



UNIVERSITÀ DEGLI STUDI DI MILANO

Facoltà di Scienze Matematiche, Fisiche e Naturali

Corso di Laurea triennale in Sicurezza dei Sistemi e delle Reti Informatiche

Polo Didattico e di Ricerca di Crema

**Tecniche per il miglioramento
della qualità della trasmissione
di voce su IP (VoIP)**

RELATORE

Prof. Damiani Ernesto

CORRELATORE

Dott. Bellandi Valerio

TESI DI LAUREA DI

Daniele Benetti

Matricola n. 671097

Anno Accademico 2006-2007

Dediche e ringraziamenti

Questo elaborato è dedicato a mia moglie Annalisa.

Un particolare ringraziamento va alle seguenti persone per i loro contributi:

il relatore di questa tesi di laurea, prof. Damiani Ernesto;

il correlatore, dott. Bellandi Valerio (che è riuscito a sopportarmi per tutto il tempo).

Indice

| | |
|---|----|
| Elenco delle figure..... | iv |
| Introduzione..... | 1 |
| | |
| Capitolo 1 – Il segnale audio | 3 |
| 1.1 - Cenni di acustica..... | 3 |
| 1.2 - L’orecchio umano..... | 5 |
| 1.3 - La trasmissione e la trasduzione del suono | 6 |
| 1.4 - La codifica del segnale audio | 7 |
| | |
| Capitolo 2 - La comunicazione telefonica..... | 10 |
| 2.1 - Evoluzione PSTN-IP | 10 |
| 2.2 - Differenze CTM - PTM..... | 10 |
| 2.3 - Protocolli VoIP | 12 |
| 2.4 - QoS | 13 |
| 2.5 - L’indice MOS (Mean Opinion Score)..... | 14 |
| 2.6 - Algoritmi per la trasmissione della voce | 16 |
| 2.7 - La famiglia di codec ITU-T (International Telecommunication Union)..... | 17 |
| 2.8 - Codec open source..... | 18 |
| iLBC..... | 18 |
| Speex..... | 19 |
| 2.9 - Il business del VoIP: Asterisk e Skype..... | 20 |
| | |
| Capitolo 3 – Analisi dei disturbi nelle comunicazioni | 22 |
| 3.1 - Il problema delle comunicazioni VoIP in condizioni critiche..... | 22 |
| 3.2 - Il problema dell’eco acustico nelle comunicazioni telefoniche..... | 23 |
| 3.3 - L’eliminazione dell’eco acustico..... | 26 |

| | |
|---|----|
| 3.4 - L'eco ibrido | 28 |
| 3.5 - L'eliminazione dell'eco ibrido | 29 |
| 3.6 - L'eco nelle trasmissioni VoIP | 30 |
| 3.7 - Algoritmi di echo cancellation | 31 |
| 3.8 - Come funziona un Acoustic Echo Canceller (AEC) | 33 |
| 3.9 - Qualità della comunicazione VoIP | 35 |
| 3.10 - Come influisce la qualità della rete sulla qualità del segnale in VoIP | 36 |
| 3.11 - VoIP e WLAN | 38 |
| 3.12 - <i>Echo cancellation</i> in VoIP su WLAN | 39 |
| 3.13 - <i>Related Work</i> | 41 |
| | |
| Capitolo 4 – simulazione di applicazione di un AEC con alta varianza RTT | 43 |
| 4.1 - Simulazione in Matlab | 43 |
| 4.2 - Applicazione del filtro per la rimozione dell'eco in condizioni di ambiente ottimali. | 45 |
| 4.3 - Applicazione del filtro per la rimozione dell'eco in condizioni di ambiente ad alta varianza del Round Trip Time. | 48 |
| 4.4 - Packet Based Echo Canceller | 52 |
| 4.5 - Simulazione in Matlab di un PBEC | 53 |
| | |
| Capitolo 5 – Conclusioni | 58 |
| 5.1 - Come migliorare la comunicazione | 58 |
| 5.2 - Considerazioni finali | 60 |
| | |
| Appendice 1. Programma Matlab | 61 |
| Glossario degli acronimi | 68 |
| Bibliografia | 70 |

Elenco delle figure

| | |
|--|----|
| Figura 1 – Grafico Spostamenti/Tempo | 3 |
| Figura 2 – Grafico teorema di Fourier..... | 5 |
| Figura 3 – Esempio di codifica analogico/digitale | 8 |
| Figura 4 – Esempi di campionamenti a diverse velocità..... | 8 |
| Figura 5 – Differenze CTM-PTM | 11 |
| Figura 6 – Tabella indice MOS | 15 |
| Figura 7 – Incapsulamento VoIP..... | 16 |
| Figura 8 – Tabella <i>Codec</i> /indice MOS/consumo banda..... | 17 |
| Figura 9 – Comparazione tra <i>Codec</i> | 20 |
| Figura 10 – Esempio di propagazione WiFi..... | 23 |
| Figura 11 – Eco acustico | 24 |
| Figura 12 - Schema AEC base..... | 27 |
| Figura 13 – Schema AEC avanzato..... | 28 |
| Figura 14 - <i>Hybrid</i> | 28 |
| Figura 15 – Rimozione eco lineare..... | 29 |
| Figura 16 – Eco PSTN/VoIP | 30 |
| Figura 17 – Schema algoritmo AEC..... | 31 |
| Figura 18 – LEC + AEC..... | 33 |
| Figura 19 - Livelli di percezione della qualità..... | 37 |
| Figura 20 – Scenario VoIP su WiFi | 40 |
| Figura 21 – Filtro FDAF..... | 43 |
| Figura 22 – Dialogo interlocutori | 46 |
| Figura 23 – Situazione ottimale..... | 46 |
| Figura 24 – Segnale in ingresso con eco | 47 |
| Figura 25 – Schema con eco cancellato..... | 47 |
| Figura 26 – Diagramma a blocchi algoritmo simulazione RTT variabile..... | 49 |
| Figura 27 – Segnale con eco in ingresso e RTT variabile..... | 50 |
| Figura 28 – Segnale in ingresso con eco cancellato e RTT variabile..... | 51 |
| Figura 29 – Comparazione eco eliminato con e senza RTT variabile..... | 52 |

Figura 30 – Schema PBEC 53

Figura 31 – Eco con blocchi da 20 byte 54

Figura 32 – Eco con blocchi da 200 byte 55

Figura 33 – Eco con blocchi da 2.000 byte 56

Figura 34 – Eco con blocchi da 20.000 byte 56

Figura 35 – Eco con blocchi da 200.000 byte 57

Figura 36 – Diagramma a blocchi di un AEC ottimale con condizioni di RTT variabile... 59

Introduzione

VoIP è il termine attualmente utilizzato per identificare la diffusione della voce su mezzi trasmissivi digitali. L'acronimo sta a identificare *'Voice over Internet Protocol'* che tradotto significa 'voce attraverso il protocollo internet'.

È intuitivo desumere che l'applicazione più diffusa del protocollo sia la trasmissione della voce attraverso reti informatiche siano esse Lan o Internet, sovrapponendosi alle soluzioni analogiche d'uso comune come le linee telefoniche tradizionali.

Questa sovrapposizione, dovuta soprattutto ai bassi costi dell'utilizzo del VoIP, apre un gamma di problematiche di carattere tecnico imposte dalla filosofia architettuale di Internet. Per esempio una comunicazione VoIP bi-direzionale, una comune telefonata, deve avvenire in tempo reale, senza ritardi tra la trasmissione del mittente e la ricezione del destinatario e/o perdite di pacchetti audio.

L'architettura strutturale di Internet sicuramente non agevola questa particolare necessità del VoIP perché il pacchetto IP contenente l'informazione deve attraversare molti router; in condizioni di trasmissione senza fili, come WiFi o Bluetooth, le problematiche concernenti la gestione dei pacchetti audio aumentano a causa dell'instabilità fisica della connessione.

La comunicazione VoIP si sta affermando in molti settori: le possibilità offerte ne permettono l'applicazione sia in settori aziendali ma anche per applicazioni diverse come l'utilizzo in condizioni critiche: in situazioni di disastro naturale o in campo di battaglia si potrebbero ripristinare le comunicazioni appoggiando il protocollo VoIP su una rete WiFi.

Le reti WiFi però non possiedono i requisiti di stabilità della trasmissione necessari per garantire una corretta comunicazione VoIP: le trasmissioni radio sono soggette ad un'elevata rumorosità ed i segnali radio ad alta frequenza sono soggetti alla riflessione e quindi vengono ricevute più volte con sfasamenti temporali dipendenti dalla lunghezza del percorso compiuto. Queste interferenze generano disturbi che, in una comunicazione VoIP, si tramutano in perdite di segnale e in degradazione del qualità della voce.

Uno dei maggiori problemi da affrontare in una comunicazione VoIP è l'eco acustico, che consiste nel disturbo rilevato tra due interlocutori che parlano simultaneamente: la voce in partenza rientra, con un ritardo indefinito, al mittente, creando un evidente disturbo di comprensione.

Per ridurre questo fastidioso problema si devono ideare degli algoritmi specifici denominati di '*Echo cancellation*', atti a riconoscere ed eliminare il ritorno dell'eco acustico.

Sono stati sviluppati diversi algoritmi per porre una soluzione al problema, alcuni di pubblico dominio ed altri invece coperti da segreto industriale. Algoritmi di pubblico dominio sono ad esempio utilizzati per Asterisk, il più conosciuto centralino VoIP software gratuito; algoritmi coperti da segreto industriale sono ad esempio sviluppati per Skype, un software di comunicazione VoIP per P2P disponibile in varie piattaforme.

In ambito WiFi però tali algoritmi non sono affidabili in quanto non tengono in considerazione l'ambiente trasmissivo: un'alta varianza del *Round Trip Time* dato dall'instabilità della rete, per esempio, rende inefficace la funzionalità dell'algoritmo di EC (*Echo cancellation*).

In questo elaborato sono analizzate le varie caratteristiche della comunicazione audio, partendo dalla struttura del suono fino ad arrivare alla comunicazione VoIP con lo studio delle problematiche attuali.

In seguito saranno descritte delle simulazioni in Matlab realizzate per verificare la variabilità nel comportamento di un algoritmo di '*Echo cancellation*' in presenza di alta variabilità del *Round Trip Time* nelle comunicazioni WiFi.

Infine sarà proposta un'idea di soluzione al problema mediante l'adozione di un *Jitter Buffer* avanzato unito ad un *Packet Based Echo Canceller* (PBEC).

Capitolo 1

Il segnale audio

1.1 - Cenni di acustica

Per la fisica, il suono è un'oscillazione, cioè un movimento nello spazio, compiuta dalle particelle, cioè atomi o molecole, in un mezzo. Il mezzo è un corpo qualunque sia esso nello stato solido, liquido o gassoso. Quindi, in condizione di vuoto assoluto, il suono non può propagarsi.

Le oscillazioni sono spostamenti delle particelle, intorno alla posizione di riposo e lungo la direzione di propagazione dell'onda, provocati da movimenti vibratori, provenienti da un determinato oggetto, chiamato sorgente del suono, il quale trasmette il proprio movimento alle particelle adiacenti, grazie alle proprietà meccaniche del mezzo; le particelle a loro volta, iniziando ad oscillare, trasmettono il movimento alle altre particelle vicine e queste a loro volta ad altre ancora; in questo modo, un semplice movimento vibratorio si propaga meccanicamente dando vita all'onda sonora (o onda acustica), che si definisce pertanto onda longitudinale. [1]

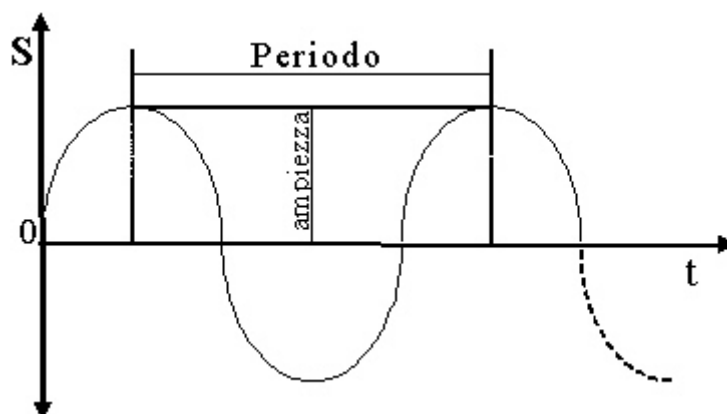


Figura 1 – Grafico Spostamenti/Tempo

Per studiare le onde sonore è possibile avvalersi un supporto visivo, utilizzando un grafico cartesiano, riportante il tempo (t) sull'asse delle ascisse, e gli spostamenti delle particelle (s) su quello delle ordinate. Il tracciato in figura 1 esemplifica gli spostamenti delle particelle: all'inizio, la particella si sposta dal suo punto di riposo (asse delle ascisse) fino al culmine del movimento oscillatorio, rappresentato dal ramo crescente di parabola che giunge al punto di massimo parabolico. Poi la particella inizia un nuovo spostamento in direzione opposta, passando per il punto di riposo e continuando per inerzia fino a un nuovo culmine simmetrico al precedente, questo movimento è rappresentato dal ramo decrescente che, intersecando l'asse delle ascisse, prosegue in fase negativa fino al minimo parabolico. In fine, la particella ritorna in dietro e ripete nuovamente la sequenza di spostamenti, così come si osserva nel tracciato del grafico.

Il periodo (nel grafico in figura 1 è il segmento tra due creste) è il tempo impiegato dalla particella per tornare nello stesso punto dopo aver cominciato lo spostamento, e indica la durata complessiva dell'oscillazione. La distanza dalla cresta all'asse delle ascisse indica, invece, l'ampiezza del movimento, in altre parole la distanza massima percorsa dalla particella dalla sua posizione di riposo durante l'oscillazione. Tuttavia, nonostante il periodo e l'ampiezza siano due grandezze che da sole sarebbero sufficienti per descrivere le caratteristiche di un'onda, non sono frequentemente utilizzate, perlomeno non in forma pura; quando si sta esaminando un'onda sonora si preferisce, infatti, usare altre grandezze da queste derivate. La quantità di periodi prodotti in un secondo misura la frequenza, in hertz, che indica il numero d'oscillazioni compiute dalla particella in un secondo. L'orecchio umano percepisce le frequenze che vanno dai 20 Hz ai 20 kHz, ed è più sensibile tra i 2 e i 4 kHz. Il range della voce umana varia dai 500 Hz ai 2 kHz

Dall'ampiezza dell'onda, invece, otteniamo l'intensità, che nel caso dell'onda sonora è intesa come pressione esercitata dalla stessa su una superficie, se, infatti, l'ampiezza misura lo spazio, tanto più sarà estesa, tanto maggiore sarà la pressione esercitata sulle particelle vicine e dunque pure la forza esercitata contro un ostacolo; la misura d'intensità delle onde sonore sono i decibel. Il range dinamico, ossia l'intervallo dal suono più basso al più alto percepibile dall'orecchio umano, è di 96 dB (*Decibel*).

Il teorema di Fourier enuncia che qualsiasi onda (nel nostro caso sonora) è riducibile a una somma di onde la cui prima è la **fondamentale** e le altre le **armoniche**.

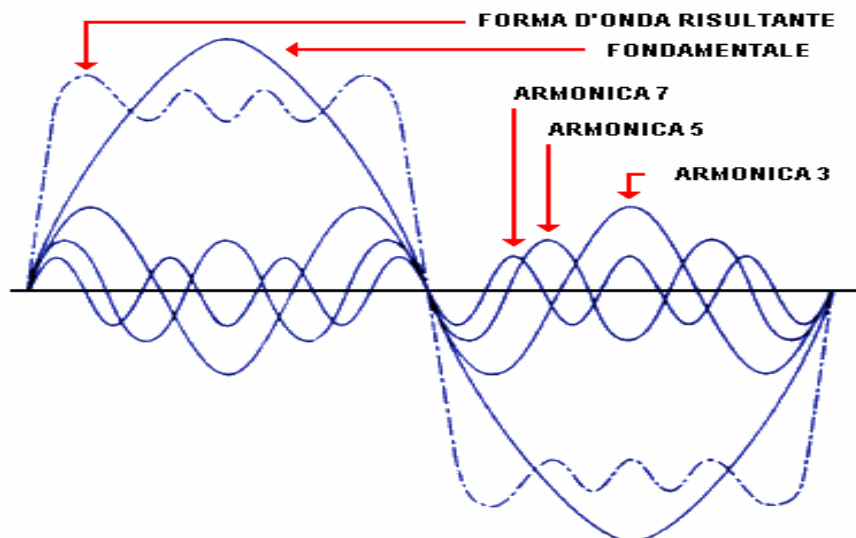


Figura 2 – Grafico teorema di Fourier

Nella figura (2) sopra riportata, si può intuitivamente capire che un'onda quadra è riducibile alla somma di un'onda fondamentale con tre armoniche che possiedono frequenze multiple della fondamentale. L'intensità, la forza che volgarmente è definita 'volume', di ogni singola armonica determina il suono. Ecco perché si riescono a distinguere suoni diversi malgrado possiedano la stessa tonalità (basti pensare a una nota musicale suonata da diversi strumenti). [2]

1.2 - L'orecchio umano

Nella traiettoria della propagazione dell'onda sonora può trovarsi l'orecchio umano. Si compone di tre parti:

orecchio esterno: composto dal padiglione auricolare e dal condotto uditivo esterno. Il padiglione auricolare è l'unica parte visibile dell'orecchio. Lo scopo del padiglione è convogliare le onde sonore nel condotto uditivo.

orecchio medio: in appena un centimetro quadrato di spazio l'orecchio medio contiene i tre più piccoli ossicini del corpo umano: martello, incudine e staffa. Il martello è internamente collegato alla membrana timpanica, mentre la staffa, l'ultimo degli ossicini, è

articolato direttamente con l'orecchio interno, perciò i movimenti della membrana, provocati dal suono, sono amplificati (circa di venti volte per effetto leva) e trasmettono così all'orecchio interno tutta la raffinata ricchezza dei suoni, da quelli singoli a quelli di un'intera orchestra.

orecchio interno, l'elemento organizzatore e recettore dei suoni. L'orecchio interno si compone di coclea e vestibolo. La prima è deputata alla rilevazione e alla trasduzione dei suoni, il vestibolo, invece, contiene i recettori dell'equilibrio.

Tramite questo complesso organo il cervello umano è in grado di riconoscere la voce di una persona nota o di riconoscere il suono di uno strumento. [3]

1.3 - La trasmissione e la trasduzione del suono

Un'onda sonora che si propaga in un mezzo è soggetta all'assorbimento acustico del mezzo stesso, cioè una caratteristica di smorzamento del segnale: con l'aumentare della distanza percorsa l'onda sonora perde di intensità fino al totale smorzamento. Per riuscire a propagarla su distanze maggiori si può ricorrere a diverse soluzioni tra cui amplificare la forza del segnale o trasformarlo in un segnale diverso adatto a viaggiare su mezzi diversi: la trasduzione.

Per trasduzione s'intende la conversione di un tipo di energia in un'altra, per esempio da energia meccanica in energia elettrica.

Nel campo sonoro la trasduzione si realizza tramite degli elementi meccanici detti 'trasduttori' che, nell'utilizzo più comune, sono il microfono e l'altoparlante. Il criterio di funzionamento è il seguente: il microfono è una bobina immersa in un campo magnetico che vibra quando è attraversata dalle onde sonore convertendole quindi in segnali elettrici; nell'altoparlante accade il contrario e cioè i segnali elettrici sono convertiti in forza elettromagnetica in grado di far muovere un corpo generando onde sonore.

1.4 - La codifica del segnale audio

La conoscenza della codifica del segnale audio è utile per la comprensione degli algoritmi di ottimizzazione e/o eliminazione dell'eco.

Da quanto appreso al paragrafo precedente, la vibrazione di una bobina immersa in un campo magnetico genera della corrente elettrica (trasduzione del suono) e tramite circuiti elettronici di amplificazione il segnale audio viene normalizzato, solitamente, in una gamma compresa tra -5 a +5 vcc (volt in corrente continua). La distanza tra le tensioni massima e minima nella gamma prende il nome di 'gamma totale delle ampiezze' e nel caso di tensioni tra -5 a +5 vcc questa gamma possiede un valore di 10 vcc. La rilevazione del valore di ampiezza (sia essa positiva o negativa) in un determinato istante prende il nome di 'campione' e la relativa quantizzazione consiste nel trasformare il valore in una stringa di bit.

Il campionamento è quindi la tecnica che consiste nel convertire un segnale continuo nel tempo in un segnale discreto, valutandone l'ampiezza a intervalli regolari. In questo modo, a seguito di una successiva operazione di quantizzazione e conversione, è possibile ottenere una stringa digitale (discreta nel tempo e nell'ampiezza) che approssimi quella continua originaria.

Il tempo che intercorre tra due singoli campionamenti si chiama 'periodo di campionamento'; la frequenza di campionamento, calcolabile come dalla formula (1), è l'inverso del periodo e si misura in Hz (Hertz) ma, generalmente, la misurazione è riportata in kHz (kilohertz).

$$frequenza = \frac{1}{periodo} \text{ (Hz)} \quad (1)$$

Per esempio in un compact disc audio la frequenza di campionamento è di 44.1 kHz.

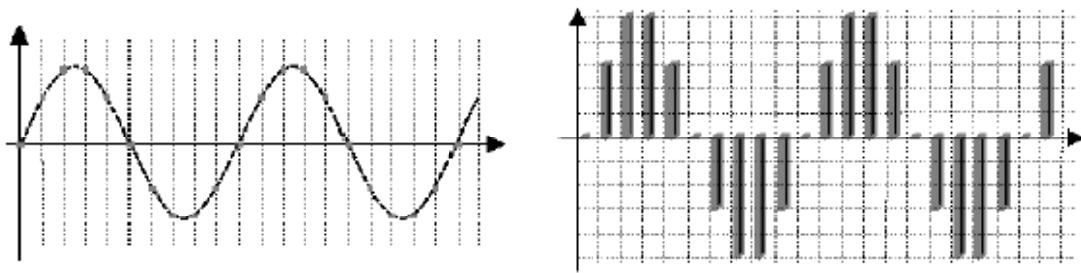


Figura 3 – Esempio di codifica analogico/digitale

Nei grafici riportati in figura 3 si può notare come a una frequenza di campionamento x , del segnale dell'onda sinusoidale rappresentata a sinistra solo alcuni valori siano riportati, e quindi campionati, nel grafico a destra. Si può quindi dedurre che ad alte frequenze di campionamento si otterrà una maggiore qualità del segnale digitale, più simile a quello originario. Nei grafici sotto riportati (4) si può verificare quanto appena esposto.

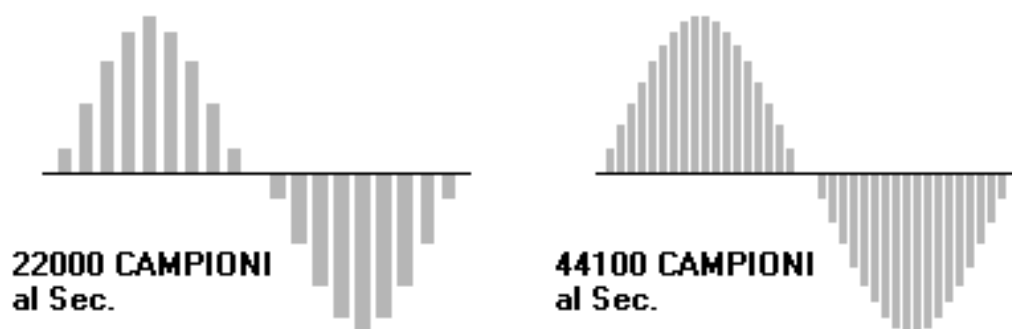


Figura 4 – Esempi di campionamenti a diverse velocità

Un altro valore da considerare è la **profondità dei bit** utilizzati per il campionamento. Questo valore rispecchia l'accuratezza con cui è stato misurato un campione. La profondità di bit, perciò, si riferisce alla lunghezza delle parole binarie (cioè sequenze di 0 e 1) usate per descrivere ciascun campione del segnale d'ingresso. Parole più lunghe permettono la rappresentazione di una serie più ampia di numeri, e quindi misurazioni più accurate e riproduzioni più fedeli di un segnale (maggior dinamica e minor distorsione). Per esempio in un sistema a 16 bit, ciascun campione è rappresentato come una parola binaria lunga 16 cifre. Poiché ciascuna di queste 16 cifre può essere uno 0 o un 1, sono possibili 65.536 (2^{16}) valori per ciascun campione.

Gli algoritmi software che eseguono il campionamento del suono trasformandolo in un segnale digitale e viceversa sono chiamati **Codec** (*Codificator / Decodificator*).

Capitolo 2

La comunicazione telefonica

2.1 - Evoluzione PSTN-IP

Dai suoi inizi la rete telefonica tradizionale realizzata per svolgere il servizio di trasporto della fonia si è evoluta nel criterio di trasferimento ora conosciuto come CTM (*Circuit Transfer Mode* – Modo di Trasferimento a Circuito) con il quale è possibile trasferire anche i dati. Una successiva evoluzione, dovuta soprattutto ai limiti imposti dal CTM in termini di velocità, tasso d'errore e allocazione di banda, è stata l'introduzione di nuove tecnologie di trasferimento alternative chiamate PTM (*Packet Transfer Mode* – Modo di Trasferimento a Pacchetto). Oggi esistono molte reti telefoniche con funzionamento PTM: alcune proprietarie, altre conformi agli standard ITU-T (X.25 – X.75). [4]

ARPANET, una delle più vecchie tecnologie di trasmissione dati su rete geografica, quella che poi si trasformò nell'attuale INTERNET, fu realizzata sfruttando lo standard X.25 per i propri PSN (*Packet Switching Nodes*) poiché il protocollo 1822, sviluppato originariamente per la trasmissione dei dati tra PSN, non fu mai uno standard industriale. Successivi sviluppi delle comunicazioni dati portarono alla nascita dei protocolli IP e TCP, sui quali si basa la comunicazione VoIP.

2.2 - Differenze CTM – PTM

Le reti di comunicazione possono essere divise in due tipi fondamentali: quelle orientate alla connessione, chiamate a commutazione di circuito e quelle senza connessione, chiamate a commutazione di pacchetto. Le prime funzionano formando un circuito (o

connessione) dedicata fra due punti; le seconde invece suddividono in piccoli pacchetti i dati da trasferire che seguono percorsi dinamici, diversi anche per la stessa comunicazione.

In figura 5 sono schematizzate le caratteristiche peculiari.

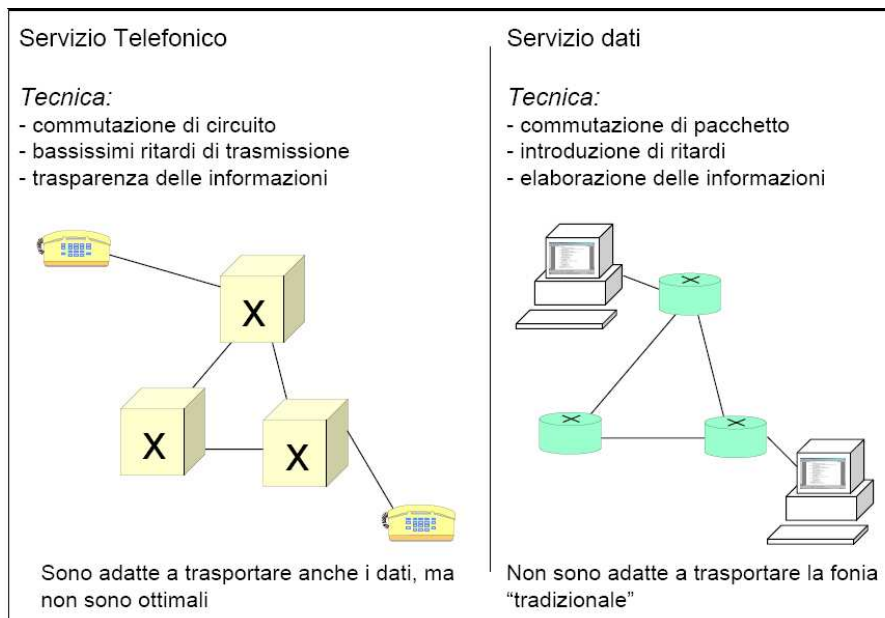


Figura 5 – Differenze CTM-PTM

Nella rete telefonica tradizionale ISDN, il trasferimento CTM consiste nel stabilire una connessione partendo dal telefono di origine attraverso la centrale telefonica di commutazione locale per arrivare alla centrale remota ed, infine, al telefono di destinazione. Mentre è in funzione l'apparecchio telefonico campiona ripetutamente il segnale vocale, codificandolo digitalmente e trasmettendolo sotto forma di dati. Questa modalità possiede sia vantaggi che svantaggi: il lato positivo è la banda di trasmissione costante, che nelle reti ISDN è di 64 Kbps; gli svantaggi però sono molteplici in quanto una banda trasmissiva di 64 Kbps è molto limitata per la trasmissione di pacchetti IP su Internet restringendo quindi le possibilità di ottenere una comunicazione VoIP stabile. [18] Le reti senza connessione, usate solitamente per collegare computer, permettono di ottimizzare i problemi di banda di trasmissione in quanto la rete non è vincolata alla velocità di 64 Kbps, come per l'ISDN, essendo di capacità molto più elevata (anche a Gbps) e viene occupata solamente per il tempo necessario alla comunicazione del pacchetto. Il vantaggio principale della commutazione a pacchetti, utilizzato nelle reti PTM, è che più comunicazioni possono verificarsi parallelamente; lo svantaggio è che

quando il traffico aumenta i singoli computer non possono disporre integralmente della capacità della banda trasmissiva ma comunque la convenienza economica e le prestazioni elevate superano il problema. Ecco quindi che le reti PTM hanno permesso l'evolversi del VoIP.

2.3 - Protocolli VoIP

La tecnologia VoIP richiede due tipologie di protocolli di comunicazione in parallelo: uno per il trasporto dei dati, cioè i pacchetti contenenti i campionamenti della voce come il protocollo RTP (*Real-time Transport Protocol*); l'altro per la codifica della segnalazione della conversazione (per segnalazione s'intende i meccanismi per instaurare, modificare e terminare una comunicazione telefonica) che utilizza altri tipi di protocolli alternativi. Ne esistono diversi, ma i più diffusi sono:

- H.323 della ITU
- Sip (*Session Initiation Protocol*) della IETF

H.323 è uno standard creato nel 1996 dallo ITU-T Study Group 16. La sua creazione fu indotta dalla crescita della comunicazione multimediale su reti a commutazione di pacchetto e fu progettato con un obiettivo: promuovere uno standard per gli utenti delle sessioni di comunicazione audio, video e dati attraverso reti orientate alla connessione dove non è garantita la qualità di servizio (QoS), come sono le reti IP. Implementando l'H.323, i prodotti e le applicazioni multimediali di differenti produttori possono operare congiuntamente senza problemi di compatibilità.

Sip è stato proposto come una possibilità a H.323 per la segnalazione adottando una tipologia di connessione client-server, dove il server è solitamente un intermediario tra due comunicazioni. Il protocollo possiede fondamentalmente le seguenti funzioni: localizzare gli utenti e acquisirne le preferenze; invitare gli utenti a partecipare a una sessione;

instaurare le connessioni di sessione; gestire eventuali modifiche dei parametri di sessione; rilasciare le parti; cancellare la sessione in qualunque momento si desideri. [18]

2.4 - QoS

Nel campo delle reti di telecomunicazioni, il termine Qualità di servizio o più semplicemente QoS (*Quality of Service*) è impiegato per indicare i parametri usati per caratterizzare la qualità del servizio offerto dalla rete (ad esempio perdita di pacchetti e ritardo), o gli strumenti per ottenere una qualità di servizio desiderata.

Nel campo della telefonia, e in generale della commutazione di circuito, la qualità del servizio prevede parametri come:

- livello di rumore sul circuito
- livello sonoro
- probabilità di trovare una linea libera per iniziare una comunicazione
- probabilità di interruzione indesiderata di una comunicazione
- disponibilità del servizio
- durata media e massima dei disservizi

Nel campo delle reti a pacchetto i parametri tipicamente considerati sono:

- **perdita di pacchetti o *dropped packets***: viene considerata la percentuale di pacchetti che la rete nel suo complesso non riesce a consegnare a destinazione. La perdita di un pacchetto viene gestita in modi diversi dai protocolli di trasporto, anche se questo esula dalla definizione della qualità di servizio della rete: in un protocollo senza riscontro, si otterrebbe la mancata trasmissione dell'informazione; in un protocollo con riscontro come TCP, il ricevente dopo aver atteso un tempo ragionevole deve chiedere che l'informazione venga ritrasmessa, causando anche gravi ritardi (*delay*) nella trasmissione complessiva.

- **ritardo (*delay*)** subito da un pacchetto dalla sua immissione nella rete alla consegna al destinatario. Vengono considerate caratteristiche come il ritardo medio ("i pacchetti in media impiegano 10 ms ad attraversare la rete"), e i suoi valori medi ("il 99% dei pacchetti viene consegnato entro 20 ms"). Viene anche considerato il *Jitter*, ovvero la variazione del ritardo tra pacchetti inviati in sequenza da un nodo ad un altro.
- **consegna fuori ordine o *out-of-order***: su alcune reti, è possibile che una sequenza di pacchetti inviati da un nodo ad un altro venga consegnata in un ordine diverso da quello originale. Questo accade tipicamente perché i pacchetti vengono instradati su percorsi diversi. Per risolvere il problema si richiede che i protocolli di trasporto riordinino i pacchetti fuori ordine una volta che sono giunti a destinazione comportando però ulteriori ritardi nella ricostruzione del flusso di dati a livello applicativo.
- **errore di trasmissione**: un pacchetto può essere consegnato a destinazione, ma non essere identico a quello inviato. Molte reti riconoscono la gran parte degli errori di trasmissione, e alcune sono anche in grado di correggere tali errori. Si considera la percentuale di pacchetti errati. Normalmente i protocolli di trasporto riconoscono un pacchetto errato e ne richiedono la ritrasmissione come se questo fosse stato perso, ma è anche possibile che l'errore raggiunga l'applicazione finale.

2.5 - L'indice MOS (*Mean Opinion Score*)

Con l'indice MOS si assegna un valore alla qualità della comunicazione audio. Tali valori, misurati empiricamente, vengono così di seguito attribuiti.

| MOS | Qualità | Disturbo |
|------------|----------------|--------------------------------|
| 5 | Eccellente | Impercettibile |
| 4 | Buono | Percettibile ma non fastidioso |
| 3 | Sufficiente | Un po' fastidioso |
| 2 | Insufficiente | Fastidioso |
| 1 | Scarso | Molto fastidioso |

Figura 6 – Tabella indice MOS

La misura è effettuata analizzando la codifica di alcune frasi convenzionali. Per la lingua Inglese, alcune di queste frasi sono:

- *You will have to be very quiet.*
- *There was nothing to be seen.*
- *They worshipped wooden idols.*
- *I want a minute with the inspector.*
- *Did he need any money?*

La valutazione MOS è un metodo soggettivo per determinare la qualità della voce, ma non è l'unico. ITU-T ha raffinato ulteriormente la raccomandazione P.861 che descrive i procedimenti in cui si può determinare oggettivamente la qualità della voce adoperando il metodo automatizzato di misura PSQM (*Perceptual Speech Quality Measurement*). PSQM presenta molti difetti quando è utilizzato nel campo della voce; uno di questi è il fatto che ciò viene ascoltato da una macchina o da PSQM, non è ciò che viene percepito

dall'orecchio umano. In altri termini si può ingannare l'orecchio umano e percepire una voce con una più elevata qualità mentre con un computer non è possibile. Questo metodo, inoltre, è stato sviluppato in modo da individuare i difetti provocati dalla compressione e decompressione e non dalla perdita di pacchetti o dal tremolio. [5]

2.6 - Algoritmi per la trasmissione della voce

La tecnologia VoIP prevede che, tramite algoritmi software o dispositivi hardware denominati DSP (*Digital Signal Processor*), la voce venga codificata, compressa, frammentata ed inserita in pacchetti dati trasferiti via protocollo IP. In figura 7 è riportato lo *stack* protocollare di una comunicazione VoIP.

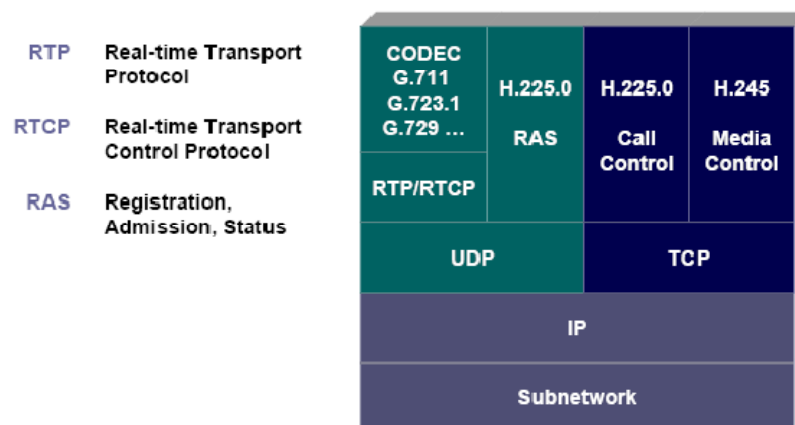


Figura 7 – Incapsulamento VoIP

Esistono diversi criteri di codifica, cioè *Codec*, che convertono il segnale analogico del suono in un segnale digitale e viceversa. Questi si differenziano per la compressione più o meno elevata del segnale vocale o per la capacità di sopprimere l'eco (vedi i punti relativi all'eliminazione dell'eco ibrido ed acustico). La scelta del *Codec* appropriato ottimizza l'impiego più o meno dispendioso della banda Internet a disposizione degli interlocutori. Generalmente, maggiore è la compressione del segnale minore risulterà la qualità della comunicazione telefonica. In figura 8 è riportata una tabella contenente i valori di consumo di banda e di indice MOS per alcuni *Codec* tra i più diffusi.

| Codec | Data rate (kbit/s) | Mean Opinion Score (MOS) |
|--------------|---------------------------|---------------------------------|
| G.711 | 64 | 4.1 |
| G.729 | 8 | 3.92 |
| G.723.1 | 6.3 | 3.9 |
| G.729a | 8 | 3.7 |
| G.723.1 | 5.3 | 3.65 |
| GSM | 13 | 3.8 |
| ILBC | 15 | 4 |

Figura 8 – Tabella *Codec*/indice MOS/consumo banda

Altre tipologie più comuni di *Codec* sono : Speex e DoD FS-1016. L'implementazione di diversi *Codec* nelle varie soluzioni di comunicazioni VoIP, siano essi hardware o software, aumentano la compatibilità tra gli apparati e l'intercomunicabilità. Alcuni *Codec* richiedono che per l'utilizzo vi sia la corresponsione di royalties, altri invece sono free.

2.7 - La famiglia di *Codec* ITU-T (International Telecommunication Union)

ITU è il settore dell'Unione Internazionale delle Telecomunicazioni che si occupa di regolare le telecomunicazioni telefoniche e telegrafiche. Fino al 1993 era noto come CCITT (Comité consultatif international téléphonique et télégraphique).

L'ITU-T fornisce delle raccomandazioni o specifiche standard riconosciute a livello internazionale. Le varie aree di lavoro corrispondono ad altrettante serie, rappresentate da una lettera maiuscola, mentre le singole specifiche interne a ciascuna serie sono rappresentate da un numero. La lettera G, chiamata G-Standard, contraddistingue le specifiche per i sistemi di trasmissione digitale e di rete (*Transmission systems and media, digital systems and networks*). I *Codec* di serie G implementano diverse soluzioni per il

campionamento e la trasmissione della voce, come il campionamento lineare, algoritmi μ -law (mu-law) a-law, ADPCM, AMR-WB, etc..

Per una lista completa degli algoritmi si può reperire abbondante documentazione sul sito ITU-T istituzionale (www.ITU.int).

2.8 - *Codec open source*

Non esistono molti *Codec 'Open Source'* a disposizione. Sono qui di seguito descritti due di particolare rilievo appositamente sviluppati per l'utilizzo in ambito VoIP: iLBC e Speex.

2.8.1 - iLBC

iLBC (Internet Low Bitrate *Codec*) è stato studiato per la trasmissione di voce con un ristretto utilizzo di banda trasmissiva, con payload di 13.33 kbit/s. Una sua caratteristica interessante è la gestione avanzata del delay o perdita di pacchetti IP, senza brusche interruzioni della qualità della voce.

Alcune caratteristiche sono:

- Bitrate di 13.33 kbps (399 bits, in pacchetti di 50 bytes) per una dimensione del frame di 30 ms
- Bitrate di 15.2 kbps (303 bits, in pacchetti di 38 bytes) per una dimensione del frame di 20 ms
- Qualità della voce superiore al *Codec* ITU-T G.729A
- Algoritmo robusto alla perdita di pacchetti TCP
- Complessità computazionale simile al *Codec* ITU-T G.729A

2.8.2 - Speex

Speex è basato sul *Codec CELP (Code Excited Linear Prediction)* e possiede delle caratteristiche aggiunte rispetto alla sola codifica della voce. Le caratteristiche di Speex sono:

- *Narrowband (8 kHz), wideband (16 kHz), and ultra-wideband (32 kHz) compression in the same bitstream*
- *Intensity stereo encoding*
- *Packet loss concealment*
- *Variable bitrate operation (VBR)*
- *Voice Activity Detection (VAD)*
- *Discontinuous Transmission (DTX)*
- *Fixed-point port*
- *Acoustic Echo Canceller*
- *Noise suppression*

La '*Linear Prediction*' consiste nel analizzare la voce per ottimizzare un basso bitrate della trasmissione, caratteristica ovviamente apprezzata in VoIP. Il principio di base sta nella assunzione secondo la quale la voce è il risultato dalla modulazione provocata da gola e bocca (detta formante) della emissione sonora da parte delle corde vocali (il residuo).

Generalmente un segnale vocale è composto da due componenti: un insieme di coefficienti per la predizione lineare e un errore di predizione.

Tra le varie caratteristiche elencate le seguenti sono molto importanti per l'ottimizzazione delle comunicazioni in VoIP:

- '*Acoustic Echo Canceller*' che è un algoritmo che permette di parametrizzare dinamicamente sia il *frame_size* del campionamento che il *filter_length* del riverbero utilizzando un filtro adattivo.

- il *'Jitter Buffer'*, che è una memoria che mantiene i pacchetti arrivati a destinazione in anticipo rispetto ad altri, situazione occorrente con lo sfruttamento del protocollo UDP.

Questo *Codec*, per le sue caratteristiche è senz'altro una delle migliori soluzioni Open Source presenti sul mercato. Il grafico in figura 9 illustra un comparazione tra Speex, iLBC ed altri *Codec* commerciali tra i più conosciuti con l'indice MOS rilevato in relazione dei bitrate impiegati.

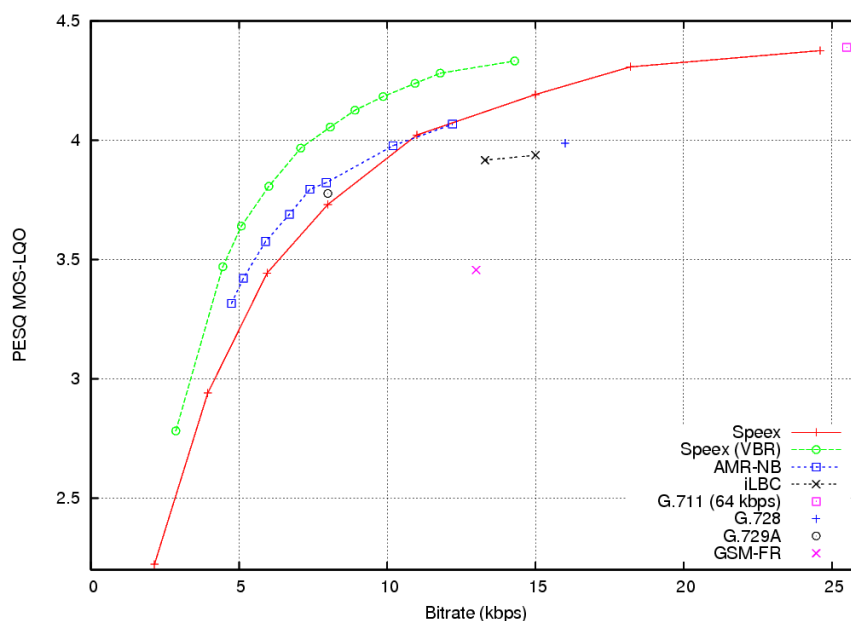


Figura 9 – Comparazione tra *Codec*

2.9 - Il business del VoIP: Asterisk e Skype

Sostanzialmente sono due gli approcci al VoIP: uno è la possibilità di integrazione tra informatica e telefonia; l'altro è l'utilizzo della tecnologia per ridurre i costi delle telefonate.

Ecco quindi che due prodotti software si distinguono nel mercato del VoIP per le potenzialità offerte nei due scenari:

- **Asterisk**, un progetto totalmente *Open Source* indirizzato alla creazione di un centralino telefonico software dalle potenzialità 'virtualmente illimitate'. Basato su

piattaforma Linux e corredato da opportune interfacce hardware, Asterisk permette di gestire telefonate sia VoIP che tradizionali non avendo nulla da invidiare ad un moderno centralino telefonico. Si riesce inoltre ad interfacciarlo a tutti i provider VoIP esistenti per inoltrare o ricevere chiamate attraverso Internet, minimizzando i costi delle telefonate, soluzione ottimale specie per le aziende che comunicano con filiali dislocate remotamente o con l'estero.

- **Skype**, un software di *instant messaging* e VoIP, introdotto nel 2002, capace di unire caratteristiche presenti nei client più comuni (chat, salvataggio delle conversazioni, trasferimento di file) ad un sistema gratuito di telefonate basato su un *network P2P*. La possibilità di far uso di un servizio a pagamento, SkypeOut, che permette di effettuare chiamate a telefoni fissi, rendono il programma competitivo rispetto ai costi della telefonia tradizionale, soprattutto per le chiamate internazionali e intercontinentali.

Sfortunatamente non è possibile integrare le due soluzioni in quanto Skype possiede un codice proprietario e quindi non modificabile. Sta all'utente finale adottare la soluzione che più soddisfa le sue esigenze: aziende e istituzioni di adeguate dimensioni che intendono sfruttare il VoIP possono adottare una soluzione basata su Asterisk; Skype invece può essere adottato come una soluzione personale per risparmiare sul traffico telefonico.

Il mercato VoIP sembra crescere in tutto il mondo per raggiungere, 196 miliardi di dollari nel 2007, gran parte dei quali provenienti dal segmento business. Anche l'Europa si muove all'interno dell'onda crescente: i minuti di conversazione VoIP, quintuplicati nel 2005 e triplicati nel 2006, indicano una crescita dell'80% nel 2007. I vantaggi dell'adozione di soluzioni IP da parte delle imprese utenti sono, oltre alla riduzione dei costi delle chiamate che può essere anche molto diversificata da impresa a impresa, il miglioramento dell'efficienza in termini di riduzione del Tco (*Total Cost of Ownership*) e il supporto al business. [19]

Capitolo 3

Analisi dei disturbi nelle comunicazioni

3.1 - Il problema delle comunicazioni VoIP in condizioni critiche

Se consideriamo alcuni ambienti di utilizzo dei servizi VoIP come quelli *mission-critical* o *safety-critical*, l'impossibilità di comunicare potrebbe portare a gravi conseguenze. Per esempio: non poter comunicare in una situazione di disastro o in un campo di battaglia potrebbe originare delle problematiche tali da causare anche la perdita di vite umane.

Una delle caratteristiche richieste dalla comunicazione digitale, ovvero il VoIP, è l'elaborazione in tempo reale della voce in modo tale da rimuovere rumori di fondo ed eco. In questi casi l'adattabilità degli algoritmi di *echo-cancellation* e *noise-cancellation* si rivela un'importante proprietà essenziale per poter impiegare terminali di comunicazione semplici e pratici, che ne svincolino l'utilizzo da cornette telefoniche o cuffiette e microfono. L'impiego di apparati simili a delle ricetrasmittenti, più maneggevoli ma con una tendenza a riportare tutti i disturbi ambientali, deve essere funzionale. Gli algoritmi di *echo-cancellation* devono essere in grado di eliminare l'eco acustico generato dall'ambiente in cui la ricetrasmittente si trova ad operare; gli algoritmi di *noise-cancellation* devono riconoscere rumori di fondo come sirene, esplosioni, motori di vario genere, etc..

Un'altra variabile da considerare nella comunicazione VoIP in ambiente *mission/safety-critical* è il mezzo trasmissivo. Quello più adatto allo scopo è il supporto WiFi dato che non è pensabile infatti che il terminale sia collegato con un cavo al centralino telefonico. Per assicurare una comunicazione funzionale VoIP su WLAN è necessario servirsi di misure speciali: se il sistema non opera in determinate condizioni il risultato è la perdita di qualità del segnale con la possibile caduta della comunicazione. Se un sistema VoIP su rete cablata dispone di sistemi di backup per l'alimentazione elettrica e degli apparati di

comunicazione, un sistema mobile possiede un sistema di alimentazione indipendente e le WLAN devono essere create ad-hoc. [6]

Si deve considerare che non sempre lo scenario operativo si realizza su terreni piatti: come si vede in figura 10, colline, montagne ed ogni elevazione potrebbero far decadere o interrompere il segnale portante.

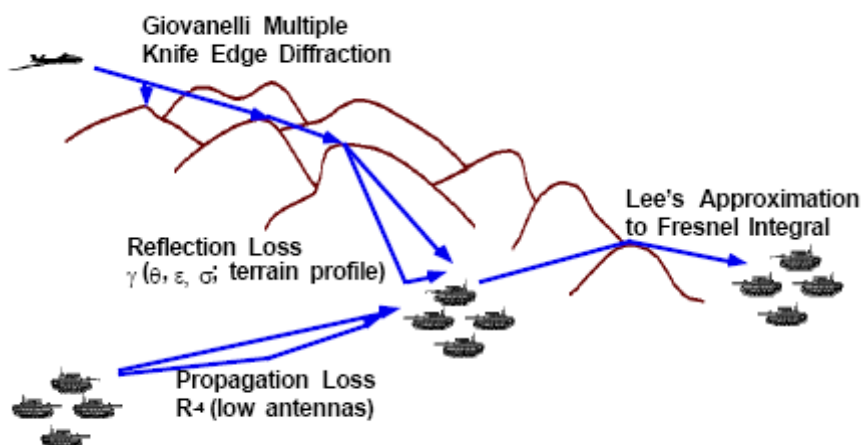


Figura 10 – Esempio di propagazione WiFi

Si deve quindi creare una rete robusta per la propagazione del segnale. [7]

Alcuni di questi concetti saranno ripresi nel proseguire del presente documento.

3.2 - Il problema dell'eco acustico nelle comunicazioni telefoniche.

Distaccandoci temporaneamente dall'analisi della comunicazione VoIP, cominciamo ad analizzare una problematica fondamentale da risolvere nell'ambito della trasmissione e trasduzione della voce: l'eco.

Perché esiste l'eco ?

Un qualsiasi sistema elettro-meccanico di comunicazione vocale si compone, essenzialmente, di due componenti: il microfono, per registrare la voce; l'altoparlante, per udirne quella dell'altro interlocutore. L'effetto eco si presenta quando il suono in uscita dall'altoparlante viene ripreso dal microfono, come rappresentato in figura 11.

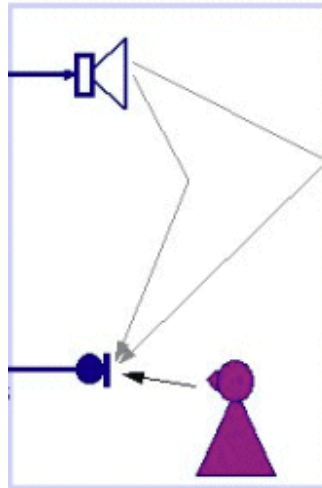


Figura 11 – Eco acustico

Cause di eco acustico si possono ottenere con l'utilizzo di diversi dispositivi di comunicazione come:

- microfono e casse audio per computer;
- sistemi vivavoce per automobili;
- telefoni standard in modalità vivavoce;
- sistemi di teleconferenza;
- cornette telefoniche di bassa qualità (le vibrazioni generate dall'altoparlante della cornetta sono propagate fino al microfono).

Il verificarsi dell'eco acustico è dovuto sostanzialmente da due cause: la propagazione della voce dall'altoparlante al microfono, per la scarsa isolamento tra i due componenti e la distanza tra gli interlocutori.

Perché l'effetto eco sia rilevabile all'orecchio umano dobbiamo trovarci in presenza di due fattori: deve essere udibile e possedere sufficiente ritardo rispetto al suono inviato. In caso contrario non si può parlare di eco ma eventualmente di riverbero.

In fisica e in acustica l'eco è un fenomeno prodotto dalla riflessione di onde sonore contro un ostacolo che vengono a loro volta nuovamente "percepiti" dall'emettitore più o meno immutate e con un certo ritardo rispetto al suono diretto. Tale ritardo non deve essere inferiore ad 1/10 di secondo. Al di sotto di tale valore si ottiene il riverbero.

In termini più generali, in fisica l'eco è definito come un'onda che viene riflessa da una discontinuità nel mezzo di propagazione, e che ritorna con una intensità e ritardo sufficiente per essere percepita.

Anche nelle comuni comunicazioni su linea telefonica tradizionale (PSTN - *Public Switched Telephone Network*) può prodursi un effetto eco. Nelle comunicazioni urbane il problema non sussiste in quanto il ritardo è così breve che al massimo si ottiene un riverbero; per chiamate su lunga distanza invece l'eco esiste ed è eliminato tramite dispositivi hardware.

Ma non sempre l'eco deve essere forzatamente eliminato: un eco 'buono' è quello che in telefonia classica viene chiamato '*sidetone*'. Il '*sidetone*' può essere paragonato ad un eco immediato della propria voce, cioè quanto viene detto al microfono passa immediatamente allo *speaker* ad un volume più basso; questo effetto è necessario per fornire agli interlocutori di chiamate telefoniche tradizionali la certezza che l'apparecchio stia correttamente funzionando. Per spiegare il concetto basti pensare a come si comporta il cervello umano durante una normale dialogo, non telefonico, tra persone: ogni parola pronunciata viene sempre ascoltata dall'orecchio ed il cervello possiede dei meccanismi di 'filtraggio' che impediscono di distogliere l'attenzione da quanto si sta dicendo aiutando a regolare la forza della propria voce. Nel caso venisse a mancare questo ritorno, come se per esempio le orecchie fossero otturate, il primo effetto che verrebbe a crearsi sarebbe l'incapacità di regolare la forza della propria voce. Il '*sidetone*' quindi è necessario all'interlocutore per assicurarsi del funzionamento dell'apparecchio e per regolare la forza della propria voce durante la comunicazione.

Ragionando sul comportamento del cervello umano si può fare un'altra riflessione: poniamo che durante una conversazione, non telefonica, invece che sentire la propria voce l'individuo udisse un'altra persona che ripete esattamente ciò che sta dicendo; accadrebbe che colui ne fosse disturbato e potrebbe perdere il filo del discorso. L'eco viene percepito

dal cervello umano proprio come la replica della propria voce come se questa fosse riprodotta da terzi e quindi non è in grado di filtrarla: ecco spiegato il motivo per il quale questo effetto è tanto fastidioso.

Come è già stato detto l'eco, per essere percepito, deve essere in ritardo e sufficientemente forte da essere udito. La mancanza di uno dei fattori elimina l'effetto. Ma quanto deve essere questo ritardo perché l'eco si presenti? E' stato calcolato che la soglia di percezione dell'eco corrisponde ad un ritardo di 50 ms (millisecondi); ad un valore di 100 ms l'eco rende la conversazione faticosa.

In una conversazione telefonica, sia essa analogica o digitale, se si desiderasse mantenere un ritardo di 50 ms non si potrebbe coprire grandi distanze nemmeno se il segnale viaggiasse alla velocità della luce. Questa affermazione la si può facilmente verificare con un calcolo: la velocità della luce è di $3 \cdot 10^8$ m/s (metri al secondo); applicando la formula (2)

$$distanza = \frac{ritardo \times velocità}{2} \quad (2)$$

(ritardo x velocità deve essere diviso a metà in quanto l'eco è un disturbo riflesso e quindi la distanza percorsa è il doppio della distanza effettiva tra gli interlocutori) il risultato è 7.500 Km.

A questo punto si comprende che il fattore di ritardo dell'eco non può essere eliminato aumentando la velocità delle trasmissioni, quindi l'unica soluzione è eliminare l'altra componente, cioè la sua forza, fino a renderlo non udibile.

3.3 - L'eliminazione dell'eco acustico

Per eliminare l'eco acustico si deve, sostanzialmente, riconoscere il suono in uscita dall'altoparlante ed eliminarlo nel momento in cui rientra nel microfono.

La difficoltà dell'operazione è dovuta dal fatto che il suono entrante nel microfono è spesso distorto ed in ritardo. Potrebbe poi sommarsi anche un rumore di fondo che

complica la procedura di eliminazione, senza considerare che a seconda dell'ambiente l'eco può essere diverso, cioè con ritardo e forza sonora sempre differente, annullandone la linearità. I dispositivi AEC (*Acoustic Echo Celler*) dovranno quindi integrare dei filtri adattivi in grado di misurare il segnale in ingresso: questi filtri possono essere modellati matematicamente, adottando dei criteri di controllo dell'errore. In figura 12 è riportato lo schema base di un AEC.

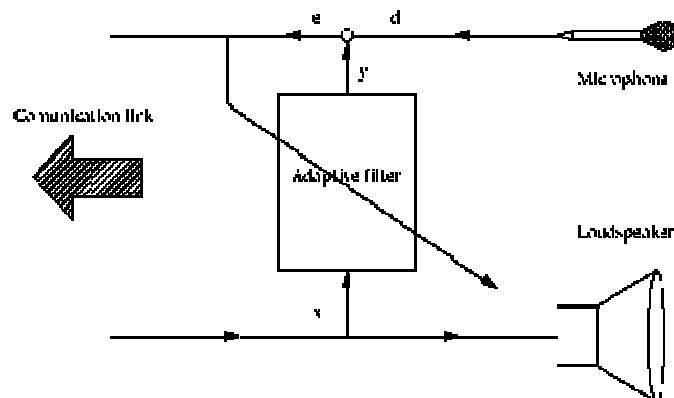


Figura 12 - Schema AEC base

Oltre al filtro adattivo un AEC può contenere anche altri componenti come:

- *Far End Speech Detector (FESD)*
- *Near End Speech Detector (NESD)*
- *Double Talk Detector (DTD)*
- *Control Unit (CU)*
- *Non Linear Processor (NLP)*
- *Noise Reduction Unit (NR)*
- *Automatic Gain Control (AGC)*

Questi componenti si integrano al filtro adattivo (vedi figura 13) in modo tale da ottimizzare e migliorare la qualità della voce. [8]

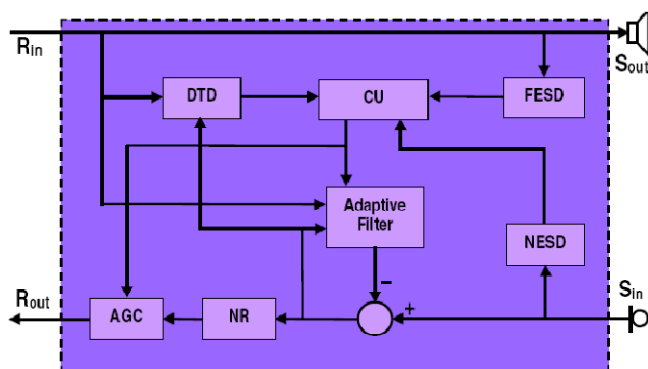


Figura 13 – Schema AEC avanzato

3.4 - L'eco ibrido

Un'altra tipologia di eco rilevabile nelle comunicazioni telefoniche è "l'eco ibrido".

L'eco ibrido viene prodotto nella rete PSTN, che è la rete analogica di telefonia tradizionale, durante la conversione del segnale elettrico da un sistema a quattro ad uno a due fili, come mostrato in figura 14.

Il disadattamento di impedenza¹ tra le due reti genera la riflessione del segnale che, unitamente al ritardo dovuto al percorso inverso, riascoltiamo sottoforma di eco.

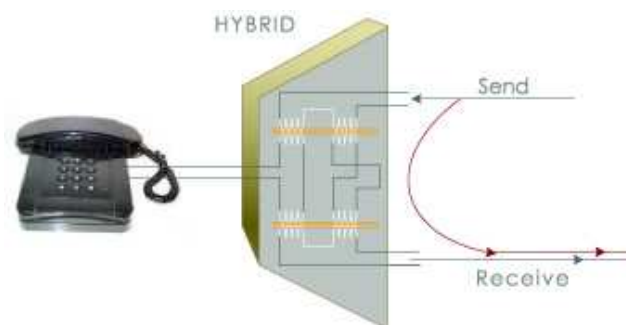


Figura 14 - Hybrid

¹ L'impedenza è la grandezza fisica che rappresenta il rapporto tra un fasore della tensione e un fasore della corrente. Comunemente è indicata con Z ed, essendo il rapporto di due fasori, è un numero complesso. Il termine fu coniato da Oliver Heaviside nel luglio del 1886. In termini classici si può dire che l'impedenza esprime la "resistenza" che un bipolo oppone al passaggio di una corrente elettrica alternata e si misura in ohm. Il concetto di impedenza generalizza la Legge di Ohm estendendola ai circuiti funzionanti in regime sinusoidale (comunemente detto corrente alternata).

Questo tipo di disturbo, in ambito VoIP, si presenta quando il segnale transita dalla rete VoIP a quella PSTN (e viceversa). Il circuito appena descritto, responsabile della generazione dell'eco, viene denominato *'Hybrid'*. [9]

3.5 - L'eliminazione dell'eco ibrido

Entrambi i tipi di disturbi, cioè l'eco acustico e l'eco ibrido, risultano comunque alquanto fastidiosi, soprattutto quando il ritardo della voce è rilevante. Non di rado inducono alla drastica interruzione della comunicazione. L'eco ibrido è un fenomeno che per la sua caratteristica di linearità è possibile eliminare completamente attraverso algoritmi abbastanza efficaci, anche se può capitare che in alcuni casi la trasmissione perda di linearità, per esempio in presenza di *'Jitter'* e quindi l'eliminazione dell'eco non è fattibile. Principalmente l'eliminazione dell'eco ibrido si realizza avvalendosi di un *'Echo Canceller'* lineare. Gli *'Echo Cancellers'* sono dei sofisticati algoritmi, integrati in processori chiamati DSP (*Digital Signal Processor*). La modalità di funzionamento è la seguente: durante il campionamento della segnale vocale, l' *'Echo Canceller'* crea un modello del percorso dell'eco da utilizzare per la stima dell'eco stesso. Questa stima viene sottratta dal segnale vocale in modo tale che transiti solo la comunicazione, senza eco. Chiaramente questa è una spiegazione semplice: gli *'Echo Cancellers'* devono possedere la capacità di campionare eco multipli rilevati in tempi e su livelli diversi. Consideriamo il seguente esempio in figura (15).



Figura 15 – Rimozione eco lineare

La trasmissione dal telefono di San Francisco attraversa: un *'Echo Canceller'*, la rete PSTN, l'*'Echo Canceller'* di Philadelphia e raggiunge il telefono di Philadelphia. Con un

percorso contrario la trasmissione dal telefono di Philadelphia arriva a San Francisco. Quando la trasmissione dal telefono di San Francisco transita attraverso l' *'Hybrid'* di Philadelphia, parte del segnale viene ripetuto e riflesso: se questo eco non fosse eliminato l'interlocutore di San Francisco lo percepirebbe. Quindi sarà responsabilità dell' *'Echo Cancellor'* di San Francisco, eliminare l'eco indotto dal proprio *'Hybrid'*; analogamente l' *'Echo Cancellor'* di Philadelphia dovrà eliminare quello di propria competenza.

3.6 - L'eco nelle trasmissioni VoIP

La telefonia VoIP rende ancor meno trattabile l'eco a causa della caratteristica di funzionamento del protocollo stesso. Comunicazioni tra interlocutori VoIP-VoIP o PSTN-VoIP sono aggravate da ulteriori disturbi. Una conversazione come quella in figura 16, di tipo PSTN-VoIP, non è 'pulita': Alice, interlocutore PSTN, riceve il proprio eco, rinviato da Bob, interlocutore VoIP, ma quest'ultimo non è affetto dal disturbo in quanto la rete PSTN elimina il suo eco.

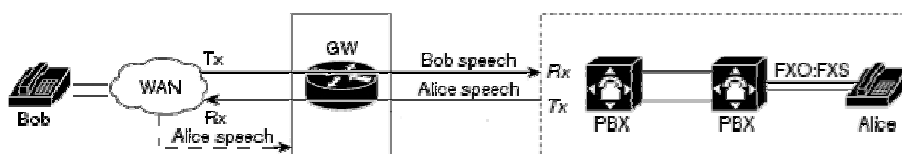


Figura 16 – Eco PSTN/VoIP

Non è semplice far fronte a questi disturbi: innanzitutto ci sono alcune problematiche di carattere tecnico nell'interazione tra l'architettura VoIP e Internet. Come si può facilmente intuire, la comunicazione della voce richiede una trasmissione dei pacchetti in tempo reale ma, a causa della struttura fisica di Internet, questi devono attraversare molti router e il *Round Trip Time* (RTT) tende ad incrementare fino a valori elevati. L'eco diventa quindi non lineare e, conseguentemente, i processori DSP lineari si scoprono inutili.

Un'altra tipologia molto importante di eco è l'eco acustico. Se i vantaggi apportati dalle comunicazioni VoIP eliminano la tradizionalità dell'apparecchio telefonico, in quanto

l'utilizzo del Personal Computer è principe, microfoni, casse audio e cuffie non possiedono caratteristiche adeguate per l'isolazione del segnale: l'eco acustico risulta quindi essere un problema ingente e difficilmente trattabile.

3.7 - Algoritmi di *Echo cancellation*

Gli *Echo Canceller* (EC), come per i *Codec*, sono degli algoritmi matematici reperibili sia in modalità open source che commerciale. Gli attuali DSP si basano su filtri adattivi che devono costantemente effettuare misurazioni sulla qualità della voce in modo tale che sappiano riconoscere ed eliminare quello che è l'eco da quella che è la voce da trasmettere. L'eliminazione dell'eco lineare è molto più semplice da realizzare rispetto all'eliminazione dell'eco acustico in quanto i LEC (*Line Echo Cancellers*) infatti sono progettati per eliminare l'eco ibrido, diverso dall'eco originato, per esempio, da pareti come auditori o teatri. Nella seguente figura 17 è riportato lo schema progettuale di un algoritmo per la cancellazione dell'eco acustico.

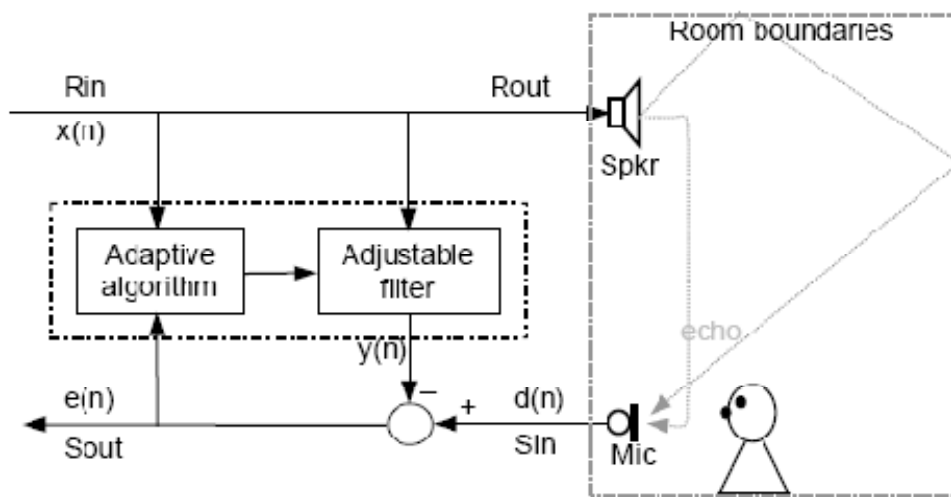


Figura 17 – Schema algoritmo AEC

Le performance di un algoritmo per la cancellazione dell'eco devono essere valutate in base a due sostanziali criteri: il rispetto delle raccomandazioni ITU-T sull'eliminazione

dell'eco; il valore di ritardo che è in grado di riconoscere a seconda dell'ambiente in cui è utilizzato.

Questo secondo valore, chiamato in gergo '*Tail length*', è difficile da determinare ed è legato direttamente al tempo di riverbero dell'ambiente. Ad un maggiore tempo di riverbero è necessario un *Tail length* maggiore. Se questo è inferiore si avvertirà sempre l'eco. Un metodo semplice per calcolare il valore di '*Tail length*' di una stanza può essere l'applicazione della seguente formula (3):

$$T = (N + 1) * \frac{d}{c} \quad (3)$$

Dove: **N** è il numero dei possibili punti di riflesso da eliminare (per esempio una stanza è composta da 6 pareti); **d** è la distanza tra l'interlocutore ed il muro (in metri); **c** è la velocità del suono ($343 \frac{m}{s}$).

Per esempio per una stanza larga 4 mt, alta 2.8 mt e lunga 5 mt si otterrà il seguente valore (4):

$$T = (6 + 1) * \frac{5}{343} = 0,102 \text{ sec} = 102 \text{ msec} \quad (4)$$

Ovviamente per il valore **d** è stata considerata la massima distanza possibile; quindi nell'esempio abbiamo appreso che l'algoritmo EC dovrà riuscire a trattare ritardi d'eco di almeno 100 msec.

La letteratura insegna che un buon algoritmo di EC deve saper trattare eco con un ritardo di almeno 200 msec.

Esistono, chiaramente, una varietà molto vasta di algoritmi per la cancellazione dell'eco: tali algoritmi si adattano ai più svariati usi e si differenziano per il campo di applicazione come la telefonia standard o IP. Attualmente i progettisti di tali algoritmi sono aziende interessate alla rivendita delle proprie soluzioni e quindi non è possibile accedere alle informazioni senza incorrere in problemi di copyright o brevetti. Le poche soluzioni di algoritmi open source disponibili nel campo della telefonia VoIP sono utilizzate in Asterisk, un software open source che permette di realizzare un centralino telefonico VoIP. In Asterisk sono implementati gli algoritmi di seguito elencati, dei quali però le modalità di

funzionamento sono perlopiù sconosciute, ma comunque è possibile ottenerne il codice sorgente.

- Asterisk mark *Echo Celler* (mec)
- Asterisk mark2 *Echo Celler* (mec2)
- Asterisk mark3 *Echo Celler* (mec3)
- Asterisk steve *Echo Celler* (sec)
- Asterisk steve2 *Echo Celler* (sec2)
- Asterisk kb1 *Echo Celler* (kb1ec)
- Asterisk mg2 *Echo Celler* (mg2ec)

3.8 - Come funziona un *Acoustic Echo Celler* (AEC)

La maggior sfida nel campo degli *Echo Cellers* sta nel permettere di effettuare conversazioni a mani libere utilizzando un microfono ed un altoparlante.

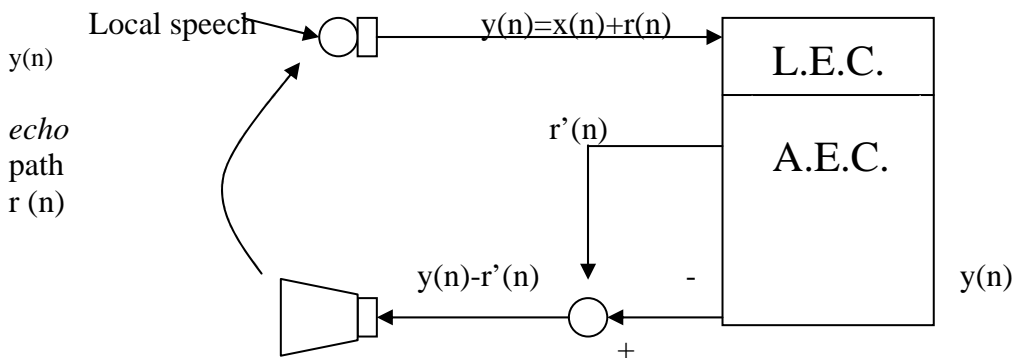


Figura 18 – LEC + AEC

In una situazione come quella evidenziata in figura 18 è utilizzato: un filtro lineare (LEC) per misurare il percorso fisico tra il microfono e l'altoparlante per eliminare l'accoppiamento diretto che possiede un ritardo fisso; un AEC per misurare ed eliminare il riverbero, tramite un algoritmo adattivo.

L'implementazione dell'algoritmo per l'AEC non è semplice: deve essere sempre campionato il segnale in ingresso al microfono ed il segnale in uscita dall'altoparlante in quanto le condizioni acustiche cambiano costantemente. Basti pensare che un interlocutore si può muovere per la stanza. [10]

L'approccio standard per l'eliminazione dell'eco acustico prevede l'utilizzo di filtri adattivi lineari a tempo discreto per caratterizzare quantitativamente l'effetto acustico indesiderato, generando una replica elettronica dell'eco. Questa replica è poi sottratta dall'eco originale per disaccoppiare casse e microfono dal punto di vista acustico.

Il tempo necessario al cancellatore d'eco per apprendere completamente le caratteristiche acustiche è chiamato "tempo di convergenza t_c ". Un buon tempo di convergenza è 50ms.

Un altro fattore importante è l'*Acoustic Echo Return Loss Enhancement* (AERLE) o, in altri termini, l'attenuazione applicata all'eco originale; un buon valore è 65 dB.

Gli algoritmi adattivi LMS (*Least Mean Square*) e NLMS (*Normalized Least Mean Square*) sono i più usati per questo scopo. La loro funzione è di minimizzare il quadrato del segnale errore (quadrato dell'eco); quadrato perché si lavora sulla potenza dei segnali. NLMS ha prestazioni scarse in presenza di rumore. Un algoritmo particolarmente utilizzato è l'APA (*Affine Projection Algorithm*). Altri algoritmi non sono adatti per via dell'elevata complessità computazionale richiesta (come per es.: RLS- *Recursive Least Square*).

Genericamente gli algoritmi lavorano nel seguente modo: a partire da un set di valori nulli per i coefficienti del filtro, dopo ogni iterazione questi valori sono adattati in modo da minimizzare l'eco acustico. La velocità con cui ciò avviene, la precisione della soluzione dopo la convergenza, il numero di calcoli necessari per ogni iterazione, la robustezza dell'algoritmo, la sua semplicità, sono tutti fattori da tenere in considerazione in fase di progettazione.

Per migliorare le prestazioni, gli algoritmi sono supportati da filtri adattivi a risposta impulsiva finita, detti FIR. (*Finite Impulse Response*). [11]

3.9 - Qualità della comunicazione VoIP

I fattori che devono essere considerati riguardanti la qualità della comunicazione, quando si utilizza il VoIP, sono:

- Delay e latenza
- *Jitter Buffer*
- *Voice Compression (coding/decoding)*
- Perdite di pacchetti (*packet loss*)
- VAD (*Voice Activity Detection*)

Delay e latenza: è il tempo che intercorre da quando il segnale audio esce dalla bocca dell'oratore ed arriva alle orecchie dell'ascoltatore. Esistono principalmente due cause di delay nelle comunicazioni: il VoIP Delay di propagazione, cioè il tempo fisico di attraversamento del mezzo trasmissivo (es: comunicazioni satellitari...); il Delay di *Handling*, cioè il tempo necessario per gestire i pacchetti IP (pacchettizzazione, compressione, instradamento...)

Jitter e Jitter Buffer: i pacchetti IP dalla rete non arrivano sempre ad intervalli regolari. Il *Jitter* è la differenza tra il tempo di arrivo e quello ideale. Per compensare questa differenza viene utilizzato il *Jitter Buffer*, che però possiede un peso computativo che rallenta ulteriormente il ritardo. Una gestione del *Jitter Buffer* dinamico, a seconda del livello di congestione e prestazioni della rete, è chiamato *Jitter Buffer* adattativo, ed è sicuramente più affidabile del *Jitter Buffer* semplice.

Voice Compression: l'utilizzo di un *coder* per trasformare il segnale da analogico in digitale e di un *decoder* per l'operazione inversa. In questa fase il segnale può essere ulteriormente compresso per ottimizzare i tempi di trasmissione a discapito però della qualità della voce.

Perdite di pacchetti: l'utilizzo del protocollo RTP nelle applicazioni VoIP assicura una continuità nella conversazione anche se alcuni pacchetti sono persi o in ritardo. Questo sicuramente influisce sulla qualità della comunicazione ma almeno non è interrotta. Nel caso un pacchetto non arrivi nel tempo previsto il client ripete l'ultimo pacchetto ricevuto: solitamente questa ripetizione non viene rilevata dall'ascoltatore.

VAD (*Voice Activity Detection*): in una comunicazione bidirezionale non sempre entrambi gli interlocutori parlano. Quindi durante queste pause di silenzio la banda, allocata ma non utilizzata, è sfruttata per trasmettere altre informazioni. Per rilevare il discorso si utilizza un algoritmo di *Voice Activity Detection*. Un buon algoritmo VAD deve essere in grado di ridurre al minimo i '*false detect*' (aprire il circuito anche non in presenza di conversazione) ed i '*miss detect*' (non aprire il circuito in presenza di conversazione). [12]

3.10 - Come influisce la qualità della rete sulla qualità del segnale in VoIP

Per valutare la qualità della voce '*over IP*' si possono considerare tre punti di vista: la qualità soggettiva, cioè come viene percepita la voce dagli interlocutori; la qualità oggettiva, che con l'aiuto della qualità soggettiva misura il funzionamento degli algoritmi preposti a migliorare la comunicazione; la qualità della rete di trasporto, che può essere facilmente misurata sui parametri come il *Jitter*, il *packet loss* o il *packet delay*. In figura 19 è riportato uno schema esemplificativo. [13]

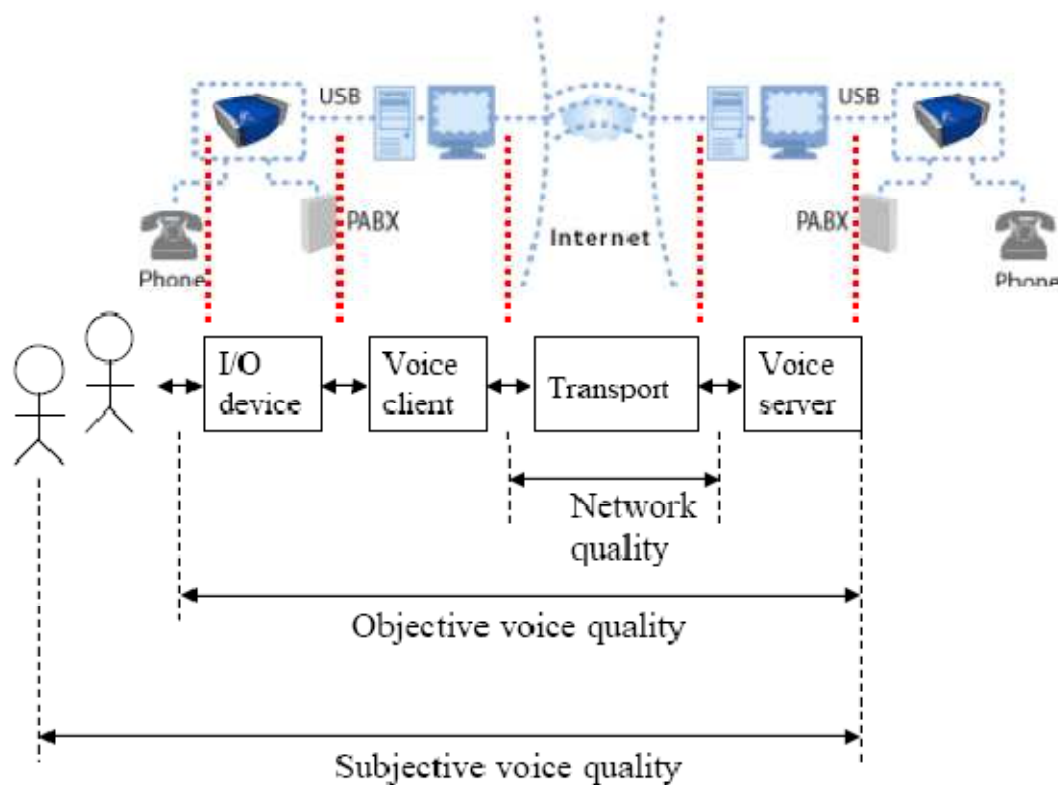


Figura 19 - Livelli di percezione della qualità

Il trasporto si rivela quindi una componente importante per la qualità della voce: a seconda dell'ambiente questo può essere più o meno veloce e più o meno protetto. Per esempio in una lan aziendale l'alta velocità di comunicazione, che attualmente è difficile trovare minore di 100 Mbit/s, influisce poco sulla qualità della voce; via Internet invece si pone il problema opposto e cioè, come già precedentemente esposto, i pacchetti possono arrivare in ritardo o nell'ordine sbagliato; infine su reti WLAN il trasporto viene complicato da molti fattori quali la crittografia o la perdita di qualità del segnale, dovuto a problematiche di propagazione.

In Internet generalmente una scarsa qualità di rete decrementa le performance di un sistema VoIP. Nelle applicazioni VoIP il delay, il *Jitter* e la perdita di pacchetti sono le principali cause che influiscono sulla qualità della voce. Il *Jitter* può essere parzialmente compensato con l'utilizzo di un *Buffer*, detto di 'payout', dal ricevente, ma questo causa altro ritardo e perdita di pacchetti. Una rete IP è composta da componenti sia logici che fisici come: i protocolli di rete, routing e di controllo del traffico; le modalità di

funzionamento dei router; la banda portante trasmissiva e l'affidabilità della rete. Quest'ultima è una componente molto importante in quanto è causa del delay e della perdita di pacchetti e può essere influenzata da due fattori principali: la riconfigurazione dinamica dei percorsi di routing e la perdita di link.

3.11 - VoIP e WLAN

I recenti disastri come lo Tsunami del 2004 o l'uragano Katrina in USA hanno dimostrato l'importanza dei sistemi di comunicazione di emergenza. Anche in campo militare le comunicazioni in battaglia sono da ritenersi critiche. Lo standard di comunicazione via cavo si rivela spesso inefficace e, quindi, un'alternativa valida da valutare è l'utilizzo delle reti wireless. Le reti wireless sono molto più stabili e decentralizzate rispetto alle altre infrastrutture di comunicazione su cavo. Quindi è possibile creare delle WLAN (*Wireless Lan*) strutturate in maniera tale da non interrompersi in situazioni critiche e/o di emergenza. [14]

Attualmente le WLAN non sono adatte alla trasmissione di protocolli real-time (RTP) per le seguenti ragioni:

- I parametri QoS possiedono un'alta variabilità a seconda dell'ambiente in cui operano e quindi possono influire significativamente sulle performance dell'applicazione VoIP. Inoltre i meccanismi QoS delle WLAN sono sviluppati per il management della portante WiFi, mentre le applicazioni con altri QoS come il VoIP e TCP condividono lo stesso canale (cioè VoIP non sfrutta un canale prioritario).
- Il roaming tra gli *Access Point* introduce delle interruzioni nella comunicazione della durata di anche un secondo, tempo inaccettabile dal punto di vista della comunicazione vocale.
- Una possibile soluzione è quindi sviluppare algoritmi di comunicazioni VoIP, come il *Coding*, il *Noise Reduction* e l'*Echo cancellation* capaci di adattarsi alle variazioni di stabilità della comunicazione dovute all'attuale modalità di

funzionamento del VoIP. Nel proseguire di questo documento ci concentreremo sull'analisi della variazione nel comportamento degli algoritmi di *Echo cancellation* a seconda della variabilità della rete portante.

3.12 - *Echo cancellation* in VoIP su WLAN

Come era stato già accennato precedentemente, migliorare la qualità della trasmissione della voce in IP richiede diversi approcci seppur mantenendo i concetti di base. Si devono prendere in considerazione le variabili introdotte dall'infrastruttura di comunicazione IP, come LAN, Internet o WLAN. Ecco quindi che algoritmi standard di *Echo cancellation*, come NMLS o LMS, non si adattano automaticamente alla varianza della comunicazione in quanto sono sviluppati per infrastrutture robuste. Gli stessi algoritmi di EC introdotti in Asterisk non sono quindi efficaci e appropriati solo per reti locali, cioè utilizzando PBX Asterisk in rete LAN con *TRUNK* su rete PSTN e telefoni IP. Si deve comunque tenere in considerazione che l'avanzare delle competenze tecnologiche permette di realizzare apparecchiature di comunicazione IP sempre più efficaci, migliorando di conseguenza la validità della trasmissione dati su rete geografica. [15]

Basti considerare l'evoluzione della distribuzione della connettività ADSL nei comuni italiani: da pochi accessi a velocità sensibilmente basse a molti ad alta velocità.

Gli accessi a banda larga hanno superato la soglia degli 8 milioni a fine settembre 2006, rispetto alle poche centinaia di migliaia di solo cinque anni fa. Il 27% delle famiglie italiane, il 69% delle imprese con 3 e più addetti (40% sul totale imprese) e il 75% delle amministrazioni pubbliche dispongono oggi di collegamenti a banda larga (fonte: Osservatorio Banda Larga – Between).

E' intuitivo pensare che problematiche di trasmissione dati su IP, come *Jitter*, *Delay*, *latenza*, etc.. , siano nettamente diminuite oggi rispetto agli anni passati e, quindi, le comunicazioni VoIP via Internet possano essere più robuste. Con le reti WLAN però le cose cambiano: la *latenza* varia continuamente, *QoS* è specificamente strutturato per la robustezza del segnale ma non per lo strato del canale applicativo ed il tutto si riflette negativamente sulla qualità della comunicazione VoIP. Ecco che in situazioni critiche o di

Battlefield, dove le comunicazioni WLAN sono da preferirsi per le motivazioni già esposte, è necessario adattare gli algoritmi di ‘*Speech Quality*’ allo scopo di rimediare a queste variazioni.

L’*Echo cancellation* torna ad essere un problema importante da risolvere per ottenere una qualità accettabile della voce. Consideriamo una situazione come quella in figura 20:

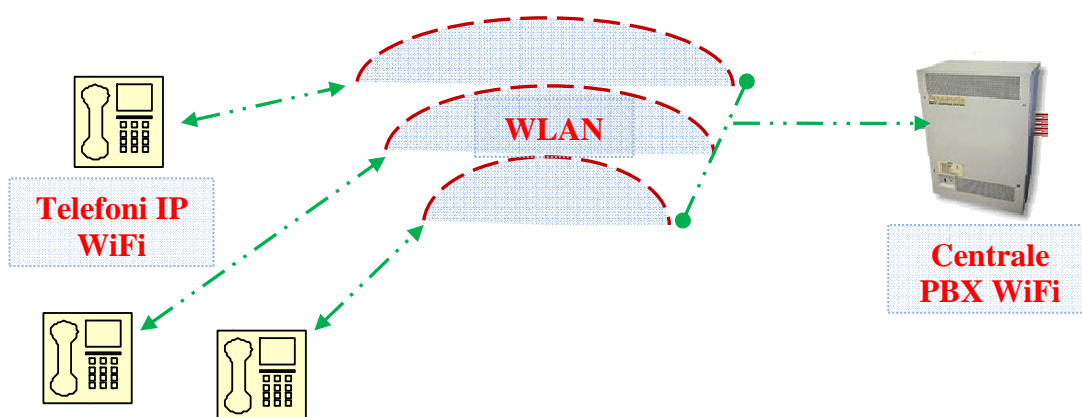


Figura 20 – Scenario VoIP su WiFi

i telefoni WiFi devono necessariamente comunicare via WLAN con la centrale PBX, che potrebbe essere anche Asterisk Based. Sicuramente gli algoritmi di *Echo cancellation* ivi integrati cercherebbero di rimuovere l’eco acustico generato ma, con molta probabilità, l’alta varianza della latenza IP li renderebbe inefficaci.

Ma perché si verificano errori nelle reti WiFi ?

Sostanzialmente perché le reti wireless sono caratterizzate da una serie di problematiche non presenti nei sistemi cablati:

- Scelta di una banda radio disponibile a livello mondiale
- Problema della stazione nascosta/esposta
- Le trasmissioni radio sono soggette ad un’elevata rumorosità
- I segnali radio ad alta frequenza sono soggetti alla riflessione e quindi vengono ricevuti più volte con sfasamenti temporali dipendenti dalla lunghezza del percorso compiuto. Questo tipo d’interferenza viene chiamata *Multipath Fading*.

- Le microonde sono ionizzanti ed è quindi necessario limitarne la potenza
- La mobilità dei computer impone l'uso di tecniche di routing specializzate
- Molte implementazioni di TCP e UDP sono ottimizzate per sole reti affidabili
- I dati trasmessi via radio sono facilmente intercettabili e quindi vanno crittografati
- Vulnerabilità alle interferenze dovute a dispositivi che lavorano alle stesse frequenze (casuali o intenzionali) [16]

Quindi è intuibile immaginare che la varianza del *Round Trip Time* TCP è quindi molto elevata a causa delle ritrasmissioni, al contrario di quello che succede per le normali connessioni TCP in rete fissa.

3.13 – *Related Work*

La stesura di questo elaborato si prefigge lo scopo di trovare una tecnica di miglioramento della comunicazione VoIP come, ovviamente, si può capire dal titolo. Per giungere a quanto voluto sono state analizzate sia le problematiche rilevate che le soluzioni già proposte.

Nell'ambiente didattico e di ricerca, soluzioni per la rimozione dell'eco e dei disturbi nella trasmissione VoIP, sono state analizzate in vari aspetti e sfaccettature prese per singoli casi e campi di applicazione.

Sono stati realizzati diversi algoritmi matematici per la rimozione dell'eco [8] [10] [15], applicati successivamente anche al VoIP.

Come è già stato descritto, il problema del VoIP su WiFi è uno scalino successivo da oltrepassare, in quanto si devono prima risolvere le questioni riguardanti l'acustica e solo successivamente adattare le soluzioni al modello di trasporto che, nel nostro caso, è il WiFi.

Ecco che quindi l'attuale approccio al problema è stato affrontato su temi diversi che possono essere elencati a piani o, meglio, livelli:

- 1) Ideare algoritmi per eliminare l'eco acustico [8] [10], dove la letteratura già cita diverse soluzioni matematiche tra cui LMS, NLMS, APA, RLS.
- 2) Eliminare l'eco acustico nelle comunicazioni telefoniche [11], dove si devono ideare '*Acoustic Echo Cancellor*' che implementino filtri adattivi basati su algoritmi di eliminazione dell'eco.
- 3) Adattare gli algoritmi al VoIP [13], dove soluzioni come il '*Packet Based Echo Cancellor*' meglio si adattano alla trasmissione della voce su IP, in quanto è tenuta in considerazione, ed analizzata, una comunicazione digitale su rete informatica.
- 4) Adattare il VoIP alle comunicazioni WiFi [14], dove ancora si possono elaborare soluzioni efficaci e performanti migliorando quanto è già disponibile per le comunicazioni VoIP.

I lavori maggiormente interessanti si ravvisano con le soluzioni basate sui PBEC, dove sono analizzate le teorie innovative di rimozione dell'eco in ambiente di rete [17] e, soprattutto, l'applicazione degli stessi in reti con forte ritardo [20]. In quest'ultimo caso il ritardo è inteso come distanza di comunicazione tra gli interlocutori: è già stato discusso che la distanza aumenta il ritardo e, maggiore è il ritardo, maggiore è l'eco percepito.

A questo punto è possibile ragionare sulle ipotesi e sulle soluzioni già formulate per poter comprendere il comportamento di un '*Echo canceller*' in condizioni ambientali critiche, come il WiFi, e cercare di adattare un PBEC ad una comunicazione VoIP su reti ad alta varianza del '*Round Trip Time*'.

Capitolo 4 – Simulazione di applicazione di un AEC con alta varianza RTT

4.1 - Simulazione in Matlab

Per poter simulare il comportamento di un algoritmo di cancellazione dell'eco in ambiente con forte varianza RTT è stato predisposto un test con Matlab.

L'algoritmo di *Echo cancellation* utilizzato per la simulazione si avvale del filtro adattivo **FDAF** (*Frequency Domain Adaptive Filter*) che possiede caratteristiche di elevata precisione in presenza di segnali stabili e continui. In figura 21 ne è schematizzato il funzionamento.

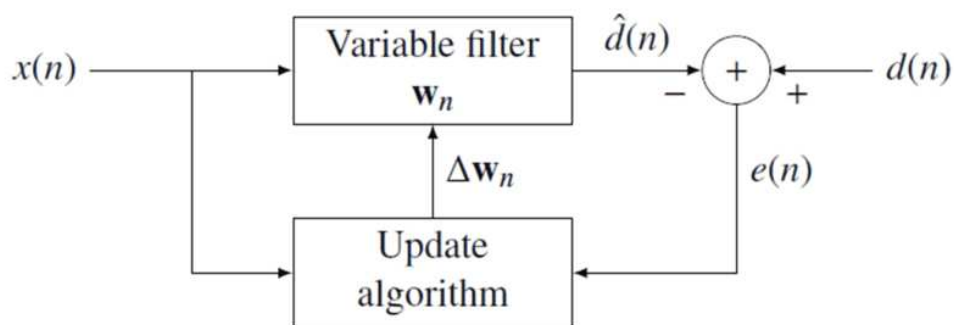


Figura 21 – Filtro FDAF

Il segnale in input $x(n)$ è composta dalla somma del segnale desiderato $d(n)$ e dal segnale da rimuovere $v(n)$; questa variabile è incognita ed è l'eco. $x(n)$ si può ricavare dalla seguente formula (5).

$$x(n) = d(n) + v(n) \quad (5)$$

La funzione del segnale di errore è la differenza fra il segnale voluto e quello valutato dall'algoritmo, come si può ricavare dalla seguente formula (6).

$$\mathbf{e}(\mathbf{n}) = \mathbf{d}(\mathbf{n}) - \hat{\mathbf{d}}(\mathbf{n}) \quad (6)$$

Il filtro variabile (*Variable filter*) possiede una struttura FIR (*Finite Impulse Response*) che consiste in una tipologia di filtro digitale la cui risposta è un impulso sempre più vicino allo zero con un uscita sempre stabile. Con il filtro variabile viene valutato il segnale voluto, in relazione all'impulso in risposta in modo tale da ottenere in uscita dal filtro il segnale $\hat{\mathbf{d}}(\mathbf{n})$, che è la replica elettronica dell'eco da sottrarre da $\mathbf{x}(\mathbf{n})$, in modo tale da ottenere un valore molto vicino a $\mathbf{d}(\mathbf{n})$. L' *Update Algorithm* aggiorna i coefficienti del filtro basandosi sui segnali di errore e sul segnale in input, passando al filtro variabile un fattore di correzione $\Delta\mathbf{w}_n$, calcolato tramite la relazione in formula (7)

$$\Delta\mathbf{w}_n = \mathbf{w}_{n+1} - \mathbf{w}_n \quad (7)$$

dove \mathbf{w}_n è calcolato dalla relazione (8)

$$\mathbf{w}_n = [\mathbf{w}_n(\mathbf{0}), \mathbf{w}_n(\mathbf{1}), \dots, \mathbf{w}_n(\mathbf{p})]^T \quad (8)$$

\mathbf{T} e \mathbf{p} sono rispettivamente il tempo di campionamento e l'ordine del filtro.

Il segnale $\hat{\mathbf{d}}(\mathbf{n})$ è calcolato tramite la relazione (9)

$$\hat{\mathbf{d}}(\mathbf{n}) = \mathbf{w}_n^T \mathbf{x}(\mathbf{n}) \quad (9)$$

dove il calcolo di $\mathbf{x}(\mathbf{n})$ si può ricavare dalla seguente formula (10)

$$\mathbf{x}(\mathbf{n}) = [\mathbf{x}(\mathbf{n}), \mathbf{x}(\mathbf{n} - \mathbf{1}), \dots, \mathbf{x}(\mathbf{n} - \mathbf{p})]^T \quad (10)$$

A differenza degli algoritmi LMS e NLMS, che possiedono scarse prestazioni in presenza di rumore, situazione ricorrente in presenza di alta varianza RTT, il filtro FDAF invece gode di alta efficacia. La scelta di questo filtro, per la simulazione in Matlab, è quindi motivata per questa caratteristica.

Le simulazioni prevedono due test: uno in ambiente ottimale ed uno in ambiente con forte varianza RTT. In ingresso all'algoritmo di simulazione sono passati due segnali audio reperibili dalla libreria di Matlab (*nearspeech* e *farspeech*) che mostrano un dialogo tra due interlocutori. La simulazione viene eseguita dal punto di vista dell'interlocutore *near*.

Tramite il filtro **DF2T** (*Discrete-time, Direct-form II Transposed Filter*) è stato creato l'eco acustico generato da un ipotetico riflesso del segnale *farspeech*: questo eco verrà sommato al *nearspeech*, che chiameremo *dhat*, così da simulare il segnale in ingresso al microfono da trattare [21]. Lo scopo finale è che l'algoritmo sia in grado di rimuovere da *dhat* l'eco creato.

4.2 - Applicazione del filtro per la rimozione dell'eco in condizioni di ambiente ottimali.

Nella seguente figura 22 è riportato il dialogo ottimale senza la presenza di eco. In verde il segnale *farspeech*, in blu il segnale *nearspeech*. Questo significherebbe che in una situazione senza eco, al microfono lato *near* il segnale in ingresso dovrebbe essere come quello in figura 23.

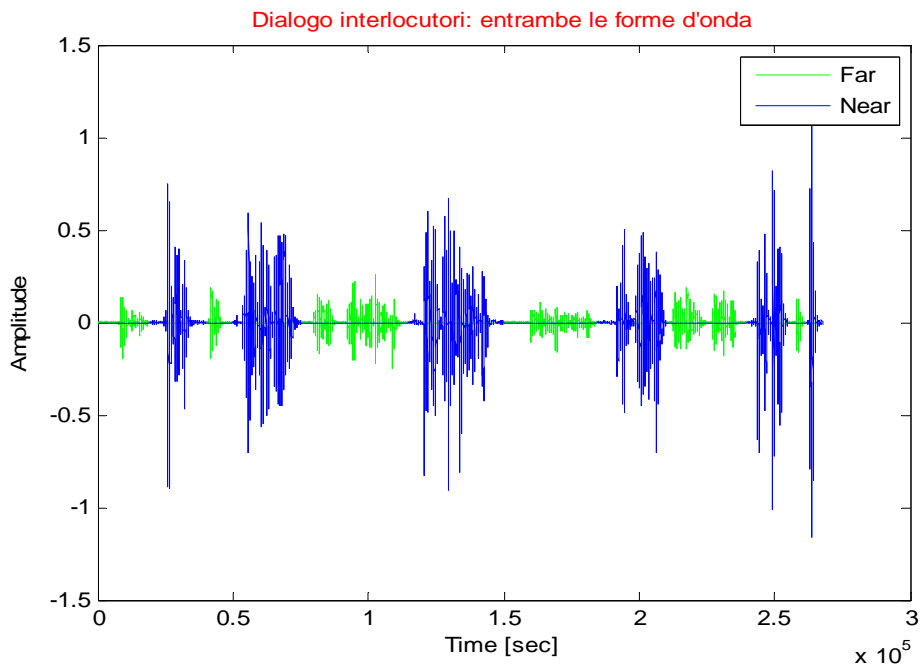


Figura 22 – Dialogo interlocutori

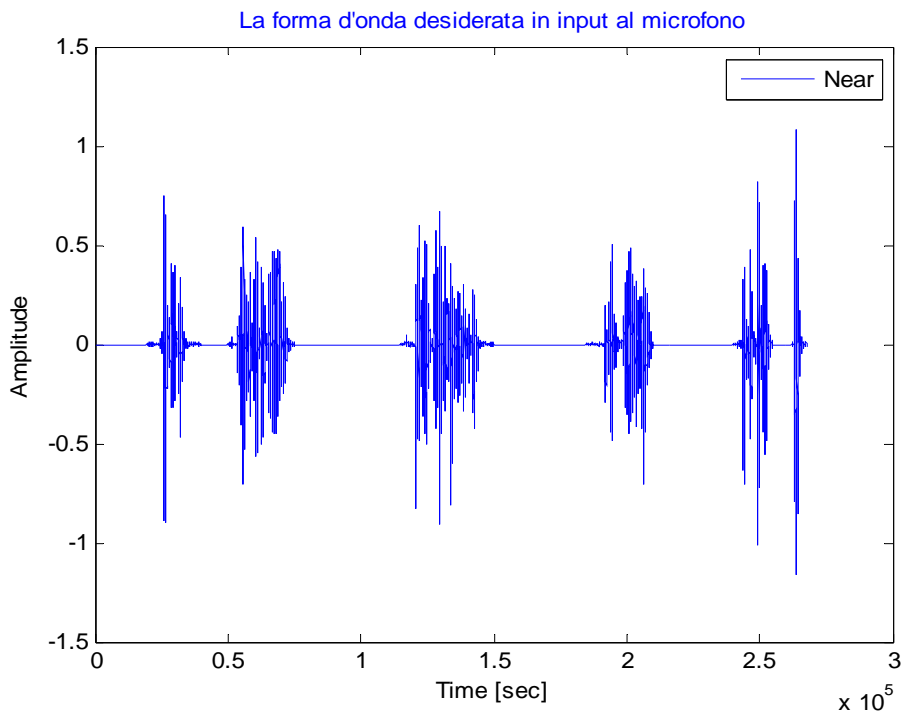


Figura 23 – Situazione ottimale

Invece in ingresso al microfono lato *near* si avrà il segnale come in figura 24, dovuto all'eco acustico del segnale *farspeech* generato dall'ambiente.

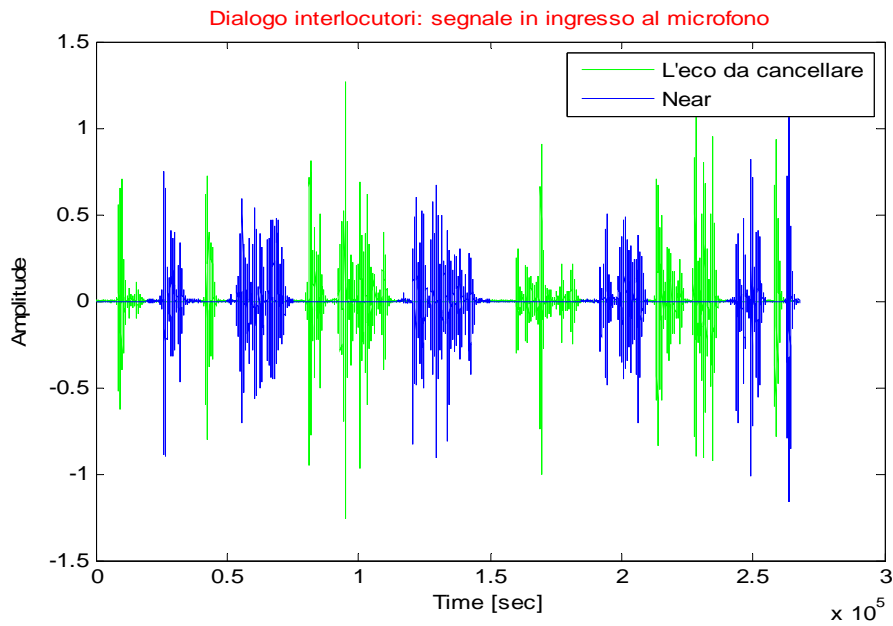


Figura 24 – Segnale in ingresso con eco

Dopo l'applicazione dell'algoritmo di *Echo cancellation* si nota, come riportato in figura 25, come il segnale di eco viene riconosciuto e ridotto fino alla sua quasi totale cancellazione (in verde).

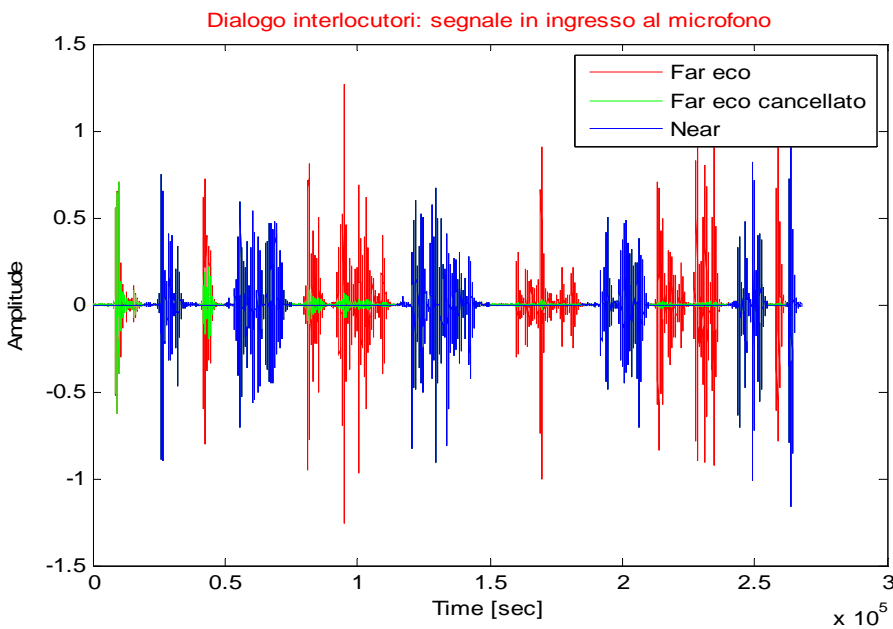


Figura 25 – Schema con eco cancellato

Il comportamento del filtro adattivo riesce ad eliminare l'eco con precisione crescente durante il dialogo, fino alla quasi totale cancellazione. In appendice al presente documento è riportato il listato del programma Matlab, che, copiato ed eseguito, permette anche di apprezzare con l'ascolto auditivo la qualità dei segnali.

4.3 - Applicazione del filtro per la rimozione dell'eco in condizioni di ambiente ad alta varianza del *Round Trip Time*.

Per simulare in Matlab il comportamento del segnale con un alta varianza RTT si è reso necessario riprodurre una trasmissione dati a pacchetti. Come precedentemente illustrato una tecnica di 'dissimulazione' della perdita dei pacchetti è l'utilizzo del *Jitter Buffer*, che consiste nel ripetere l'ultimo pacchetto arrivato in sostituzione del primo perso e inserire pacchetti vuoti per i successivi, finché non riprende il normale flusso audio. Per riprodurre il più fedelmente possibile una simulazione di trasmissione di dati a pacchetto sono stati utilizzati i seguenti dati in ingresso:

- Supposizione di utilizzo del *Codec G.729* 8Kbps con un payload di 20 bytes
- *Bitrate* = 8192 bps (8 Kbps)
- *Payload* 20 Bytes = 160 bit
- Supposizione che il tempo di elaborazione del pacchetto audio dal *Codec* ≈ 0 ms
- tempo massimo di attesa del prossimo pacchetto = $160 \text{ bit} / 8192 \text{ bps} = 0.0195$ secondi = 19,5 millisecondi
- differenza massima di RTT = $19.5 * 2 = 39$ ms (perché RTT misura andata e ritorno)
- se $RTT > 39$ allora il *Jitter Buffer* duplica il primo pacchetto e scarnerà i successivi finché la differenza RTT non ritorna inferiore o uguale a 39

L'algoritmo di simulazione dell'alta varianza RTT è stato realizzato come nel diagramma a blocchi in figura 26.

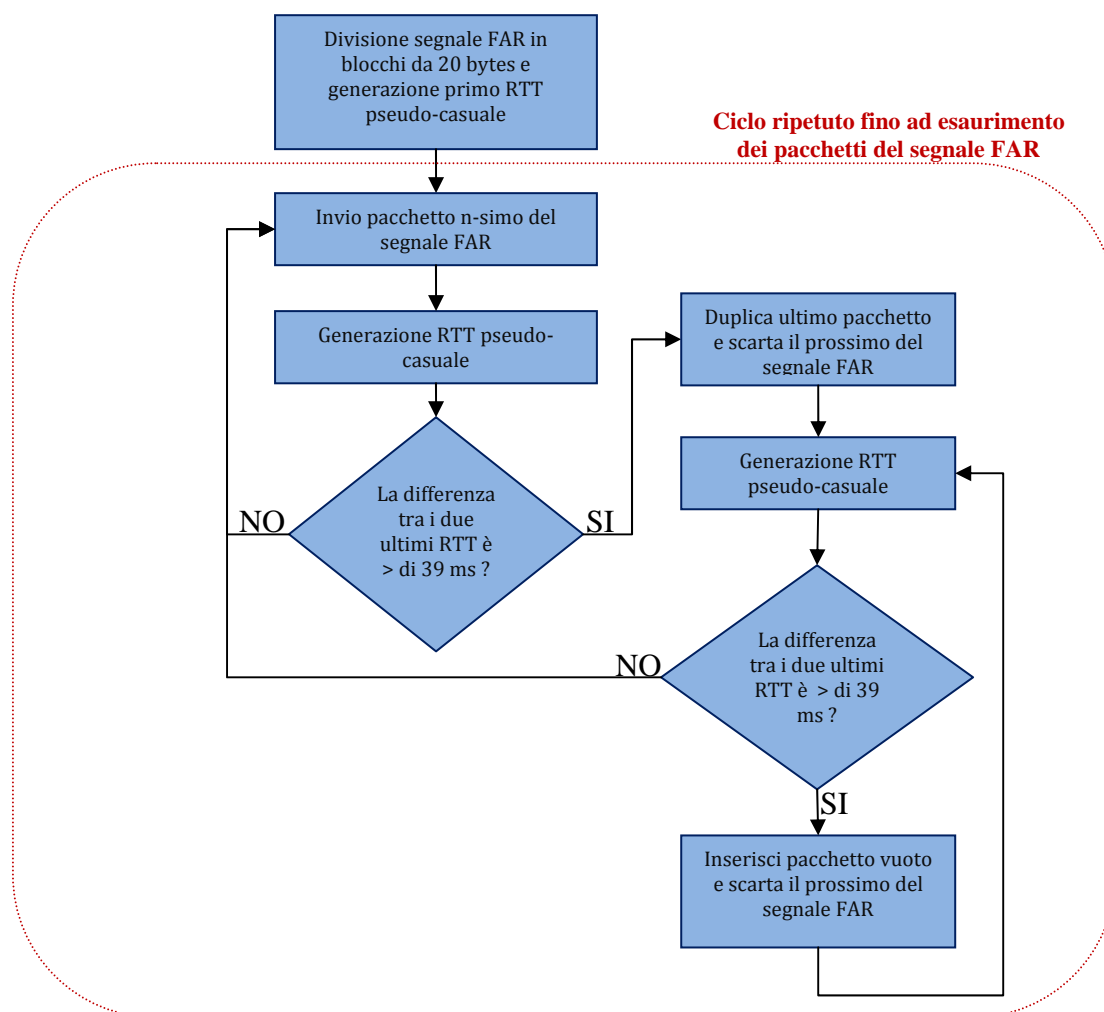


Figura 26 – Diagramma a blocchi algoritmo simulazione RTT variabile

Immaginando uno scenario di comunicazione VoIP su WiFi, dove gli interlocutori sono in movimento e quindi il segnale è soggetto a molti disturbi dovuti alla rumorosità ed al *Multipath Fading*, si è considerata un'elevata varianza RTT. Prendendo sempre in esame il segnale in ingresso al microfono lato *near*, il segnale *farspeech* è stato diviso in pacchetti da 20 bytes; tramite un algoritmo iterativo è stata simulata la trasmissione di questo segnale e l'utilizzo del *Jitter Buffer* con valori casuali di RTT compresi tra 0 e 100. Ha poca importanza che il valore iniziale sia 0, valore praticamente impossibile, dato che si misura la varianza e quindi la differenza tra due RTT consecutivi. Si sarebbe potuto scegliere una gamma tra 10 e 110 ms e non sarebbe cambiato alcunché in quanto un insieme di 100 valori di RTT implica sempre una percentuale del 18% di possibilità che il

pacchetto vada perso durante la trasmissione: se la differenza tra i due ultimi RTT è maggiore di 39 ms, è generato forzatamente un ‘buco’ nella trasmissione a causa del superamento del valore del massimo tempo di attesa imposto dal *Codec* audio. La percentuale probabilistica è calcolata dalla seguente formula (11).

$$Prob = \frac{\sum_{RTT=minRTT}^{maxRTT-39} RTT}{maxRTT \times maxRTT} \quad (11)$$

Dove ***maxRTT*** è il valore massimo di RTT e ***minRTT*** il valore minimo che, in questo caso, corrispondono rispettivamente a 100 e 0.

Come si può notare dalla figura 27, anche in presenza di un’alta varianza RTT, ovviamente, otteniamo dell’eco acustico. Ma, come evidenziato in figura 28, la varianza RTT non influisce sulla qualità dell’algoritmo di cancellazione dell’eco.

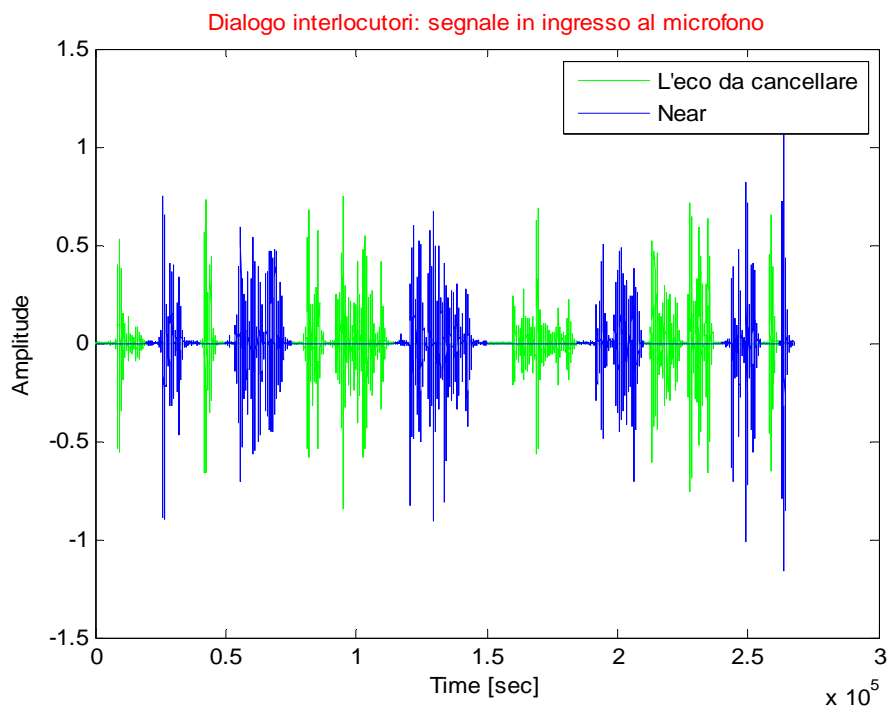


Figura 27 – Segnale con eco in ingresso e RTT variabile

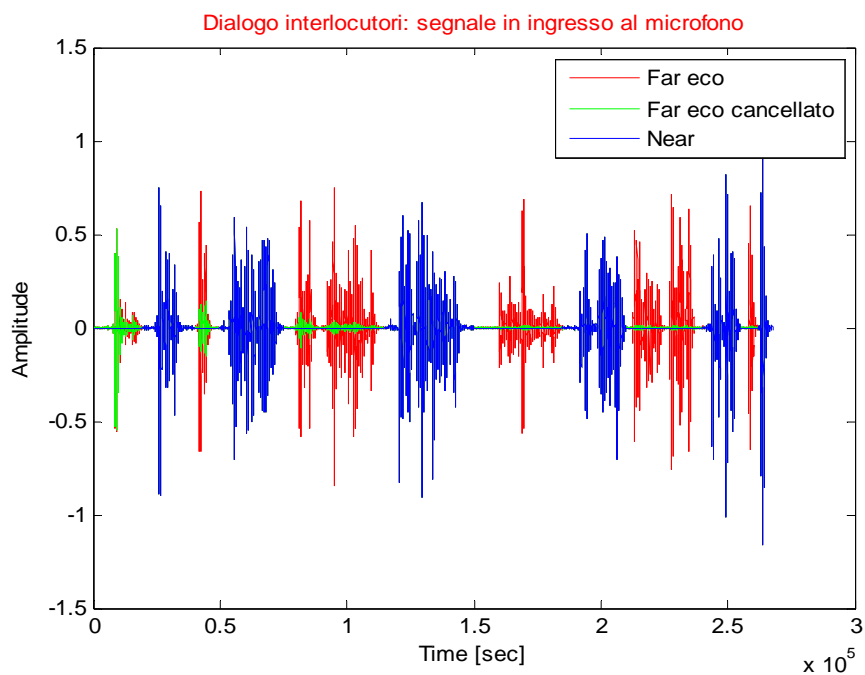


Figura 28 – Segnale in ingresso con eco cancellato e RTT variabile

La differenza si nota in fase di ascolto del segnale: la qualità è scarsa, la voce risulta metallica e poco comprensibile. Risultati con test effettuati incrementando la varianza RTT a valori sempre più elevati, rivelano come la degradazione della qualità del segnale aumenti ma l'efficacia dell' algoritmo di cancellazione dell'eco rimanga invariata.

Infatti la differenza di cancellazione di eco tra le due simulazioni è praticamente impercettibile, come si può verificare dalla seguente figura 29.

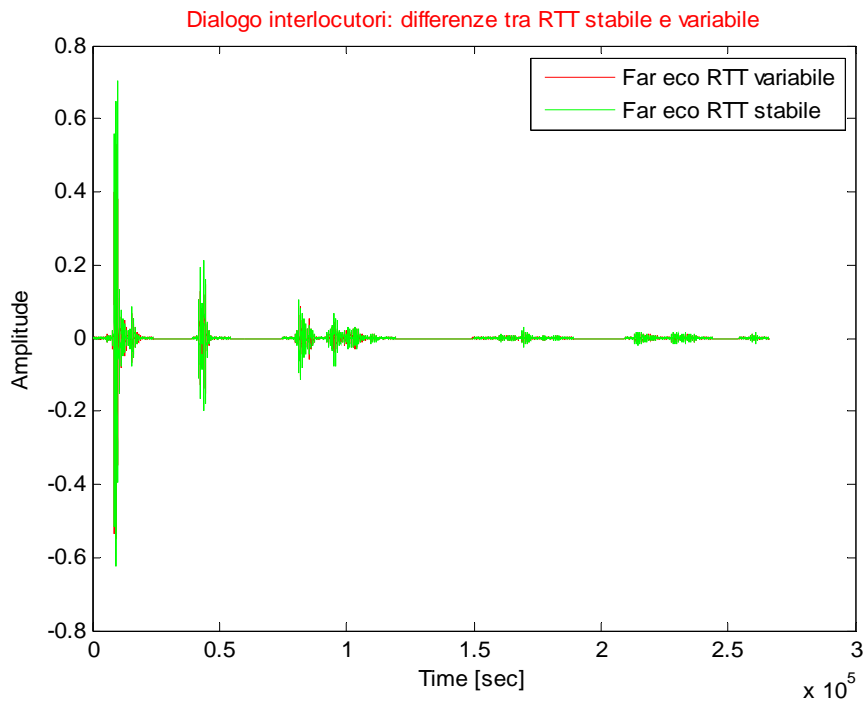


Figura 29 – Comparazione eco eliminato con e senza RTT variabile

Una tecnica emergente per eliminare l'eco in ambito VoIP adatta gli algoritmi standard alla trasmissione a pacchetti. Per poter fare ciò si realizzano i **PBEC** (*Packet Base Echo Canceller*) che sostanzialmente sono degli algoritmo EC che analizzano i singoli pacchetti e non il flusso di dati.

4.4 - *Packet Based Echo Canceller* (PBEC)

Il *Packet Based Echo Canceller* è un algoritmo di *Echo cancellation* sviluppato appositamente per il VoIP. Nella figura 30 è riportato un semplice schema di funzionamento.

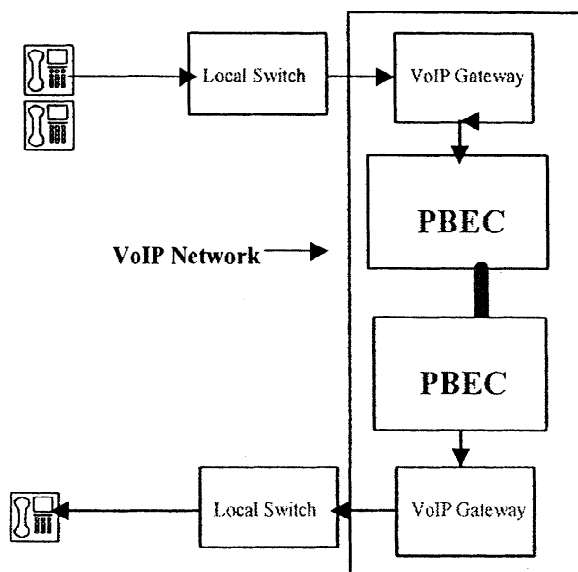


Figura 30 – Schema PBEC

Come si può intuitivamente apprendere il PBEC deve trovarsi all'interno del gateway VoIP, che potrebbe essere un PBX hardware o software (tipo Asterisk). Quando un pacchetto VoIP è elaborato dal PBEC, il suono dell'interlocutore remoto è campionato e ne viene estratto un modello di *'echo replica'*. Questo modello sarà utilizzato per sottrarre l'eco. Il principio di funzionamento è il seguente: quando il pacchetto VoIP remoto arriva al VoIP gateway viene separato dall'*Header* ed il *Payload*, che contiene il segnale audio, viene elaborato dall'*Echo Canceller*. Quando l'operazione di rimozione dell'eco è completata l'*Header* riviene collegato ed il pacchetto è inviato al successivo VoIP gateway. Questo procedimento è soggetto ad una perdita di pacchetti, ma comunque è ottimizzato mediante l'inserimento di valori casuali che non dovrebbero essere percepiti dagli interlocutori. [17]

4.5 - Simulazione in Matlab di un PBEC

Una successiva simulazione in Matlab è la realizzazione di un rudimentale PBEC allo scopo di eliminare il segnale d'eco da singoli gruppi di pacchetti IP. La simulazione è stata realizzata modificando l'algoritmo creato precedentemente per la trasmissione a pacchetti

in presenza di alta variabilità RTT. Avendo già verificato che un'alta varianza RTT non influisce sulla efficacia nella cancellazione dell'eco, con la presente simulazione si verifica il comportamento dell'algorithm EC sui singoli pacchetti. Sono state eseguite 5 simulazioni, ognuna utilizzando un input con gruppi sempre più numerosi di pacchetti, cominciando con singoli pacchetti di 20 byte fino ad un gruppo di 10.000 pacchetti, con una dimensione totale di 200.000 byte. Il funzionamento dell'algorithm FDAF richiede che il parametro *Blocklength*, cioè la dimensione in byte del blocco di segnale elaborato, sia un multiplo della dimensione totale in byte del gruppo di pacchetti: dai test effettuati, per ottenere la maggiore efficacia dell'algorithm, il parametro deve possedere un valore di 1/10 della dimensione del gruppo di pacchetti considerato, ma comunque mai maggiore di 2.048 .

Test con blocchi da 20 byte

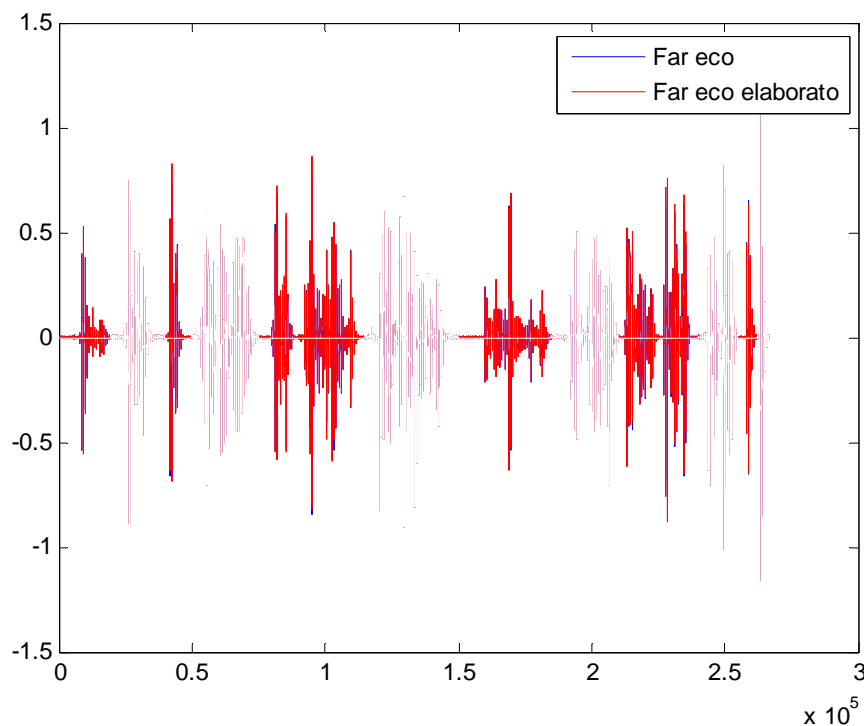


Figura 31 – Eco con blocchi da 20 byte

Il blocco da 20 byte combacia con la dimensione del pacchetto. Conseguentemente la simulazione è stata quindi realizzata su gruppi contenenti un singolo pacchetto. Il test

rivela la totale inefficacia nella rimozione dell'eco. L'eco elaborato in figura 31 (in rosso) è identico a quello da rimuovere e, nel grafico, vi si sovrappone completamente.

Test con blocco da 200 byte

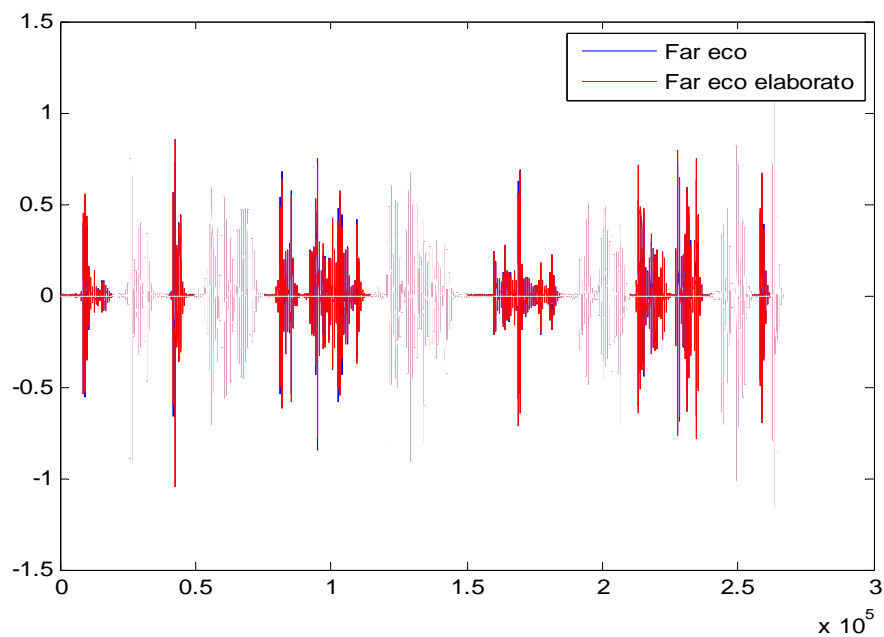


Figura 32 – Eco con blocchi da 200 byte

In questo test si nota un quasi impercettibile miglioramento: nel grafico in figura 32 l'eco elaborato si sovrappone ancora quasi completamente all'eco originale, lasciando però intravedere una minima eliminazione del segnale (punti blu sulle creste del segnale).

Test con blocco da 2.000 byte

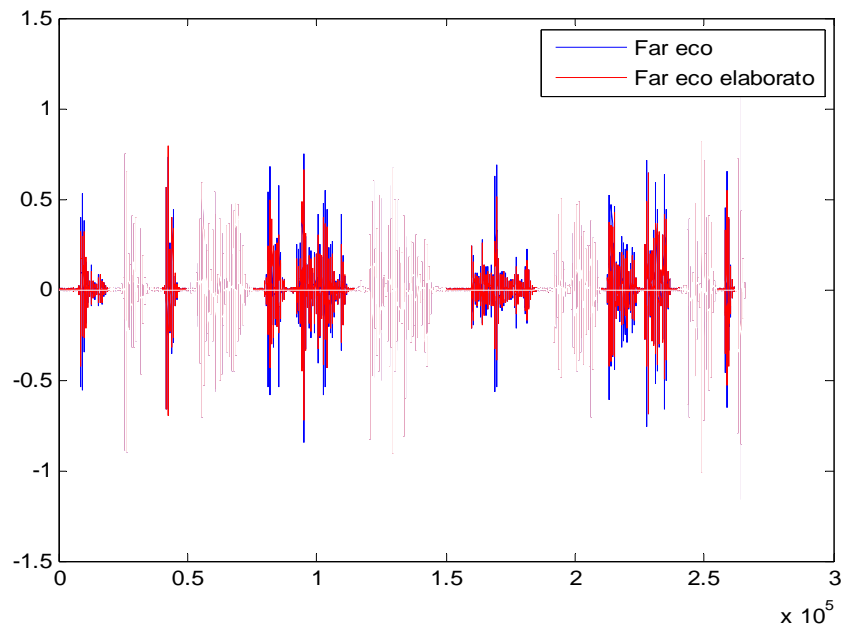


Figura 33 – Eco con blocchi da 2.000 byte

In figura 33 si può notare un sensibile miglioramento nella rimozione dell'eco.

Test con blocco da 20.000 byte

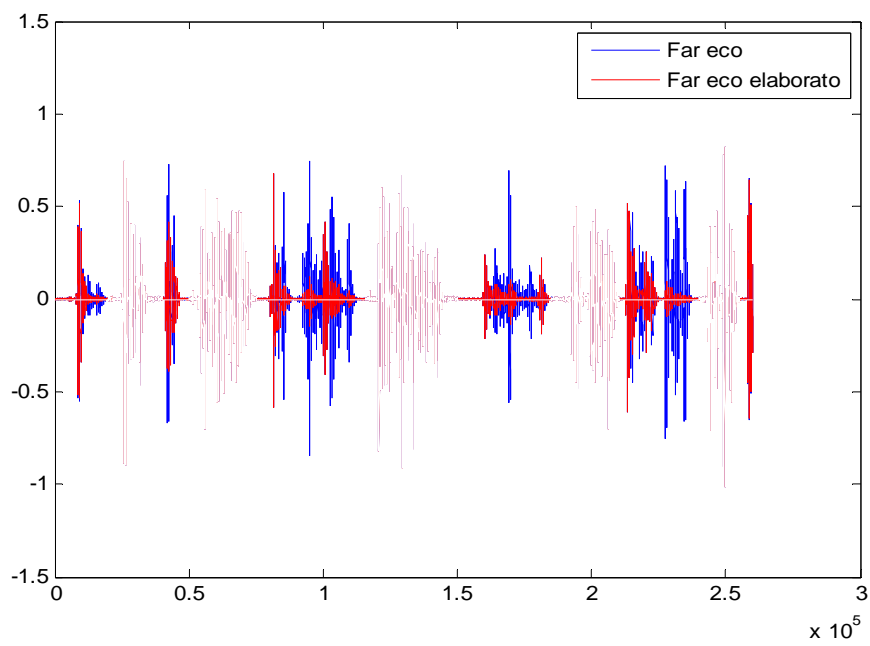


Figura 34 – Eco con blocchi da 20.000 byte

In figura 34 si nota che ad ogni incremento nella dimensione del blocco elaborato si ottengono miglioramenti nell'efficacia dell'algoritmo.

Test con blocco da 200.000 bytes

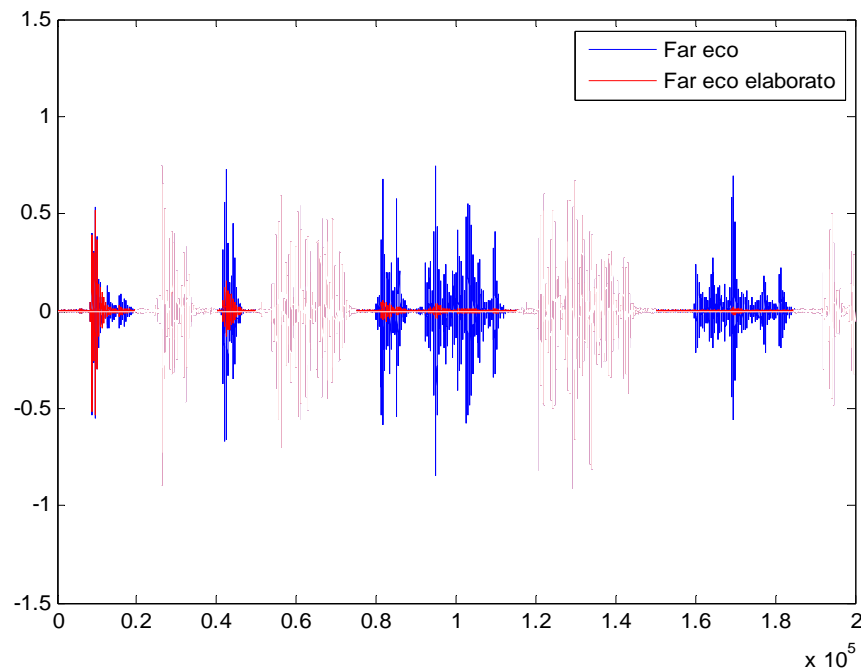


Figura 35 – Eco con blocchi da 200.000 byte

Infine, considerando un blocco che comprende la quasi totalità del segnale di eco da elaborare, si ritorna, come si può intuitivamente immaginare, ad una simulazione simile a quella descritta al capitolo '4.2-Applicazione del filtro per la rimozione dell'eco in condizioni di ambiente ottimali'

In definitiva, a seguito della simulazione, si può evincere che l'efficacia di un PBEC dipende comunque dalla dimensione del blocco interessato all'elaborazione. In blocchi di piccole dimensioni un algoritmo di *Echo cancellation* non riesce a riconoscere l'eco nel suono elaborato e, di conseguenza, non si è in grado di eliminarlo efficacemente. Questo è facilmente intuibile: uno dei requisiti per percepire l'eco è che sia in ritardo: ovviamente un filtro adattivo non può riconoscere l'eco in presenza di pochi o singoli pacchetti, ma deve considerarne diversi per ottimizzare l'elaborazione.

Capitolo 5 – Conclusioni

5.1 - Come migliorare la comunicazione

Dalle simulazioni del capitolo precedente si può dedurre la seguente conclusione: i problemi di comunicazione VoIP in presenza di un'alta varianza RTT convergono in un decadimento della qualità del segnale audio rendendolo incomprensibile agli interlocutori, ma non influiscono sull'efficacia degli algoritmi di *Echo cancellation*. Si è anche visto che per migliorare l'efficacia di un algoritmo PBEC è sufficiente aumentare la dimensione del blocco del segnale audio elaborato.

Ma come si può migliorare la comunicazione VoIP sfruttando queste due informazioni ?

In presenza di alta varianza RTT la comunicazione risulta frammentata a causa della prolungata attesa nella successione dei pacchetti: lo streaming audio deve essere continuativo, anche se vi è una perdita di informazioni. Una soluzione al problema può essere l'adozione di una gestione avanzata del *Jitter Buffer*: un *Buffer* di adeguate dimensioni può raccogliere ed ordinare i pacchetti audio in arrivo e rilasciarli solo quando ne è raggiunta una quantità accettabile.

La realizzazione di una soluzione richiede l'introduzione di un ritardo (Delay) nella comunicazione il cui valore sarà di maggior dimensione proporzionalmente alla capienza del *Buffer*. Quindi un algoritmo di '*Denoise*' potrebbe campionare una media della varianza RTT e regolare la dimensione del *Jitter Buffer* appropriatamente.

Ecco dove l'applicazione di un algoritmo di PBEC potrebbe rivelarsi utile nella rimozione dell'eco in questa soluzione: il *Jitter Buffer* accumula, per le sue caratteristiche, pacchetti audio che, come abbiamo visto nella simulazione, maggiore è il loro numero migliore è l'efficacia dell'algoritmo PBEC.

Nel diagramma a blocchi in figura 36 è riassunto quanto appena esposto:

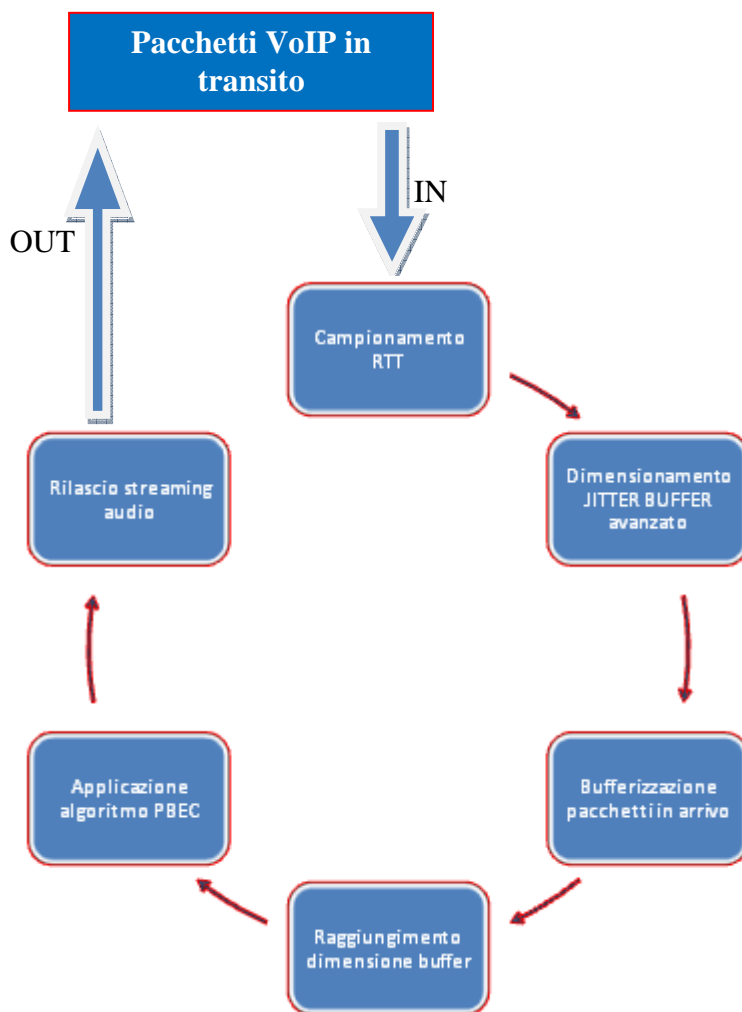


Figura 36 – Diagramma a blocchi di un AEC ottimale con condizioni di RTT variabile

Un possibile svantaggio che si otterrebbe con questa soluzione riguarda la percezione da parte degli interlocutori di un ritardo (delay) nella conversazione, ritardo direttamente proporzionale alle condizioni critiche di trasmissione dati, cioè alla varianza RTT, ma che comunque assicurerebbe una continuità nel segnale sonoro senza frammentazione che, frammentato, convoglierebbe in una difficoltosa comprensione del discorso.

5.2 - Considerazioni finali

Dalle simulazioni in Matlab sono stati restituiti degli ottimi spunti per la realizzazione di algoritmi di miglioramento della qualità nella comunicazione VoIP.

Le idee fornite con questo elaborato possono essere ampliate con la realizzazione di un algoritmo di PBEC avanzato, algoritmo che per una concretizzazione richiede uno studio specifico sul funzionamento degli algoritmi di un '*Echo Cancellor*', studio non affrontato in questa tesi.

In definitiva si è cercato di dimostrare che per utilizzare la tecnologia VoIP in condizioni critiche, come la trasmissione WiFi, è richiesto un adattamento degli attuali algoritmi di '*noise cancellation*' e '*Echo cancellation*' già esistenti, senza doverne forzatamente elaborare di nuovi o investire nella ricerca di tecnologie specifiche.

Appendice 1. Programma Matlab

```
%% Il dialogo perfetto !! Simulazione senza ECO
fprintf '\nIl dialogo perfetto: non esistono ne eco ne disturbi';
%pulizia totale
clear all;
% Carica il segnale 'vicino' nearspeech
load nearspeech;
% Mette nell'array near il contenuto di nearspeech
near = v;
% Carica il segnale 'lontano' farspeech
load farspeech;
% Copia nell'array far il contenuto di farspeech
far=x;
% un po' di pulizia
clear v;
clear x;
% creiamo il dialogo
talk=far+near;
% visualizziamo il grafico del dialogo
plot (far,'g');
hold on;
plot (near,'b');
xlabel('Time [sec]');
ylabel('Amplitude');
title('Dialogo interlocutori: entrambe le forme d'onda',...
      'Color',[1 0 0]);
legend ('Far','Near');
hold off;
fprintf '\npremi un tasto per continuare';
pause;
fprintf '\nok, continuo';
plot (near,'b');
xlabel('Time [sec]');
ylabel('Amplitude');
title('La forma d'onda desiderata in input al microfono',...
      'Color',[0 0 1]);
legend ('Near');
fprintf '\npremi un tasto per continuare';
pause;
fprintf '\nok, continuo';
%% Simulazione con ECO

% Al microfono, lato near, causa l'eco della stanza al microfono arriva
% questo segnale come ritorno d'eco
fprintf '\nCreazione del segnale di eco rientrante nel microfono';
M = 4001;
fs = 8000;
[B,A] = cheby2(4,20,[0.1 0.7]);
Hd = dfilt.df2t([zeros(1,6) B],A);
hFVT = fvtool(Hd); % Analyze the filter
set(hFVT, 'Color', [1 1 1])
```

```

H = filter(Hd,log(0.99*rand(1,M)+0.01).* ...
    sign(randn(1,M)).*exp(-0.002*(1:M)));
H = H/norm(H)*4;    % Room Impulse Response
n = 1:length(near);
t = n/fs;
x = far(1:length(far));
% dhat è il segnale elaborato come eco rientrante al microfono
% dell'interlocutore far; x è il segnale far, Notare le differenze
% ascoltando i due suoni
dhat = filter(H,1,x);
%p8 = audioplayer(dhat,fs);
%playblocking(p8);
fprintf ('\npremi un tasto per continuare');
pause;
fprintf ('\nok, continuo');
%p8 = audioplayer(x,fs);
%playblocking(p8);

%% The Microphone Signal
%
% The signal at the microphone contains both the near-end speech
% and the far-end speech that has been echoed throughout the room.
% The goal of the acoustic echo canceler is to cancel out the
% far-end speech, such that only the near-end speech is transmitted
% back to the far-end listener.
% d contiene entrambi i segnale near e dhat che sono in ingresso al
% microfono
d = dhat + near;
plot (dhat,'g');
hold on;
plot (near,'b');
xlabel('Time [sec]');
ylabel('Amplitude');
title('Dialogo interlocutori: segnale in ingresso al microfono',...
    'Color',[1 0 0]);
legend ('L'eco da cancellare','Near');
hold off;
fprintf ('\npremi un tasto per continuare');
pause;
fprintf ('\nok, continuo');
p8 = audioplayer(d,fs);
%playblocking(p8);

%% The Frequency-Domain Adaptive Filter (FDAF)
%
% The algorithm that we will use in this demonstration is the
% Frequency-Domain Adaptive Filter (FDAF). This algorithm is very
% useful
% when the impulse response of the system to be identified is long. The
% FDAF uses a fast convolution technique to compute the output signal and
% filter updates. This computation executes quickly in MATLAB. It also
% has
% improved convergence performance through frequency-bin step size
% normalization. We'll pick some initial parameters for the filter and
% see
% how well the far-end speech is cancelled in the error signal.

```

```
fprintf '\nCreazione ed applicazione del filtro adattivo sul segnale in
ingresso al microfono (near+eco far)';
mu = 0.025;
W0 = zeros(1,2048);
del = 0.01;
lam = 0.98;
echodiag = d;
x = x(1:length(W0)*floor(length(x)/length(W0))); %x contiene il segnale
far pieno
d = d(1:length(W0)*floor(length(d)/length(W0))); %d contiene il dialogo
con l'eco far

% Construct the Frequency-Domain Adaptive Filter
hFDAF = adaptfilt.fdaf(2048,mu,1,del,lam);
%hFDAF = adaptfilt.fdaf(2048,mu,1,del,0.98,2048,0.0001);
[y,e] = filter(hFDAF,x,d);
plot (d,'r');
hold on;
plot (e,'g');
plot (near,'b');
xlabel('Time [sec]');
ylabel('Amplitude');
title('Dialogo interlocutori: segnale in ingresso al microfono',...
      'Color',[1 0 0]);
legend ('Far eco','Far eco cancellato','Near');
hold off;
%y contiene il solo eco rilevato e quindi da cancellare
%e contiene il dialogo con l'eco cancellato
%p8 = audioplayer(e,fs);
%playblocking(p8);
%pause

%% E adesso vediamo cosa succede con la varianza RTT

% Supponiamo un payload di 20 bytes (Codec G.729 8Kbps)
% Bitrate = 8 Kbps = 8192 bps
% 20 Bytes = 160 bits
% tempo max di attesa del prossimo pacchetto (senza considerarne
% l'elaborazione) = 160 / 8192 = 0.0195 secondi = 19,5 millisecondi
% quindi possiamo tollerare una differenza di RTT massima di 19.5 * 2
% (perchè RTT misura andata e ritorno) = 39 millisecondi
% se la differenza positiva di RTT > 39 allora Jitter duplica il primo
pacchetto e poi arriverà
% e, basandosi sul valore di RTT, scarta i pacchetti (inseriti a valore
0)
% una differenza negativa di RTT sarebbe un miglioramento della velocità
% del link e quindi non sono persi pacchetti

% creiamo un nuovo far con RTT random
fprintf '\nIl segnale far arriva disturbato se la varianza di RTT\nsupera
i 39 ms: creazione del segnale';
numBlockFar = length (far) / 20; %dividiamo far per blocchi di 20 bytes
clear newFar;
tempFar = 0; %variabile temporanea con un valore che poi verrà tolto
zero=[0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0;0]; %vettore che contiene 20
bytes a zero per emulare un pacchetto perso
```

```
newFar=far(1:20); %il primo pacchetto arriva sempre senza problemi di
latenza
i=1; %inizializziamo la variabile i per il ciclo

variab=100; %questa variabile contiene i limiti della varianza della
latenza in millisecondi che andrà da 0 a variab

%se inferiore a 39 allora il segnale è sempre perfetto
fprintf ('\nVariabilità latenza RTT da 1 a %d millisecondi',variab);
rtt1=rand*variab; %generiamo un primo valore casuale di RTT tra 0 e 300
millisecondi
while (i < (numBlockFar))
    rtt2=rand*variab; %secondo valore casuale di RTT
    diffrrtt=rtt2-rtt1; %differenza tra le latenze
    if (diffrrtt<=39) %se la differenza di latenza è contenuta nei limiti
        newFar=cat (1,tempFar,far((1+20*i):(20*(i+1))));
        tempFar = newFar;

    else
        persi=round (diffrrtt/39); %calcoliamo quanti pacchetti sono
andati persi a causa della differenza di latenza
        if ((persi-(diffrrtt/39)) < 0); %bisogna comunque arrotondare ad
un numero di pacchetti persi più alto
            persi=persi+1;
        end

        newFar=cat (1,tempFar,far((1+20*(i-1):(20*i))); %il primo
pacchetto è una copia del precedente
        tempFar = newFar;
        if (persi > 1) %se abbiamo perso più di un pacchetto
            for ii = 1 : (persi-1)
                newFar=cat (1,tempFar,zero); %aggiunge un pacchetto vuoto
                tempFar = newFar;
            end
        end
        i=i+persi-1; %incrementiamo la variabile i per il numero dei
pacchetti persi (-1 perchè un incremento è effettuato da MATLAB)
    end
    rtt1=rtt2; %il nuovo RTT (2) viene usato come indice per il
successivo ciclo for
    i=i+1;

end

newFar=cat (1,(tempFar(2:length(tempFar))),zero); %togliamo lo 0 iniziale
e riempiamo il vettore
tempFar=newFar(1:(length(far))); %portiamo il vettore alla lunghezza di
far
newFar=tempFar;

fprintf ('\npremi un tasto per continuare');
pause;
fprintf ('\nok, continuo');

%% Ora ripetiamo il ciclo di test dell'eco
oldfar=far; %copia di far
```

```

far=newFar; %copio le variabili perchè riutilizzo il codice
olddhat=dhat;
oldd=d;
olde=e;
oldy=y;

%% Simulazione con ECO
fprintf '\nSimulazioni con FAR disturbato dalla latenza';
fprintf '\nCreazione del segnale di eco rientrante nel microfono';
% Al microfono, lato near, causa l'eco della stanza al microfono arriva
% questo segnale come ritorno d'eco

M = 4001;
fs = 8000;
[B,A] = cheby2(4,20,[0.1 0.7]);
Hd = dfilt.df2t([zeros(1,6) B],A);
hFVT = fvtool(Hd); % Analyze the filter
set(hFVT, 'Color', [1 1 1])
H = filter(Hd,log(0.99*rand(1,M)+0.01).* ...
    sign(randn(1,M)).*exp(-0.002*(1:M)));
H = H/norm(H)*4; % Room Impulse Response
n = 1:length(near);
t = n/fs;
x = far(1:length(far));
%dhat è il segnale elaborato come eco rientrante al microfono
%dell'interlocutore far; x è il segnale far, Notare le differenze
%ascoltando i due suoni
dhat = filter(H,1,x);
%p8 = audioplayer(dhat,fs);
%playblocking(p8);
fprintf '\npremi un tasto per continuare';
pause;
fprintf '\nok, continuo';
%p8 = audioplayer(x,fs);
%playblocking(p8);

%% The Microphone Signal
%
% The signal at the microphone contains both the near-end speech
% and the far-end speech that has been echoed throughout the room.
% The goal of the acoustic echo canceler is to cancel out the
% far-end speech, such that only the near-end speech is transmitted
% back to the far-end listener.

% d contiene entrambi i segnale near e dhat che sono in ingresso al
% microfono
d = dhat + near;
plot (dhat,'g');
hold on;
plot (near,'b');
xlabel('Time [sec]');
ylabel('Amplitude');
title('Dialogo interlocutori: segnale in ingresso al microfono',...
    'Color',[1 0 0]);
legend ('L'eco da cancellare','Near');
hold off;
fprintf '\npremi un tasto per continuare';

```

```
pause;
fprintf '\nok, continuo';
%p8 = audioplayer(d,fs);
%playblocking(p8);

%% The Frequency-Domain Adaptive Filter (FDAF)
% The algorithm that we will use in this demonstration is the
% Frequency-Domain Adaptive Filter (FDAF). This algorithm is very
useful
% when the impulse response of the system to be identified is long. The
% FDAF uses a fast convolution technique to compute the output signal and
% filter updates. This computation executes quickly in MATLAB. It also
has
% improved convergence performance through frequency-bin step size
% normalization. We'll pick some initial parameters for the filter and
see
% how well the far-end speech is cancelled in the error signal.
fprintf '\nCreazione ed applicazione del filtro adattivo sul segnale in
ingresso al microfono (near+eco far)';
mu = 0.025;
W0 = zeros(1,2048);
del = 0.01;
lam = 0.98;
echodiag = d;
x = x(1:length(W0)*floor(length(x)/length(W0))); %x contiene il segnale
far pieno
d = d(1:length(W0)*floor(length(d)/length(W0))); %d contiene il dialogo
con l'eco far

% Construct the Frequency-Domain Adaptive Filter
%hFDAF = adaptfilt.fdaf(2048,mu,1,del,lam); %in teoria non serve creare
più
%volte il filtro
%hFDAF = adaptfilt.fdaf(2048,mu,1,del,0.98,2048,0.0001);
[y,e] = filter(hFDAF,x,d);
% prova: [y,e] = filter(hFDAF,x,e);
plot (d,'r');
hold on;
plot (e,'g');
plot (near,'b');
xlabel('Time [sec]');
ylabel('Amplitude');
title('Dialogo interlocutori: segnale in ingresso al microfono',...
'Color',[1 0 0]);
legend ('Far eco','Far eco cancellato','Near');
hold off;
%y contiene il solo eco rilevato e quindi da cancellare
%e contiene il dialogo con l'eco cancellato
%p8 = audioplayer(e,fs);
%playblocking(p8);
fprintf '\npremi un tasto per continuare';
pause;
fprintf '\nok, continuo';
%% Verifichiamo le differenze di cancellazione dell'eco con

soloOldE=olde-near(1:(length(olde))); %soloOldE e soloE sono i segnali
%Far elaborati (cioè eliminati)
```



```

soloE=e-near(1:(length(e)));;
plot (soloE,'r');
hold on;
plot (soloOldE,'g');
xlabel('Time [sec]');
ylabel('Amplitude');
title('Dialogo interlocutori: differenze tra RTT stabile e variabile',...
      'Color',[1 0 0]);
legend ('Far eco RTT variabile','Far eco RTT stabile');

fprintf '\npremi un tasto per continuare';
pause;
fprintf '\nok, continuo';
% Proviamo ad applicare l'algoritmo di Echo cancellation ai singoli
% pacchetti
mu = 0.025;
del = 0.001;
lam = 0.98;
fprintf '\nApplichiamo l'algoritmo di Echo cancellation
considerando\nporzioni di segnale e non tutto';
pkt=20000; %questa variabile identifica la lunghezza del blocco da
%considerare per l'algoritmo di cancellazione dell'eco: maggiore è il
%blocco + efficiente è la cancellazione

numBlockX = round (length (x) / pkt); %dividiamo far per blocchi di pkt
bytes
hFDAF = adaptfilt.fdaf(2000,mu,1,del,lam) %creiamo il filtro: per
ottenere la massima efficienza
%il block length deve essere sempre circa 1/10 della dimensione del
pacchetto
% ma comunque mai superiore a 2048
tempe=0;
tempy=0;
fprintf '\n';
for i = 0:(numBlockX-1);
    [newy,newe]
    filter(hFDAF,x((1+pkt*i):(pkt*(i+1))),d((1+pkt*i):(pkt*(i+1))));
    tempe=cat (1,tempe,newe);
    tempy=cat (1,tempy,newy);
    fprintf ('\rTermine dell'elaborazione: %d',(numBlockX - i));
end;

newe=tempe(2:length(tempe));%togliamo lo 0 iniziale
newy=tempy(2:length(tempy));%togliamo lo 0 iniziale
%newe contiene il dialogo con l'eco cancellato
%newy contiene l'eco rilevato da cancellare
plot (d(1:length(newe)),'b');
hold on;
plot (newe(1:length(newe)),'r');

plot (near(1:length(newe)),'w');
legend ('Far eco','Far eco elaborato');
hold off;

```

[21]

Glossario degli acronimi

| | |
|--------------|--|
| AEC | <i>Acoustic Echo Canceller</i> |
| APA | <i>Affine Projection Algorithm</i> |
| CELP | <i>Code Excited Linear Prediction</i> |
| CODEC | <i>Codificator/Decodificator</i> |
| CTM | <i>Circuit Transfer Mode</i> |
| DF2T | <i>Discrete-time, Direct-form II Transposed Filter</i> |
| DSP | <i>Digital Signal Processor</i> |
| EC | <i>Echo cancellation</i> |
| FDAF | <i>Frequency Domain Adaptive Filter</i> |
| FIR | <i>Finite Impulse Response</i> |
| Gbps | <i>Gigabits Per Second</i> |
| iLBC | <i>Internet Low Bitrate Codec</i> |
| IP | <i>Internet Protocol</i> |
| ISDN | <i>Integrated Service Digital Network</i> |
| Kbps | <i>Kilobits Per Second</i> |
| LAN | <i>Local Area Network</i> |
| LEC | <i>Line Echo Cancellers</i> |
| LMS | <i>Least Mean Square</i> |
| Mbps | <i>Megabits Per Second</i> |
| MOS | <i>Mean Opinion Score</i> |
| NLMS | <i>Normalized Least Mean Square</i> |
| P2P | <i>Peer to Peer</i> |
| PBEC | <i>Packet Based Echo Canceller</i> |
| PBX | <i>Private Branch Exchange</i> |
| PC | <i>Personal Computer</i> |
| PSN | <i>Packet Switching Nodes</i> |
| PSQM | <i>Perceptual Speech Quality Measurement</i> |
| PSTN | <i>Public Switched Telephone Network</i> |
| PTM | <i>Packet Transfer Mode</i> |

| | |
|-------------|---|
| QoS | <i>Quality of Service</i> |
| RAS | <i>Registration Admission Status</i> |
| RLS | <i>Recursive Least Square</i> |
| RTCP | <i>Real-time Transport Control Protocol</i> |
| RTP | <i>Real-time Transport Protocol</i> |
| RTT | <i>Round Trip Time</i> |
| SIP | <i>Session Initiation Protocol</i> |
| TCO | <i>Total Cost of Ownership</i> |
| UDP | <i>User Datagram Protocol</i> |
| VAD | <i>Voice Activity Detection</i> |
| VoIP | <i>Voice over Internet Protocol</i> |
| WiFi | <i>Wireless Fidelity</i> |
| WLAN | <i>Wireless LAN</i> |

Bibliografia

- [1] Wikipedia – http://it.wikipedia.org/wiki/Onda_sonora
- [2] Marcello Guidotti -
http://www.supportimusicali.it/tutorials/Articoli/codificapcm/codifica_pcm.asp
- [3] Wikipedia – <http://it.wikipedia.org/wiki/Udito>
- [4] Paolo Prandoni - Audiovisual Communications Laboratory
<http://lcavwww.epfl.ch/~prandoni/dsp/echo/node1.htm>
- [5] Oilproject - <http://www.oilproject.org/>
- [6] Răzvan Beuran – 20/04/2006 - VoIP over Wireless Lan
- [7] Stephen R. Kolek - Steven J. Rak - Paul J. Christensen - Battlefield Communication
Network Modeling
- [8] John Garas - A engineering guide to acoustic Echo Cancellers
- [9] Adaptive Digital Technologies -
http://www.adaptivedigital.com/products/echo_cancel_explained.htm
- [10] John F. Doherty - A Robust Echo Canceler for Acoustic
- [11] Federico Brucciani - Algoritmi per filtri adattivi per la cancellazione dell'eco acustico
in un sistema formato da altoparlanti e microfono all'interno di una stanza -
http://web.tiscali.it/Federico_Brucciani/
- [12] Michele Scarpiniti - Miglioramento della qualità del segnale audio in VoIP -
<http://ispac.ing.uniroma1.it/scarpiniti/>
- [13] Kauffman, Andre Neumann - An Algorithm to Evaluate the Echo Signal and the
Voice Quality in VoIP Networks - <https://drum.umd.edu/dspace/handle/1903/3410>
- [14] Razvan Beuran – Yoichi Shinoda – Shin-ichi Nakagawa - Performance Analysis of
VoIP over WLAN.
- [15] Jan Radecki – Zeljko Zilie – Katarzyna Radecka - Echo cancellation in IP Networks
- [16] Marco Benvegnù - Panoramica sulle reti wireless
- [17] Ganesan Periakarruppan - Hairul Azhar b Abdul Rashid - Packet based Echo
cancellation for Voice over IP protocol

- [18] Douglas Comer – 2002 - Internetworking con TCP/IP
- [19] Elisabetta Bevilaqua – VoIP: Un mercato in crescita (2006) -
http://www.zerounoweb.it/index.php?option=com_content&task=view&id=1483&id_tipoologia=3
- [20] Ganesan Periakarruppan, Andy L.Y.Low, Hairul Azhar b Abdul Rashid - Packet based echo cancellation for voice over internet protocol simulated with variable amount of network delay time
- [21] Scott C. Douglas, “Acoustic Echo Cancellation (AEC)”, MathWorks Inc. 2005

Altre fonti di riferimento

- [Christof Faller - New Approaches for Acoustic Echo Control]
http://lcvwww.epfl.ch/research/topics/acoustic_echo_control.html
- [Speex] <http://www.speex.org>
- [iLBC] <http://www.ilbcfreeware.org/>
- [Wikipedia – www.wikipedia.org]
http://en.wikipedia.org/wiki/Mean_Opinion_Score
http://it.wikipedia.org/wiki/Linear_predictive_coding
http://it.wikipedia.org/wiki/Session_Initiation_Protocol
<http://it.wikipedia.org/wiki/QoS>
<http://it.wikipedia.org/wiki/H.323>
- [Prof. Vincenzo Lombardo- Dispense di elaborazione digitale del suono]
<http://www.di.unito.it/~vincenzo/ElabSuoImm3/>
- [ATS Training Services] <http://www.atsweb.it>
- [Fabrizio Coccetti – Università di Trieste] Strumenti per misurare le prestazioni IP
- [Cisco System]
http://www.cisco.com/en/US/tech/tk652/tk701/technologies_white_paper09186a00800d6b68.shtml
- [Prof. Rossano Gaeta – Dispense informatica generale]
<http://www.di.unito.it/~rossano/DIDATTICA/suism-0304/lezione5-suism.pdf>
- [Scott C. Douglas – Acoustic Echo cancellation (AEC)]

[Berkeley Design Technology Inc – Optimizing DSP Software for the Latest Processors]

<http://www.bdti.com/articles/optimize/sld001.htm>

[Asterisk - the world's leading open source telephony engine] <http://www.asterisk.org>

[Skype] <http://www.skype.com/intl/it>