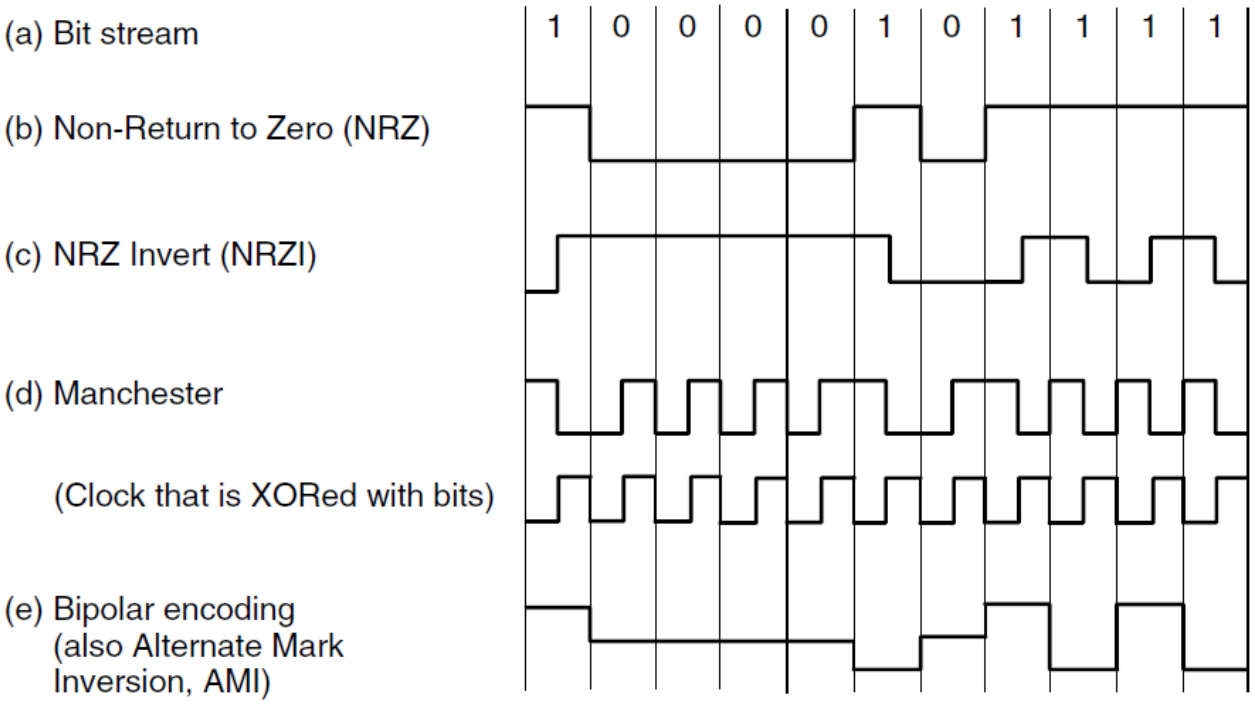
MODULAZIONE DIGITALE

TRASMISSIONE IN BANDA BASE



NRZ (Non-ritorno-a-zero) b: è la forma più basilare di modulazione digitale. Principalmente il segnale segue l'andamento dei dati, e, nella forma più tradizionale una tensione positiva è interpretata come 1 e una negativa come 0. Ovviamente al ricevente arriverà un segnale distorto e attenuato, ma in base al simbolo di tensione più vicino può interpretare correttamente i bit.
È una strategia semplicissima, che ovviamente presenta alcuni problemi:

- Il segnale potrebbe oscillare ogni 2 bit tra valori positivi e negativi (alternanza di 0 e 1). Abbiamo quindi bisogno di una banda di almeno B/2 HZ se il tasso di invio dei bit è B bit/s. Non possiamo fare andare NRZ più veloce di così.
- Con NRZ, una lunga sequenza di 0 o 1 lascia invariato il segnale! A meno di non disporre di un clock molto preciso (che spreca tante risorse nel caso di alta velocità), diventa difficile distinguere i bit dopo un po'.

Manchester d: questa codifica è più strategica. Ci siamo accorti che con NRZ ci vuole un clock per non fare confusione.. vediamo come implementarlo: mandare un messaggio contentente il nostro clock sprecherebbe banda (potremmo usare la linea che usiamo per il clock per spedire dati!), quindi un'idea è mandare il clock in XOR con i dati senza creare una seconda connessione. Il ricevente è in grado di distinguere i dati e il clock indipendentemente con lo stesso segnale perchè quando il clock è messo in XOR con un segnale codificato con NRZ semplicemente lo 0 è una transizione dal basso verso l'alto (che è il clock stesso) e l'1 invece sarà una transizione dall'alto verso il basso. Il problema è:

- a casa del clock richiede il doppio della banda per la stessa quantità di dati che potrebbe essere spedita con NRZ.

NRZI (Non-ritorno-a-zero invertito) c: abbiamo visto che le "non transizioni" sono un problema non da poco. Questa strategia prevede di codificare i dati in modo tale da avere abbastanza transizioni nel segnale.
Il NRZI codifica un 1 come una transizione e uno 0 come una situazione stazionaria (tensione alta o bassa non importa), in questo modo lunghe sequenze di 1 non causano problemi ma resta ancora il problema di lunghe sequenze di zeri.

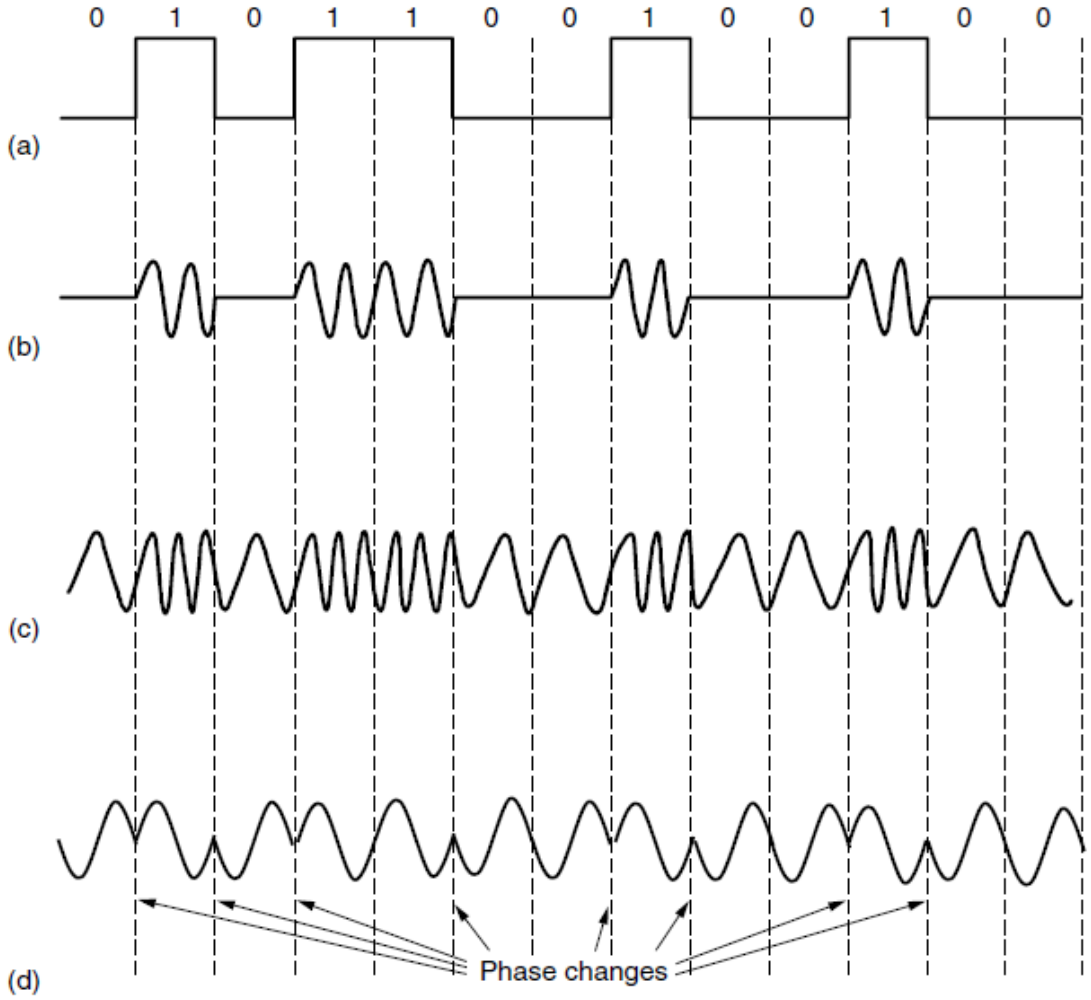
Come risolvere il problema definitivamente? Potremmo spezzare lunghe sequenze di zeri ripetuti assegnando loro piccoli gruppi di bit da trasmettere: in questo modo lunghe sequenze di 0 sono associate a sequenze leggermente più lunghe che però non hanno troppi 0 ripetuti.
La codifica **4B/5B** è la più famosa e associa a ogni combinazione di 4 bit una combinazione di 5 bit con pochi 0 ripetuti (visto che vengono usate solo 16 combinazioni di 5 bit vengono scelte proprio le migliori possibili, le restanti, sempre escludendo quelle con troppi 0, possono essere usate come segnali di controllo per il livello fisico). L'overhead è del 25% che è meglio del 100% della codifica manchester.

Scrambling: questa forma di modulazione funziona un po' come l'algoritmo stream cyper mode della crittografia. Uno scrambler opera applicando, prima della trasmissione, un'operazione di XOR tra i dati e una sequenza di numeri pseudocasuali: in questo modo i dati sembreranno casuali quanto la sequenza casuale assegnata alla trasmissione. Il ricevente DEVE AVERE la stessa sequenza di numeri pseudocasuali e applicare uno XOR al messaggio per riottenere i dati. Questa tecnica non aggiunge alcun overhead. Problemi:

- Niente assicura che non compaiano lunghe sequenze di 0 ripetute perchè quando si ha a che fare con numeri casuali può sempre capitare di essere sfortunati

Codifica bipolare e: Un segnale è bilanciato quando la media della tensione è 0. Il bilanciamento aiuta ad aggiungere transizioni per il clock recovery in quanto siamo in presenza di una mescolanza di valori positivi e negativi.
Questa codifica è un modo diretto per formulare un codice bilanciato: usa due livelli di tensione per rappresentare un 1 (per esempio +1 V e -1 V) e nessun livello per rappresentare uno 0 (0 V appunto) in maniera tale che questi ultimi si annullino in media. Anche con questa codifica è utile utilizzare una codifica di linea come 4B/5B o 8B/10B.

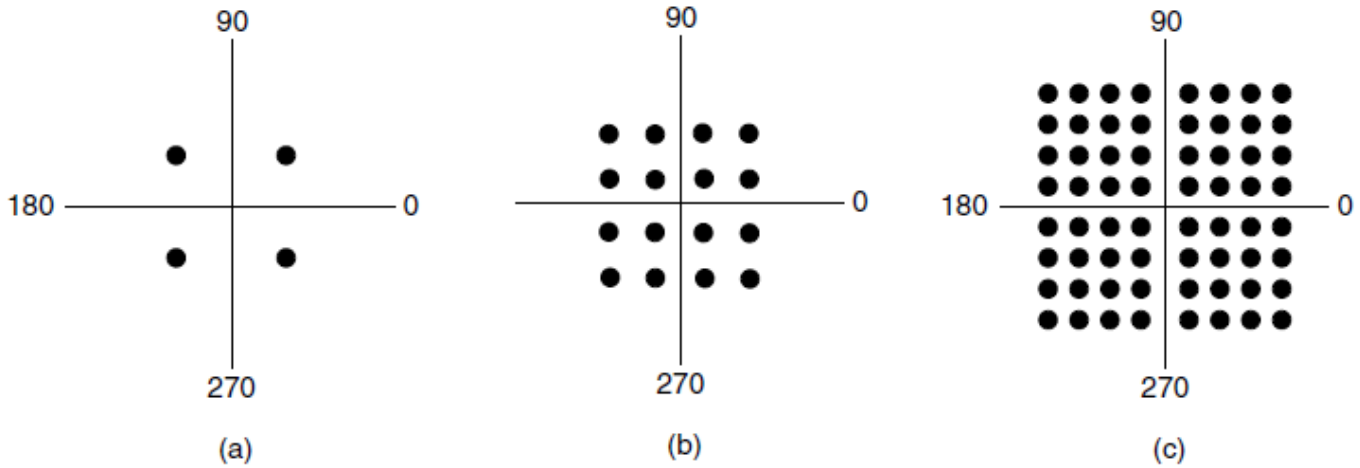
TRASMISSIONE IN BANDA PASSANTE



Una banda di frequenza arbitraria può essere usata per far passare il segnale e trasmettere dati. Possiamo modulare l'ampiezza, la frequenza o la fase del segnale portante e ognuno di questi metodi ha un suo nome:

ASK (Amplitude shift keying) b: usa la MODULAZIONE DI AMPIEZZA. Con due diverse ampiezze della forma d'onda vengono rappresentati lo 0 el'1.
FSK (Frequency shift keying) c: usa la MODULAZIONE DI FREQUENZA. Con due diverse frequenze della forma d'onda vengono rappresentati lo 0 e l'1.
PSK (Phase shift keying) b: usa la MODULAZIONE DI FASE. L'onda portante è sistematicamente traslata di 0 o 180 gradi all'inizio della trasmissione di uno 0 o un 1. In questo caso, essendoci due fasi, il sistema viene chiamato BPSK (binary phase shift keying).

Schemi più efficienti:



QPSK (quadrature phase shift keying) a: è uno schema migliore e più efficiente dal punto di vista della banda. Fa uso di 4 traslazioni di una forma d'onda (usa quindi la modulazione di fase): a 45, 135, 225 e 315 gradi per trasmettere 2 bit di informazione con un unico simbolo (2^2 = 4 possibili combinazioni con 2 bit).

Possiamo combinare gli approcci descritti in precedenza per ottenere più simboli e rappresentare quindi più bit con un simbolo solo. Tipicamente moduliamo in maniera combinata l'ampiezza e la fase (visto che frequenza e fase sono impossibili da combinare assieme). Nel diagramma a costellazione del QPSK si può osservare che la fase è rappresentata dalla retta che congiunge un punto all'origine dell'asse (grado della circongerenza goniometrica), mentre la distanza dall'origine rappresenta l'ampiezza della forma d'onda. Questo tipo di codifiche prendono il nome di QAM(quadrature amplitude modulation).

QAM-16 b: usa 16 combinazioni di ampiezza e fase per ottenere 4 bit per simbolo.
QAM-64 c: usa ben 64 combinazioni di ampiezza e fase per ottenere 6 bit per simbolo.

Gray code (codifica di Gray): è una tecnica che si basa sulla "vicinanza" dei vari simboli che codificano i bit. In pratica vengono scelti bit in modo tale che simboli adiacenti differiscano di un solo bit. Se il ricevente commette un errore di decodifica su di un simbolo (cosa che grazie alle distorsioni di segnale potrebbe benissimo capitare, soprattutto all'aumentare esponenziale dei simboli), verrà fatto un errore su un solo bit.