

UNIVERSITÀ DEGLI STUDI DI PADOVA
DIPARTIMENTO DI INGEGNERIA DELL'INFORMAZIONE
TESI MAGISTRALE IN INGEGNERIA DELLE TELECOMUNICAZIONI

STUDIO DI FATTIBILITÀ PER LA
TRASFORMAZIONE IN TECNOLOGIA VOIP DI UN
IMPIANTO TELEFONICO ANALOGICO AD ANELLO

RELATORE: Chiar.mo prof. Michele Zorzi

CORRELATORE: Dott. Franco Righetti

LAUREANDO: Francesco Triolo

A.A. 2012/2013

Ai miei genitori

Sommario

Il sistema telefonico ad anello al centro dello studio di questa tesi è parte integrante di un'infrastruttura di telecomunicazioni sviluppata in ambito ferroviario. In particolare è costituito da una rete di telefoni interconnessi in parallelo tramite una struttura ad anello chiamata *circuito di piazzale*. Il protocollo di comunicazione usato dai telefoni di piazzale è basato sulla trasmissione di un segnale analogico in banda fonica. Lo scopo di questa tesi è l'analisi e lo studio delle condizioni per la trasformazione da sistema analogico a sistema completamente digitale basato su VoIP (Voice over Internet Protocol) utilizzando l'infrastruttura, ovvero il cavo, già presente e in uso. La suddetta trasformazione in sistema digitale permetterebbe di poter effettuare chiamate, anche in multiconferenza, indipendenti tra loro.

Indice

1	Il sistema STSI	3
1.1	Il circuito di piazzale attuale	5
1.1.1	I segnali in linea	6
1.1.2	I telefoni	7
1.1.3	I cavi	8
2	Il progetto di trasformazione	9
3	Il livello fisico	12
3.1	Analisi del cavo telefonico	12
3.2	Architettura del trasmettitore	18
3.2.1	La modulazione QAM	18
3.2.2	Il filtro di trasmissione	20
3.2.3	Il codice a correzione d'errore	24
3.2.4	Riepilogo livello fisico del trasmettitore	27
3.3	Modello del canale	28
3.4	Architettura del ricevitore	29
3.4.1	L'Equalizzatore	29
3.4.2	Il Costas loop	32
3.4.3	Il Demodulatore	34
3.4.4	Il Decoder Viterbi	35
3.4.5	Riepilogo ricevitore	38

<i>INDICE</i>	vii
4 Il livello link layer	40
4.1 Composizione del frame	40
4.2 Accesso al mezzo	41
4.2.1 Rimozione, Aggiunta o Sostituzione di un telefono	42
4.3 Buffer	43
4.4 Risparmio energetico, chiamata e risposta	43
5 I livelli superiori	45
5.1 Real-time Transport Protocol e UDP	45
5.2 Livello applicazione: i codec	46
5.2.1 G.711	47
5.2.2 G.729	48
5.3 La pacchettizzazione	49
6 Risultati e considerazioni	52
6.1 Cavo 4 km	52
6.2 Cavo 7 km	54
6.3 Conclusioni	55
Bibliografia	59

Introduzione

La rete ad anello oggetto di questa tesi è detta *circuito di piazzale* e fa parte di un'architettura di telecomunicazioni proprietaria di RFI (Rete Ferroviaria Italiana), chiamata STSI (Sistema di Telefonia Selettiva Integrata). Questo *circuito di piazzale*, facente parte di un CTS (Concentratore Telefonia Selettiva) di stazione, permette agli operatori ferroviari, in servizio all'esterno dei locali di una stazione, di comunicare con qualsiasi altro operatore o dirigente ferroviario situato in qualsiasi punto della medesima stazione o di un'altra stazione (ivi compresi gli operatori lungo una linea ferroviaria). I segnali trasmessi e ricevuti dai telefoni di piazzale sono di tipo analogico e collocati spettralmente in banda fonica, quindi hanno un'occupazione in frequenza molto ristretta.

La maggiore limitazione di questo sistema è la possibilità di effettuare una sola chiamata che impegna la linea in modo esclusivo. E' possibile tuttavia sfruttare *l'inclusione in linea*, cioè inserirsi nella comunicazione e partecipare alla conversazione, passando così in modalità multiconferenza, ma non è possibile effettuare un'altra chiamata indipendente tra altri due o più interlocutori. L'obiettivo di questa tesi è analizzare e valutare le condizioni di fattibilità per la trasformazione di questo sistema analogico in sistema completamente digitale basato su tecnologia VoIP.

L'azienda che progetta e sviluppa questi sistemi telefonici, e che mi ha gentilmente ospitato permettendomi di svolgere l'attività di tirocinio, è la Telefin S.p.A. con sede a Verona.

La struttura della tesi è divisa in 6 capitoli:

1. Nel Capitolo 1 verrà illustrato a grandi linee il sistema ad anello attuale e il contesto nel quale è installato;

2. Nel Capitolo 2 verrà introdotto brevemente il progetto di trasformazione;
3. Nel Capitolo 3 verranno analizzate le caratteristiche di funzionamento dei componenti del livello fisico;
4. Nel Capitolo 4 verranno analizzate la composizione dei frame a livello link layer e le procedure di funzionamento del sistema proposto;
5. Nel Capitolo 5 verranno introdotti brevemente i livelli superiori, dove si vedranno anche i codec scelti;
6. Nel Capitolo 6 infine verranno riportati i risultati delle simulazioni e le considerazioni sulla fattibilità dell'impianto proposto.

Il sistema STSI

Nell'ambito delle telecomunicazioni ferroviarie è presente una varietà di tipologie di impianti molto vasta. Ognuno di questi impianti impiega una combinazione di tecnologie diversa a seconda dello scopo di utilizzo a cui è destinato, il tutto è progettato in maniera da consentire il più possibile l'interoperabilità tra i sistemi e garantire il funzionamento anche in condizioni di errori o imprevisti che possano compromettere il corretto funzionamento.

Diversamente dal passato, i sistemi sviluppati e prodotti da diverse aziende devono rispettare direttive e specifiche redatte dagli uffici tecnici atti a tal scopo, così da uniformare le tipologie dei dispositivi, le caratteristiche di trasmissione e la conformazione dei segnali trasmessi. Il *circuito di piazzale*, l'impianto oggetto di studio in questa tesi, fa parte di un sistema di telecomunicazioni incentrato sulla telefonia chiamato STSI (*Sistema di Telefonia Selettiva Integrata*). Quest'ultimo rappresenta il nuovo standard per gli impianti telefonici selettivi di servizio nelle stazioni e lungo le linee ferroviarie, è basato su soluzioni flessibili ed innovative che consentono di renderlo idoneo per varie condizioni di impiego tramite semplici interventi di riconfigurazione.

Le funzioni e le caratteristiche principali del sistema STSI permettono di:

- consentire l'accesso alle comunicazioni radio terra-treno realizzate mediante sistemi radiomobili cellulari pubblici (GSM), che ora non sono più utilizzati, e proprietari (GSM-R¹);

¹Il sistema GSM-Railway è uno standard internazionale proprietario per telecomunicazioni mobili ferroviarie sviluppato specificatamente per garantire comunicazioni senza interruzioni alle alte velocità, fino a 500 km/h [3]

- integrare su un'unica console operatore le funzioni di accesso ai vari ambienti di comunicazione (telefonia selettiva, telefonia automatica, telefonia mobile, telediffusione sonora);
- migliorare gli aspetti di affidabilità/disponibilità di impianto e di manutenzione in linea, attraverso l'impiego esteso di criteri di tele-diagnostica;
- garantire la compatibilità tra sistemi prodotti da fornitori diversi.

L'architettura generale del STSI viene illustrata in Figura 1.1 .

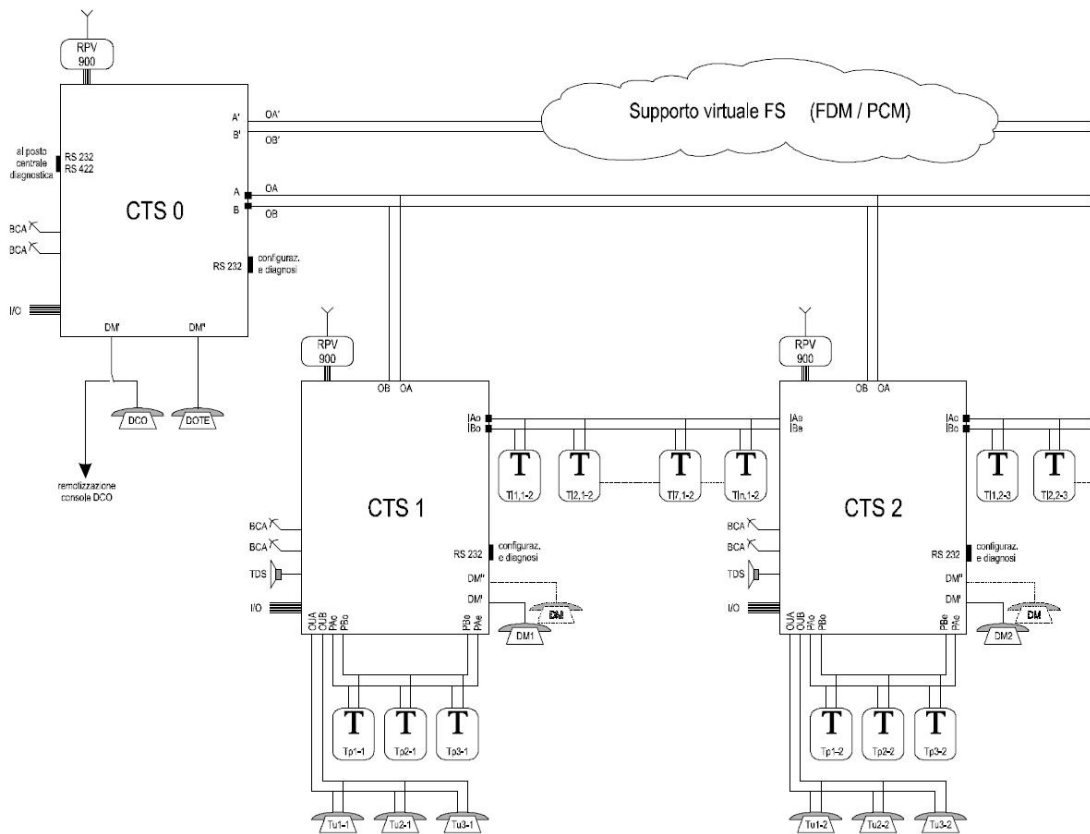


Figura 1.1: Schema generale del sistema STSI

I macroblocchi CTS0, CTS1, CTS2, ecc... rappresentano gli impianti nelle stazioni ferroviarie, ognuno è chiamato *Concentratore Telefonia Selettiva* con una sua numerazione progressiva. Come si vede dallo schema, l'organizzazione della struttura è di tipo gerarchico, per cui il CTS0 è la stazione principale di un tratto

di linea che comprende altre stazioni ‘minori’. In generale nella CTS0 sono presenti vari operatori ferroviari che si occupano di monitorare e gestire un gruppo di sistemi e servizi, tra cui la linea elettrica ferroviaria (DOTE, *Dirigente Operativo Trazione Elettrica*) e la circolazione dei treni nelle stazioni (DCO, *Dirigente Centrale Operativo*). Ciascuna stazione secondaria può essere *presenziata* o no, cioè può essere localmente presente un *Dirigente Movimento* (DM) che dirige e gestisce il traffico ferroviario transitante nella stazione di sua competenza. In caso di assenza di un DM in una certa stazione, sarà il DCO nella stazione principale ad occuparsi di questa mansione.

Le linee di comunicazione seguono la struttura gerarchica del sistema: quelle di primo livello collegano la stazione principale CTS0 con quelle secondarie e sono chiamate *Omnibus*, quelle di secondo livello collegano le stazioni secondarie tra loro adiacenti e sono dette *Interstazionali*, poi ci sono altre linee di secondo livello che collegano i telefoni locali e sono chiamate *di piazzale*, proprio perché sono atte a interconnettere i telefoni installati nei locali esterni (piazzali) delle stazioni, oppure lungo la linea. In ogni stazione sono inoltre presenti anche le interfacce GSM-R per la telefonia mobile e le interfacce per la *Tele Diffusione Sonora* (TDS) per gli annunci sonori informativi emessi dagli altoparlanti dei piazzali. Nella Figura 1.1 ogni linea è rappresentata come se fosse un solo conduttore ma è costituita da una coppia di cavi, quindi si vede come ogni linea abbia una ‘gemella’ di backup per prevenire blocchi e cadute di connessione dovuti ad interruzioni accidentali.

Lo scopo dell’adozione di quest’architettura è quello di interconnettere qualsiasi telefono con qualsiasi altro apparecchio situato in un altro punto della stessa stazione, in altre stazioni, oppure lungo la linea ferroviaria, comprendendo anche l’instradamento su rete mobile proprietaria GSM-R in alcuni casi.

1.1 Il circuito di piazzale attuale

I circuiti telefonici di piazzale sono progettati con una topologia ad anello, per cui il riscontro della disponibilità al servizio degli stessi può essere realizzata monitorando i messaggi di servizio impiegati dal protocollo di segnalazione/chiamata all’estremo opposto del circuito rispetto al punto di immissione di tali messaggi di servizio. L’assenza in ricezione all’altro estremo dell’anello dei messaggi trasmessi origina il rilevamento di una condizione di anomalia e la contemporanea

immissione dai due estremi dei messaggi di servizio sui due tronchi di anello che si vengono a determinare. Il riscontro di un'anomalia nell'anello dà avvio a una fase diagnostica di ricerca guasti che può essere configurata a seconda delle necessità.

Anche gli anelli dei circuiti di piazzale, come altre infrastrutture adottate nel sistema STSI, utilizzano una doppia coppia di cavi: una viene usata per comunicazioni da e verso il DCO, che si occupa della circolazione dei treni, mentre l'altra coppia viene utilizzata per comunicazioni con il DOTE, che si occupa di trazione elettrica. La doppia coppia ha anche una funzione di backup: se una linea ha un guasto, l'altra può essere utilizzata quando libera.

Questa infrastruttura è variamente configurabile in base al luogo e all'uso a cui è destinata, il numero di telefoni installati nell'anello non è fisso: possono essere usati fino a 7 telefoni secondo la specifica ferroviaria [1], più il centralino che diventa l'ottavo dispositivo e costituisce anch'esso un elemento attivo dell'anello poiché le telefonate vengono instradate sia da un telefono dell'anello verso l'esterno (ad esempio per chiamare un'altra stazione, o anche un ufficio presente nella stessa stazione), sia dall'esterno verso i telefoni di piazzale.

Come già accennato nell'introduzione, la principale limitazione dell'attuale sistema del circuito di piazzale è il fatto che sia possibile una sola comunicazione alla volta; è tuttavia possibile usufruire dell'*inclusione in linea*, cioè la possibilità che un terzo interlocutore possa alzare il microtelefono e inserirsi nella comunicazione in corso trasformandola in una multiconferenza. La linea rimane quindi occupata in maniera esclusiva da una sola comunicazione. Altre telefonate indipendenti da quella in corso non sono possibili contemporaneamente.

1.1.1 I segnali in linea

I segnali trasmessi e ricevuti dai telefoni di piazzale, detti *tonitel*, sono sia di tipo analogico che digitale e permettono la presenza contemporanea, sulla stessa coppia, di fonia e dati tramite la creazione di due canali indipendenti, ottenuta tramite divisione dello spettro dei segnali in due bande di frequenza differenti. Si hanno così un canale dedicato per la fonia e un canale dedicato per i dati. La banda complessiva utilizzata è compresa tra 300 e 3400 Hz.

La porzione di banda dedicata alla fonia è compresa tra 300 e 2400 Hz, mentre il canale dati è confinato nella banda da 2500 a 3400 Hz e viene impiegato usando una modulazione con tecnica FSK (*Frequency Shift Keying*). Questo tipo

di modulazione è di semplice implementazione: viene trasmessa una portante la cui frequenza viene variata ad ogni periodo di simbolo in base al bit da trasmettere, la frequenza di *mark* (bit '1') è 2650 Hz mentre quella di *space* (bit '0') è 2850 Hz [1]. La banda per il canale dati è molto stretta e consente di raggiungere un symbol-rate di 300 baud, che corrisponde ad un bit-rate a livello fisico di 300 bit/s, con una durata di bit di 3.33 ms; nonostante il tasso di trasmissione sia molto basso, è comunque sufficiente per il suo scopo: il canale dati viene usato per inviare segnalazioni di chiamata, che comprendono la priorità, la richiesta di *inclusione in linea*, e altri tipi di messaggi; vi è inoltre un sistema di telediagnostica per la segnalazione di eventuali guasti o *warning*, ad esempio per segnalare un malfunzionamento del microtelefono.

Le prestazioni che devono essere garantite dai modem integrati negli apparecchi in termini di errori sui bit sono vincolate da un BER (Bit Error Rate) imposto uguale o minore di 10^{-5} con un SNR (rapporto segnale-rumore) di 15 dB.

1.1.2 I telefoni

I telefoni di piazzale sono installati in casse stagne che li proteggono dalle intemperie climatiche e ambientali tipiche dei locali esterni, ma anche da manomissioni e/o modifiche involontarie, per cui sono dotate anche di serratura. Tenendo conto delle condizioni tipiche dei luoghi dove vengono installati i telefoni, cioè nelle vicinanze di linee elettriche aeree a 3KVcc o 25KVac e passaggio sui binari di mezzi con trazione elettrica di potenza, gli involucri degli alloggiamenti devono garantire un'adeguata protezione per gli ingressi e le uscite da sovratensioni e sovracorrenti e garantire il funzionamento in un range di temperature da -25°C a $+70^{\circ}\text{C}$ e umidità relativa da 0% a 100%. Inoltre deve essere garantito il funzionamento per almeno 15 anni anche con periodi intermedi di magazzinaggio, esposizione a clima marino e/o smog intenso. Sui medesimi cavi di segnale degli anelli di piazzale viene iniettata una tensione continua che funge da telealimentazione per i telefoni. Le specifiche stabiliscono che ci debbano essere due tensioni sommate: una da +70Vcc e l'altra da -70Vcc, fornendo quindi 140Vcc nel punto di iniezione; non sono previste particolari limitazioni di corrente, ad eccezione delle protezioni per il personale di manutenzione. Il centro delle due tensioni è collegato a terra e le protezioni intervengono se una persona entra in contatto con la coppia di cavi o con un polo e la terra.



Figura 1.2: Telefono di piazzale *Tonitel* installato in cassa stagna da esterno

I telefoni sono collegati ai cavi in parallelo tramite piccole scatole di derivazione. In Figura 1.2 è riportata una foto di un telefono *Tonitel* utilizzato nei piazzali.

1.1.3 I cavi

I supporti fisici di norma utilizzati rispettano delle specifiche tecniche definite in ambito ferroviario [2]. Sono coppie di cavi intrecciati, comunemente chiamate *doppini*, sono tipicamente fabbricati in rame rosso, hanno un diametro di 0,7 mm e hanno un isolamento in polietilene.

Le lunghezze tipiche degli anelli di piazzale vanno tipicamente dai 3 ai 6-7 Km e nel sistema attuale, avendo i segnali una larghezza di banda ristretta, non vi è dispersione né eccessiva distorsione dei segnali, non è quindi necessario l'impiego di un equalizzatore. Non vi è neppure un sistema di riezione dei disturbi derivanti da diafonia vicina e/o lontana poiché questi tipi di disturbi sono in genere irrilevanti, sia per la banda di funzionamento, che è quasi immune ai disturbi indotti, sia perché le ubicazioni di installazione dei cavi tipicamente non hanno sorgenti di disturbo persistenti né particolarmente intense.

Il progetto di trasformazione

L'obiettivo di questo progetto è, come preannunciato, lo studio delle condizioni per poter attuare la conversione dell'attuale sistema analogico in sistema completamente digitale basato su tecnologia VoIP.

Un requisito fondamentale è il mantenimento dell'attuale infrastruttura, cioè dei cavi già fisicamente presenti e posati, e della topologia ad anello. Un cambio di topologia, adottando cioè uno schema *a stella*, consentirebbe di utilizzare una infrastruttura standard collegando tutti i telefoni ad uno switch e usare un protocollo simile ad Ethernet per la trasmissione punto-punto, tuttavia non sarebbe vantaggioso poiché porterebbe ad un incremento delle difficoltà e dei costi di installazione dovute alla necessità di posare nuovi cavi, la cui quantità e lunghezza diventerebbero proporzionali al numero di telefoni necessario all'impianto. Inoltre non è detto che ci sia fisicamente lo spazio nelle canaline per una tale quantità di cavi, dato che i telefoni sono stati originariamente installati in sequenza seguendo uno schema ad anello.

L'idea generale è quindi impiegare un sistema VoIP utilizzando come mezzo trasmissivo il *doppino* telefonico presente sugli anelli su distanze medie. Lo schema dello stack protocollare dei livelli alti (dal livello di rete al livello applicativo) rimane quello standard impiegato diffusamente nei sistemi VoIP, ciò che cambia è appunto il processo trasmissivo vero e proprio ai bassi livelli.

Una prima ipotesi su come sfruttare il canale trasmissivo era quella di adottare un'architettura a divisione di frequenza in cui ogni telefono aveva una banda dedicata al proprio canale, ma un'implementazione di questo tipo sarebbe piuttosto complicata dal punto di vista dei filtri di ricezione/trasmissione per il numero

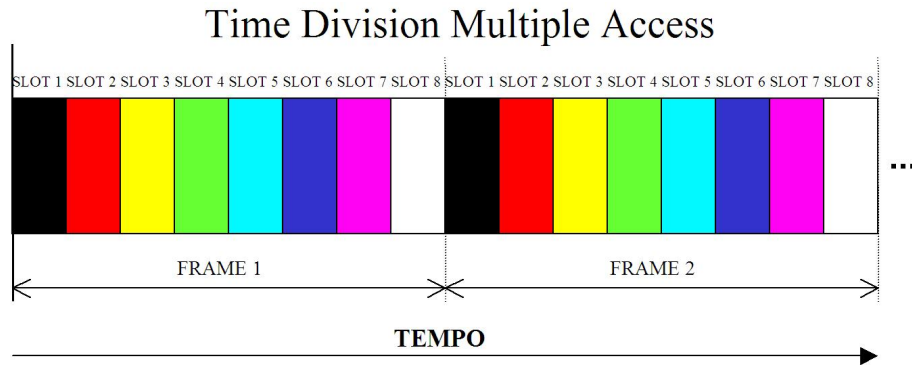


Figura 2.1: Tecnica di accesso al mezzo TDMA

necessario di bande distinte (fino a 7 telefoni più il centralino, quindi 8 dispositivi totali) e soprattutto non avrebbe consentito una sufficiente larghezza di banda per ogni apparecchio tale da supportare un traffico VoIP con un certo bit-rate.

Uno schema adottabile è invece un modello trasmissivo a divisione di tempo, noto come TDMA (*Time Division Multiple Access*), poiché permetterebbe ad ogni telefono di sfruttare tutta la larghezza di banda disponibile. Durante la trasmissione da un apparecchio, tutti gli altri ricevono sulla stessa banda di frequenze, quindi non vi è necessità dell'uso di particolari filtri per la suddivisione dello spettro del segnale ricevuto associato ai telefoni. Questa tecnica divide in turni temporali l'accesso al mezzo condiviso, il tempo di utilizzo del mezzo è diviso in *frame* della stessa durata, che a loro volta vengono suddivisi in *slot* uguali tra loro (vedi Figura 2.1). Ogni slot viene assegnato ad un dispositivo che potrà trasmettere in quel breve lasso di tempo. Il numero di slot in un frame è pari al numero di apparecchi presenti nell'anello, in questo caso sono in totale 8 (che è il numero massimo dichiarato nelle specifiche). Un frame ha quindi durata temporale pari alla somma dei tempi di trasmissione impiegati da tutti i dispositivi presenti, cioè è l'intervallo che un dispositivo deve attendere tra una trasmissione a quella successiva.

La durata di un frame viene determinata dall'intervallo di pacchettizzazione, cioè dal periodo di tempo che intercorre tra l'arrivo di un pacchetto da trasmettere dai livelli superiori e quello successivo, questo periodo è dettato principalmente dal codec scelto per la compressione della voce, come si vedrà nella Sezione 5.3.

Usando TDMA si ha quindi un solo telefono trasmettitore in ogni slot, tutti gli altri sono ricevitori nello stesso slot e sulla stessa banda, ovviamente solo uno o alcuni saranno i destinatari della trasmissione. Ogni dispositivo dovrà memorizzare

l'associazione dei 'turni' di trasmissione degli altri telefoni così da sapere quando è il proprio turno in cui trasmettere. L'assegnazione è dinamica, cioè i telefoni possono essere assegnati a qualsiasi turno 'libero' in qualsiasi ordine.

Nei prossimi capitoli verranno spiegate in dettaglio le caratteristiche e le specifiche che dovrà avere il sistema ai bassi livelli dello stack protocollare, con un approccio di tipo *bottom-up*, cioè si tratterà prima il livello fisico (PHY), parlando delle caratteristiche del mezzo trasmissivo, della modulazione e del codice a correzione d'errore proposti, poi si analizzeranno i livelli superiori.

Il livello fisico

La ‘sfida’ principale di questo progetto è trovare un’architettura efficiente ma al contempo di semplice implementazione, due caratteristiche che in genere sono inversamente proporzionali, quindi lo scopo è trovare un compromesso ottimo tra queste due peculiarità. La limitazione maggiore è data dalle prestazioni del cavo: un *semplice*, seppur con un certo standard qualitativo, doppiino telefonico di diametro 0,7 mm, che deve trasmettere un segnale digitale, modulato e con una certa larghezza di banda, ad una distanza non banale, anche fino a 6-7 km a seconda dell’impianto.

La larghezza di banda è sicuramente il problema principale: oltre una certa frequenza infatti il cavo ha un’attenuazione troppo elevata che introduce una distorsione tale da non poter recuperare una forma d’onda che sia simile a quella trasmessa e ciò peggiora con la distanza: si rende perciò obbligatorio l’impiego di un equalizzatore.

Per poter raggiungere buone prestazioni è inoltre necessario l’uso di un codice a correzione d’errore FEC (*Forward Error Correction*).

3.1 Analisi del cavo telefonico

Per poter implementare una simulazione completa a livello fisico che rispecchi il vero comportamento del canale di comunicazione è necessario conoscere le caratteristiche e i parametri elettrici del cavo presente nei piazzali, purtroppo però non è stato possibile recuperarli, perciò sono state fatte delle misure per poter ricostruire

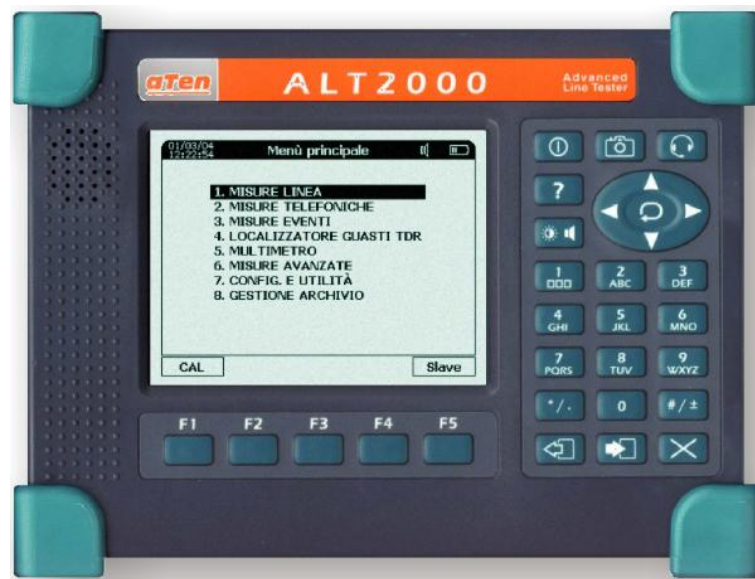


Figura 3.1: Strumento usato per l'attenuazione, *Advanced Line Tester 2000*

il comportamento sia in frequenza, in termini di attenuazione e fase, e nel tempo (risposta impulsiva).

Per l'attenuazione è stato usato lo strumento di misura automatica *ALT2000* (Figura 3.1) prodotto da aTen S.r.l. [5]. Esso ha due porte a disposizione (trasmissione e ricezione) e si è sfruttato il fatto che il capo e la coda del cavo fossero nello stesso punto, così da ottenere una misura completamente automatica piuttosto precisa. Prima di effettuare la misura si è spenta la telealimentazione e si sono scollegati i capi di entrambe le linee degli anelli. Purtroppo non è stato possibile scollegare fisicamente i telefoni dai cavi, il che ha portato ad una piccola alterazione della misura, che comunque non ha avuto un impatto deleterio sui risultati.

La *spazzolata* in frequenza va da 1 kHz e arriva ai 1100 kHz a passi di 4,5 kHz, mentre la lunghezza del cavo misurato non si conosce in maniera precisa, ma è approssimativamente di 4 km. La curva ottenuta è riportata in Figura 3.2 .

Dai risultati della misura si nota che, eccetto una piccola perdita a 130 kHz dovuta ai circuiti di ingresso dei telefoni ancora collegati nell'anello, l'andamento dell'attenuazione in frequenza è quello atteso, cioè distorto in ampiezza, caratteristico dei cavi telefonici intrecciati, ed è piuttosto regolare (non ha zone ristrette di forte attenuazione).

Per le informazioni sulla fase purtroppo non erano disponibili né una funzio-

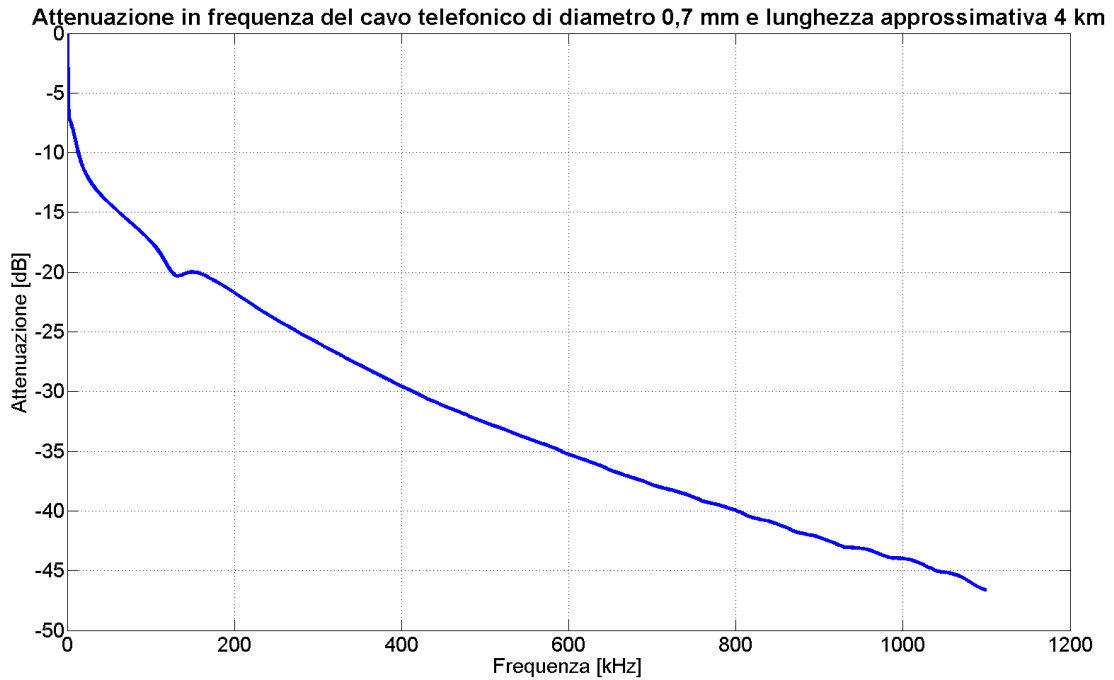


Figura 3.2: Curva di attenuazione misurata del cavo di piazzale

ne dell'ALT200 né un altro strumento automatico altrettanto completi e precisi, perciò per ricostruire il comportamento reale si è ricorsi all'uso di un generatore di funzioni per creare una sinusoide a frequenza variabile partendo dai 10 kHz e arrivando a 1100 kHz con passo variabile. Ad ogni frequenza corrispondeva un certo ritardo tra il segnale trasmesso e quello ricevuto dall'altro capo del cavo, quindi essendo una sinusoide e misurando con un oscilloscopio questo ritardo tra due punti equivalenti del segnale, è stato possibile ricavare il corrispondente sfasamento introdotto dal cavo, come viene riportato di seguito: il ritardo misurato è ΔT_0 , il segnale trasmesso è $S_{TX} = \cos(2\pi f_0 t)$ e ha fase iniziale nulla, mentre il segnale ricevuto è del tipo

$$S_{RX} = \cos(2\pi f_0(t + \Delta T_0)) = \cos(2\pi f_0 t + 2\pi f_0 \Delta T_0) = \cos(2\pi f_0 t + \phi_0)$$

quindi per ogni frequenza f_0 , la sinusoide generata fornisce l'informazione sulla fase che introduce il cavo, ottenuta calcolando $\phi_0 = 2\pi f_0 \Delta T_0$.

I risultati delle misurazioni hanno evidenziato un ritardo di ricezione abbastanza costante, scoprendo quindi un comportamento quasi perfettamente lineare in frequenza (eccetto una impercettibile deviazione alle frequenze iniziali fino ai

50-60 kHz) come si può osservare nella Tabella 3.1; questa considerazione porta ad affermare che il tipo di cavo non è dispersivo (non ha distorsione di fase), cioè non tende a differenziare le componenti frequenziali in maniera tale da corrompere il segnale. La linearità in fase è visibile in Figura 3.3 dove sono raffigurati i valori della Tabella 3.1.

Frequenza	Ritardo	Sfasamento
10 kHz	26.8 μs	1.6839 rad
20 kHz	23.8 μs	2.9908 rad
30 kHz	22.0 μs	4.1469 rad
40 kHz	21.6 μs	5.4287 rad
50 kHz	21.3 μs	6.6916 rad
60 kHz	21.0 μs	7.5429 rad
70 kHz	20.6 μs	9.0472 rad
80 kHz	20.4 μs	10.2542 rad
90 kHz	20.2 μs	11.4228 rad
100 kHz	20.2 μs	12.6920 rad
150 kHz	19.6 μs	18.4726 rad
200 kHz	19.4 μs	24.3788 rad
250 kHz	19.4 μs	30.4734 rad
300 kHz	19.2 μs	36.1911 rad
350 kHz	19.1 μs	42.0031 rad
400 kHz	19.0 μs	47.7522 rad
450 kHz	19.0 μs	53.7212 rad
500 kHz	19.0 μs	59.5332 rad
550 kHz	19.0 μs	65.4865 rad
600 kHz	19.0 μs	71.4398 rad
650 kHz	19.0 μs	77.3931 rad
700 kHz	18.9 μs	83.1265 rad
800 kHz	18.9 μs	94.8133 rad
900 kHz	18.9 μs	106.500 rad
1000 kHz	18.9 μs	118.187 rad
1100 kHz	18.9 μs	129.873 rad

Tabella 3.1: Ritardi e corrispondenti sfasamenti introdotti dal cavo di piazzale

Una volta ottenuti i suddetti dati sul comportamento in frequenza, *ribaltando* simmetricamente i dati per poter ricostruire anche le frequenze negative, è stata creata una risposta bilatera, simmetrica e complessa coniugata dato che il canale

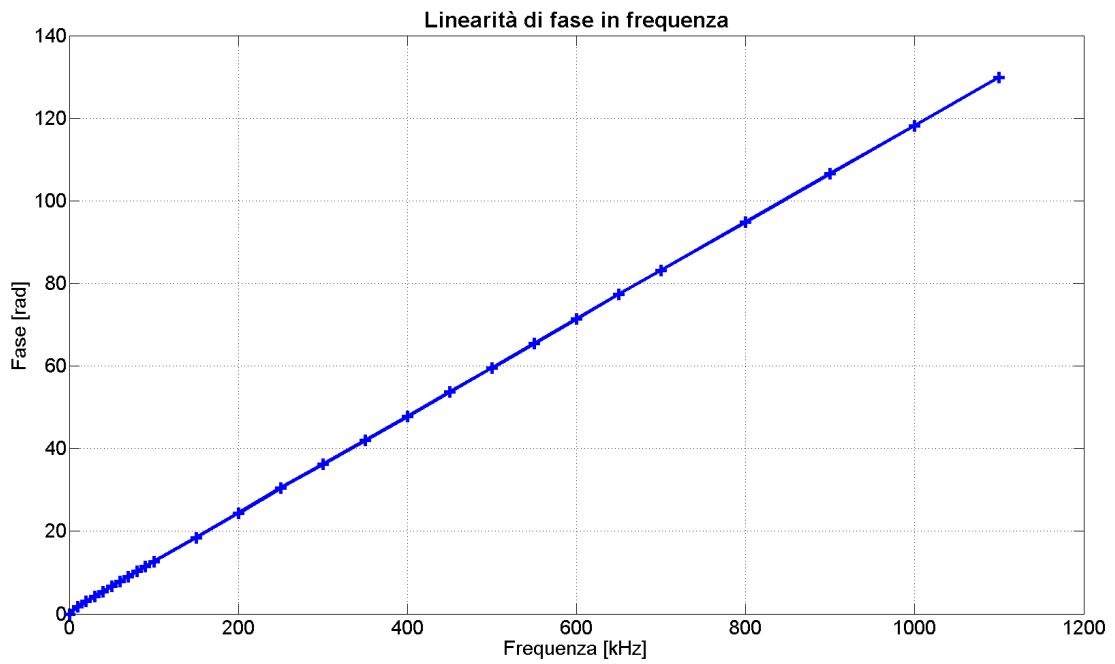


Figura 3.3: Linearità in frequenza della fase

è reale. Poi calcolando la trasformata di Fourier inversa si è ottenuta la risposta impulsiva nel tempo, riportata in Figura 3.4.

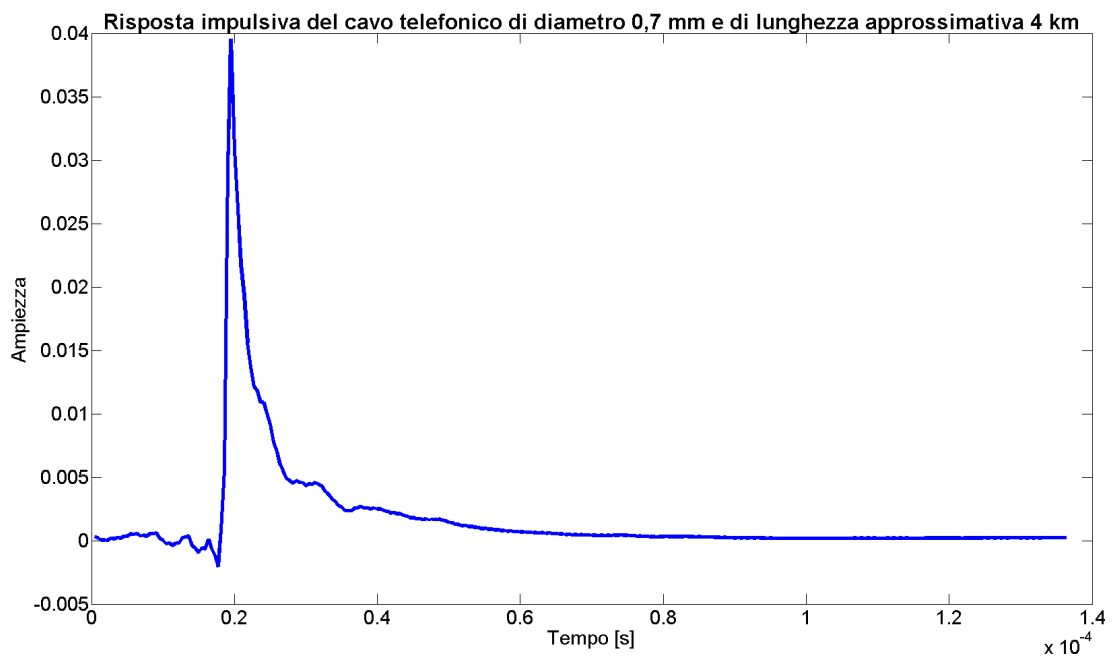


Figura 3.4: Risposta impulsiva cavo piazzale

Successivamente sono state effettuate le stesse prove su un cavo di lunghezza maggiore (ipotizzata come la massima lunghezza degli anelli dei piazzali), cioè 7 km. Purtroppo però non era disponibile lo stesso tipo di cavo, era invece a disposizione un altro cavo con caratteristiche lievemente diverse, ma costruito con lo stesso materiale conduttore, le differenze sono il diametro leggermente maggiore (0,9 mm invece di 0,7 mm) e isolamento in carta (invece del polietilene). Il comportamento è leggermente diverso: all'aumentare della frequenza l'attenuazione ha una progressione maggiore, cioè il *gap* (presente per la diversa lunghezza) tra il cavo da 4 km e quello da 7 km non è costante. L'andamento della fase è ugualmente lineare come per il cavo precedente. In Figura 3.5 si possono vedere le differenze di comportamento in frequenza per quanto riguarda l'ampiezza del segnale ricevuto.

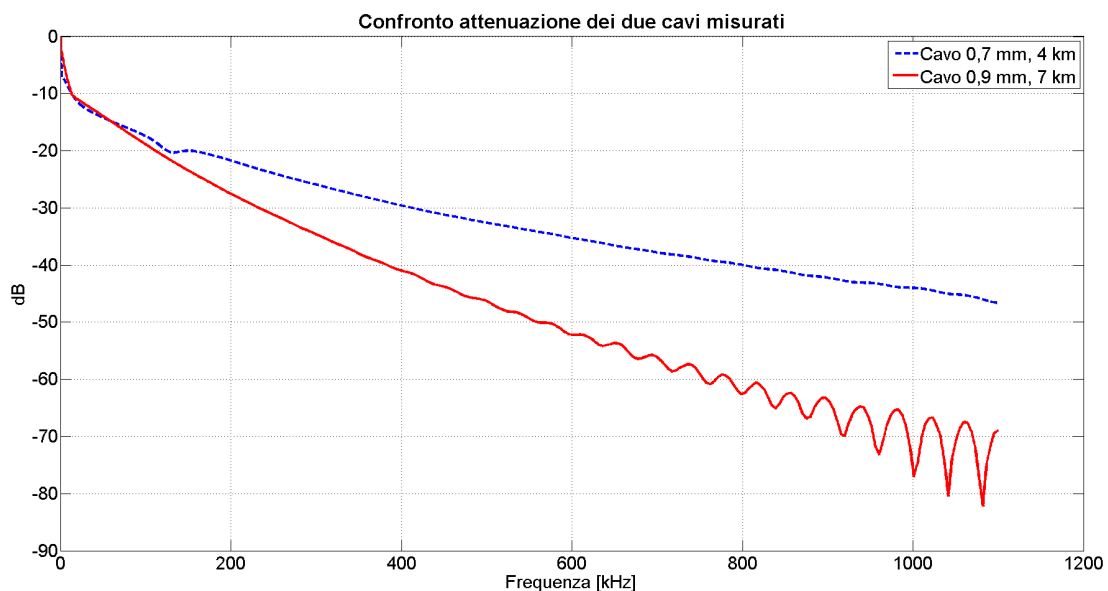


Figura 3.5: Curve delle attenuazioni dei due cavi messe a confronto

In Figura 3.6 sono riportate le risposte impulsive. Si può notare come il cavo più lungo sia più dispersivo, il picco principale ha ampiezza molto minore del cavo più corto, di circa il 34 %, e le oscillazioni pre e post cursori sono più accentuate, quindi questo porterà ad avere una piccola quantità di interferenza di intersimbolo (ISI). Come si vedrà, l'equalizzatore andrà configurato con parametri *ad hoc* per ogni tipo di canale. Si ricorda comunque che il cavo con diametro 0,9 mm non è

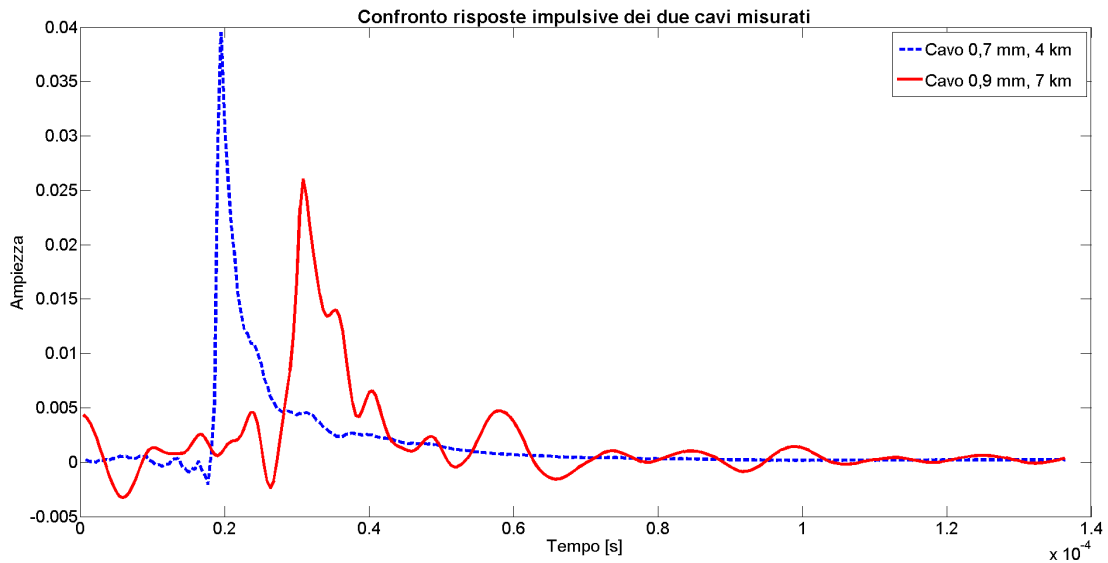


Figura 3.6: Confronto risposte impulsive

lo stesso di quello usato nei piazzali, quindi verrà usato come prova non specifica per il suo reale utilizzo, ma come indicazione sugli effetti a lunghe distanze.

3.2 Architettura del trasmettitore

3.2.1 La modulazione QAM

Per questo tipo di impianto la scelta migliore è la modulazione 4-QAM (*Quadrature Amplitude Modulation*), che associa la sequenza di bit in ingresso ad una sequenza di simboli con costellazione di cardinalità $M = 4$, cioè vengono mappati $\log_2(M) = 2$ bit per simbolo. Il segnale in uscita dal modulatore trasmesso sul canale è di tipo passabanda e si appoggia ad una portante sinusoidale.

La scelta è ricaduta su questa modulazione perché, come si vede dalla mappatura, è efficiente in banda, infatti trasporta 2 bit per simbolo, quindi il bit-rate è doppio del symbol-rate, ed è robusta nei confronti del rumore: le regioni di decisione sono 4 e sono ben distinte, la distanza minima tra i simboli è $\sqrt{2}$, come si vede in Figura 3.7. A parità di efficienza spettrale e di SNR, 4-QAM, rispetto alle modulazioni PAM (*Pulse Amplitude Modulation*) e FSK (*Frequency Shift Keying*), garantisce un tasso d'errore sui simboli inferiore [4], nonostante abbia un'architettura leggermente più complessa.

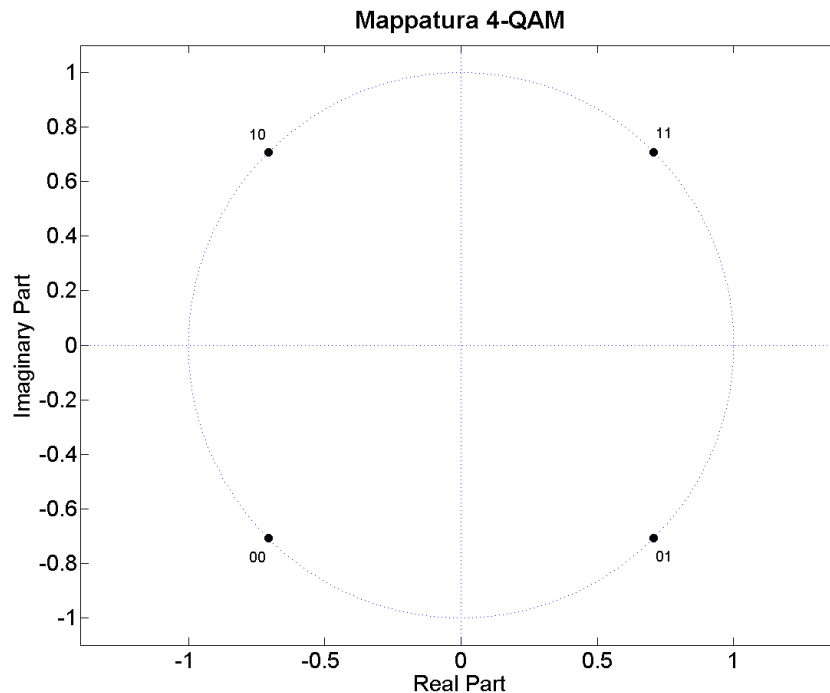


Figura 3.7: Schema mappatura dei bit in simboli complessi 4-QAM

I bit da trasmettere vengono mappati secondo una codifica Grey¹ nei 4 simboli complessi per poter avere un solo bit di differenza tra simboli adiacenti a due a due. Con questa associazione i simboli vengono rappresentati anche dai bit stessi: il primo bit indica se la parte immaginaria del simbolo è positiva o negativa ('1' se è positiva) e il secondo bit indica il segno della parte reale. Il modulo dei simboli trasmessi è sempre 1, come si vede dalle coordinate XY che hanno entrambe valore $\pm \frac{\sqrt{2}}{2}$.

La modulazione QAM è a simboli complessi, ma in questa implementazione del modello si è voluto costruire un'architettura più vicina a quella che potrà essere poi l'implementazione fisica del sistema, perciò in uscita dal modulatore, al posto di un unico flusso di valori complessi, sono presenti due sequenze distinte di valori reali: una trasporta l'informazione sulla parte reale e l'altra sulla parte immaginaria dei simboli complessi.

¹La codifica Grey ha la peculiarità di avere un solo bit di differenza rispetto ai simboli precedente e successivo, contrariamente alla mappatura binaria tradizionale che associa parole binarie con valore incrementale. Questo permette di semplificare e rendere meno soggette a errori lunghe sequenze di bit. Vedi pag. 531 di [4]

3.2.2 Il filtro di trasmissione

La scelta dei filtri di trasmissione è molto importante perché determina la larghezza di banda effettiva del segnale trasmesso e a seconda della conformazione in frequenza, aiuta ad annullare l'ISI. Nell'architettura sono presenti due filtri di trasmissione identici e paralleli funzionanti per i due flussi di dati sui simboli.

Per ottenere in ricezione una esatta replica dei simboli trasmessi, eccetto per l'ampiezza e per un ritardo, esiste una famiglia di filtri di trasmissione detta *classe di Nyquist*, che rispetta le condizioni di non distorsione, cioè che, in teoria, è in grado di annullare l'interferenza di intersimbolo. Chiamando i coefficienti del filtro di trasmissione h_k con $k \in \{\dots, -2, -1, 0, 1, 2, \dots\}$ e considerando un intervallo di modulazione T , le *condizioni di non distorsione secondo il criterio di Nyquist* nel dominio del tempo sono:

$$\begin{cases} h_0 = 1 \\ h_i = 0 \quad i \neq 0 \end{cases}$$

mentre in frequenza diventano:

$$\sum_{l=-\infty}^{+\infty} H\left(f - \frac{l}{T}\right) = T$$

dove H indica la trasformata di Fourier di h_k . Questa classe di filtri ha inoltre banda limitata ma una breve durata nel tempo.

Il tipo di filtri più diffuso appartenente a questa classe è quello a impulso a *coseno rialzato*, in banda base, il cui nome deriva dal fatto che lo spettro ha la parte discendente (sulle frequenze positive) sagomata come un coseno, con pendenza variabile a seconda del parametro di *roll-off* ρ , visibile in Figura 3.8 [4].

$$H_{RC}(f) = T \operatorname{rcos}(fT, \rho)$$

In realtà il filtro largamente impiegato non è esattamente quello appena visto, ma è leggermente diverso, ovvero ha lo spettro che è la radice quadrata di quello del coseno rialzato, per questo è chiamato impulso a *radice di coseno rialzato* (abbreviato in inglese SRRC, *Square-Root Raised-Cosine*). Quindi lo spettro risulta essere:

$$H_{SRRC}(f) = T \sqrt{\operatorname{rcos}(fT, \rho)}$$

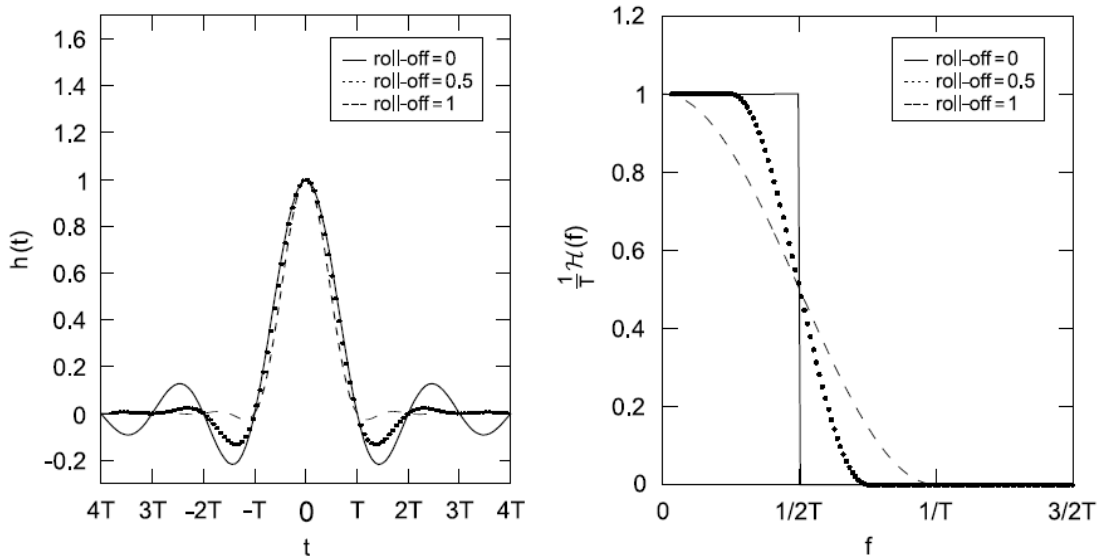


Figura 3.8: Impulso a coseno rialzato nel tempo a sinistra e in frequenza a destra, immagine presa da [4]

Il motivo di questa scelta diffusa è la più facile implementazione fisica, poiché ha un andamento più *morbido* e tempi di risposta più lunghi; conseguentemente ha una occupazione di banda leggermente maggiore. Però per la diversa costruzione del filtro (la presenza della radice quadrata) la conseguenza è che non ha più i requisiti che rispettano i criteri di Nyquist, questo accade poiché negli istanti multipli del periodo di modulazione l'ampiezza del filtro non è esattamente zero, tuttavia, avendo comunque l'impulso risultante una forma d'onda molto simile al coseno rialzato standard e non introducendo una quantità di ISI importante, viene preferito per il motivo sopra citato.

Nelle simulazioni quindi è stato adottato quest'ultimo tipo di filtro con un parametro di roll-off ρ fissato a 0.25.

Per simulare il comportamento di un DAC (*Digital-to-Analog Converter*) prima del canale, il periodo di campionamento viene ridotto da T a $T_C = \frac{T}{Q_0} = \frac{T}{8}$ usando il filtro di trasmissione h_{TX} anche come filtro interpolatore, con fattore di interpolazione $Q_0 = 8$. Un'implementazione efficiente del filtro è quella polifase rappresentata in Figura 3.11, in cui il flusso di simboli (in Figura è rappresentato il flusso di dati sulla parte reale dei simboli $a_{k,I}$, in cui I indica *Inphase*) viene parallelizzato in ingresso ad un banco di 8 filtri a tempo T ognuno decimato con fattore Q_0 e ritardato di un coefficiente rispetto al filtro precedente.

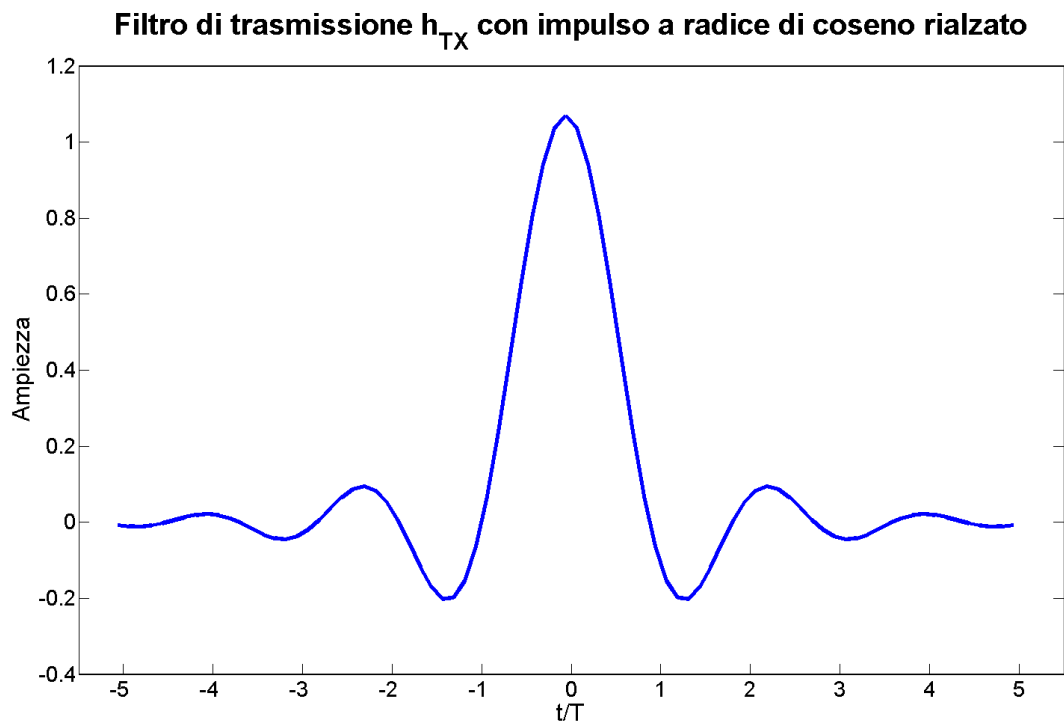


Figura 3.9: Filtro di trasmissione interpolatore a radice di coseno rialzato

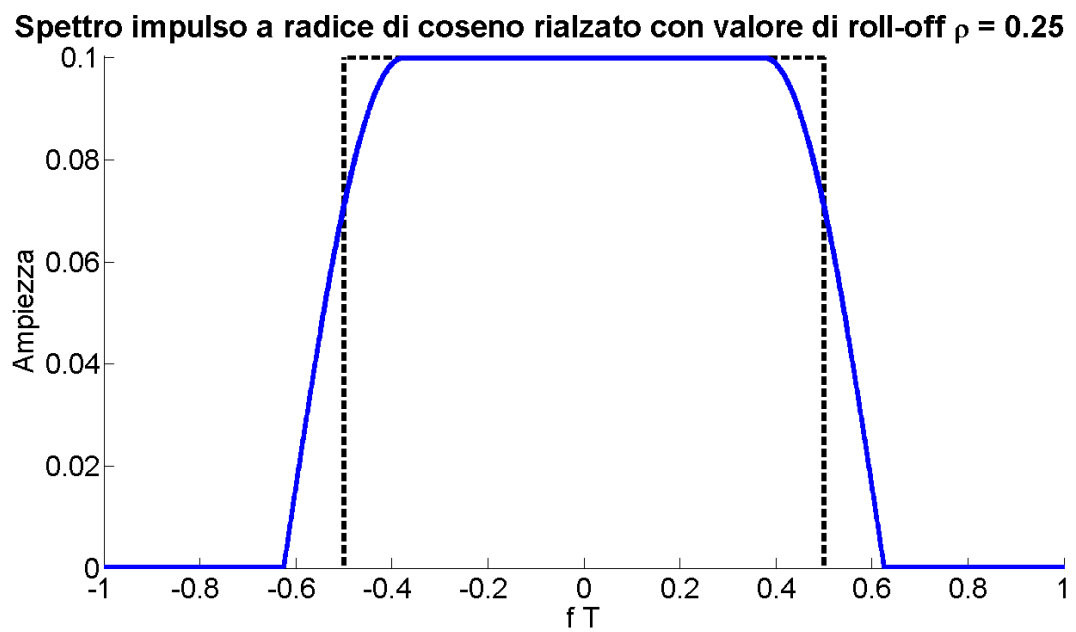


Figura 3.10: Spettro filtro di trasmissione a radice di coseno rialzato con $\rho = 0.25$

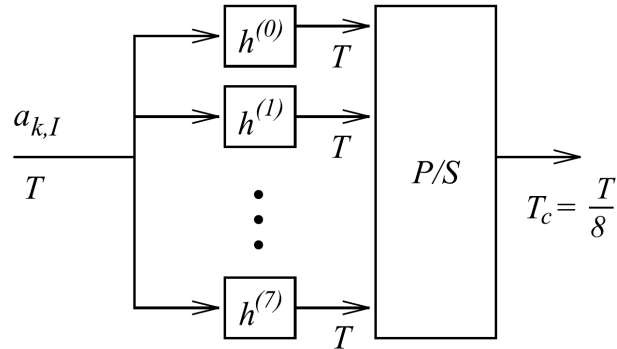


Figura 3.11: Implementazione polifase di un filtro interpolatore con fattore $Q_0 = 8$, immagine presa da [4]

Il numero di coefficienti scelto per i filtri è $N_h = 81$, una quantità sufficientemente grande da avere una risposta in frequenza quasi senza lobi laterali, e avere al contempo una risposta impulsiva piuttosto breve. La risposta impulsiva e lo spettro del filtro di trasmissione usato h_{TX} sono riportati rispettivamente in Figura 3.9 e in Figura 3.10.

Dopo il filtraggio sui filtri di trasmissione, i due flussi di dati vengono modulati con un oscillatore a *frequenza intermedia* f_0 , in particolare vengono moltiplicati ciascuno per una sinusoide: la parte reale, che è in fase, viene moltiplicata per $\cos(2\pi f_0 n T_C)$, mentre la parte immaginaria, in quadratura, viene moltiplicata per la stessa sinusoide, ma sfasata di 90° . In questo modo si trasla lo spettro da banda base a banda passante centrandolo sulla frequenza intermedia. Infine i due flussi modulati vengono sommati costituendo così il segnale trasmesso sul canale, nella parte inferiore di ogni ramo è riportato il periodo di simbolo: T_b indica il periodo di bit, corrispondente a $\frac{1}{R_b}$ cioè l'inverso del bitrate, T indica il periodo di simbolo in uscita dal modulatore QAM, T_C è il periodo di campionamento dei segnali filtrati e interpolati. Quindi, chiamati S_I e S_Q i segnali in uscita dai filtri di trasmissione, il segnale trasmesso sarà:

$$S_{TX} = S_I \cos(2\pi f_0 n T_C) - S_Q \sin(2\pi f_0 n T_C)$$

In Figura 3.12 è riportato lo schema a blocchi completo del trasmettitore.

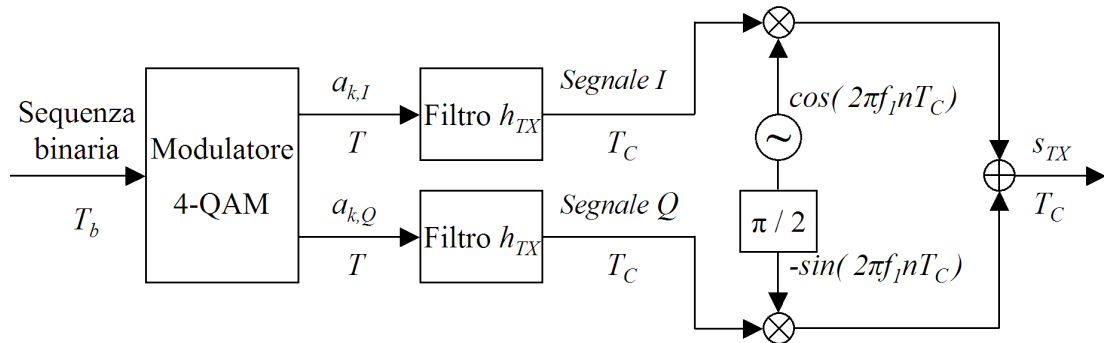


Figura 3.12: Schema a blocchi del trasmettitore

3.2.3 Il codice a correzione d'errore

Data la lunghezza del cavo, il segnale non arriva al ricevitore con una forma d'onda uguale a quella trasmessa, ma arriva alterato per la distorsione di ampiezza introdotta dal canale, tuttavia la larghezza di banda utilizzabile è piuttosto ampia. Oltre all'equalizzatore, per poter raggiungere il tasso d'errore designato di 10^{-5} per un SNR di 15 dB è necessario l'impiego di un codice a correzione d'errore (FEC) in modo da riuscire a recuperare l'informazione sui bit corrotti.

Il codice scelto è di tipo convoluzionale, una famiglia di codici a correzione d'errore molto diffusa per la sua semplicità d'implementazione, infatti l'encoder usa semplici registri e sommatore binari. Una delle caratteristiche principali di questo tipo di codice è che i bit in uscita codificati dipendono sempre dal bit attuale e dai bit precedenti, quindi l'encoder può essere implementato come una macchina a stati finiti in cui da ogni stato è possibile passare ad un numero limitato di altri stati.

I parametri di questi codici sono i seguenti:

- k è il numero di bit informativi per ogni parola di codice prodotta;
- n è il numero di bit per ogni parola di codice;
- m è la memoria dell'encoder, cioè il numero di registri usati.

Per questo progetto sono stati scelti come parametri $(k,n,m) = (1,2,4)$ ottenendo quindi un codice con $rate = \frac{1}{2}$, che rappresenta appunto il rapporto tra numero di bit informativi e numero di bit codificati; il $rate$ indica l'efficienza del codice, quindi

più è elevato il rapporto, maggiore sarà la quantità di bit informativi presenti in una parola di codice, a scapito però della capacità di correzione; in questo caso quindi vengono impiegati la metà dei bit effettivamente trasmessi per poter recuperare l'informazione sui dati utili trasmessi. Nonostante questo 'spreco' di risorse, i vantaggi sono notevoli. La memoria dell'encoder m è la lunghezza binaria con cui sono rappresentati gli stati: più lunghi sono gli stati, maggiore sarà la capacità di correzione del codice in ricezione, ma al contempo aumenta il numero di stati (raddoppia per ogni bit aggiuntivo) e questo comporta un aumento di complessità del decoder. Il numero di bit scelto per rappresentare gli stati è un buon compromesso tra semplicità dell'architettura e capacità di correzione. Lo schema a blocchi dell'encoder per il codice utilizzato è riportato in Figura 3.13, dove si notano i due bit di uscita C_1 e C_2 , che vengono poi accodati in serie, per ogni bit in ingresso che scandisce anche i cicli con cui l'encoder lavora.

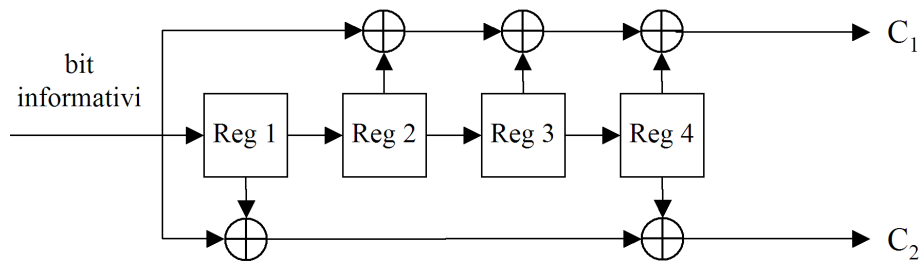


Figura 3.13: Schema encoder convoluzionale

Essendo $m = 4$ il numero di registri, il numero di stati possibili in cui si può trovare l'encoder in ogni istante è $2^4 = 16$, mentre il numero di stati in cui può andare il sistema è $2^1 = 2$ poiché ad ogni ciclo arriva un solo bit informativo, che può essere '0' o '1', ugualmente in ogni stato si può arrivare da due possibili stati precedenti. Un motivo di semplicità di questo tipo di encoder è che ogni stato, rappresentato con 4 bit, ha gli ultimi 3 bit uguali ai primi 3 bit dello stato precedente, mentre, come si è visto, il primo bit è sempre il bit informativo.

La procedura di funzionamento è ben schematizzabile in un diagramma a traliccio (*trellis*): gli stati vengono elencati ad ogni passo di codifica e per ogni bit informativo c'è una transizione al blocco successivo, così facendo si crea un *percorso* che attraversa tutto il trellis passando per stati diversi a seconda della sequenza di bit informativi che arriva. Quindi viene creata una unica sequenza di stati univoca e legata a quella determinata sequenza di bit in ingresso; una sequenza di bit con

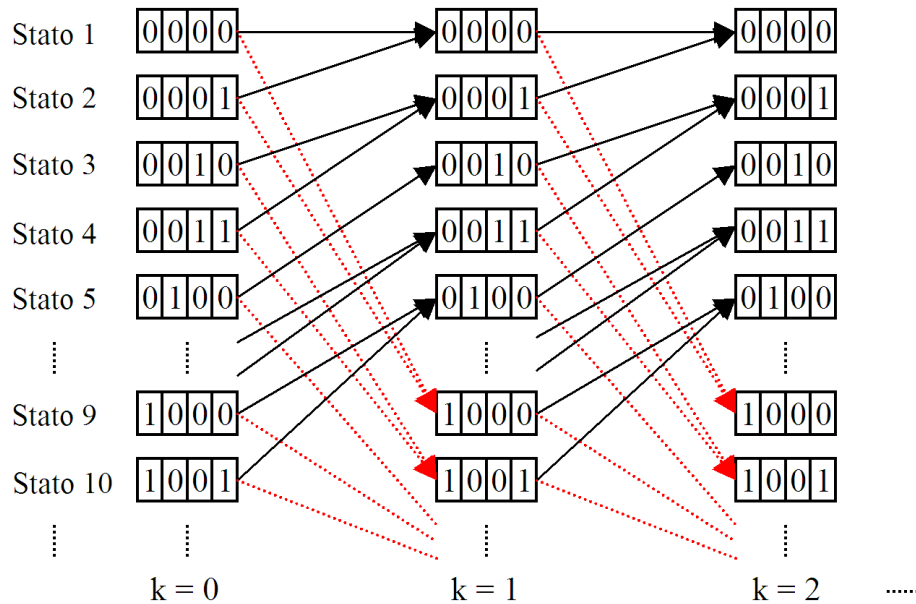


Figura 3.14: Funzionamento a passi dell'encoder, k indica i cicli di clock, le transizioni dovute all'arrivo di uno '0' sono rappresentate con una freccia nera solida, mentre le transizioni per l'arrivo di '1' sono rosse tratteggiate. Non tutte le transizioni sono rappresentate.

anche un solo bit di differenza determinerà una diversa sequenza di stati, sempre univoca. Questo fa sì che la sequenza sia decodificabile al ricevitore. Lo schema di funzionamento è riportato in Figura 3.14.

L'interleaver

I codici convoluzionali sono robusti contro gli errori statisticamente indipendenti, ma non contro gli errori *a raffica* (*burst* in inglese), che sono causati da fenomeni improvvisi con una durata più o meno breve che perturbano il canale corrompendo il segnale durante quel lasso di tempo. Siccome il decoder, come si vedrà, decodifica per sequenze, risulterebbe molto dannoso ricevere una parte di bit corrotti adiacenti, poiché falserebbe l'individuazione dell'esatta sequenza trasmessa. Genericamente questi errori sono più incisivi sui canali wireless (tipici esempi sono il passaggio di un oggetto metallico tra trasmettitore e ricevitore, oppure un fulmine durante un temporale, ecc...), ma in questo specifico caso in studio fenomeni simili sono ugualmente presenti nelle vicinanze del cavo, possono essere ad esempio il passaggio di un treno (che con la motrice di potenza può creare una induzione di

corrente improvvisa), oppure l'attivazione di uno scambio, o anche più facilmente la vicinanza di cavi di potenza vicini che aprono e chiudono circuiti, e altri.

Si rende perciò utile l'impiego dell'*interleaver* ('mescolatore') che permuta l'ordine dei bit della sequenza binaria secondo uno schema ben preciso, in modo da rendere tutti gli errori del canale quasi perfettamente indipendenti. Al ricevitore ci sarà il *de-interleaver* che, con uno schema inverso, riassegnerà la stessa successione dei dati binari trasmessi prima di decodificarli.

La tecnica di *interleaving* consiste nello scrivere i bit per righe in una matrice $M_1 \times M_2$ e costruire la sequenza di uscita leggendo per colonne, così facendo gli eventuali errori a raffica sul canale che andrebbero a corrompere alcuni bit consecutivi, in ricezione verrebbero 'distribuiti' intervallandosi ai bit corretti, consentendo una migliore decodifica.

Le dimensioni della matrice, che può anche non essere quadrata, possono essere dimensionate a seconda del luogo dove verrà installato l'impianto in base alla cadenza e alla durata dei disturbi improvvisi. Per le simulazioni la dimensione scelta è 16×16 .

3.2.4 Riepilogo livello fisico del trasmettitore

Viene proposto uno schema a blocchi completo del livello fisico del trasmettitore. La sequenza completa quindi è:

1. i bit informativi del livello 2 arrivano con rate $R_b \frac{\text{bit}}{s}$ e corrispondente periodo di bit di $T_b = \frac{1}{R_b}$;
2. l'interleaver non modifica il rate quindi in uscita il periodo di bit è ancora T_b ;
3. l'encoder produce 2 bit codificati per ogni bit informativo, quindi il rate di uscita sarà doppio diventando $R_{cod} = 2R_b$, così anche il periodo di bit dimezzerà $T_{cod} = \frac{1}{R_{cod}} = \frac{T_b}{2}$;
4. il modulatore 4-QAM associa coppie di bit ad un simbolo che ha una parte reale $a_{k,I}$ e una immaginaria $a_{k,Q}$, entrambe con periodo di simbolo T che corrisponde al periodo di bit informativo T_b , il legame tra periodo di simbolo e banda occupata (minima) è $B = \frac{1}{T}$;

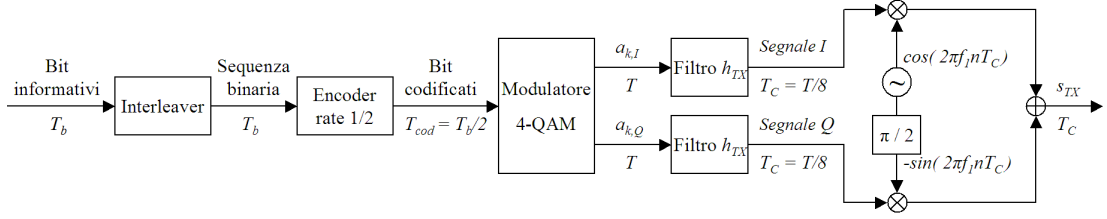


Figura 3.15: Schema a blocchi completo del livello fisico

5. i due filtri di trasmissione h_{TX} fungono sia da modellatori di impulso, sia da interpolatori portando il periodo di campionamento del segnale a $T_C = \frac{T}{8}$;
6. infine i segnali vengono traslati con le rispettive portanti e sommati, mantenendo il periodo di campionamento T_C .

3.3 Modello del canale

Come visto nella Sezione 3.1, il modello del canale utilizzato nelle simulazioni è riportato in Figura 3.4. A valle del canale viene sommato il rumore bianco gaussiano, che è caratterizzato da uno spettro costante a banda larga, convenzionalmente indicato con $\frac{N_0}{2}$. La banda su cui viene applicato il rumore ha larghezza pari a $\frac{1}{T_C}$ (in banda passante), quindi risulta che la potenza statistica del rumore, che coincide con la sua varianza essendo a media nulla, è $M_W = \sigma_W^2 = \frac{N_0}{2T_C}$. La potenza statistica del segnale S_C , che è il segnale all'uscita del canale, viene indicata con M_{S_C} . Il rapporto segnale/rumore è definito come il rapporto tra la potenza statistica del segnale e la potenza statistica del rumore, nel caso di un segnale QAM usato nelle simulazioni risulta

$$SNR = \Gamma = \frac{M_{S_C}}{N_0(1/T)} = \frac{M_{S_C}}{2\sigma_W^2 T_C(1/T)} = \frac{M_{S_C} Q_0}{2\sigma_W^2} \quad \text{dove } T_C = \frac{T}{Q_0}.$$

Da questo si determina la deviazione standard che deve avere il rumore bianco gaussiano generato nelle simulazioni, che risulta essere

$$\sigma_W = \sqrt{\frac{M_{S_C} Q_0}{2\Gamma}}.$$

3.4 Architettura del ricevitore

Il segnale trasmesso S_{TX} passa attraverso il canale visto precedentemente e arriva al ricevitore. In questo progetto, essendo lo studio di fattibilità incentrato sulle prestazioni del cavo a medie-lunghe distanze e sull'architettura digitale, si dà per certa e implicita la presenza, oltre che del DAC in trasmissione prima del canale, di un *front end* composto da un circuito di sincronizzazione analogico a monte del ricevitore e al campionatore ADC (*Analog-to-Digital Converter*) che avrà un certo offset temporale t_0 di campionamento a tempo T_C , fornendo così il segnale ricevuto con periodo di campionamento uguale a quello del segnale trasmesso; in Figura 3.16 sono riportati i blocchi dei circuiti hardware la cui presenza è sottintesa ma non analizzata poiché non sono incisivi sulle prestazioni valutate in questo progetto.

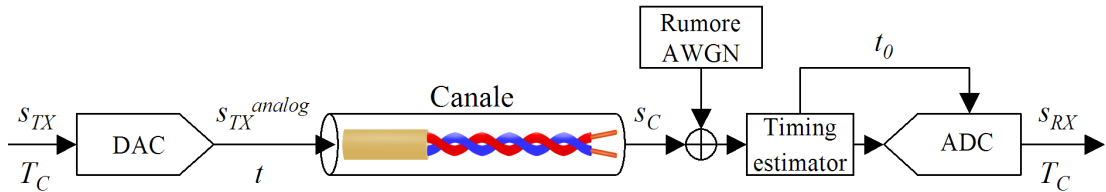


Figura 3.16: Schema a blocchi del canale e della parte analogica

3.4.1 L'Equalizzatore

A valle del convertitore ADC si riceve un segnale campionato a tempo T_C , l'architettura proposta pone l'equalizzatore in questo punto con ottimi risultati: la distorsione del segnale, composto dalla somma delle componenti reale e immaginaria dei simboli trasmessi, viene compensata tramite la moltiplicazione per i coefficienti dell'equalizzatore. Il tipo scelto è lineare adattivo, poiché fornisce buone prestazioni in presenza sia di ISI che di rumore [4]. Non avendo una conoscenza a priori del canale, il funzionamento dell'equalizzatore è diviso in due fasi:

- fase di training, in cui adatta i coefficienti in base ad una forma d'onda nota ricevuta;
- fase di funzionamento a regime, detta *Decision directed mode*.

In modalità di training viene trasmessa una sequenza di bit nota, che verrà modulata in una forma d'onda che il ricevitore conosce (possibilmente memorizzata precedentemente in una memoria flash da cui legge i valori dei campioni e li confronta

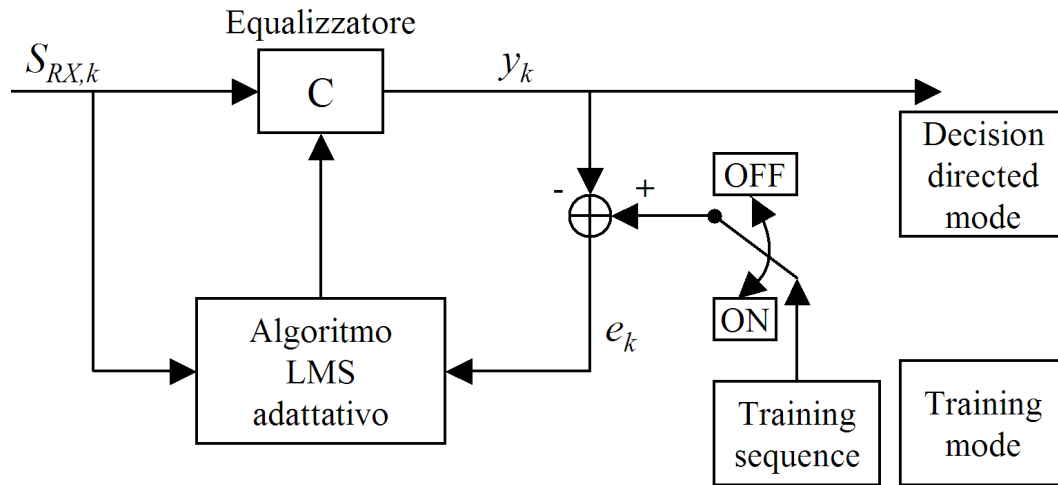


Figura 3.17: Schema a blocchi dell'equalizzatore

con quelli ricevuti) adattando i coefficienti in modo da ottenere una forma d'onda equalizzata il più possibile simile a quella nota pre-memorizzata. Il funzionamento della fase di training viene riassunto di seguito:

1. viene calcolata la correlazione mutua tra il segnale ricevuto e quello pre-memorizzato, il picco massimo della correlazione indica il ritardo D introdotto dal canale, espresso in numero di campioni;
2. la differenza di lunghezza (in numero di campioni) tra il segnale ricevuto e la sequenza nota è la lunghezza del canale N ;
3. una volta ottenuti i dati di cui sopra, inizia l'algoritmo adattativo a minimi quadrati LMS (*Least mean-square*).

Il nucleo funzionale è l'algoritmo adattativo: il numero di coefficienti C_i viene impostato pari alla lunghezza del canale determinata prima N , i campioni ricevuti vengono inseriti a scorrimento in N registri; ad ogni ciclo, ogni campione dei registri viene moltiplicato per il coefficiente corrispondente e i risultati vengono sommati producendo il campione y_k dove k indica l'istante temporale; da qui viene determinato l'errore attuale calcolando la differenza tra y_k e il campione di riferimento corrispondente della sequenza nota, vengono così modificati i coefficienti per il prossimo ciclo. La formula di aggiornamento dei coefficienti, derivante

dall'algoritmo LMS è in forma vettoriale:

$$\overline{C}_{k+1} = \overline{C}_k + \mu e_k \overline{S_{RX,k}}$$

dove $\overline{S_{RX,k}}$ indica il vettore dei campioni del segnale S_{RX} dall'istante $k - N + 1$ a k . In Figura 3.18 viene schematizzata la struttura in fase di training.

Schema equalizzatore in modalità training

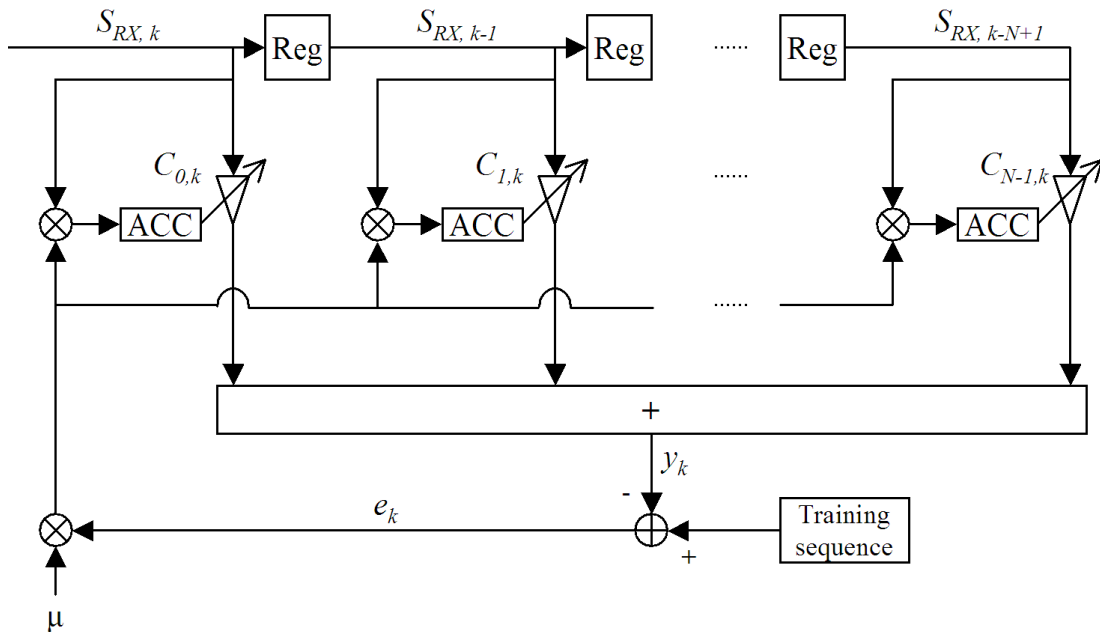


Figura 3.18: Struttura funzionale dell'equalizzatore lineare adattivo

Durante la fase di training ogni telefono riceve la stessa *training sequence* a turni dagli altri telefoni, che si trovano a distanze diverse, quindi deve adattare i coefficienti per gli altri 7 apparecchi presenti nell'anello, per fare ciò è prevista una memoria RAM in cui organizza i dati come una matrice, in cui in ogni riga scriverà i coefficienti associati ad un determinato telefono, che quindi saranno diversi appunto per la distanza differente.

Terminata la fase di training, nella quale i coefficienti C_i arrivano a convergenza con una sufficiente lunghezza della sequenza nota, l'equalizzatore passa in modalità di decisione diretta usando i coefficienti precedentemente calcolati, oltre che al parametro D , indicante il ritardo introdotto dal canale.

Schema equalizzatore in modalità diretta

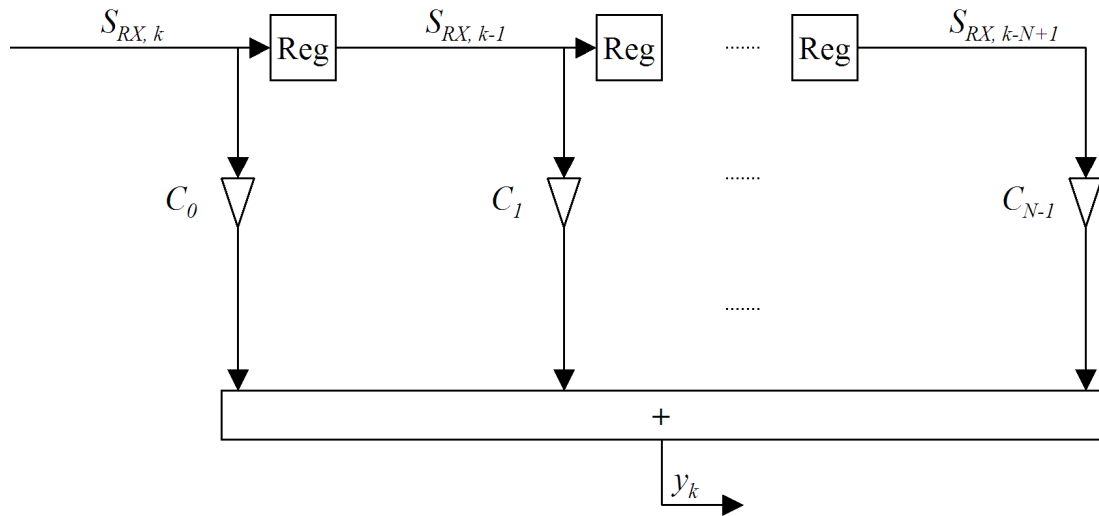


Figura 3.19: Equalizzatore in modalità decisione diretta

3.4.2 Il Costas loop

L'implementazione QAM scelta fa uso di filtri in banda base, il ricevitore quindi, dopo l'equalizzatore, trasla lo spettro dalla banda passante moltiplicando il segnale ricevuto per una sinusoide che deve essere identica in frequenza e fase alla portante del segnale ricevuto.

Il segnale che arriva dal canale non contiene componenti periodiche con periodo $\frac{n}{f_0}$ con n intero, quindi non può essere usato un semplice PLL (*Phase-Locked Loop*) per mantenere l'aggancio di fase tra l'oscillatore locale e il segnale ricevuto, quindi si ricorre all'impiego del più complesso circuito *Costas loop*, che svolge due funzioni: compensa piccole variazioni di fase introdotte da possibili disturbi che intervengono casualmente sul canale e separa la parte reale e quella immaginaria dal segnale ricevuto [4]. Prima della demodulazione, lo spettro del segnale ricevuto deve essere riportato in banda base, per ottenere ciò viene moltiplicato per una sinusoide alla stessa frequenza f_0 e poi viene filtrato come si vedrà.

Si supponga che il segnale S_{EQ} , in uscita dall'equalizzatore e in banda passante, abbia subito una variazione di fase, indicata con ϕ , della portante a frequenza f_0 ,

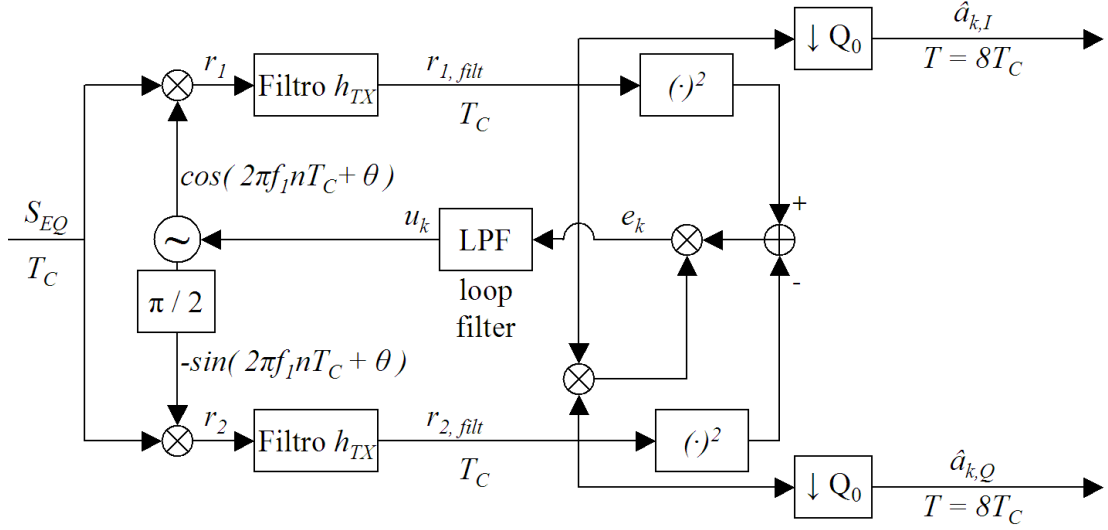


Figura 3.20: Circuito Costas loop

avendo così un segnale del tipo:

$$S_{EQ} = S_I \cos(2\pi f_0 n T_C + \phi) - S_Q \sin(2\pi f_0 n T_C + \phi) + W$$

dove W è il rumore. Come anticipato, il compito del Costas loop è allineare la fase della portante del segnale ricevuto con quella dell'oscillatore locale. L'oscillatore locale ha una sua fase variabile θ che viene continuamente modificata proporzionalmente all'errore di fase del segnale ricevuto, quindi anche se durante la trasmissione avvenisse una perturbazione del segnale, la sincronia verrebbe mantenuta. Si vede ora il funzionamento (analizzando il segnale della parte reale): moltiplicando S_{EQ} per la sinusoide dell'oscillatore del ricevitore si ottiene

$$\begin{aligned} r_1 &= S_I [\cos(2\pi f_0 n T_C + \phi) \cos(2\pi f_0 n T_C + \theta)] - S_Q [\sin(2\pi f_0 n T_C + \phi) \cos(2\pi f_0 n T_C + \theta)] + W \\ &= S_I \left[\frac{1}{2} \cos(2\pi 2f_0 n T_C + \phi + \theta) + \frac{1}{2} \cos(\phi - \theta) \right] - S_Q \left[\frac{1}{2} \sin(2\pi 2f_0 n T_C + \phi + \theta) + \frac{1}{2} \sin(\phi - \theta) \right] \end{aligned}$$

Dopo la traslazione in banda base sono tuttavia ancora presenti le immagini dello spettro a frequenza doppia, come si vede in $\cos(2\pi 2f_0 n T_C + \phi + \theta)$, quindi è necessario l'impiego del filtro *matched* passa-basso h_{TX} che è identico al filtro

usato nel trasmettitore; a valle del filtro il segnale è

$$r_{1, filt} = \frac{1}{2}S_I \cos(\phi - \theta) - \frac{1}{2}S_Q \sin(\phi - \theta)$$

che, in condizioni di perfetto allineamento della fase del segnale ricevuto e di quella dell'oscillatore (cioè quando $\phi = \theta$), diventa un segnale composto dalla parte reale dei simboli ricevuti:

$$r_{1, filt} = \frac{1}{2}S_I$$

Discorso analogo duale si ha per il segnale della parte immaginaria, ottenendo così

$$r_{2, filt} = \frac{1}{2}S_Q$$

Ad ogni ciclo di clock i due segnali vengono consegnati al demodulatore campione per campione, mentre il resto del circuito è composto da varie operazioni, tra cui il calcolo del quadrato, moltiplicazioni e somme, per poter costruire il segnale e_k che ha la componente 'DC', cioè a frequenza nulla, uguale a zero solo se le fasi coincidono, mentre se ha un valore diverso da zero, questo è proporzionale allo sfasamento tra le due sinusoidi; per ottenere un valore preciso quindi si filtra il segnale e_k con il *loop filter* che è un filtro passa-basso con banda strettissima, composto da un numero di coefficienti pari a 128, così da ottenere il segnale u_k che comanda l'oscillatore controllato numericamente. Ad ogni ciclo quindi la differenza di fase viene corretta fino ad arrivare ad una perfetta coincidenza che permetterà poi di avere una *detection* ottima dei simboli.

I segnali in uscita dal Costas loop vengono decimati con fattore Q_0 per ottenere così le parti reale e immaginaria dei simboli ricevuti a tempo T , rispettivamente $\hat{a}_{k,I}$ e $\hat{a}_{k,Q}$.

3.4.3 Il Demodulatore

La robustezza contro il rumore dei simboli della modulazione 4-QAM è data dalla loro disposizione: in ricezione è sufficiente determinare in quale quadrante del piano complesso si trova il simbolo corrente ricevuto analizzando il segno delle parti reale e immaginaria ricevute, che vengono rispettivamente associate ad un bit '0' se sono positive e a '1' se negative, come si può vedere nella Figura 3.7 che riporta lo schema del *bitmapper*. Viene così creato il *bitstream* ricevuto, che

verrà riordinato dal *de-interleaver* secondo la sequenza con cui sono stati generati dall'encoder di trasmissione.

3.4.4 Il Decoder Viterbi

Per decodificare il flusso di bit in uscita dall'interleaver è stato scelto l'algoritmo di Viterbi poiché molto efficiente nell'implementazione del criterio di massima verosimiglianza, che consiste nel massimizzare la probabilità che una sequenza ricevuta sia 'più vicina' possibile a quella effettivamente trasmessa.

L'algoritmo di Viterbi non è particolarmente costoso in termini di risorse hardware, nel senso che richiede una modesta capacità di calcolo ma una buona quantità di memoria per la necessità di memorizzare lunghe sequenze di stati; il funzionamento si basa sull'emulazione del comportamento dell'encoder che, come si è visto nella Sezione 3.2.3, è una macchina a stati finiti che passa da uno stato all'altro creando la sequenza di bit codificati secondo lo schema per cui ad ogni bit informativo crea una coppia di bit in uscita effettuando una transizione di stato inserendo il bit di informazione come primo bit dello stato di arrivo. Per la configurazione dell'encoder, il numero di possibili stati successivi è 2 e ad ogni stato è possibile arrivare da 2 stati precedenti.

Il concetto base dell'algoritmo, che permette di selezionare la sequenza a massima verosimiglianza, è l'uso di due variabili: una è la *branch metric*, che è la metrica con cui si misura la 'distanza' tra due parole di bit in una transizione, l'altra è il cosiddetto *path cost* di ogni percorso, composto dalla somma delle *branch metric* precedenti che hanno portato allo stato corrente. Inizialmente vengono assegnati costi alti ad ogni percorso per non creare un percorso 'privilegiato'. L'inizializzazione dell'algoritmo ha due fasi di scrittura nella memoria: prima vengono creati i 16 stati (che sono composti da tutte le possibili combinazioni dei 4 bit che li rappresentano), poi vengono calcolate tutte le possibili transizioni da tutti gli stati associando ad ognuna la parola di codice di due bit e il bit di informazione corrispondente replicando il comportamento dell'encoder. Inoltre vengono create 16 variabili che saranno i *path cost* dei possibili percorsi (ognuno parte da ciascuno dei 16 stati diversi).

Finita la fase iniziale, l'algoritmo crea passo-passo il trellis di tutti i 'percorsi' che passano attraverso gli stati: per ogni istante k arrivano due bit codificati e comincia un ciclo nel quale viene analizzato uno stato alla volta osservando quali

sono i due possibili stati precedenti e per ognuna delle due transizioni viene confrontata la parola attuale di bit codificati con le parole codificate corrispondenti, viene calcolata la *branch metric* per ciascuna parola e viene scartata quella con costo più alto, sommando così il costo della transizione ‘vincente’ al precedente *path cost* di quel percorso. Quindi ad ogni ciclo vengono aggiornati i costi e vengono memorizzati gli stati a minor costo dei 16 percorsi che si sono creati fino a quell’istante.

Ci sono due metodi per determinare la distanza tra la parola di codice corrente e tutte le possibili parole codificate, questi due metodi differenziano anche il tipo di decodifica e parte dell’architettura del decoder stesso:

Hard input Questo tipo di decodifica è la più semplice sia per quanto riguarda la complessità computazionale, sia per la facilità di implementazione fisica. La *branch metric* viene determinata in base alla *distanza di Hamming* tra le parole di codice corrente da confrontare, che consiste nella quantità di bit differenti ci sono, quindi ad esempio due sequenze binarie uguali avranno 0 come distanza. Il decoder con architettura *Hard input* riceve in ingresso la sequenza di bit codificati provenienti dall’uscita del demodulatore 4-QAM e non ha bisogno di ulteriori calcoli o operazioni per decodificare. Ha una complessità computazionale bassa grazie all’uso di XOR binari per calcolare le distanze di Hamming, questo porta anche ad una più semplice implementazione fisica.

Soft input Questa modalità di decodifica ha una complessità leggermente maggiore ma offre prestazioni superiori. Si basa sulla distanza euclidea dei simboli complessi, in ingresso non riceve una sequenza di bit, ma a monte c’è un detector al posto del demodulatore 4-QAM che in base ai simboli ricevuti calcola le metriche che saranno poi le distanze dei simboli codificati. Il calcolo di queste metriche avviene considerando anche il livello di rumore del canale, consentendo di fornire un indice di attendibilità della metrica, in particolare viene sfruttato il fatto che la parte reale di un simbolo è rappresentato dal primo dei due bit codificati (il secondo bit rappresenta la parte immaginaria): esso infatti è ‘1’ quando la parte reale è positiva, mentre è ‘0’ quando è negativa, come si può vedere in Figura 3.7. Quindi i simboli ricevuti, corrotti dal rumore e dalla distorsione del canale, possono dare un’indicazione del bit

codificato che rappresentano, perciò la decodifica *Soft input* usa un rapporto, detto *Likelihood ratio*, proporzionale a $\frac{\hat{a}_{k,I}}{\sigma_W^2}$ (per la parte reale, mentre per la parte immaginaria è proporzionale a $\frac{\hat{a}_{k,Q}}{\sigma_W^2}$) per creare le metriche che poi verranno inviate al decoder. σ_W^2 indica la potenza statistica, nonché la varianza, del rumore: più rumore c'è, più la metrica di un determinato simbolo sarà piccola, 'avvicinando' il simbolo corrente agli altri simboli, cioè andando verso una equidistanza dai vari simboli indicando così una maggiore incertezza. Mentre in caso di rumore molto basso, la distanza del simbolo sarà 'rinforzata' dalla metrica che indicherà la vicinanza netta ad un determinato simbolo rispetto agli altri 3. In questa modalità quindi il decoder non riceve in ingresso né bit né simboli codificati, ma direttamente l'elenco di metriche pesate calcolate prima dal detector. La complessità è leggermente superiore all'Hard input poiché richiede il calcolo aritmetico delle metriche.

Arrivato alla fine del *bitstream* di bit codificati si avranno 16 percorsi diversi ciascuno con l'elenco degli stati a minor costo, detti *survivors*, si seleziona il percorso a minore costo totale, che quindi contiene una determinata sequenza di stati, e viene 'ricostruito' a ritroso il percorso leggendo il primo bit di ogni stato e scrivendolo in uscita dal decoder: questa operazione è chiamata *traceback*, si crea così la sequenza più vicina, secondo il concetto di massima verosimiglianza, a quella trasmessa.

Un inconveniente di questo algoritmo è l'assenza di una limitazione dell'uso della memoria: per una sequenza in ingresso continua di bit codificati il decoder aggiunge gli stati vincenti a ciascuna sequenza senza limite. Per ovviare a ciò, arrivato ad una certa *profondità* del trellis sufficientemente grande, vengono memorizzati i costi dei percorsi e vengono scritti in uscita i bit decodificati fino a quel momento, poi ricomincia la costruzione del trellis assegnando agli stati come costi iniziali quelli raggiunti precedentemente, procedendo nella stessa maniera.

Il rate di uscita del decoder sarà la metà di quello di ingresso, riportando così il periodo di bit uguale a quello di trasmissione dei bit di informazione: $R_b = \frac{R_{cod}}{2}$.

Il deinterleaver

Il *deinterleaver* esegue semplicemente l'operazione duale dell'interleaver: scrive i bit in una matrice $M_1 \times M_2 = 16 \times 16$ (uguale a quella di trasmissione) per

colonne e poi li legge per righe riportandoli in uscita, ricomponendo così l'ordine di trasmissione dei bit informativi.

3.4.5 Riepilogo ricevitore

Riepilogando, il ricevitore, riportato in Figura 3.21, segue i seguenti passi:

1. arriva dal canale il segnale S_{RX} campionato a tempo T_C distorto dal canale e corrotto dal rumore;
2. l'equalizzatore moltiplica linearmente i campioni ricevuti per i coefficienti ottimi trovati nella precedente fase di training costruendo così il segnale equalizzato S_{EQ} con periodo di campionamento uguale all'ingresso;
3. il Costas loop moltiplica il segnale per una portante uguale a quella di trasmissione e filtra con un filtro *matched* per ottenere così un segnale in banda base come quello trasmesso, inoltre mantiene allineata la fase della portante e dell'oscillatore, in uscita decima con un periodo 8 volte più alto, fornendo così due segnali separati che sono i campioni delle parti reale e immaginaria dei simboli;
4. il demodulatore 4-QAM esegue una mappatura inversa (da simboli a bit) nel caso di decodifica con Hard input fornendo i bit in uscita, mentre in caso di Soft input c'è un detector che fornisce le metriche in uscita;
5. il decoder decodifica il bitstream con l'algoritmo di Viterbi, essendo il rate del codice $\frac{1}{2}$ in uscita fornisce 1 bit ogni 2 di ingresso, quindi il rate di uscita sarà dimezzato: $R_b = \frac{R_{cod}}{2}$;
6. il de-interleaver esegue l'operazione inversa dell'interleaver del trasmettitore riposizionando i bit nel corretto ordine, consegnandoli così al livello superiore.

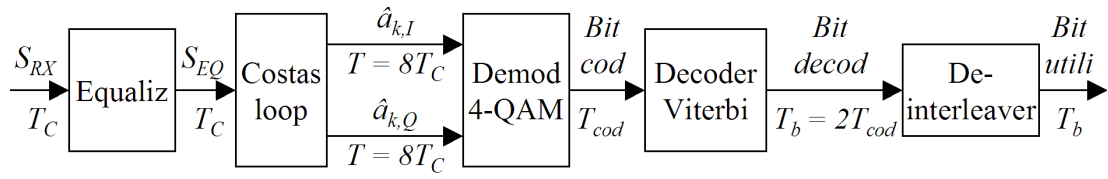


Figura 3.21: Schema a blocchi completo del ricevitore con decodifica Hard Input, nella versione con decodifica a *Soft detection* invece il demodulatore è sostituito con un detector che calcola le metriche

Capitolo 4

Il livello link layer

Il *link layer* è il secondo livello dello stack protocollare, si interfaccia con il livello fisico e con il livello di rete (livello 3, IP). Il compito principale è gestire l'accesso al mezzo, che è a turni temporali ciclici con tecnica *Time Division Multiple Access*, come visto nel Capitolo 2.

4.1 Composizione del frame

La composizione del *frame* a livello Link Layer, riportata in Figura 4.1, prende spunto dalla specifica Ethernet 802.3 [8], ma è stata modificata mantenendo i campi utili in questo ambito ed eliminando i campi non necessari per la riduzione della dimensione totale:

- *Preamble* ha la funzionalità hardware per la sincronizzazione del clock del ricevitore, è composto da 7 parole di 8 bit ciascuna contenente bit '0' e '1' alternati, con lunghezza complessiva di 56 bit;
- SFD *Start-of-Frame Delimiter* serve a indicare l'inizio dell'header, è formato da una parola di 8 bit con bit alternati eccetto gli ultimi due bit posti a '1';
- *MAC Source* e *MAC destination* sono gli indirizzi fisici degli adattatori di rete rispettivamente del trasmettitore e del ricevitore, la somma delle due lunghezze è 96 bit¹;

¹La lunghezza degli indirizzi può essere ridotta considerato che gli apparecchi nell'anello sono 8, quindi basterebbero 3 bit ad indirizzo consentendo così di ridurre a 6 bit la lunghezza complessiva degli indirizzi del mittente e destinatario

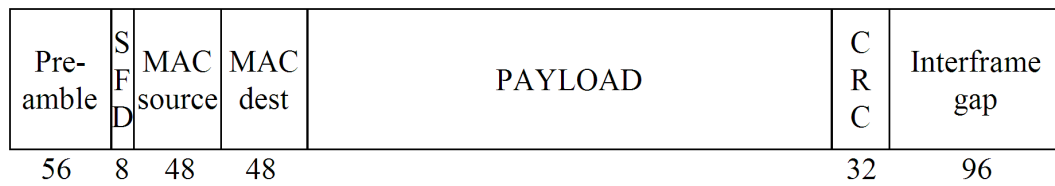


Figura 4.1: Struttura di un pacchetto di livello 2 link layer, i numeri in basso rappresentano la lunghezza di ciascun campo espressa in numero di bit

- *Payload*, sono i bit informativi provenienti dal livello superiore da trasmettere, ha una lunghezza diversa a seconda del periodo di pacchettizzazione impostato ai livelli superiori, che verrà studiato successivamente;
- *CRC* di 32 bit per il controllo di integrità del pacchetto;
- *Interframe gap*, di lunghezza 96 bit, è impiegato per far ‘assestare’ i circuiti hardware e liberare il canale da eventuali segnali ancora in volo prima della successiva trasmissione.

L’overhead a livello 2 quindi ha una lunghezza di 288 bit.

4.2 Accesso al mezzo

La tecnica di accesso al mezzo TDMA prevede la suddivisione dei frame temporali in slot in cui ogni apparecchio può trasmettere in maniera esclusiva.

La procedura di assegnazione dei turni (slot) inizia dal centralino: nella fase iniziale nessun dispositivo dell’anello ha l’assegnazione ad uno slot e tutti i telefoni rimangono in ricezione, solo il centralino trasmette e il suo slot diventa il primo dando avvio all’inizio del *frame* e creando così il riferimento per la sincronizzazione del sistema. I primi dati che vengono trasmessi sono costituiti dalla *sequenza di training* necessaria agli equalizzatori dei telefoni per adattare i parametri e arrivare a convergenza. Dalle simulazioni è risultato che una convergenza stabile e ottima è raggiunta con una lunghezza in numero di bit pari a 30000, quindi, come si vedrà, in base al bitrate disponibile saranno necessari più turni di trasmissione del centralino per trasmettere tutta la sequenza di training. Finita la fase di training, il centralino informa i telefoni sulla possibilità di iniziare a trasmettere a loro volta a partire dal turno successivo.

I telefoni quindi non trasmettono fino a quando non ricevono il pacchetto di fine training. Inizia così la fase di assegnazione dinamica degli slot: ciascun telefono attende un tempo casuale e trasmette nello slot successivo la stessa sequenza di training, il centralino rimane in ascolto ricevendo tutte le trasmissioni per tutta la durata del frame temporale, in cui può accadere che due o più telefoni trasmettano nello stesso slot creando così una collisione; quando arriva nuovamente il suo turno di trasmissione, il centralino informa i telefoni delle assegnazioni avvenute con successo indicando in quale slot viene assegnato ciascun MAC address. I telefoni non ancora assegnati ad uno slot ricevono le informazioni su quali slot sono ancora liberi e, dopo aver atteso un tempo di *backoff*, riprovano a trasmettere negli slot rimasti liberi. Procedendo così a ripetizione tutti gli slot verranno assegnati ai telefoni presenti. Durante la fase iniziale, appena ciascun telefono viene assegnato ad uno slot, invia come prima cosa la sequenza di training nota a tutti, cosicché ciascun altro telefono (e il centralino) in ricezione apprenda i parametri dell'equalizzatore da quel determinato trasmettitore.

4.2.1 Rimozione, Aggiunta o Sostituzione di un telefono

L'impianto telefonico in studio non è destinato ad un uso intenso, ma piuttosto sporadico, perciò non si avrà un traffico continuo di dati lungo l'anello, ma ci saranno lunghi periodi di inattività. Per questo motivo se un telefono dovesse smettere improvvisamente di funzionare o venisse scollegato, gli altri dispositivi non se ne accorgerebbero, quindi dovrebbe essere segnalato esplicitamente in modo da informarli che quella destinazione non è più disponibile. Per questo scopo i dispositivi, quando sono in stato di inattività, dovranno trasmettere a intervalli regolari, ad esempio 30 secondi o 1 minuto o altro, un pacchetto breve di *keep alive* col quale informano gli altri dispositivi della loro presenza, successivamente trasmettono nuovamente la sequenza di training per eventuali nuovi telefoni, come si vedrà nella prossima sezione. Se non viene ricevuto il pacchetto *keep alive* di un determinato telefono per 2 volte consecutive (quindi ogni 1 o 2 minuti o altro), gli altri cancellano dalla tabella delle presenze il suo MAC address.

Nel caso di aggiunta di un telefono all'anello questo rimane in ascolto senza trasmettere per almeno 1 o 2 minuti (a seconda della configurazione scelta), apprendendo e sincronizzandosi intanto con gli slot e ricevendo quindi le sequenze di training da tutti gli altri. Dopo il periodo di ascolto individua il primo slot vuoto

e trasmette la sequenza di training per segnalare la sua presenza e far adattare gli equalizzatori degli altri dispositivi, il centralino confermerà l'assegnazione dello slot.

In caso di necessità di sostituzione di un telefono è quindi sufficiente scollegare il 'vecchio' telefono e collegare quello nuovo, in questo modo la rete si accorgerà della scomparsa del primo e registrerà la presenza di quello nuovo dopo pochi minuti.

4.3 Buffer

Nella durata di uno slot tutti i 7 dispositivi che ricevono inseriscono il pacchetto in un buffer (con capacità che può essere di un solo pacchetto), solo dopo la fine della completa ricezione il pacchetto verrà letto ed elaborato così da 'distribuire' il carico di lavoro e non avere un picco di assorbimento di corrente elevato di breve durata.

Il livello di link layer avrà un altro buffer secondario in cui dovrà mantenere l'ultimo pacchetto corretto finché non ne arriva un altro senza errori, questo è necessario poiché se durante una conversazione viene ricevuto un pacchetto che il CRC rivela essere corrotto, questo viene scartato e al suo posto viene passato ai livelli superiori l'ultimo pacchetto precedente corretto, così una eventuale raffica di errori sul canale avrebbe un periodo 'tampone' nel quale non si annulla la comunicazione. Questa tecnica è preferibile piuttosto che passare ai livelli superiori un pacchetto vuoto poiché offre una continuità migliore per il codec tra un pacchetto di campioni e quello successivo.

Se ci dovesse essere una lunga sequenza di pacchetti errati, o un'assenza di pacchetti, la comunicazione verrà chiusa automaticamente.

4.4 Risparmio energetico, chiamata e risposta

In stato di rete configurata, dopo la fase di training, ogni telefono attua un piano di risparmio energetico che consiste nel disattivare il ricevitore fisico per 2 o 3 frame consecutivi, in modo da non consumare energia per demodulare ogni volta che viene trasmesso qualche pacchetto in rete. Per questo è importante che nei periodi di inattività (nessuna comunicazione in corso) i pacchetti di *keep alive* dovranno

essere trasmessi nello stesso frame (ogni 30 secondi o 1 minuto, come si è visto in precedenza) concordato così da disattivare il risparmio energetico durante quel frame temporale (in modo che tutti i telefoni vengano informati di quali apparecchi sono ancora presenti). Quando il risparmio energetico è attivo, un telefono è detto *in stato di sleep mode*.

Durante una conversazione la regola non cambia: tutti gli altri telefoni non impegnati invieranno il pacchetto di *keep alive* nel frame concordato.

La procedura di chiamata e risposta è semplice: il chiamante inizia a trasmettere a ripetizione un pacchetto di chiamata con il MAC address del destinatario (o i destinatari trasmettendo in più frame) finché non riceve un pacchetto di risposta, che può arrivare con 2 o 3 frame in ritardo poiché il destinatario può trovarsi in *sleep mode* nel momento della prima trasmissione e così si instaura la conversazione. Ovviamente il risparmio energetico in conversazione viene disattivato.

Capitolo 5

I livelli superiori

Per *livelli superiori* si intendono i livelli dal 3 al 7 dello stack protocollare standard ISO/OSI. Grazie all'architettura modulare, è possibile modificare ciascun livello e utilizzarlo in maniera trasparente agli altri livelli con cui si interfaccia, e utilizzare modelli già pronti per gli altri. Per questa caratteristica, per questo progetto è possibile utilizzare dal livello IP in su protocolli standard.

In particolare il livello IP non ha bisogno di implementare particolari algoritmi di routing quindi viene adottato il protocollo standard per quanto riguarda il formato degli indirizzi, la composizione dell'header e le funzionalità. Ogni dispositivo sarà quindi configurato con un proprio indirizzo IP.

In seguito verranno brevemente introdotti i possibili protocolli dei livelli superiori per una implementazione del sistema in esame, fino al livello di applicazione.

5.1 Real-time Transport Protocol e UDP

Per l'applicazione in esame (la trasmissione della voce) il protocollo a livello applicazione adottabile è RTP *Real-time Transport Protocol*, creato apposta per lo *streaming* di contenuti audio/video con l'obiettivo di ridurre al minimo la latenza anche a scapito di una perdita di affidabilità. Il protocollo si occupa di scrivere l'header dei pacchetti provenienti dall'applicazione (prodotti dal codec) inserendo, oltre ad alcune informazioni funzionali, il numero di sequenza e il *timestamp* per poter ricostruire la corretta sequenza e la temporizzazione dei campioni audio in ricezione [10].

RTP viene impiegato principalmente in combinazione con UDP (*User Datagram Protocol*) a livello di trasporto per la trasmissione dei campioni vocali, in modo da introdurre un ritardo minimo nella consegna dei pacchetti e non aggiungere informazioni ridondanti non necessarie. Per le sessioni di controllo invece viene usato il protocollo TCP (*Transmission Control Protocol*), molto più affidabile ma con latenze maggiori.

Compressione degli header

Il ROHC (*Robust Header Compression* [18]) è un metodo standardizzato per la compressione degli header dei livelli di trasporto e di rete, sviluppato specificatamente per applicazioni di trasmissione video e audio su rete IP. La necessità di un metodo di compressione è dettata dal ‘peso’ che hanno gli header, che contengono informazioni non utili agli utenti, dei pacchetti rispetto al payload. L’applicazione VoIP di interesse fa uso del protocollo RTP a livello di applicazione e di IP a livello 3, la cui somma delle lunghezze minime degli header è pari a 40 byte (320 bit), che, a seconda della lunghezza del payload, può rappresentare dal 10 al 20 % della dimensione totale di un pacchetto a livello Link Layer. Il funzionamento del metodo ROHC si basa sul fatto che, nell’ambito delle applicazioni voce in questo caso, la maggior parte dei pacchetti trasmessi abbiano gli header con molti campi uguali, che quindi possono essere omessi e trasmettere le informazioni costanti e ridondanti solo nel primo pacchetto di una sessione di trasmissione.

Con la compressione degli header RTP e IP è possibile ottenere una riduzione della lunghezza da 40 a soli 4 byte e ottenere così un overhead del 1 - 2 % su un pacchetto a livello Link Layer.

5.2 Livello applicazione: i codec

Negli ultimi anni, per l’elevata richiesta di apparati per la trasmissione della voce che sfruttano la tecnologia digitale, si è sviluppata e diffusa una varietà di codec audio tale da soddisfare tutte le esigenze di qualità e compressione per ridurre la banda occupata, che si traduce in un insieme di algoritmi più o meno complessi ed efficienti.

La caratteristica fondamentale di interesse per la trasmissione vera e propria è il *bitrate* con cui vengono prodotti i campioni audio, determinato dalla precisione

con cui vengono quantizzati (espresso in numero di bit) e dalla frequenza di campionamento, che determina la larghezza di banda occupata sul canale. A parità di bitrate, a seconda dell'efficienza del codec, la qualità audio cambia e le differenze per poterla determinare sono basate sia su una tecnica deterministica e oggettiva (misurazione della differenza quadratica tra la forma d'onda audio originale e quella ricostruita) che su un dato soggettivo mediato chiamato *Mean Opinion Score*, un test condotto in ambito telefonico durato molti anni che ha raccolto le opinioni soggettive di persone sulla qualità di una telefonata di prova su una scala da 1 a 5 [12], un codec con un MOS maggiore di 4 è considerato eccellente.

5.2.1 G.711

Uno dei codec di alta qualità più diffusi è il G.711 [14], che nonostante l'anzianità (è stato ideato nel 1972 e rivisto nel 1988) è ancora oggi largamente utilizzato poiché garantisce eccellenti prestazioni senza la necessità di un'alta capacità di calcolo (è classificato come il codec a minor complessità [13]), considerando anche che con le odierne tecniche di modulazione digitale si può contare su alti bitrate disponibili.

Il codec, di tipo *lossy* (cioè con una perdita di fedeltà nella forma d'onda ricostruita), processa sorgenti audio nello spettro 300 - 3400 Hz (range di frequenze in cui tipicamente rientra la voce) campionando a 8 kHz, e usa una quantizzazione logaritmica che produce campioni in uscita con precisione 8 bit (256 livelli), creando un *bitstream* in uscita con una velocità di 64 kbit/s. Il MOS nel caso peggiore (rete altamente impegnata) è 4.11 [12].

Esistono due versioni dell'algoritmo di compressione usato dal codec chiamate μ -law e A-law, la prima è usata soprattutto in Nord America e Giappone, la seconda in Europa e nel resto del mondo. Il funzionamento di entrambi gli algoritmi si basa sul raggruppamento di alcuni valori di ampiezza del segnale PCM lineare in ingresso (con precisione di 13/14 bit per campione) in valori approssimati in uscita, ottenendo così una compressione; il funzionamento dettagliato è consultabile nella specifica ITU-T [15].

G.711.0

Una variante molto più recente, approvata nel 2009, ed efficiente del codec visto in precedenza è il G.711.0, la cui differenza consiste nell'algoritmo di compressione dei dati, di tipo *lossless*, che consente di avere una precisione maggiore della forma d'onda in uscita rispetto all'originale e quindi la qualità risultante è molto elevata. La frequenza di campionamento rimane la medesima e il bitrate d'uscita rimane pari 64 kbit/s.

L'adozione di quest'ultima variante è quindi oggi preferibile rispetto alla sua versione iniziale.

Il codec G.711, in entrambe le sue varianti, può essere configurato per produrre pacchetti in uscita di lunghezza diversa, andando quindi a 'regolare' l'intertempo tra un pacchetto e l'altro. In Tabella 5.1 sono riportate le lunghezze dei pacchetti, con le corrispondenti dimensioni in bit e durata temporale.

Il problema fondamentale dei codec appena visti è il bitrate di uscita: 64 kbit/s per un'applicazione di tipo voce è di un certo peso in termini di occupazione di banda necessaria per la trasmissione e, se quest'ultima non fosse sufficiente, rappresenta un impedimento nella realizzazione dell'impianto. Perciò verrà ora introdotto un altro codec molto diffuso che produce un bitrate di uscita molto inferiore.

5.2.2 G.729

Un altro codec di più recente nascita (la prima versione è del 1996, mentre l'ultima revisione è del gennaio 2007) e con una notevole diffusione è il G.729. Il successo di questo codec è dato dal ridottissimo bitrate in uscita rispetto alla maggioranza degli altri codec *lossy* disponibili, pari a 8 kbit/s, cioè otto volte inferiore a quello del G.711.

La frequenza di campionamento è la medesima, di 8 kHz, così come la larghezza di banda audio confinata all'interno dei 4 kHz: da 300 a 3400 Hz. Nonostante il ridotto bitrate, l'aspetto fondamentale del codec G.729 è la qualità di poco inferiore al G.711, testimoniata da un MOS 3.92 [12], che ne consente l'uso senza particolari difficoltà da parte degli utenti. La complessità è superiore al G.711 ma con l'adozione della variante *Annex A*, a virgola fissa, si riescono a mantenere

contenuti l'esosità di risorse hardware e il ritardo di codifica, considerando anche il fatto che al giorno d'oggi la potenza di calcolo a questi livelli non è una limitazione.

Il codificatore riceve in ingresso una sequenza di campioni audio lineari PCM a 16 bit e per mezzo dell'algoritmo *conjugate-structure algebraic-code-excited linear prediction* (CS-ACELP) produce in uscita un flusso compresso a 8 kbit/s. Per i dettagli sul funzionamento fare riferimento a [17].

Contrariamente al codec G.711, il G.729 non ha possibilità di configurazione della lunghezza dei pacchetti prodotti: processa segmenti di durata fissa di 10 ms, producendo così pacchetti lunghi 80 bit (vedi Tabella 5.1).

5.3 La pacchettizzazione

Il periodo di pacchettizzazione è molto importante poiché determina la durata temporale di un *frame* e, conseguentemente, degli *slot* di trasmissione del TDMA. Questi due aspetti sono strettamente legati poiché per garantire la continuità di fonia senza fastidiose interruzioni periodiche è necessario che un pacchetto prodotto dal codec venga trasmesso in uno slot: quando è in corso una conversazione, il codec del trasmettitore produce i pacchetti da trasmettere che vengono mandati ai livelli inferiori fino al livello fisico che, quando arriva il turno di trasmissione, trasmette in linea il pacchetto attuale; successivamente i turni proseguono per gli altri telefoni e quando finisce il frame temporale (cioè si è ritornati nuovamente al turno del trasmettitore) un nuovo pacchetto è pronto per la trasmissione. In questo modo l'interlocutore ricevente non 'sente' interruzioni di comunicazione ma riceve (ai livelli superiori) un flusso continuo di pacchetti, nonostante la ricezione avvenga a turni, poiché durante la decodifica di un pacchetto da parte del codec, contemporaneamente il livello fisico riceve il successivo pacchetto.

Il codec G.729, come si è visto, permette di creare i pacchetti più piccoli di tutte le possibili configurazioni viste, in particolare a livello applicazione ogni 10 ms vengono prodotti 80 bit, che vengono incapsulati livello per livello secondo la procedura standard dello stack protocollare fino alla trasmissione fisica dei dati.

Come previsto dallo standard ISO/OSI, il payload di ogni livello dello stack protocollare è composto dalla composizione dell'header e del payload del livello superiore, portando ad avere nel pacchetto a livello link layer una somma di in-

formazioni non utili all'utente, detta *overhead*, ma necessari per l'indirizzamento e l'identificazione del pacchetto stesso alla destinazione.

Considerando il payload di applicazione di 80 bit e sommando gli header compressi di RTP/UDP e IP, lunghi complessivamente 4 byte come visto in precedenza, si ottiene un payload totale a livello link layer pari a 112 bit, portando così ad avere un totale di 400 bit (comprendendo anche l'header link layer di 288 bit) da trasmettere in un singolo slot temporale.

Come precedentemente detto visto, l'intervallo di pacchettizzazione del codec determinerà anche la durata temporale dei frame, quindi nel caso di utilizzo del codec G.729 sarà pari a 10 ms, comportando quindi una durata temporale di ciascun slot (8 in un frame) di 1.25 ms. Queste considerazioni portano quindi a necessitare di un canale che possa trasmettere bit utili ad un rate non inferiore a 320 kbit/s .

Nella Tabella 5.1 vengono riportate le possibili configurazioni di entrambi i codec e i valori che ne derivano:

Codec G.711 o G.729 (quest'ultimo non configurabile, ha un solo valore);

Campioni Indica il numero di campioni vocali inclusi in un pacchetto del livello applicazione;

P Payload: espresso in bit, indica la quantità di dati utili in un pacchetto a livello applicazione;

H + P Header + Payload a livello link layer: indica la lunghezza in bit del pacchetto completo che verrà effettivamente trasmesso nel canale, l'header totale è composto dai 4 byte compressi dei protocolli RTP/UDP/IP più i 288 bit del livello link layer, per un totale di 320 bit di header;

Durata Indica la durata temporale di un frame, stabilito dalla corrispondente configurazione del codec scelta;

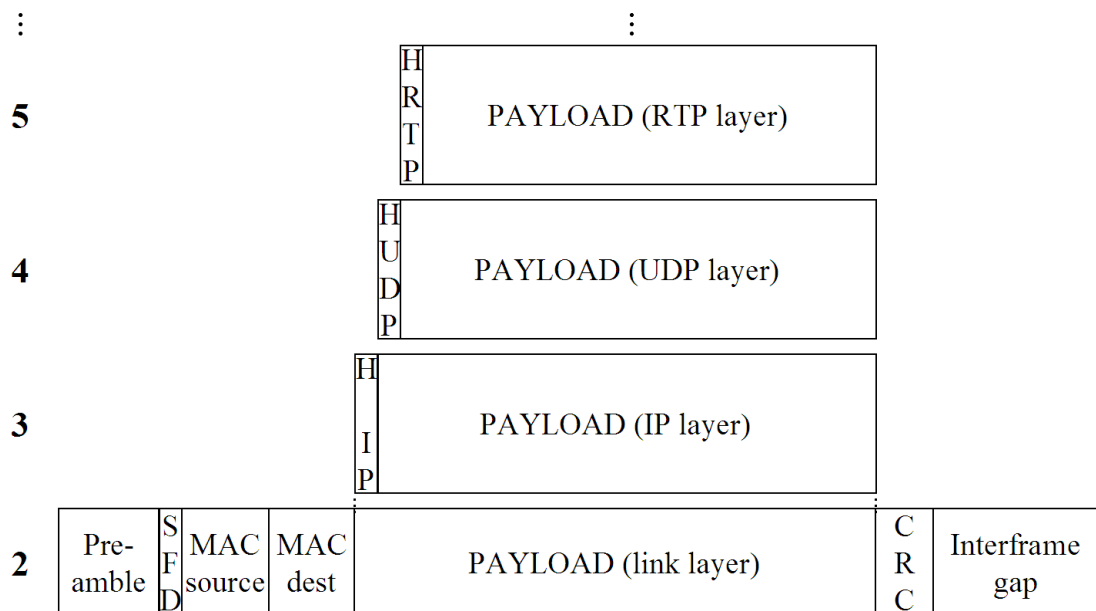
Durata slot E' la durata del frame divisa per 8 ottenendo così il tempo a disposizione nel turno di ogni telefono per trasmettere il pacchetto attuale;

Bitrate E' la velocità di trasmissione che deve garantire il canale per poter trasmettere l'intero pacchetto entro uno slot.

Codec	Campioni	P	H + P	Durata	Durata slot	Bitrate
G.711	40	320 bit	640 bit	5 ms	0.625 ms	1024 kbit/s
	80	640 bit	960 bit	10 ms	1.25 ms	768 kbit/s
	160	1280 bit	1600 bit	20 ms	2.5 ms	640 kbit/s
	240	1920 bit	2240 bit	30 ms	3.75 ms	597 kbit/s
	320	2560 bit	2880 bit	40 ms	5 ms	576 kbit/s
G.729	80	80 bit	400 bit	10 ms	1.25 ms	320 kbit/s

Tabella 5.1: Dimensioni configurabili dei pacchetti prodotti dai codec e relativi valori per garantire la trasmissione

Nella Figura 5.1 è rappresentato uno schema riepilogativo della composizione dei pacchetti dal livello 5 al livello 2.



Nota: le proporzioni dei payload non sono in scala

Figura 5.1: Composizione dei pacchetti ai vari livelli secondo lo standard ISO/OSI

Risultati e considerazioni

La modulazione 4-QAM occupa una banda minima pari a

$$B_{min} = \frac{1}{T}$$

cioè pari all'inverso di un periodo di simbolo, quindi la larghezza di banda occupata dal segnale modulato è pari al baud rate (salvo una quantità di banda aggiuntiva occupata dal filtro di trasmissione). Mappando due bit per simbolo quindi, il bitrate *grezzo* a livello fisico è pari al doppio del baud rate, ma, come si è visto, siccome la codifica di canale è affidata ad un codice con rate $\frac{1}{2}$, si ottiene a livello di link layer un bitrate 'disponibile' di dati utili pari al baud rate, nonché alla larghezza di banda occupata dal segnale trasmesso: ad esempio, se il bitrate desiderato di dati utili è pari a 400 kbit/s, allora la banda occupata sarà 400 kHz.

Sono state eseguite numerose simulazioni sui modelli di entrambi i cavi e su larghezze di banda diverse, dettate dalla Tabella 5.1 per poter valutare le prestazioni raggiungibili del sistema ideato per questo progetto: sono state impiegate larghezze di banda pari a 400, 600, 700, 800 e 1050 kHz, ricordando che ad ogni banda impiegata corrisponde una dimensione di pacchetto diversa.

6.1 Cavo 4 km

Il cavo lungo 4 km è attualmente utilizzato in un piazzale, misura 0,7 mm di diametro e ha una risposta impulsiva molto regolare, vista in Figura 3.4, ben definita, con un singolo impulso di breve durata e assenza di interferenza di intersimbolo.

Si riportano il tasso d'errore sui bit e sui pacchetti rispettivamente nelle figure 6.1 e 6.2.

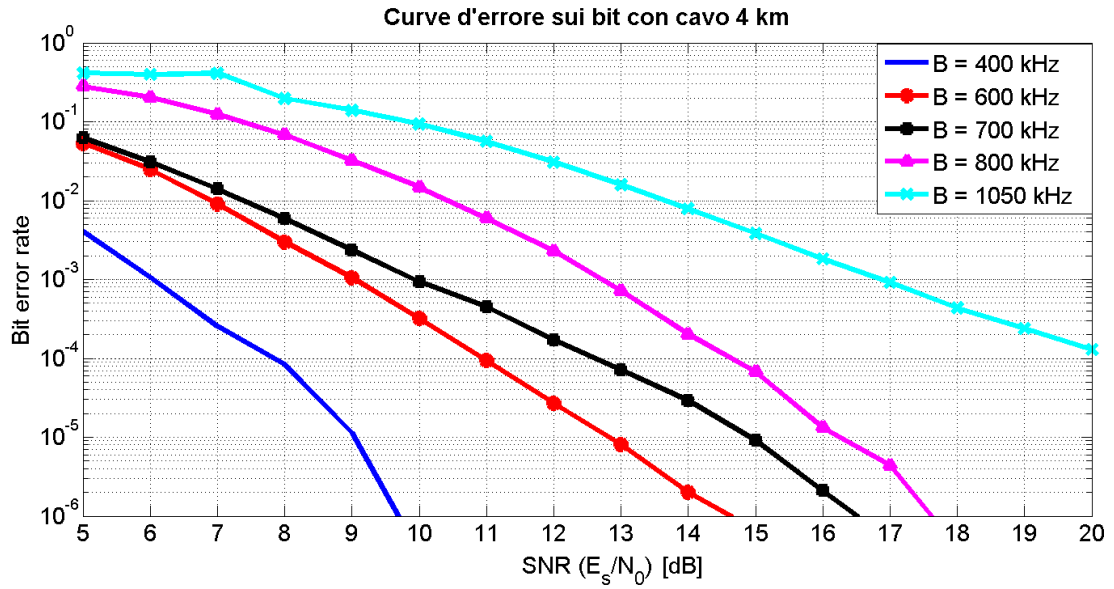


Figura 6.1: Tasso di errore sui bit del cavo di piazzale da 4 km

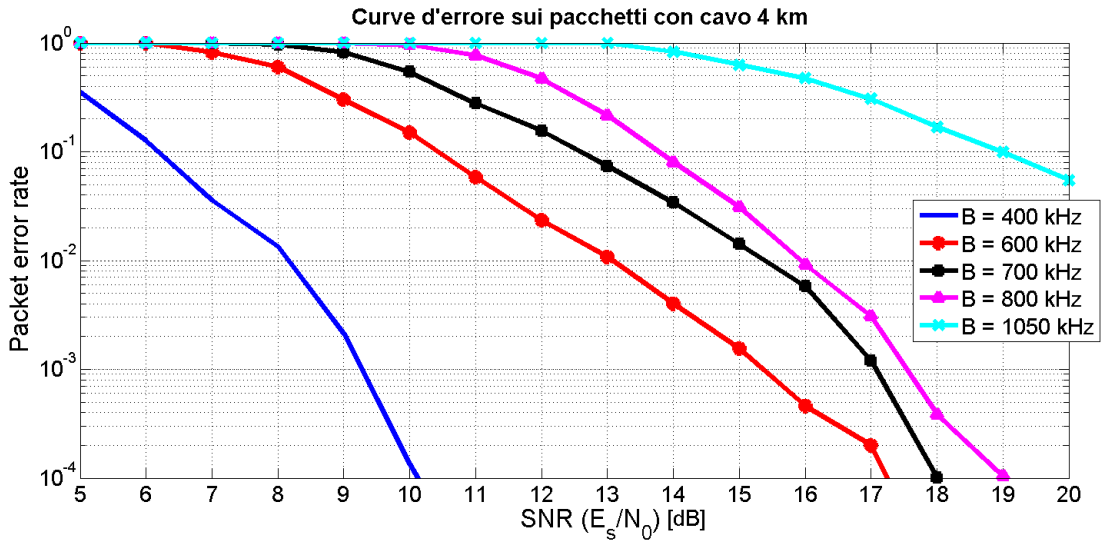


Figura 6.2: Tasso di errore sui pacchetti del cavo di piazzale da 4 km

Prevedibilmente, le curve d'errore sulla larghezza di banda minore sono molto buone garantendo un tasso di perdita dei pacchetti molto basso e quindi una

possibilità di comunicazione senza problemi di alcun genere. Le altre bande, che sarebbero impiegate con il codec di alta qualità G.711, sono comunque utilizzabili, ma la comunicazione risulterebbe piuttosto disturbata a causa del tasso di perdita di pacchetti abbastanza elevato.

6.2 Cavo 7 km

Il cavo in questione, come visto, non è impiegato in un impianto, ma è un cavo ‘da laboratorio’ di prova. E’ stato usato per valutare le eventuali prestazioni raggiunte da questo sistema in un caso limite.

La risposta impulsiva si è vista in Figura 3.6: ha un impulso più irregolare ed ‘allargato’, inoltre presenta una certa quantità di ISI e un’attenuazione molto più marcata dai 700 kHz ai 1100 kHz.

Sono state effettuate le medesime simulazioni dell’altro cavo.

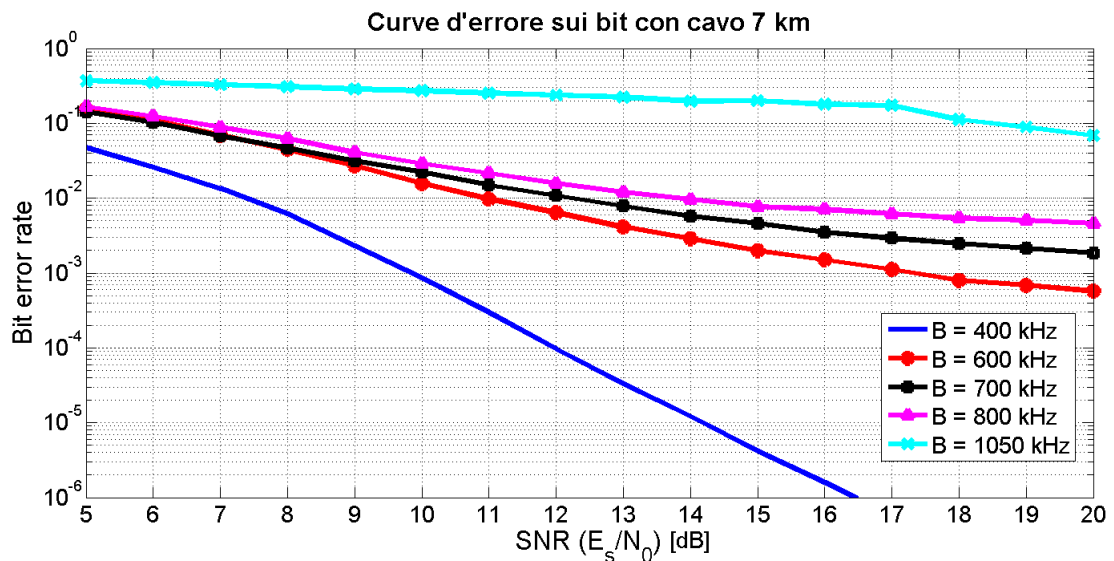


Figura 6.3: Tasso di errore sui bit del cavo di prova da 7 km

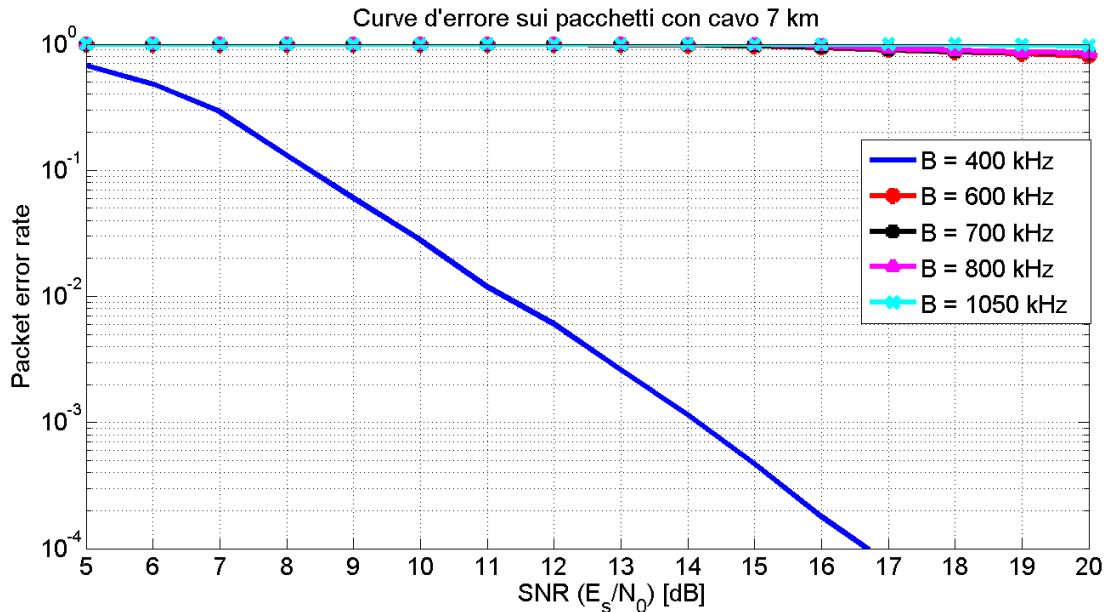


Figura 6.4: Tasso di errore sui pacchetti del cavo di prova da 7 km

Le prestazioni ottenute rivelano come l'uso di un codec G.711 (bande 600, 700, 800 e 1050 kHz) non sia possibile: i pacchetti più grandi e il tasso d'errore sui bit piuttosto alto non permettono la consegna di pacchetti integri.

Nel caso dei 7 km, quindi, l'unica configurazione adottabile è un codec G.729 con lunghezza dei pacchetti di 400 bit (a livello link layer) e un'occupazione di banda sul canale di circa 400 kHz (che diventano circa 500 kHz per la banda del filtro di trasmissione). In questo caso adattando la potenza di trasmissione per raggiungere un rapporto segnale/rumore superiore a 10 dB si può avere un tasso di perdita di pacchetti accettabile che non pregiudica la comprensibilità della comunicazione.

6.3 Conclusioni

Dalle curve d'errore si evince che un sistema così ideato è facilmente realizzabile (non utilizza una tecnologia costosa né complessa) e può garantire una linea di comunicazione basata su trasmissione digitale su distanze di 4 km senza nessun problema; su distanze di 7 km invece ci sarà qualche sporadica perdita di pacchetto che però non pregiudica la comprensione dell'interlocutore: ad esempio per un SNR di 11 dB mediamente si perderà 1 pacchetto ogni 100, dal punto di vista della

comunicazione vocale questo significa che verranno ‘persi’ 10 ms ogni secondo e questa interruzione viene appena percepita dall’interlocutore ricevente, quindi può considerarsi comunque una buona comunicazione.

Elenco delle figure

1.1	Schema generale del sistema STSI	4
1.2	Telefono di piazzale <i>Tonitel</i> installato in cassa stagna da esterno	8
2.1	Tecnica di accesso al mezzo TDMA	10
3.1	Strumento usato per l'attenuazione, <i>Advanced Line Tester 2000</i>	13
3.2	Curva di attenuazione misurata del cavo di piazzale	14
3.3	Linearità in frequenza della fase	16
3.4	Risposta impulsiva cavo piazzale	16
3.5	Curve delle attenuazioni dei due cavi messe a confronto	17
3.6	Confronto risposte impulsive	18
3.7	Schema mappatura dei bit in simboli complessi 4-QAM	19
3.8	Impulso a coseno rialzato nel tempo a sinistra e in frequenza a destra, immagine presa da [4]	21
3.9	Filtro di trasmissione interpolatore a radice di coseno rialzato	22
3.10	Spettro filtro di trasmissione a radice di coseno rialzato con $\rho = 0.25$	22
3.11	Implementazione polifase di un filtro interpolatore con fattore $Q_0 = 8$, immagine presa da [4]	23
3.12	Schema a blocchi del trasmettitore	24
3.13	Schema encoder convoluzionale	25
3.14	Funzionamento a passi dell'encoder, k indica i cicli di clock, le transizioni dovute all'arrivo di uno '0' sono rappresentate con una freccia nera solida, mentre le transizioni per l'arrivo di '1' sono rosse tratteggiate. Non tutte le transizioni sono rappresentate.	26

<i>ELENCO DELLE FIGURE</i>	58
3.15 Schema a blocchi completo del livello fisico	28
3.16 Schema a blocchi del canale e della parte analogica	29
3.17 Schema a blocchi dell'equalizzatore	30
3.18 Struttura funzionale dell'equalizzatore lineare adattivo	31
3.19 Equalizzatore in modalità decisione diretta	32
3.20 Circuito <i>Costas loop</i>	33
3.21 Schema a blocchi completo del ricevitore con decodifica Hard Input, nella versione con decodifica a <i>Soft detection</i> invece il demodulatore è sostituito con un detector che calcola le metriche	39
4.1 Struttura di un pacchetto di livello 2 link layer, i numeri in basso rappresentano la lunghezza di ciascun campo espressa in numero di bit	41
5.1 Composizione dei pacchetti ai vari livelli secondo lo standard ISO/OSI	51
6.1 Tasso di errore sui bit del cavo di piazzale da 4 km	53
6.2 Tasso di errore sui pacchetti del cavo di piazzale da 4 km	53
6.3 Tasso di errore sui bit del cavo di prova da 7 km	54
6.4 Tasso di errore sui pacchetti del cavo di prova da 7 km	55

Bibliografia

- [1] Rete Ferroviaria Italiana - Divisione Infrastruttura, *Specifica tecnica TT 575: fornitura per nuovo Sistema di Telefonia Selettiva Integrata*.
- [2] Ferrovie dello Stato - Area ingegneria e costruzioni, *Specifiche tecniche per la fornitura di cavo per telecomunicazioni a 4 coppie da 0,7 mm isolate con materiale termoplastico*, 1996.
- [3] Wikipedia, <http://en.wikipedia.org/wiki/GSM-R>.
- [4] Nevio Benvenuto, Giovanni Cherubini, *Algorithms for Communications Systems and their Applications*, John Wiley & Sons Ltd, Chichester, 2002.
- [5] aTen sistemi elettronici S.r.l, http://www.aten.it/Indice_Ita.html .
- [6] John G. Proakis, *Square Root Raised Cosine Filter*, <http://140.117.160.140/CommEduImp/pdfdownload/9222/BBIC-2-SRRC.pdf> , National Sun Yat-sen University.
- [7] Chih-Peng Li, *Convolutional Codes* [http://wits.ice.nsysu.edu.tw/course/pdfdownload/9221/CC-ConvolutionalCode\(II\).pdf](http://wits.ice.nsysu.edu.tw/course/pdfdownload/9221/CC-ConvolutionalCode(II).pdf) , National Sun Yat-sen University.
- [8] Composizione di un frame Ethernet, http://en.wikipedia.org/wiki/Ethernet_frame.
- [9] *Viterbi Decoding of Convolutional Codes*, <http://web.mit.edu/6.02/www/f2010/handouts/lectures/L9.pdf>, 2010, Massachusetts Institute of Technology.

- [10] IETF Group, Real-time Transport Protocol, RFC 3550, <http://tools.ietf.org/html/rfc3550>.
- [11] Alessandro Falaschi, *Voice over IP*, <http://infocom.uniroma1.it/alef/labints/Text/voip.html#RTP>, Università *La Sapienza*, Roma.
- [12] Mean Opinion Score, http://en.wikipedia.org/wiki/Mean_opinion_score.
- [13] Adaptive Multi-Rate audio codec http://en.wikipedia.org/wiki/Adaptive_multi-rate_compression#cite_note-3gpp-26090-1.
- [14] Common VoIP Codecs, <http://voip.about.com/od/voipbasics/a/voipcodecs.htm>.
- [15] ITU-T, *Codec G.711 : Pulse code modulation (PCM) of voice frequencies*, <http://www.itu.int/rec/T-REC-G.711-198811-I/en>.
- [16] ITU-T, *Codec G.711.0 : Lossless compression of G.711 pulse code modulation*, <http://www.itu.int/rec/T-REC-G.711.0-200909-I/en>.
- [17] ITU-T, *Codec G.729 : Coding of speech at 8 kbit/s using conjugate-structure algebraic-code-excited linear prediction (CS-ACELP)*, <http://www.itu.int/rec/T-REC-G.729-201206-I/en>.
- [18] Alan U. Kennington, *Robust Header Compression*, <http://www.rohc.net/links.html>.

Ringraziamenti

Questa esperienza di tirocinio nell'azienda veronese Telefin S.p.A. mi ha dato molto, sia in termini lavorativi che in termini umani. Il mio primo ringraziamento va all'ing. Francesco Fogliani per aver dato questa opportunità di 'finestra sul mondo del lavoro' non solo a me, ma anche ad altri studenti universitari che hanno potuto saggiare la qualità dell'ambiente di lavoro offerto da questa azienda.

Ringrazio di cuore il dott. Franco Righetti, responsabile tecnico e mio tutore aziendale, che mi ha proposto questo progetto di tesi e mi ha seguito, consigliato e ascoltato in questi sei mesi con una disponibilità che raramente si può ottenere da persone così impegnate nel lavoro.

Un altrettanto sentito ringraziamento di cuore va ai colleghi per avermi accolto in modo così disponibile da farmi sentire subito a mio agio, in particolar modo ringrazio Magda, Enrico e Marco G. per aver condiviso le pause caffè e le bellissime passioni come i viaggi, di quelli che non si dimenticano e che è doveroso condividere. Ringrazio Marco C., Andrea R., Riccardo per aver condiviso la nostra passione in comune: i motori, e aver passato una splendida domenica in pista alla guida di bolidi che difficilmente si dimenticheranno.

Ringrazio i miei colleghi di stanza Paolo, Marco O., Andrea M. e Simone per avermi ospitato nel loro spazio vitale (lavorativo!) e aver riso su parodie e sulla vita di tutti i giorni.

Ringrazio tutti gli altri colleghi dell'azienda che purtroppo ho incrociato poco, ma con cui mi sarebbe piaciuto condividere di più: Damiano e Mara, Antonella, Luigi, Federica e l'altra Federica, Michele, Francesco, il Gian, Gabriel, Sergio, Max, Antonio, Marco C., Maurizio, Nerina, Giorgia e tutti gli altri.

Ringrazio il prof. Michele Zorzi per avermi dato la possibilità di svolgere questo

progetto e avermi seguito e consigliato durante tutto l'arco di tempo, nonostante gli innumerevoli impegni in tutto il mondo.

Ultimi ma non ultimi, ringrazio i miei genitori Peppe e Anna per avermi sostenuto sempre e comunque durante tutti questi anni di carriera universitaria.