**(Contesto)**

Tradizionalmente lo sviluppo di apparati radio viene effettuato utilizzando hardware specializzato. L'aumento della potenza di calcolo nei computer moderni ha consentito di poter effettuare attraverso il software operazioni precedentemente troppo onerose. I computer sono finalmente in grado di eseguire tutta la parte di elaborazione delle informazioni ma necessitano di una scheda hardware che permetta poi la trasmissione e la ricezione: La piattaforma SDR.

**(Obiettivo)**

-La prima parte è consistita nella valutazione del funzionamento di SDR economici, la scelta della tipologia di tecnica di trasmissione è ricaduta su OFDM visto che rappresenta lo standard per le telecomunicazioni moderne. Infatti trova utilizzo in ADSL, Powerline, Wifi, radio e televisione digitale.

Per l'elaborazione delle informazioni al fine di creare un canale di trasmissione OFDM è stato scelto di utilizzare l'ambiente gratuito e open-source Gnuradio. Questo ambiente è il più utilizzato per la commercializzazione e la ricerca svolta sulle piattaforme SDR. Inoltre dispone di un insieme di moduli già predisposti per l'implementazione di OFDM.

-Successivamente è stato possibile, grazie all'esperienza acquisita, analizzare la possibilità di integrare un livello di sicurezza aggiuntivo mediante la creazione di un prototipo per la crittografia RSA.

**(SDR)**

Esistono vari SDR disponibili, da quelli completi da laboratorio che costano più di mille euro in dotazione all'università (in alto) utilizzato per la trasmissione

A versioni economiche da 8 euro (in basso) vendute come ricevitori per DVB-T e DAB.

La differenza principale sta nella configurabilità della velocità di campionamento e la frequenza su cui possono trasmettere/ ricevere.

Entrambi i modelli dispongono di driver per interfacciarsi con Gnuradio.

**(TX1)**

La comunicazione inizia preparando le informazioni in un formato adatto per la trasmissione OFDM vera e propria. I dati vengono raggruppati e vengono aggiunte informazioni per il controllo dell'errore. Viene generato un header a cui viene concatenato il payload.

**(OFDM1)**

A questo punto per capire meglio il funzionamento della seconda parte della trasmissione è utile ricordare alcuni concetti teorici:

OFDM ha la caratteristica di allocare in modo efficiente la banda grazie all'ortogonalità delle sottoportanti, l' ortogonalità garantisce che i simboli modulati sulle sottoportanti adiacenti non interferiscano fra loro.

OFDM suddivide le informazioni su molte sottobande che possono essere anche migliaglia, le sottoportanti sono di 3 tipi: quelle destinate alle informazioni (che sono la maggiorparte), quelle destinate ai simboli pilota ed infine una detta DC sulla quale non viene modulato nulla. Quest'ultime due sono utili al ricevitore per la sincronizzazione.

**(TX2)**

Alle informazioni già sotto forma di costellazioni viene aggiunto un preambolo e successivamente vengono allocate. Attraverso la trasformata di Fourier inversa si passa al dominio del tempo dove al simbolo OFDM viene aggiunto il prefisso ciclico. Infine i campioni vengono passati al blocco driver per l'sdr nel quale vengono configurati i parametri per la trasmissione reale come frequenza, larghezza di banda e velocità di campionamento.

**(OFDM2)**

La ricezione è più complessa della trasmissione e necessita di alcuni ulteriori passaggi per il funzionamento.

La sincronizzazione deve essere effettuata su due fronti distinti: sulla frequenza di campionamento e sullo sfasamento fra le sottoportanti. I problemi di sincronizzazione vengono associati principalmente a differenze sui clock fra trasmettitore e ricevitore, all'effetto doppler oppure nel caso delle reti wireless all'effetto della propagazione del segnale su percorsi multipli.

L'equalizzazione ha il compito di ribilanciare le componenti del segnale nella maniera più fedele possibile a come è stato trasmesso. Per riuscirci è necessaria una stima del comportamento del canale che viene effettuata dal preambolo e viene corretta alla ricezione di ogni simbolo OFDM mediante i piloti.

Il ricevitore effettua un controllo per verificare la presenza di errori ed eventualmente scarta l'intero pacchetto. Alcune implementazioni di OFDM fanno uso di algoritmi più avanzati ed onerosi per la notifica e la successiva correzione dell'errore se possibile.

**(RX1)**

La parte centrale del ricevitore è il blocco Header/Payload Demux che ha il fondamentale compito di separare i campioni contenenti l'header da quelli contenenti le informazioni ancora da portare nel dominio delle frequenze.

La procedura di ricezione inizia quando viene rilevato l'inizio della trasmissione, a quel punto i campioni relativi ai primi tre simboli ofdm vengono inoltrati alla seconda sezione del ricevitore. per ora ci basta sapere che i dati decodificati contengono le informazioni contenute nell'header e servono al blocco Demux per il successivo inolto dei campioni corrispondenti al payload.

**(RX2)**

La seconda sezione del ricevitore riceve i campioni e ricava le informazioni in maniera analoga fra header e payload. La lettura consiste nel passaggio al dominio delle frequenze, applicare l'equalizzazione e decodificare le costellazioni ottenute. Nella parte del payload è presente il blocco channel estimation che effettua una stima del canale utile al blocco successivo per l'equalizzazione.

Infine viene controllata la presenza di errori e scritte le informazioni sul file audio.

**(Test)**

Sono stati effettuati alcuni test per osservare il comportamento del sistema. Come previsto all' aumentare della potenza trasmessa diminuiscono le informazioni perse, inoltre dai seguenti grafici emergono un altro paio caratteristiche interessanti: La percezione dei pacchetti persi per l'orecchio umano è poco sensibile se in piccole quantità (meno 3%) ma la qualità percepita crolla sopra il 4% fino a rendere impossibile la comprensione sopra al 10%. Risultati perfettamente in linea con quelli presenti in alcuni studi riguardanti la tolleranza d'errore in sistemi streaming audio. Il grafico a destra è molto utile per capire le limitazioni di questo sistema, la sezione blu rappresenta il range dentro il quale c'è scambio di dati. E' interessante notare che il computer con l'SDR in trasmissione limita la generazione dei campioni una volta arrivati a 4Msps. questo collo di bottiglia è dovuto alla potenza della cpu su singolo core, Gnuradio ha un algoritmo automatico per il multi-threading ma in questa particolare applicazione risulta in un miglioramento solo del 4%.

In basso è possibile notare la differenza fra i file audio duvuta alla mancanza di pacchetti.

**(RSA)**

Una volta presa dimestichezza con gnuradio e gli sdr è stata valutata la possibilità di un aggiunta al sistema di un layer di crittografia, sono stati sviluppati due moduli come prototipo e sono stati validati all'interno dell'ambiente. Lo svliluppo di un modulo è diviso in tre parti fondamentali:

* un file contenente tutte le informazioni sulla modalità di interazione del modulo con il flusso dati (ad esempio, tipo di file in ingresso, numero ingressi, parametri faccoltativi o oblligatori per l'utilizzo).
* La definizione di un metodo work all'interno del quale avviene il funzionamento vero e proprio del blocco
* Codice di test utile durante la fase di sviluppo che permette un feedback sul funzionamento senza dover compilare e rendere disponibile il blocco all'interno dell'ambiente

Infine la validazione finale mostrata a schermo, nella parte centrale dell'interfaccia grafica sono presenti i blocchi collegati al fine di codificare e decodificare un file. nella parte inferiore si può notare sulla sinistra un feedback restituito dai blocchi durante il funzionamento mentre sulla destra il file inviato e quello ricevuto. Il modulo è stato utile per analizzare alcune criticità di un eventuale utilizzo.

**(Conclusioni)**

Gli sdr economici sono in grado di funzionare con ofdm in in ambiente gnuradio, la principale limitazione riscontrata è causata dalla ridotta frequenza di campionamento massima. Inoltre Gnuradio ha mostrato un limite sulla divisione del lavoro fra i core limitando l'esecuzione del sistema a 4Msps.

Per quanto riguarda l'integrazione del modulo di crittografia

* è necessario approfondire il sistema delle meta-informazioni contenuto in gnuradio e modificare l'algoritmo sviluppato affinchè si codifichi a blocchi dell'ordine dei 100Byte anzichè 8bit.
* E' d'obbligo utilizzare chiavi più lunghe nell'ordine di 1000bit al fine di garantire resistenza contro attacchi brute force