Alma Mater Studiorum · Università di Bologna

SCUOLA DI SCIENZE

Corso di Laurea in Informatica

Studio e realizzazione di una segreteria telefonica VoIP

Relatore:
Chiar.mo Prof.
Renzo Davoli

Presentata da:
Fabio Trigari

Marzo
2017/2018

Ah smettetela di lamentarvi che
qualche spione ascolti le vostre telefonate.
Non siete così interessanti.
Persino la persona a cui state
telefonando non vi sta ascoltando.
Cit. Daniele Luttazzi

Capitolo 1

Introduzione

Come tutti gli oggetti, anche le segreterie telefoniche col trascorrere del tempo subiscono delle trasformazioni diventando di sempre più facile impiego, innovative ed indispensabili.

Con questo progetto di tesi non si pone solo l'obiettivo di migliorare le segreterie telefoniche, rendendole più funzionali per la famiglia e/o azienda, ma affinare anche lo stile di chiamata. Le segreterie telefoniche dei privati, non hanno subìto molte trasformazioni nel tempo, dal momento che questo strumento ricopre già un ruolo soddisfacente. Il software che ho progettato invoglierà molte persone ad utilizzarlo perchè permetterebbe di aver maggiore comodità nell'uso dello strumento stesso, inoltre sarà possibile implementare in questa segreteria nuovi servizi di Internet of Things.

L'idea di sviluppare questo progetto è nata dall'esigenza di non dover costantemente rispondere al telefono fisso. Rispondere, informarsi di chi ha bisogno il chiamante, e successivamente interpellare il diretto interessato. Con questo software chiamato VoiceMessage, si evita di rispondere; solo il chiamato lo farà e nel momento in cui non risponde, verrà inviato attraverso un BOT un messaggio come promemoria con la registrazione del chiamante sul gruppo Telegram.

Per apprezzare appieno questo software, bisognerebbe avere un numero VoIP. Nel caso in cui non lo si possiede, è possibile utilizzare VoiceMessage

anche sulle reti fisse tradizionali, semplicemente acquistando un apparecchio che metta in collegamento telefono, computer e modem.

La particolarità del VoIP, visto che è un "numero virtuale" è facilmente configurabile con un computer e, installando particolari software che gestiscono questo numero è possibile creare la propria segreteria telefonica.

Indice

Elenco delle figure

Capitolo 2

Contesto

2.1 La rivoluzione del telefono: Voice Over IP

La telefonia, rispetto a 20 anni fa, ha subito una grossa trasformazione, non solo attraverso la diffusione degli smartphone ma anche alla tecnologia di comunicazione, che si appoggia sempre di più sulle comunicazioni costruite sull'Internet Protocol (IP) [1] [2].

Apparentemente non si nota questo cambiamento, ma con la diffusione dei telefoni cellulari, un numero telefonico non stabilisce più un luogo ma collega una singola persona. I numeri fissi determinano una zona, infatti ogni prefisso corrisponde a una provincia italiana e i numeri a seguire corrispondono a una zona delimitata, mentre i numeri dei cellulari collegano un solo individuo. Il numero fisso, grazie anche alla fibra ottica ormai presente in quasi tutte le abitazioni, si è trasformato in un numero Voice Over IP (VoIP).

La particolarità di questa nuova tecnologia, come dice il nome, è la chiamata effettuata tramite Internet oppure qualsiasi altra rete che utilizzi il protocollo IP. L'interesse di questa tecnologia è data dalla riduzione dei costi in quanto, specialmente nell'infrastruttura delle aziende, non ci sarà più il doppio cablaggio (uno per gestire i servizi telefonici e l'altro per gestire la connessione a Internet) ma solo uno. A differenza dei numeri della Public

Switched Telephone Network (PSTN) che è la rete con cui i numeri fissi comunicano, quello VoIP non ha un vero e proprio indirizzo fisico, ma ha un indirizzo "generale", infatti il numero telefonico è associato ad un indirizzo di rete IP, ciò non definisce il luogo esatto dell'abitazione ma definisce il luogo dove è stato richiesto il numero.

I costi per trasferire un numero tradizionale da una zona all'altra ha un costo elevato, per esempio il privato può cambiare numero senza oneri, mentre per un'azienda medio-grande è un costo che deve sostenere. Richiedendo il numero VoIP si ha la possibilità di utilizzarlo senza doverlo modificare, sia fuori dal territorio provinciale, sia fuori il territorio nazionale.

Le reti telefoniche tradizionali forniscono un canale di comunicazione di 64 Kbps. Tuttavia, dato che le chiamate spesso sono unidirezionali, in quanto un utente parla e l'altro ascolta, si presenta un enorme dispendio di dati, che attraverso algoritmi specifici è possibile ridurre. Come ad esempio algoritmi di codifica audio che utilizzano circa 8 Kbps, mantenendo sempre una qualità del servizio accettabile. Esistono altri algoritmi importanti per risparmiare dati, come quelli della soppressione del silenzio, cioè trasmettono i dati solamente quando l'utente parla, il che produce un guadagno del 50%.

Questo servizio VoIP viene utilizzato da molteplici utenti ma senza esserne a conoscenza, ad esempio mentre effettuano una chiamata con Skype. Questo famoso programma di comunicazione utilizza questa tecnologia in maniera diretta, ma il servizio VoIP viene utilizzato anche indirettamente, come per esempio mentre si effettuano chiamate con la rete fissa tradizionale: il segnale, potrebbe arrivare alla centralina e da li convertirsi in dati a pacchetto. Ciò sottolinea la trasparenza di questa nuova tecnologia, in quanto l'utente lo utilizzata senza esserne a conoscenza.

Visto che è fattibile collegare il numero VoIP con i computer, è possibile effettuare, oltre alle chiamate semplici, anche videochiamate, condividere documenti, inviare FAX, invio di SMS e MMS. Inoltre lo si può considerare anche un terminale autonomo, in quanto può automaticamente riagganciare le chiamate indesiderate, cosa non eseguibile con i telefoni tradizionali.

Questa rivoluzione del numero fisso VoIP include sia i nuclei familiari sia i gruppi aziendali:

- relativamente alla famiglia risulta facile trasferirsi in qualsiasi altra città o paese mantenendo sempre lo stesso numero VoIP, così si evita di comunicare il numero VoIP ai conoscenti e ai vari enti, ciò non può avvenire con un numero fisso tradizionale;
- discorso analogo vale per le aziende che trasferendosi da un luogo all'altro, potranno conservare il proprio numero VoIP, mantenendo in questo modo il contatto con clienti, enti e società. Infatti possedere un unico numero identifica l'azienda stessa e facilita la comunicazione. Attraverso l'uso del numero VoIP è possibile chiamare o ricevere telefonate contemporaneamente. Se un'azienda non volesse utilizzare questo sistema telefonico, sarebbe costretta a fornire SIM telefoniche ai singoli dipendenti. Questo darebbe ai clienti un senso di frammentarietà dell'azienda stessa.

Esistono molteplici vantaggi della telefonia VoIP:

1. trasferimento di abitazione privata o aziendale senza dover cambiare il numero;
2. facile da configurare;
3. effettuare più chiamate contemporaneamente;
4. costi di infrastruttura contenuti;
5. possibilità di aggiungere più apparecchi telefonici in maniera semplice e veloce;
6. possibilità di essere contattati anche fuori dall'abitazione o dall'azienda, occorre aver configurato il telefono cellulare o un telefono VoIP e avere una connessione Internet;
7. utilizzare un unico numero per tutta l'azienda;

8. possibilità di gestire le filiali con un unico numero di telefono;

9. utilizzando la connessione Internet, se si dispone di una buona banda, la chiamata risulterebbe più "limpida".

Il servizio VoIP presenta tuttavia degli svantaggi:

1. è una tecnologia costruita sul protocollo IP e dato che utilizza un tipo di trasporto Best Effort, implica che i pacchetti possano arrivare a destinazione con un ritardo non controllabile, l'importante è mantenere questo ritardo al di sotto del secondo;

2. un altro punto da tenere in considerazione è la sicurezza, poichè VoIP è compreso tra i servizi offerti da Internet, occorre conoscere i problemi di sicurezza delle trasmissioni nelle reti IP.

Esistono varie aziende che effettuano contratti con numeri VoIP come OpenVOIP e Messagenet, però Messagenet ha un pacchetto gratuito che include la numerazione delle varie province italiane.

2.2 Come avviene la comunicazione

I telefoni VoIP utilizzano due diversi metodi per comunicare: l'architettura H.323 e il protocollo Session Initation Protocol (SIP). Questi sono protocolli di segnalazione telefonica adoperati per stabilire, modificare e concludere telefonate VoIP.

L'architettura H.323 è formata da un'insieme di raccomandazioni che indicano i protocolli di segnalazione, controllo di chiamata, degli standard audio e video [3]. Questa architettura piuttosto complicata è stata elaborata dalla ITU Telecomunication Standardization Sector (ITU-T).

Nella prima versione sono stati rilasciati tutti i protocolli necessari per il funzionamento di questa architettura:

- H.225 che corrisponde al Registration, Admission and Status(RAS) e alla segnalazione di chiamata; il RAS fornisce le informazioni di registrazione al gatekeeper, mentre la segnalazione tra due telefoni con architettura H.323, viene utilizzata per stabilire una connessione. La funzione di segnalazione è un sottoinsieme degli standard Q.931, che è un protocollo di segnalazione sviluppata per le reti Integrated Services Digital Network (ISDN) (reti che per la comunicazione non utilizzano le onde ma utilizzano il codice binario, permettono una comunicazione molto veloce utilizzando solo 64 Kbps), il quale collega attraverso la chiamata i due diretti interessati;

- H.245 Control Protocol for multimedia Communication, è la segnalazione di verifica per unificare la tipologia di media da adottare durante la chiamata;

- Real Time Protocol (RTP) e Real Time Control Protocol (RTCP), utilizzati per il trasporto dei dati (audio e video).

Dalla seconda versione sono stati rilasciati altri protocolli, come H.235 che corrisponde allo standard per le procedure di sicurezza.

L'architettura H.323 definisce alcuni elementi, alcuni obbligatori e altri opzionali:

- terminali: componenti necessari per poter comunicare, in quanto sono i telefoni IP o computer; devono effettuare comunicazioni real-time almeno di tipo audio;

- Multipoint Control Unit (MCU), è un componente opzionale, gestisce le conferenze multi-punto;

- Gateway, è un componente opzionale, se si comunica solo con telefoni H.323; diventa un componente obbligatorio quando si vuole comunicare con altri telefoni su reti differenti, come telefoni fissi tradizionali;

- Gatekeeper, è un componente opzionale e offre servizi ai terminali, Gateway e MCU. Il servizio più importante che mette a disposizione è la traduzione degli indirizzi, consente a due terminali di mettersi in contatto senza che un terminale conosca l'indirizzo IP dell'altro.

2.2.1 H.323

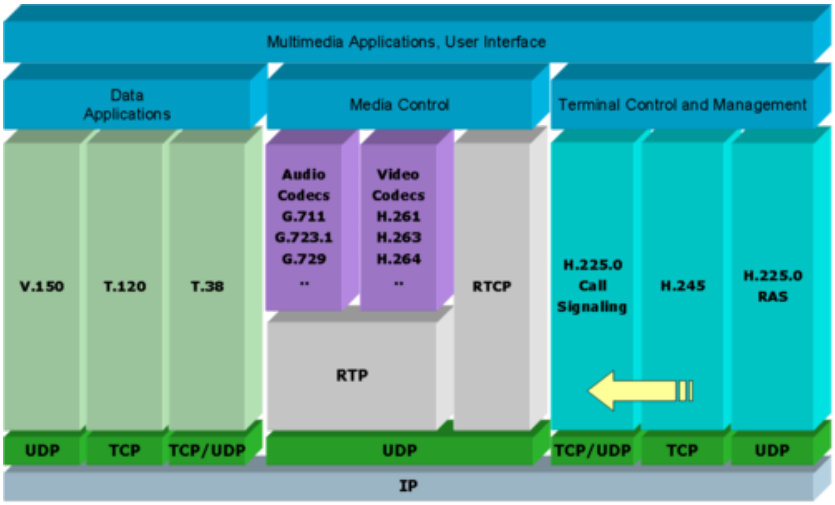

Figura 2.1: architettura di H.323 [4]

2.2.2 SIP

Il Session Initiation Protocol (SIP) è un protocollo di rete definito per creare, modificare e terminare sessioni tra uno o più partecipanti e agisce a livello di applicazione [5]. Questo protocollo è formato da due grandi protocolli utilizzati: Hyper Text Transport Protocol (HTTP) adottato per inviare e ricevere messaggi in formato testuale, la forma client-server e l'utilizzo del URL e URI per la localizzazione e le risorse del servizio; mentre dal Simple Mail Transport Protocol (SMTP) viene utilizzato lo schema di codifica del testo e le informazioni base su come inviare i dati.

Questo protocollo introduce un nuovo sistema, chiamato polite calling il quale permette al chiamante di identificare lo stato del chiamato. Questo per-

mette al chiamante di scegliere il media più opportuno per quel determinato caso.

Nell'architettura SIP esistono due gruppi: i Client e i Server, tra i client fanno parte lo User Agent (UA), il Back-to-Back (B2BUA) ed il Gateway SIP. mentre tra i server c'è il Proxy, il Redirect ed il Registration.

- User Agent (UA), è una terminazione che permette ai partecipanti alla chiamata di interagire. Consente di ricevere e/o inviare i messaggi SIP e gestire la sessione SIP. Sono di due tipi: User Agent Client (UAC) colui che effettua la richiesta e il User Agent Server (UAS) colui che la riceve;

- Back-to-Back User Agent (B2BUA), ha il compito di ricevere, modificare ed inviare le richieste SIP, per stabilire una connessione;

- Gateway, viene utilizzato per comunicare con telefoni che si avvalgono di reti differenti;

- Proxy Server, ha la funzione di intermediario tra lo UAC e UAS. La differenza tra B2BUA e il Proxy Server è data dal fatto che quest'ultimo non utilizza la crittografia end-to-end;

- Redirect Server, si differenzia dal Proxy perchè si limita a fornire solo una risposta per individuare il server successivo a cui inoltrare la richiesta;

- Registration Server, ha la funzione di accettare le richieste REGISTER, registrando indirizzo ed altri parametri dell'utente.

2.2.3 Differenze tra H.323 e SIP

Inizialmente questi due protocolli presentavano molte differenze iniziali, ma si sono modificati progressivamente.

La prima grande differenza è dovuta dagli enti che hanno sviluppato i seguenti protocolli. L'architettura H.323 è stata sviluppata da ITU-T e utilizza un approccio top-down, cioè inizialmente costruisce la struttura generale, dopodiché inizia a descrivere in maniera dettagliata i protocolli. Mentre l'architettura SIP sviluppata da IETF utilizza un approccio uguale a tutti i servizi di Internet. Inoltre ogni protocollo è indipendente dalla specifica applicazione, quindi è possibile riutilizzarlo per altre applicazioni.

Nella prima versione dell'architettura H.323, prima di aprire la chiamata, i terminali inviavano e ricevevano un numero elevato di messaggi andando a peggiorare il Quality of Service (QoS), di conseguenza aveva un Round Trip Time (RTT) pari a 7 RTT. Successivamente utilizzando il protocollo UDP per i messaggi di segnalazione, il RTT è stato ridotto notevolmente fino a 1,5 RTT. Il protocollo SIP fin dalla sua prima versione impiegava già 1,5 RTT per instaurare la chiamata.

I due protocolli sono entrambi molto validi, tuttavia l'architettura H.323 è sul mercato da più tempo rispetto SIP, quindi utilizza una tecnologia più matura. D'altra parte, il protocollo SIP essendo basato su un testo aperto e flessibile, permette di essere più facile per sviluppare nuovi servizi, infatti il protocollo H.323 è più complesso rispetto al SIP. Ciò permette al protocollo SIP ad essere preferito rispetto al protocollo H.323.

Il protocollo SIP ha rimpiazzato lo standard H.323 data la sua somiglianza con il protocollo HTTP.

2.2.4 Collegamento tra VoIP PSTN

Per poter comunicare tra le due reti, un Gateway deve convertire il segnale su rete IP e su una rete a commutazione di circuito e viceversa. Ci sono tre Gateway che adattano la segnalazione e il formato del segnale audio: Signaling Gateway (SG) converte la segnalazione da ss7 a VoIP e viceversa; Media Gateway che converte il segnale audio da ss7 a un formato adatto per la rete IP e viceversa e il Media Gateway Controller (MGC) controlla le operazioni eseguite dalle MG.

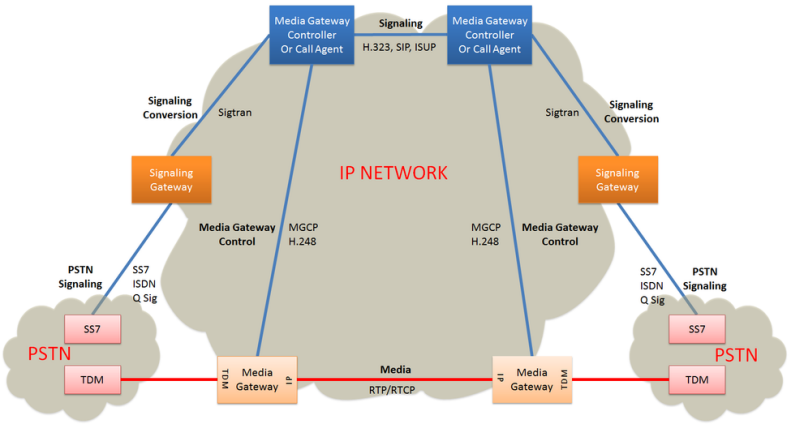

Figura 2.2: Collegamento tra la rete IP e la rete PSTN [6]

2.3 La segreteria telefonica ieri

"Answering machine" e "Voicemail" sono termini inglesi con cui si traducono le parole "segreteria telefonica". Presentano, tuttavia caratteristiche diverse:

- Answering machine: è stata inventata da William Muller nel 1935, ma la prima versione commerciale è stata rilasciata da Joseph James Zimmerman Jr. nel 1949. Il compito di questa macchina consistono nel rispondere alle chiamate e registrarle [7];
- Voicemail: la società Televoice International nel 1980, ha rilasciato sul mercato la prima Voicemail. Le Voicemail hanno inglobato le "Answering machine" e hanno il compito di poter ascoltare il messaggio registrato anche in remoto [8].

"Voicemail" che tradotto letteralmente significa "voce mail", non corrisponde appieno al concetto di "segreteria telefonica" come viene concepita oggi. Infatti quando si pensa alla segreteria telefonica si fa riferimento soltanto a una macchina che registra messaggi se non vi è risposta, invece l'idea iniziale era concepita per inviare anche le registrazioni. Tuttavia le moderne segreterie telefoniche si sono evolute notevolmente in quanto è possibi-

le installare software che permetta la gestione dei messaggi su un personal computer.

Il collegamento tra la segreteria telefonica, il telefono e la rete avviene in questa maniera: la rete telefonica è collegata alla segreteria e quest'ultima è collegata al telefono.

Esistono delle apparecchiature Hardware, chiamate Cisco, che permettono di creare una segreteria telefonica anche se si utilizza la rete PSTN. Chiaramente non si potranno sfruttare al pieno tutti i vantaggi che porta il VoIP, perchè per esempio, non è possibile gestire più chiamate contemporaneamente.

2.4 La segreteria telefonica installata su un computer

Un numero VoIP è facilmente configurabile su Asterisk, un software di VoiceMail Open Source creato da Mark Spencer nel 1999. Il fatto che sia Open Source fa sì che Asterisk sia sempre gratuito e aggiornato, questo è il motivo per cui ha ottenuto tanta importanza e considerazione.

Questo software è ideale per un uso domestico o per una piccola impresa installato su un Raspberry Pi 3. Questa macchina è un single-board computer sviluppato dalla Raspberry Pi Foundation con architettura ARM, pur rimanendo acceso costantemente ha consumi elettrici contenuti, considerato che per alimentarlo serve un comune caricabatterie del telefono. Per questo motivo lo rendono la scelta più appropriata per un uso domestico rispetto a un normale computer.

Asterisk non soltanto svolge le operazioni di una segreteria telefonica normale, bensì ha molte più funzionalità come personalizzare la chiamata, è facilmente collegabile con script di ogni genere e altre caratteristiche che lo rendono attuale.

Avendo collegato il numero VoIP a una macchina è sempre possibile raccogliere, modificare ed elaborare i dati creando un vero e proprio "Internet

of Things". Per esempio, dopo aver installato Asterisk su un Raspberry Pi o un' altra macchina e configurato il numero VoIP, risulta facile collegarlo a qualsiasi script DIY che permetta di aprire, per esempio il cancello, oppure accendere o spegnere le luci, o anche avviare qualche elettrodomestico con una semplice chiamata. Si può utilizzare anche per conoscere la temperatura interna ed esterna di un'abitazione, avere la possibilità di ascoltare i messaggi in segreteria dal proprio telefono cellulare, conoscere quante persone hanno chiamato in segreteria nell'ultimo giorno, o mese o anni.

Chiaramente la chiamata deve essere controllata con una condizione if sul numero chiamante. Non è conveniente che chiunque possa accedere a questi script perchè possono recare gravi danni. Inoltre non è possibile che un qualsiasi utente o un hacker possa chiamare la segreteria telefonica con un numero non suo, a meno che il telefono chiamante è stato smarrito. È sempre possibile aggiungere un codice di sicurezza, da digitare sul tastierino numerico prima di eseguire gli script.

Figura 2.3: Raspberry Pi collegato con: un DHT11 (sensore utilizzato per rilevare la temperatura e umidità), con uno schermo 2004 e attraverso il jack 3.5 è collegato ad una cassa audio.

2.5 Servizi maggiormente usati oggi: Telegram

Negli ultimi anni l'istant message si è sviluppato notevolmente a tal punto da sostituire le mail. Infatti è possibile inviare messaggi di testo, documenti, immagini e altro ancora, non solo utilizzando le mail ma anche con programmi di messaggistica istantanea. Attualmente i più famosi sono WhatsApp e Telegram. La particolarità di Telegram, essendo Open Source, ha la possibilità di creare bot, programmi atti ad inviare, salvare ed elaborare i dati ricevuti.

Attraverso questi bot è possibile inviare le registrazioni presenti in segreteria, in una chat personale o di gruppo, così da potere ascoltare sempre gli ultimi messaggi in segreteria.

2.6 Vantaggi di VoiceMessage

I benefici che porta questo software sono:

1. è Open Source e sarà presto disponibile su GitHub, sarà sempre aggiornato da tutti gli utenti che vorranno apportare modifiche per il miglioramento del programma;

2. l'utente evita di interloquire con presunti spam, l'utente saprà sempre chi è il chiamante e se ci fosse anche solo il sospetto di uno spam, non ci sarebbe nessuna risposta. Inoltre prima di poter chiamare un interno, una voce meccanica riproduce i vari menù della segreteria telefonica e di solito, i centralinisti quando sentono questa voce, riagganciano la chiamata;

3. è un sistema di segreteria moderna, poichè il messaggio viene inviato via mail e Telegram, ciò permette di conoscere sempre chi ha chiamato, anche se si è dall'altra parte del mondo;

4. se nessuno risponde al telefono il messaggio viene inviato tramite bot su un gruppo Telegram, quindi creando gruppi familiari e/o aziendali con questo bot, oltre a poter comunicare con gli altri membri del gruppo, si ha la possibilità di ascoltare i messaggi in segreteria;

5. evita di rispondere al telefono se la chiamata è indirizzata ad un altro utente, sarà il diretto interessato a rispondere o a richiamare;

6. è atto a migliorare le segreterie telefoniche attuali;

7. se non si risponde a una chiamata, perchè momentaneamente occupati, il messaggio registrato verrà inviato come promemoria su Telegram o mail, così sarà sempre possibile rivolgersi al chiamante in un secondo momento.

Capitolo 3

Lavoro Originale

3.1 Il progetto

Questo lavoro di tesi consiste nel creare una nuova interfaccia nelle conversazioni delle segreterie telefoniche VoIP, in quanto questo software permetterà di conoscere anticipatamente sempre chi sta chiamando ancora prima di rispondere al telefono. Infatti prima di interloquire direttamente con un utente, il chiamante si trova in una condizione iniziale a senso unico, cioè egli dovrà presentarsi e comunicare il nome dell'utente con cui desidera conversare, semplicemente parlando al telefono e, attraverso a delle casse audio configurate con l'Advanced Linux Sound Architecture(ALSA), la chiamata verrà ascoltata dai presenti. Il chiamante non riceverà per il momento nessun suono, o sentirà discorsi effettuati dai presenti che ascoltano l'audio dalla segreteria (per questo motivo è a senso unico). Nel momento in cui un utente risponderà al telefono, la chiamata verrà reindirizzata solo su quel dispositivo telefonico.

Asterisk utilizza due software audio: Advanced Linux Sound Architecture (ALSA) ed Open System Sound (OSS). ALSA è un framework Open Source, fa parte del kernel di Linux e presenta diversi vantaggi rispetto ad OSS rendendolo obsoleto. I vantaggi che porta ALSA sono molteplici come un maggior supporto per dispositivi USB, bluetooth, AC'97 e HD AUDIO.

Inoltre supporta JACK Audio Connection Kit (JACK), non supportato da OSS [9].

L'obiettivo di Tesi è quello di creare un programma che svolga le seguenti operazioni:

1. il chiamante chiama la segreteria;
2. la segreteria riproduce un messaggio al chiamante;
3. inizia la registrazione;
4. la chiamata viene spostata sul jack audio del Raspberry Pi;
5. il chiamante che si trova nella fase a senso unico, inizia a parlare.

   Si verificano le seguenti condizioni:

   a. un utente risponde alla chiamata ma prima ancora di essere in collegamento con il chiamante la registrazione viene interrotta e cancellata, da questo momento la chiamata diventa confidenziale;
   b. nessuno risponde alla chiamata, quindi la registrazione termina e il messaggio viene inviato come file audio sui social network configurati e al termine di tale operazione la registrazione viene eliminata dalla macchina;

6. la chiamata termina.

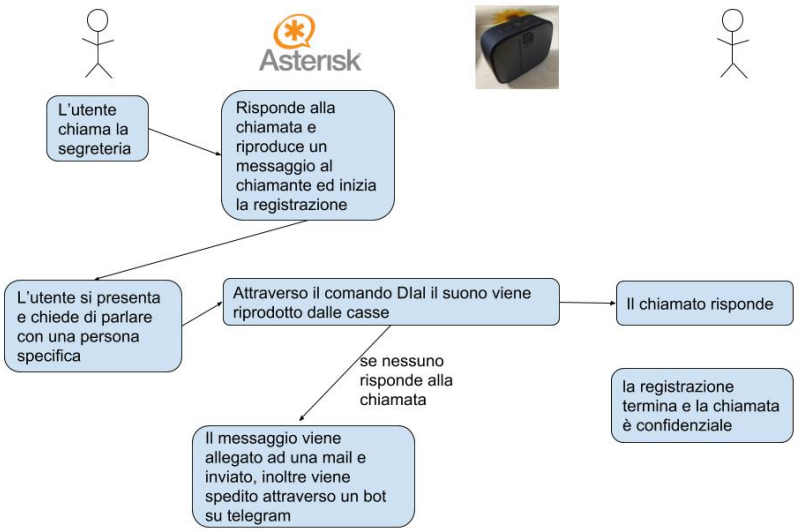

Figura 3.1: fasi che si verificano durante una chiamata

3.2 Come funziona

Gli avvenimenti temporali di VoiceMessage sono i seguenti:

1. il chiamante telefona la segreteria, in questo momento la chiamata si trova nel context di ricezione cioè il secondo context;

2. si controlla se ALSA è disponibile attraverso la variabile booleana "SIPALSA" e alla variabile di Asterisk "AVAILSTATUS", inoltre si verifica che ci siano telefoni disponibili per ricevere la chiamata;

3. se ALSA è disponibile, le operazioni procedono, vengono modificate le variabili e viene creato un file con l'ID del chiamante. Altrimenti viene riprodotto un messaggio che informa il chiamante di riprovare in un secondo momento;

4. dato che ALSA è libero, si riproduce un messaggio che spiega le istruzioni, seguito da un segnale acustico e inizia la registrazione, dopodiché si cambia context;

5. in questo nuovo context la prima operazione che viene eseguita è la chiamata ad ALSA, che rimarrà attiva per 30 secondi;

6. sono presenti tre possibili scenari di terminazione della chiamata:

   - se nessuno dovesse rispondere alla chiamata, ALSA riaggancia e le operazioni continuano, al contrario se è l'utente a riagganciare prima dei 30 secondi, sarà ALSA a continuare le operazioni. Le operazioni sono le seguenti: la registrazione termina e il messaggio viene inviato via mail e via bot Telegram, grazie a uno script bash eseguito attraverso il comando System;
   - se un utente risponde alla chiamata, le operazioni eseguite saranno le seguenti: interruzione ed eliminazione della registrazione, modifica del file creato nel primo context e collegamento alla chiamata tramite bridge, dopodichè, a seconda dell'utente che ha riagganciato (chiamante o chiamato), le operazioni continuano, cioè vengono resettate alcune variabili e viene cancellato il file precedentemente modificato.

La registrazione viene cancellata per evitare di riempire la memoria del dispositivo.

3.3 Le variabili globali

Le variabili globali in Asterisk sono molto vantaggiose, specialmente quando si vuole mantenere qualche valore attivo anche dopo la chiusura della chiamata, oppure per condividere i dati tramite i vari context.

Le variabili globali vengono inizializzate nel context "globals" ed ogni variabile viene utilizzata per uno scopo specifico:

1. sipFILE: contiene il nome del canale adoperato dal chiamante, è il nome di un file creato dal programma in real-time, viene utilizzato anche nel context callme, per andare a recuperare il file creato, inoltre

viene adoperato dall'ultimo context per decidere se inviare o meno il file audio;

2. uniqueID: contiene il nome dell'ultima registrazione effettuata, così da poterla rintracciare, per inviarla e/o cancellarla;

3. SIPALSA: contiene il valore "1" quando il chiamante sta o è in procinto di utilizzare ALSA, "0" altrimenti;

4. SIP* : l'asterisco corrisponde ai numeri dei telefoni VoIP configurati, anche in questo caso quando un utente risponde al telefono in automatico la variabile verrà aggiornata a "1";

5. ENTERED: viene utilizzata per indicare quanti utenti stanno chiamando contemporaneamente; tuttavia è utilizzata per evitare di settare a "0" la variabile SIPALSA da un altro chiamante.

3.4 h extension, il problema di chiusura della chiamata

Questo progetto di Tesi si divide in tre context: il primo context riguarda il chiamato, per rispondere alla chiamata in arrivo, l'utente dovrà chiamare il context mentre il secondo e terzo context interessano il chiamante, questi gestiscono la chiamata in ingresso. La divisione dei context tra chiamante e chiamato è necessaria, in quanto eseguono operazioni completamente diverse.

Il codice del chiamato è strutturato in maniera tale che ogni telefono VoIP associato ad Asterisk, per rispondere deve chiamare lo stesso numero. Invece la divisione tra il secondo e il terzo context è dovuta al fatto che, se il chiamante riaggancia o la linea risulta occupata, bisogna eseguire le seguenti operazioni di terminazione chiamata:

- se il chiamante dovesse terminare prima della funzione "DIAL" ad ALSA e nessun altro sta chiamando, il context deve solo resettare le variabili;

- se il chiamante arriva alla condizione di chiamata di ALSA e riaggancia la chiamata, grazie all'opzione "g" del DIAL, ALSA continua a svolgere le operazioni. Resetta le variabili necessarie per essere richiamati e il messaggio registrato viene inviato via e-mail o Telegram bot. Successivamente il messaggio viene cancellato e termina l'operazione di chiamata;

- se il chiamante riaggancia durante il messaggio di occupato, riprodotto da Asterisk, e se qualcuno sta parlando nella sezione ALSA, Asterisk non deve modificare la variabile SIPALSA;

- se la chiamata inizia e termina immediatamente, perchè per esempio è stato chiamato il numero sbagliato, occorre modificare solo alcune variabili.

Il caso base della chiusura di chiamata avviene quando la segreteria stessa automaticamente riaggancia la chiamata dopo aver inviato i file audio.

La versione precedente del software, il context del chiamante si presenta unito, però durante la chiamata si possono verificare alcune condizioni che portano ad errori, che si verificano se la chiamata termina prima o dopo il dial.

Se il chiamante riaggancia prima del DIAL, il programma deve settare alcune variabili a "0" come "SIPALSA", altrimenti le chiamate successive risulteranno occupate. Questo viene gestito attraverso la "h" extension; ogni qualvolta che un utente o il programma riaggancia prima di chiudere definitivamente la chiamata, Asterisk esegue questi ultimi comandi.

Il problema subentra quando il chiamato risponde al telefono: in questo caso la variabile "h" del context viene eseguita due volte, una dall'utente (se è il chiamato a riagganciare) e l'altra da ALSA; entrambi vanno a resettare la variabile SIPALSA a "0". Per questo motivo il context è stato diviso.

Questa divisione è necessaria per evitare come alcune operazioni vengano eseguite più volte come il reset della variabile, non solo quando il chiamato risponde alla chiamata, ma anche quando il chiamante riaggancia.

Si effettuava il reset di SIPALSA prima dell'invio delle registrazioni ma anche al termine.

È sbagliato in quanto, visto che per inviare questi dati il sistema impiega una decina di secondi, se qualcuno chiama la segreteria ed entra nella chiamata con ALSA, durante l'ultima esecuzione di h, le variabili vengono resettate nuovamente a "0". Di conseguenza se un altro utente chiama la segreteria, la chiamata ad ALSA viene sovrapposta a quella precedente.

Se il chiamante non riaggancia mai prima della chiamata ad ALSA, non emerge la necessità di settare le variabili tramite l'estensione h e di conseguenza dividere i context.

Per procedere in in questo lavoro occorre gestire anche l'eventuale possibilità in cui l'utente riaggancia quando sente il messaggio che la segreteria è momentaneamente occupata, in questo caso le operazioni vanno a resettare la variabile SIPALSA, anche se è utilizzato da un altro utente. Di conseguenza se un chiamante chiama la segreteria subito dopo, essa lo reindirizzerà sul canale di ALSA, già in uso. Il problema descritto in precedenza si risolve aggiungendo una variabile. L'obbiettivo consiste nel capire se bisogna resettare la variabile SIPALSA oppure no, perchè in quel preciso istante qualcuno potrebbe essere in procinto di, o sta utilizzando ALSA.

Creando la variabile contatore ENTERED, che conta quanti utenti stanno chiamando la segreteria, un utente è libero di poter riagganciare quando lo ritiene più opportuno, senza fare perdere la priorità assoluta a chi sta utilizzando ALSA. Infatti prima di settare la variabile SIPALSA a "0", ENTERED dovrà essere anch'esso a "0".

Così facendo se un utente riaggancia durante la riproduzione del messaggio di introduzione ad ALSA oltre a decrementare la variabile "ENTERED" di "1", verifica che quest'ultima sia a "0" e se così fosse, resetta la variabile SIPALSA a "0".

Prima di poter resettare SIPALSA, tutti i chiamanti presenti devono terminare la chiamata. La segreteria sarà nuovamente disponibile dopo 15 secondi, a meno che non ci sia un afflusso di chiamate notevoli.

3.5 Text2wave

```
same => n,System(echo "Siamo momentaneamente occupati
        la preghiamo di riprovare piú tardi"
        | /usr/bin/text2wave -eval "(language_italian)"
        -eval "(Parameter.set 'Duration_Stretch 1.5)"
        -scale 1.5 -F 8000 -o /tmp/busy.wav)
same => n,Playback(/tmp/busy)
```

Una segreteria ben fatta deve avere la possibilità di riprodurre audio e testi; la voce di una persona, può essere registrata con delle impostazioni specifiche adatte per l'uso di Asterisk, oppure utilizzare qualche software text2speech Open Source.

Vi è la possibilità di utilizzare Festival, un software Open Source compatibile con Asterisk. Tuttavia la voce riprodotta dal file audio creato con questo programma, non è per niente limpida e spesso molte parole risultano incomprese; peraltro il comando Festival, converte e riproduce il testo direttamente in chiamata. Per questi motivi non rientra tra le scelte più adatte.

Il programma utilizzato si chiama text2wave ed è inglobato in Festival: è possibile modificare la lingua, il tempo di riproduzione e altre opzioni per rendere la voce, pur sempre meccanica, ma almeno comprensibile.

In un primo momento il testo viene convertito in un file audio e successivamente riprodotto attraverso il comando "Playback".

Per poter riprodurre i file occorre innanzitutto crearli. Si possono creare direttamente da terminale oppure chiamando la segreteria telefonica. Se si decide di crearli utilizzando la segreteria telefonica, una volta creati è necessario commentare la riga che crea questi file audio, perchè vi è la possibilità di gestire più chiamanti contemporaneamente senza dover sostituire il file precedente. Conseguentemente si evita che il messaggio venga riprodotto in maniera incompleta perchè potrebbe essere cancellato e sostituito da un file identico al precedente da un altro utente in chiamata.

Esistono vari software di text2speech come eSpeak, ma l'utilizzo di text2wave è davvero semplice e la qualità della voce è accettabile.

3.6 Creazione di un file

```
same => n,System(echo "${CHANNEL}" > /home/pi/asterisk/${CHANNEL})
```

Nel secondo context (che corrisponde al primo context del chiamante) Asterisk crea un file, dove nome e contenuto sono uguali, e corrispondono al nome del canale in uso. Quest'ultimo è utilizzato dal programma che verificherà se dovrà inviare o meno la registrazione.

Questo file verrà modificato se, e solo se un utente risponde alla chiamata, il contenuto viene poi sostituito con il nome identificativo del telefono con il quale l'utente ha risposto e, prima di mettere in collegamento i due utenti, rimuove il file audio. In questo modo se la chiamata raggiunge il terzo context, ciò avviene quando è il chiamato a riagganciare la chiamata, prima di inviare il messaggio verifica che il contenuto del file sia uguale a "SI" (il contenuto se non viene modificato, è sip.Messagenet seguito dal handler, se qualcuno ha risposto alla chiamata il contenuto è SIP+codice del telefono), cioè constata che qualcuno abbia risposto alla chiamata. Se così fosse il file viene cancellato, altrimenti viene fornito il nome dell'audio allo script send.sh assieme al nome del file testo il quale, invia la mail, chiama uno script python che a sua volta invia il file audio al gruppo Telegram e in ultimo cancella entrambi i file.

Il nome del canale viene passato attraverso la variabile globale sipFILE, questo serve per conoscere il nome del canale del chiamante nel context del chiamato.

Creando il file si ha la possibilità che avvengano altre chiamate contemporaneamente, senza che il programma inverta i chiamati con i chiamanti.

È possibile utilizzare solo le variabili senza creare nessun file, però è intuitivamente più facile strutturato in questa maniera.

3.7 Il DIAL

```
exten => 1,1,Dial(CONSOLE/ALSA,,g S(30) F(chiamata,n,33))
```

Questa è la riga iniziale dell'ultimo context e rappresenta una delle funzioni più importanti del codice, perchè è la prima azione con la quale il chiamante si mette in contatto con gli utenti.

Il Dial si può dividere in tre parti, la prima parte compresa tra la prima parentesi tonda e la virgola, la seconda parte che è omessa si trova esattamente tra le due virgole adiacenti, e la terza parte tutto il resto fino alla chiusura delle parentesi.

La prima parte si riferisce al soggetto che deve chiamare la segreteria, la seconda parte stabilisce il tempo massimo della durata dello squillo del telefono (in questo caso, ALSA risponde immediatamente per questo motivo è stato omesso) e la terza parte comprende tutte le opzioni.

3.7.1 CONSOLE/ALSA

La chiamata ad ALSA consiste nel riprodurre la chiamata sulla stessa macchina che sta eseguendo Asterisk. Ad esempio Ekiga, che è un software Open Source per le chiamate e/o videochiamate, non può essere installata sulla stessa macchina dove viene eseguito Asterisk; di conseguenza se si vuole utilizzare Ekiga è indispensabile l'utilizzo di un altro computer sempre attivo, ciò non è affatto conveniente.

La versione di Asterisk 13.14.1 presenta dei problemi nella configurazione di alsa.conf. Una volta installato ALSA, occorre modificare e non aggiungere due righe, cioè bisogna invertire il caricamento tra chan_alsa.so e chan_oss.so, perchè Asterisk gestisce solo un software audio. Inoltre è necessario indicare il nome di questi device modificando l'output e l'input in alsa.conf.

Un altro problema che emerge per chi utilizza un Raspberry Pi, è che l'uscita jack da 3,5 mm non supporta l'audio in ingresso, di conseguenza se si vuole collegare una cassa audio al jack, non è possibile inviare le chiamate ad ALSA perchè il jack 3,5 del Raspberry è configurato solo per l'audio in uscita. Di conseguenza bisogna creare un file "dummy" il quale fa

credere ad ALSA che si ha un dispositivo in input collegato e funzionante ma questo non emette alcun suono. Dopodiché eseguendo questa riga "DIAL(CONSOLE/ALSA)" la chiamata verrà indirizzata alla scheda audio della macchina, di conseguenza la chiamata verrà ascoltata da tutti i presenti.

Al termine del DIAL, se l'utente non ha riagganciato, l'esecuzione riprende lasciando come sottofondo una suoneria, che si interrompe nel momento in cui la chiamata stessa termina. Questo sottofondo musicale non è necessario ma aiuta l'utente a capire quando può riagganciare senza aver ricevuto risposta, oppure per evitare di dare ulteriori informazioni importanti dal momento in cui la registrazione verrà interrotta dopo qualche istante.

3.7.2 Le opzioni

Questa riga contiene 3 opzioni:

- "g": se il chiamato (in questo caso il chiamato è ALSA) termina la chiamata, sarà il chiamante a proseguire l'esecuzione del programma;
- "S(30)": la chiamata ha durata massima di 30 secondi, al termine dei quali il chiamato riaggancia la chiamata;
- "F(chiamata,n,33)": se la chiamata viene interrotta dal chiamante, il chiamato(quindi ALSA) continuerà ad eseguire il codice nel contesto chiamata, estensione n e priorità 33.

Queste tre opzioni fanno sì che terminati 30 secondi, il chiamato o il chiamante continuerà nell'esecuzione del programma.

L'idea iniziale era quella di non utilizzare una chiamata normale come questa, ma creare una conferenza tra il chiamante, ALSA e successivamente il chiamato. Tuttavia si sono presentate più difficoltà con una chiamata con più persone, rispetto al metodo di chiamata attualmente in uso, perchè alcuni comandi fondamentali, come ad esempio aggiungere un utente alla conferenza oppure eliminarne un altro, venivano sì eseguiti, ma il livello di velocità di

esecuzione non sempre era costante. Potevano emergere dei problemi nel gestire il software della conferenza.

Il problema principale però, erano le chiamate che si sovrapponevano. Ad esempio, quando venivano eseguite più chiamate contemporaneamente, all'inizio della seconda chiamata, i chiamati si invertivano; di conseguenza il chiamato che rispondeva andava a interloquire con il primo chiamante e viceversa.

3.8 Risposta da parte dell'utente

3.8.1 Cosa avviene prima del collegamento tra i due utenti

Per rispondere a una chiamata in arrivo basta chiamare il context 800.

Nelle versioni precedenti, per rispondere alla chiamata, bisogna semplicemente alzare la cornetta. Viene utilizzato l'auto-dial. Tuttavia non è possibile per l'utente effettuare altre chiamate nel momento in cui vi è una chiamata in arrivo. Oltre a questo problema si sono verificati anche i seguenti bug:

- non sempre i telefoni squillano entro i primi 3 secondi;
- gestire in maniera efficace e intelligente l'auto-dial, mettendo un timeout di 30 secondi massimi. Se un dispositivo risponde alla chiamata, non vi è alcun modo di fermare gli altri dispositivi dallo squillare; anche se i dispositivi squillano per 5 secondi, con tempi di richiamata pari a 0 secondi per un massimo di 6 volte, emergerebbero difficoltà nel rispondere perchè trascorsi i 5 secondi non è detto che la chiamata in ingresso sia rapida (problema del punto precedente), inoltre spesso le chiamate sullo stesso dispositivo vengono sovrapposte.

In ogni caso il motivo più importante per il quale si ha abbandonato lo sviluppo dell' auto-dial è causato dal fatto che un utente può utilizzare il

telefono senza essere obbligato a rispondere prima, oppure non deve essere obbligato ad aspettare che termini la chiamata.

Se la chiamata viene accettata da un utente, prima di connettere il chiamante con il chiamato, l'esecuzione del codice del chiamato modifica alcune variabili come SIPALSA che viene aggiornato a "0". Successivamente viene aggiornata la variabile corrispondente al nome del dispositivo con il quale l'utente ha risposto. Questo è utile non solo per capire se ci sono telefoni VoIP disponibili a ricevere chiamate (perchè se fossero tutti occupati verrebbe riprodotto al chiamante il messaggio "occupato"), ma anche per raccogliere dati statistici che potranno essere analizzati in un secondo tempo.

Il file audio viene cancellato perchè non verrà più utilizzato, l'unica cosa necessaria è il file creato contenente il nome del canale che viene salvato in una variabile locale; questo perchè se giunge una nuova chiamata, il contenuto della variabile globale verrà modificato.

3.8.2 Il comando Bridge

```
same => n,Bridge(${contenuto})
```

La seconda riga importante è il collegamento tra il chiamante e il chiamato attraverso il comando "Bridge".
La particolarità di Bridge è che riesce a mettere in collegamento due utenti: il chiamante col chiamato.

Per poter effettuare il collegamento tra i due utenti, è necessario conoscere il canale del chiamante, visto che questa operazione viene svolta nel contesto del chiamato. Il nome del canale si trova all'interno del file creato dal context del chiamante e, attraverso ad una operazione si copia il contenuto del file in una variabile. Il Bridge viene chiamato con il valore della variabile.

La chiamata può terminare per due motivi differenti e le operazioni elaborate sono uguali ma svolte in contesti diversi:

- è il chiamante che riaggancia la chiamata, quindi le operazioni continuano nell'estensione del chiamato, eliminando il file creato in precedenza;

- il chiamato termina la chiamata riagganciando, in questo caso l'esecuzione continua nel context del chiamante, dopo la chiusura dell'operazione bridge, eliminando il file precedentemente creato.

3.9 Controllo e invio del messaggio registrato

```
same => n,System(/usr/share/asterisk/agi-bin/send.sh
                 ${uniqueID} ${sipFile})
```

Prima di inviare un messaggio, la segreteria deve controllare che ci sia un requisito adatto per farlo. Se il contenuto del file è il nome di un telefono VoIP, per esempio SIP2000, la registrazione non deve essere inviata e il programma procede con altre operazioni. Altrimenti se il contenuto è sip.Messagenet* (dove l'asterisco rappresenta il handler) il messaggio deve essere inviato.

Questo controllo si effettua con un GotoIf che controlla solo le prime due lettere del file, dato che la verifica è case-sensitive.

Prima di effettuare il GotoIf, grazie ad un insieme di operazioni, le prime due lettere del file vengono copiate in una variabile locale, su cui effettuare il controllo.

Il comando di invio è molto semplice in quanto è un codice bash, chiamato da Asterisk attraverso il comando System; vengono passati come parametri il nome del file audio e il nome del file contenente il nome del canale.

La prima operazione che esegue questo script è inviare una mail con allegato il file audio, il comando utilizzato è mail -s.

Per la configurazione di mail è stato creato un altro account mail, con una password diversa dalle altre, perchè nel file di configurazione, la password è in chiaro, e bisogna disattivare i controlli degli accessi sul nuovo account, rendendolo meno sicuro.

Terminata questa operazione viene chiamato il codice python, con parametr: il nome della registrazione e il nome del canale in uso, che permette l'invio dei dati su Telegram e successivamente la cancellazione. Questo script

python configurato con Telepot, viene eseguito una sola volta poi termina. Questi script di solito rimangono in loop, ma per inviare solo le registrazioni della segreteria, basta eseguirlo una sola volta, per evitare anche spreco di risorse.

Le operazioni che svolge sono: salvare in una variabile il token del bot, inviare un messaggio di informazione ed infine inviare la registrazione.

Concluse queste operazioni lo script termina e riprende l'esecuzione delle ultime righe del codice chiamante.

Terminati tutti gli invii, i file vengono cancellati.

Se la chiamata, da parte dell'utente, dovesse terminare mentre Asterisk sta eseguendo questi script, l'invio dei dati non viene sospeso, ma appena termina l'esecuzione della riga "System" la chiamata termina. Il tempo impiegato ad arrivare ad eseguire questo comando è veramente esiguo, è necessario qualche istante.

Inizialmente il software inviava una mail attraverso un comando di sistema e poi, con la funzione AGI chiamava direttamente il file python, ma se la chiamata terminava durante l'invio della mail, l'esecuzione non continuava quindi non veniva spedito il messaggio su Telegram. Questo problema si è risolto creando uno script unico che ha la funzione di svolgere le ultime operazioni anche se la chiamata termina.

Si potrebbero utilizzare altri programmi per inviare messaggi via mail e Telegram bot, però questi sono risultati i più facili e veloci da modificare e personalizzare.

Capitolo 4

Valutazioni e Sviluppo

4.1 Valutazioni e considerazioni

L'obiettivo di questa tesi è quello di programmare e realizzare una segreteria funzionale, moderna, originale ed innovativa.

Questo progetto ha avuto un esito positivo: è stato necessario affrontare i vari problemi sorti in fase di progettazione e realizzazione che hanno consentito il raggiungimento degli obiettivi prefissati.

La particolarità è data dal fatto che esistono svariate modalità per sviluppare questo software, ma sono stati adottati altri metodi per renderlo più semplice da leggere e da interpretare.

Per esempio:

- si può creare una conferenza anzichè utilizzare il dial, però come ribadito in precedenza è stato più facile e leggibile attraverso un dial;

- si potrebbe evitare di creare i file ed utilizzare solamente delle variabili e, facendo un controllo su di esse è possibile stabilire se il programma deve inviare o meno la registrazione.

4.2 Sviluppi futuri

Questo software è solo uno scheletro di quello che potrebbe diventare, infatti può essere migliorato attraverso due operazioni differenti: la prima risolvendo alcuni bug conosciuti, aumentare la sicurezza e apportare piccole migliorie, mentre la seconda consiste nell'aggiungere funzionalità.

Fanno parte della prima operazione i seguenti punti:

- risolvere un bug che si presenta molto raramente: quando sia il chiamante sia ALSA riagganciano contemporaneamente. In questo caso la registrazione non viene inviata perchè entrambi gli utenti terminano la chiamata;
- mettere una lista d'attesa se tutti i telefoni o ALSA risultano occupati;
- possibilità di rispondere alla chiamata dal momento che squillano i telefoni VoIP, chiaramente deve comparire il numero del chiamante sul display del telefono;
- anzichè utilizzare le variabili dei telefoni in uso, creare un array;
- utilizzare la funzione LOCK anzichè la variabile SIPALSA per avere un corretto uso della mutua esclusione.

Mentre per il secondo metodo si evidenziano questi ulteriori punti:

- possibilità che la segreteria richiami i chiamanti che hanno trovato lo stato di occupato, quando un apparecchio telefonico ritorna disponibile;
- mettere un flag, che ha la funzione di stabilire l'utilizzo della versione software in modalità casalinga o aziendale: nella versione casalinga quando ALSA è occupato, riaggancia semplicemente. Mentre nella versione aziendale quando ALSA è occupato, riproduce un messaggio di attendere in linea per non perdere la priorità di chiamata;

- avere la possibilità da parte del chiamante, di interloquire con le persone desiderate, con una chiamata diretta e in assenza di risposta, il messaggio verrà recapitato solo al diretto interessato;

Capitolo 5

Conclusioni

L'esito di questo progetto, potrà risultare utile nel futuro metodo di chiamata verso i telefoni fissi.

I componenti di una famiglia, per esempio, non avranno più l'esigenza di rispondere tempestivamente a una chiamata, perchè il messaggio verrà registrato ed inviato su Telegram e/o via mail.

Questo metodo gioverà anche alle aziende che potranno impiegare il software VoiceMessage per gestire al meglio le chiamate, inoltre avranno la possibilità di conservare un promemoria sugli smartphone delle chiamate senza risposta.

Sia i privati che le aziende potranno utilizzare questo software Open Source e modificarlo, migliorando gli aspetti di sicurezza e aggiungendo nuove funzionalità a seconda delle loro esigenze.

Con questo VoiceMessage le segreterie telefoniche dei privati subiranno una delle prime grandi trasformazioni.

Appendice A

Appendice

A.0.1 extension.conf

```
[globals]
zero=0
sipFile=""
uniqueID=""
SIP2000=0
SIP2001=0
SIP2002=0
ENTERED=0
softhangup=0

[callme]
exten => 800,1,NoOp("SIP ${CALLERID(num)} ANSWERED")
same => n,GotoIf($["${SIPALSA}" = "1" ]?1000,1:5000,1)
            ; 2 means busy, check and redirect

exten => 1000,1,Set(contenuto=${FILE(/home/pi
            /menage/${sipFile},0,30)})
same => n,StopMixMonitor()
```

```
same => n,Set(WavID=${uniqueID})
same => n,System(rm /home/pi/menage/${WavID}.wav)
same => n,Set(GLOBAL(softhangup)=1)
same => n,Set(GLOBAL(${CALLERID(num)})=1)
same => n,Set(GLOBAL(SIPALSA)=0)
same => n,Set(sipHandler=${sipFile})
same => n,System(echo "${CALLERID(num)}" > /home
        /pi/menage/${sipHandler})
same => n,ChanIsAvail(SIP/sip.messagenet.it)
same => n,Bridge(${contenuto})
same => n,Set(GLOBAL(${CALLERID(num)})=0)
same => n,System(rm /home/pi/menage/${sipFile})
same => n,Hangup()
same => n,Set(CHAN=${SHELL(asterisk -rx "core show
        channels" |awk -F'-' -vORS='&' ' /^SIP/ {print $1}')
same => n,SoftHangup(${CHAN})

exten => 5000,1,Hangup()

[da_messagenet]
exten => _0510217162,1,ChanIsAvail(CONSOLE/ALSA,as)
exten => _0510217162,n, Set(GLOBAL(ENTERED)=
        ${MATH(${ENTERED}+1,int)})
exten => _0510217162,n,NoOp(${ENTERED})
exten => _0510217162,n,Verbose(0,${AVAILORIGCHAN})
exten => _0510217162,n,GotoIf($[$["${AVAILSTATUS}" = "0" ] &
        $["${GLOBAL(SIPALSA)}" = "0"]
        & $[$["${GLOBAL(SIP2000)}" = "0"] |
        $["${GLOBAL(SIP2001)}" = "0" |
        $["${GLOBAL(SIP2002)}" = "0"]]]?call:busy)
exten => _0510217162,n(busy),System(echo "richiama
```

```
            fra qualche minuto" | /usr/bin/text2wave
            -eval "(language_italian)"
            -eval "(Parameter.set 'Duration_Stretch 1.5)"
            -scale 1.5 -F 8000 -o /tmp/unvailable.wav)

same => n,Playback(/tmp/unvailable)
same => n,Hangup()

exten => _0510217162,n(call),NoOp("SIP ALSA is available")
exten => _0510217162,n, Set(GLOBAL(SIPALSA)=1)
same => n,ChanIsAvail(SIP/sip.messagenet.it)
same => n,System(echo "${CHANNEL}" >
            /home/pi/menage/${CHANNEL})
same => n,Set(GLOBAL(sipFile)=${CHANNEL})
same => n,Set(localChannel=${CHANNEL})
same => n,Set(GLOBAL(uniqueID)=${UNIQUEID})
same => n,System(echo "dopo il bip inizia a
            parlare e il diretto interessato
            rispondera oppure richiamera il prima possibile"
            | /usr/bin/text2wave -eval "(language_italian)" -eval
            "(Parameter.set 'Duration_Stretch 1.5)" -scale 1.5
            -F 8000 -o /tmp/ready2speak.wav)
same => n,Playback(/tmp/ready2speak)
same => n,Playback(beep)
same => n,MixMonitor(/usr/local/etc/menage
               /${uniqueID}.wav,ab)
same => n,Goto(chiamata,1,1)

exten => h,1,Set(GLOBAL(ENTERED)=${MATH(${ENTERED}-1,int)})
exten => h,2,NoOp(${ENTERED})
exten => h,3,GotoIf($["${ENTERED}" = "0"] ?h,10)
```

```
exten => h,4,Hangup()

exten => h,10,Set(GLOBAL(SIPALSA)=0)
exten => h,11,System(rm /home/pi/menage/${CHANNEL})
exten => h,13,Hangup()


[chiamata]
exten => 1,1,Dial(CONSOLE/ALSA,,g S(30) F(chiamata,n,33))
same => n,StartMusicOnHold()
same => n,Goto(chiamata,n,33)
exten => n,33,Set(GLOBAL(SIPALSA)=0)
same => n,Set(GLOBAL(ENTERED)=${MATH(${ENTERED}-1,int)})
same => n,GotoIf($["${softhangup}" = "1"]?n,37:n,40)
exten => n,37,Set(GLOBAL(softhangup)=0)
same => n,Hangup()
exten => n,40,System(test -e /home/pi/menage/${CHANNEL})
same => n,GotoIf($["${SYSTEMSTATUS}" = "SUCCESS"]?:_0510217162,200)
same => n,Set(contenuto=${FILE(/home/pi/menage/${CHANNEL},0,2)})
same => n,GotoIf($["${contenuto}" = "SIP"] ?_0510217162,200)
same => n,Set(contenuto=${FILE(/home/pi/menage/${CHANNEL},0,4)})
same => n,Set(GLOBAL(SIP${contenuto})=0)
same => n,System(rm /home/pi/menage/${CHANNEL})
same => n,Hangup()
exten => _0510217162,200,NoOp("I'm ready to send an Email")
same => n,StopMixMonitor()
same => n,System(/usr/share/asterisk/agi-bin/send.sh
        ${uniqueID} ${sipFile})
same => n,Hangup()
```

A.0.2 send.sh

```
#!/bin/bash
echo "An user left this message in your voicemail" |
    mail -s "Asterisk RPI" -A /usr/local/etc/menage/$1.wav
    "email@mail.com"
python /home/pi/telegramBOT.py $1".wav"
rm "/home/pi/menage/"$2
rm "/home/pi/menage/"$1".wav"
```

A.0.3 telegramBot.py

```
#!/usr/bin/env python
import telepot
import time
import sys
from random import randint
bot = telepot.Bot('BOT-TOKEN')
bot.sendMessage('group or user ID', 'A user left this
    message on your voicemail')
bot.sendAudio('group or user ID', open('/home/pi/
    menage/'+sys.argv[1], 'rb'), title='Asterisk call')
```

Per la creazione del progetto è stato consultato, oltre al libro "Asterisk the definitive guide" i seguenti siti:

[1] Stackoverflow, URL: https://stackoverflow.com

[2] Voip-info, URL: https://www.voip-info.org/

[3] Wiki.asterisk, URL: https://wiki.asterisk.org/wiki/dashboard.action

Bibliografia

[1] R. Bryant, L. Madsen, J. Van Meggelen, Asterisk the definitive guide, O'REILLY, Sebastopol CA, 2013.

[2] Rosario G. Garroppo, Voice over IP (VoIP) aspetti architetturali e di progettazione, Pisa, Edizioni Plus, 2007.

[3] H.323, URL: https://en.wikipedia.org/wiki/H.323.

[4] File:Typical H.323 Stack.png, URL: https://commons.wikimedia.org/wiki/File:Typical_H.323_Stack.png

[5] Session Initiation Protocol, URL: https://en.wikipedia.org/wiki/Session_Initiation_Protocol.

[6] File:Converged Network Architecture.png, URL: https://commons.wikimedia.org/wiki/File:Converged_Network_Architecture.png

[7] Answering machine, URL: https://en.wikipedia.org/wiki/Answering_machine.

[8] Voicemail, URL: https://en.wikipedia.org/wiki/Voicemail.

[9] Open Sound System, URL: https://wiki.archlinux.org/index.php/Open_Sound_System.

Ringraziamenti

Al termine di questo mio percorso universitario, desidero ringraziare tutti coloro che mi hanno aiutato nella realizzazione della mia Tesi.

In particolare il Professore Renzo Davoli, relatore di questa Tesi di Laurea per la grande disponibilità e per l'aiuto fornito durante la stesura. Oltre ad avermi guidato nella realizzazione del lavoro, mi ha avviato nel mondo del Open Source, di Arduino e di Raspberry Pi, e mi ha introdotto con grande interesse su questo coinvolgente studio delle segreterie telefoniche.

Ringrazio tutti i Docenti universitari che, con la loro professionalità e disponibilità hanno contribuito alla mia formazione.

Proseguo ringraziando i miei colleghi universitari, per aver condiviso momenti di studio e di amicizia.

Un ringraziamento speciale alla mia famiglia che mi ha sostenuto ed aiutato in questa fase importante della mia vita.

Infine, desidero ringraziare con affetto i miei parenti ed amici per il sostegno dato e per essermi stati vicino durante tutto il percorso universitario.