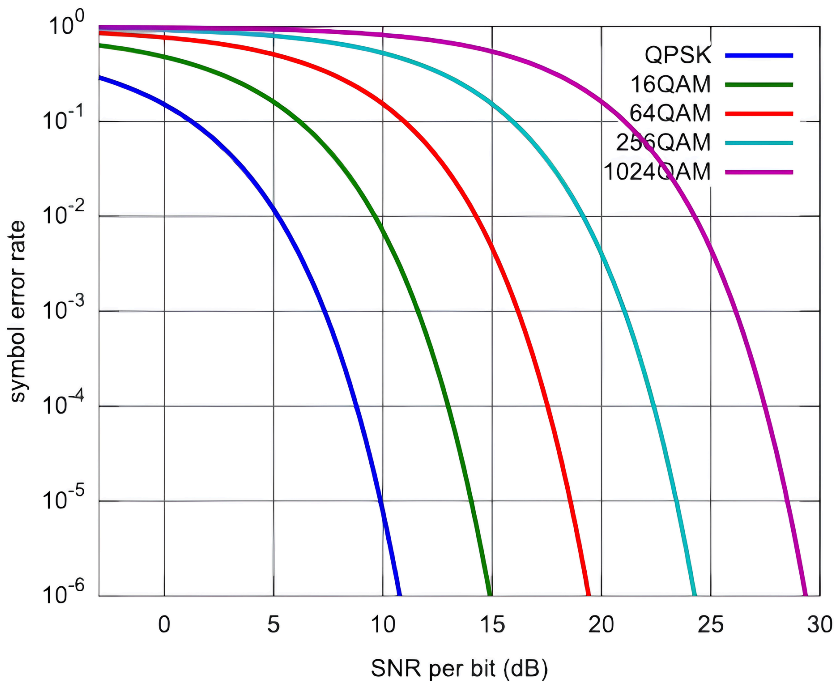
La codifica X-QAM in generale ha un **data rate maggiore** rispetto a QPSK perché usiamo **meno simboli** per codificare gli stessi dati, quindi nell'unità di tempo ci stanno più simboli di X-QAM che di QPSK.

La soluzione sorge spontanea: carichiamo tantissimi bit per simbolo, così abbiamo un data rate altissimo e abbiamo una comunicazione fenomenale.

Questo non si può fare, e lo possiamo dimostrare con le **curve di BER** (Bit Error Rate). Queste curve rappresentano la **probabilità di errore di un bit** in funzione del rapporto tra la densità di energia del segnale per bit ed il livello di rumore.



Mano a mano che il canale migliora noi abbassiamo la probabilità di errore, ma non tutte le codifiche lo fanno allo stesso modo e con la stessa velocità.



Come vediamo, le codifiche più dense, a parità di probabilità di errore, richiedono una **qualità del canale** molto più alta.

Per questo noi dobbiamo cercare un **compromesso** tra codifica e qualità del canale: se la qualità è bassa e usiamo una codifica densa allora rischiamo di sbagliare il centroide della costellazione.

In poche parole usiamo uno schema **Adaptive Modulation and Coding** (AMC).

2.2.7.3. Forward Error Correction

Finiamo con la **Forward Error Correction** (FEC).

Se una funzione di **error detection** (lato ricevente) identifica la presenza di un errore allora il blocco dati viene ritrasmesso. Il problema è che nel mondo **wireless** sbagliare un bit è all'ordine del giorno, **è più facile sbagliare che farlo giusto**, quindi rischiamo di trasmettere lo stesso blocco all'infinito.

Per evitare questo loop infinito usiamo una **forward error correction**: andiamo ad abbassare il data rate aggiungendo della **ridondanza** tramite una serie di bit, che devono essere usati per verificare se ci sono stati errori.

Una misura di quanta ridondanza stiamo inserendo è il **coding rate**, definito come

$$\mathrm{CR} = \frac{k}{n}$$

dove k è il numero di **bit utili** e n è il numero di bit totali.

A seconda delle **condizioni del canale** noi dobbiamo scegliere

1. quale **codifica** utilizzare;
2. quale **coding rate** utilizzare.