

Esercitazione di laboratorio su Asterisk

Marco Romagnoli, Davide Quaglia

14 gennaio 2011

Indice

1 Introduzione	2
2 Installazione	2
3 Configurazione di Asterisk	5
3.1 Gli utenti in asterisk	5
3.2 Il file sip.conf	6
3.3 Il file extensions.conf	8
3.3.1 Regole	9
3.3.2 Estensioni particolari	9
3.3.3 Applicazioni	10
3.4 Pattern Matching	11
3.5 Utilizzo dei contesti Macro	12
3.6 Voicemail	13
3.7 Asterisk e NAT	14
4 Esercitazione	18
4.1 Creazione degli utenti	18
4.2 Dialplan	18
4.3 Configurazione softphone	19
4.4 Dialplan avanzati	23
4.5 Asterisk e Google talk	24

1. Introduzione

Asterisk è un PBX (Private Branch Exchange) cioè un centralino che permette di creare una propria rete telefonica privata con notevole flessibilità.

- Supporta i protocolli SIP e H.323
- Può essere configurato per lavorare con altri strumenti SIP come Sip Express Router (SER) (<http://www.iptel.org/ser/>)

2. Installazione

L'esercitazione di Laboratorio consisterà nell'installare Asterisk su una distribuzione Linux Ubuntu installata su macchina virtuale utilizzando VMware. I passi da svolgere sono quindi i seguenti, a prescindere dal S.O. di partenza sia esso Windows, MAC o Linux:

I.) Installare VMware Player (Download dal sito http://downloads.vmware.com/d/info/desktop_downloads/vmware_player/3_0)

II.) Creare una macchina virtuale e installarci sopra Linux Ubuntu 10.10 a partire dal file ISO scaricabile dal sito <http://www.ubuntu.com>.

NOTA: è possibile usare direttamente una macchina virtuale con Ubuntu 10.10 se è stata messa a disposizione dal docente.

Ora passiamo ad installare Asterisk su Ubuntu 10.10.

I.) Accedere a Synaptic Packet Manager e digitare : asterisk come keyword nel campo search di Synaptic. Al comparire dei vari pacchetti selezionare per l'installazione:

- 1) Asterisk
- 2) Asterisk-prompt-it
- 3) Asterisk-doc
- 4) Asterisk-config
- 5) Asterisk-sounds-main
- 6) Asterisk-dbg
- 7) Dahdi
- 8) Dahdi-dkms
- 9) Dahdi-linux

II.) A questo punto, tenendo presente che tutte le operazioni andranno

effettuate come root, si può avviare Asterisk e collegarsi alla console, con il comando:

```
>sudo asterisk -vvvc
```

Asterisk tipicamente gira in background come un servizio di rete o daemon. Se Asterisk sta già girando, occorre digitare il comando seguente per connettersi alla command-line interface.

```
[root@server ~]# asterisk -r
```

Il parametro `-r` dice al sistema che ci si vuole ricollegare alla console. Si ottiene allora una schermata analoga a questa (CLI sta per Command Line Interface):

```
[root@server ~]# asterisk -r
Asterisk version, Copyright (C) 1999 - 2010 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty'
for details.
This is free software, with components licensed under the GNU General
Public
License version 2 and other licenses; you are welcome to redistribute
it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk version currently running on server (pid =
11187)
server*CLI>
```

I comandi prendono la forma di **module action parameters**. Per esempio **core show uptime** è il comando per vedere da quanto tempo il servizio sta girando.

```
*CLI> core show uptime
System uptime: 1 hour, 34 minutes, 17 seconds
Last reload: 1 hour, 34 minutes, 17 seconds
```

Si può utilizzare il built-in help per avere maggiori informazioni riguardo ai vari comandi. Digitando **core show help** al prompt per una lista completa di comandi, oppure **core show help comando** per l' help su un particolare comando.

Esempi:

Per uscire dalla shell :

```
*CLI> quit
```

Per riavviare Asterisk:

```
*CLI> reload
```

Per chiudere/fermare il processo Asterisk:

```
*CLI> core stop now
```

Come si ripete, per avviare Asterisk:

```
*CLI>sudo asterisk -vvvc
```

Le “v” stanno per “verbose” e il loro numero indica il livello di verbosity cioè la quantità di informazioni riguardanti le chiamate.

Asterisk allows you to control the verbosity level of the command-line interface. At a verbosity level of zero, you'll receive minimal information about calls on your system. As you increase the verbosity level, you'll see more and more information about the calls. For example, if you set the verbosity level to three or higher, you'll see each step a call takes as it makes its way through the dialplan. There are very few messages that only appear at verbosity levels higher than three.

Altri comandi importanti per fermare e riavviare Asterisk:

Per fermare il servizio:

1. **core stop now** This command stops the Asterisk service immediately, ending any calls in progress.
2. **core stop gracefully** This command prevents new calls from starting up in Asterisk, but allows calls in progress to continue. When all the calls have finished, Asterisk stops.
3. **core stop when convenient** This command waits until Asterisk has no calls in progress, and then it stops the service. It does not prevent new calls from entering the system.

Per riavviare il servizio:

1. **core restart now** This command restarts the Asterisk service immediately, ending any calls in progress.

2. **core restart gracefully** This command prevents new calls from starting up in Asterisk, but allows calls in progress to continue. When all the calls have finished, Asterisk restarts.
3. **core restart when convenient** This command waits until Asterisk has no calls in progress, and then it restarts the service. It does not prevent new calls from entering the system.

3. Configurazione di Asterisk

Tutti i file di configurazione di asterisk terminano con l'estensione “.conf” e si trovano nella cartella /etc/asterisk/. I più importanti sono:

- **sip.conf**: file di configurazione dove vengono specificate le caratteristiche degli utenti che utilizzano il centralino mediante il protocollo SIP.
- **extensions.conf**: questo file di configurazione contiene il dialplan, specifica le operazioni che vengono eseguite da asterisk a seguito di una chiamata.
- **iax.conf**: file di configurazione dove vengono specificate le caratteristiche degli utenti che utilizzano il centralino mediante il protocollo IAX.

In tutti i file di configurazione, le righe di commento sono precedute da “;”.

3.1 Gli utenti in asterisk

Gli utenti vengono suddivisi in tre categorie:

- **user**: gli user possono solo fare telefonate.
- **peer**: i peer possono solo ricevere telefonate.
- **friend**: i friend possono sia ricevere che fare telefonate, è uno “short-cut” per la definizione contemporanea di uno user e un peer.

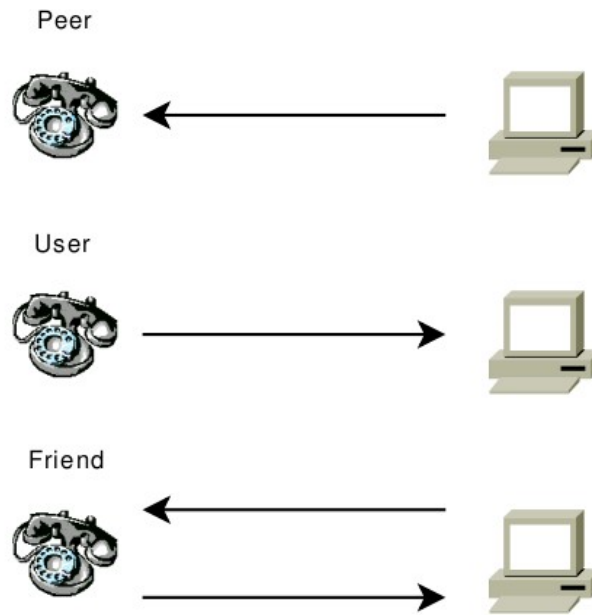


Figura 1. tipologie di utenti.

3.2 Il file sip.conf

All'interno del file `/etc/asterisk/sip.conf` vengono specificati tutti gli utenti che hanno accesso al centralino asterisk mediante il protocollo SIP. Questo file di configurazione è suddiviso in sezioni il cui nome è contenuto tra parentesi quadre. La prima sezione è chiamata "general" e contiene le opzioni di default che deve avere ogni utente, queste opzioni possono comunque essere sovrascritte dalle sezioni successive all'interno del file. Ogni altra sezione descrive l'utente corrispondente.

```
[general]
context = default
bindport = 5060
bindaddr = 0.0.0.0
srvlookup = yes
```

In questo esempio la sezione `general` contiene quattro opzioni:

- `context`: definisce quale contesto gestisce l'utente in questione, all'interno del file `extensions.conf`.
- `bindport`: Determina su quale porta UDP locale Asterisk è in ascolto per connessioni.
- `bindaddr`: Indirizzo dell'interfaccia di rete sul quale asterisk ascolta per connessioni

(di default 0.0.0.0 si mette in ascolto su tutte le interfacce di rete locali).

- `srvlookup`: Se abilitata e come argomento dell'applicazione `Dial()` viene specificato un nome Internet allora vengono interrogati i DNS per cercare l'associazione nome logico - indirizzo.

Di seguito si definisce un esempio di utente con le opzioni minime:

```
[user] ; spesso è un numero di telefono.  
type=friend  
secret=password  
qualify=yes  
host=dynamic  
canreinvite=no  
context=internal
```

In questa sezione si definiscono le opzioni dell'utente `user`: In mancanza del campo `username`, per l'autenticazione viene usato "user" (la stringa tra parentesi quadre).

- `type`: `friend` | `user` | `peer` : definisce che tipo di utente è "user", in base alla figura di pagina precedente.
- `secret`: La password che permette di registrare l'utente "user".
- `qualify`: `yes` | `no` | `val`: determina quanto devono essere distanti i peers classificati raggiungibili, di default (con `yes`) dopo due secondi i peers non vengono più considerati raggiungibili e quindi non potranno più ricevere chiamate. Con `val` è possibile specificare manualmente un valore.
- `host`: `dynamic` | `IP` | `static`: consente all'endpoint dell'utente di registrarsi con asterisk in maniera tale che asterisk sappia come raggiungere l'endpoint dell'utente. In questo caso asterisk agisce in un certo qual senso come un registrar server del protocollo SIP. Per limitare l'endpoint ad un singolo indirizzo IP è possibile specificarlo qui al posto del valore "dynamic". Mentre con `static` si dice ad Asterisk che non è necessario che l'utente si registri prima di ricevere telefonate.
- `context`: definisce quale context all'interno del file `extensions.conf` gestisce l'utente. In questo caso viene ridefinita l'opzione `context` definita nella sezione `[general]`.
- `canreinvite`: `yes` | `no` : Questa opzione di default è settata a `no`: normalmente asterisk non utilizza i messaggi di reinvio.

I messaggi di `INVITE` all'interno del protocollo sip vengono usati per stabilire comunicazioni. All'interno di un messaggio di `INVITE` si trovano le informazioni per rendere noto all'altro endpoint dove inviare il media stream della chiamata.

Asterisk normalmente si pone lui stesso come endpoint della chiamata anche per il flusso RTP ma quando si vuole forzare il passaggio diretto dei pacchetti RTP tra i

due endpoint allora Asterisk manda un messaggio di REINVITE ai due endpoint dando loro le informazioni necessarie per la comunicazione diretta.

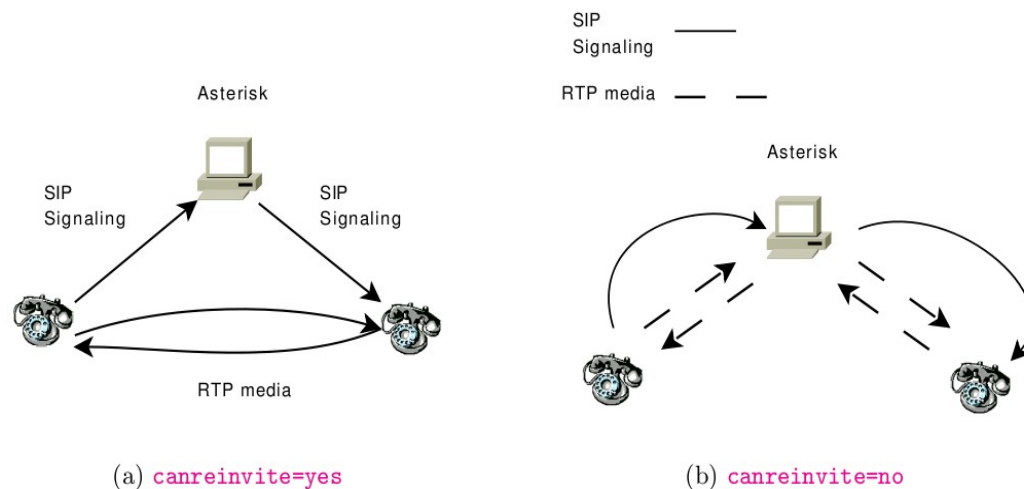


Figura 2: effetto dell'opzione canreinvite

I parametri qui descritti sono lo stretto necessario per la definizione di un utente, per la lista completa delle opzioni si rimanda al file `/etc/asterisk/sip.conf` generato durante l'installazione di asterisk o al manuale: <http://www.asterisk.org/docs>.

3.3 Il file `extensions.conf`

All'interno del file `/etc/asterisk/extensions.conf` è contenuto il dialplan del centralino asterisk. Un dialplan specifica come devono essere gestite da asterisk le chiamate in ingresso e in uscita, da e verso gli endpoint.

Il file `extensions.conf` è suddiviso in sezioni chiamati contesti. Nei contesti sono contenute le regole di chiamata raggruppate in dialplan, le regole di un contesto sono totalmente scorrelate dalle regole di un altro contesto, a meno che questo non sia richiamato con il comando `include`.

Il nome di un contesto viene specificato inserendolo all'interno di parentesi quadre. Esistono due contesti particolari all'interno del file `extensions.conf`:

- **general:** in questo contesto possono essere definite opzioni che riguardano l'intero dialplan.
- **globals:** in questo contesto vanno definite le variabili globali che possono essere referenziate all'interno del dialplan mediante la funzione `global`, con la seguente sintassi:

```

${GLOBAL(VARIABLE)}

```


3.3.1 Regole

Il dialplan è formato da regole aventi la seguente forma:

```
exten => nome,priorità,applicazione()
```

- nome: è l'identificativo di un utente del centralino (stringa o classico numero di telefono).
- priorità: è un numero intero che identifica il “passo” della sequenza di regole relative ad un nome. Ogni sequenza è composta da più passi, ogni passo ha una sua priorità che deve essere data in ordine sequenziale a partire da 1.
- applicazione(): rappresenta l'azione che deve compiere Asterisk quando quel passo della sequenza viene eseguito.

Vediamo un esempio di dialplan formato da tre regole in sequenza:

```
exten => 123,1,Playback(hello-world)
exten => 123,2,Dial(SIP/123)
exten => 123,3,Hangup()
```

Quando arriva una chiamata su asterisk al numero (estensione) 123, viene eseguito il primo passo del dialplan: l'utente sentirà il messaggio “Hello-world”. Al termine del messaggio asterisk esegue in sequenza il secondo passo del dialplan: la priorità 2, che prevede la chiamata effettiva di 123 tramite il protocollo SIP in quanto l'utente 123 utilizza tale protocollo ed è stato definito nel file sip.conf. Al termine della conversazione viene eseguito il terzo passo del dialplan con l'applicazione Hangup(), applicazione che non riceve argomenti e permette ad asterisk di chiudere la comunicazione (altrimenti il chiamante deve per forza riattaccare lui).

3.3.2 Estensioni particolari

Esistono delle estensioni che hanno un significato particolare e che non corrispondono a stringhe digitate dal chiamante:

- estensione s: si usa questa estensione quando si vuole mettere una regola che venga eseguita sempre come punto di partenza di un contesto. Ad esempio supponiamo che il nostro centralino asterisk sia connesso ad una linea esterna PSTN e che dall'esterno sia solo possibile chiamare tale centralino e non direttamente i numeri interni all'organizzazione. Allora può essere utile, ogni volta che dalla linea PSTN viene digitato il numero del centralino asterisk, usare l'estensione s per scrivere una regola che faccia in modo che il centralino risponda automaticamente alla chiamata

chiedendo con che interno si desidera essere messi in comunicazione e infine inoltra la chiamata all'interno digitato.

```
exten => s,1,Background(enter-ext-of-person)
exten => 123,1,Dial(SIP/123)
exten => 321,1,Dial(SIP/321)
```

- estensione i: l'estensione i (invalid) viene utilizzata per gestire automaticamente la digitazione di estensioni non valide cioè non descritte nei file di configurazione (da usare assieme all'applicazione Background che chiede la digitazione di un numero di telefono).

```
exten => s,1,Background(enter-ext-of-person)
exten => 123,1,Dial(SIP/123)
exten => 321,1,Dial(SIP/321)
exten => i,1,Playback(invalid-ext)
exten => i,2,Hangup()
```

In questo esempio esistono solo gli utenti 123 e 321 e se il chiamante digita un numero differente saranno eseguiti in successione i passi 1 e 2 dell'estensione i.

- estensione t: viene usata per rendere noto al chiamante che l'estensione (utente) digitata tramite l'applicazione Background non è al momento disponibile.

Ad esempio:

```
exten => s,1,Background(enter-ext-of-person)
exten => 123,1,Dial(SIP/123,5)
exten => 321,1,Dial(SIP/321,5)
exten => t,1,Playback(all-busy)
exten => t,2,Hangup()
```

Dopo 5 secondi che il telefono chiamato squilla e nessuno risponde, il controllo passa all'estensione t, eseguendo in sequenza le priorità 1 e 2.

Per una lista completa delle estensioni particolari:

<http://www.voip-info.org/wiki/view/Asterisk+standard+extensions> .

3.3.3 Applicazioni

Le applicazioni più usate all'interno di un dialplan sono:

- Dial: permette di chiamare una estensione (utente)

```
exten => 123,1,Dial(SIP/123,10)
```

L'applicazione dial risponde in modi differenti in base alle risposte ottenute: Quando l'estensione viene digitata l'applicazione Dial farà suonare il telefono connesso alla linea SIP/123. La linea chiamata viene sempre identificata con protocollo/utente. Il secondo argomento specifica di far suonare il telefono per 10 secondi.

Se l'utente chiamato risponde, al termine della conversazione l'applicazione dial esce senza alcun stato di errore. Se invece nessuno risponde, passati i 10 secondi verrà eseguito lo statement con priorità successiva.

Se l'applicazione Dial ottiene un segnale di occupato dal telefono remoto allora viene aggiunto 101 alla priorità che si sta eseguendo e si passa ad eseguire quella priorità se definita.

Per una lista completa dei possibili argomenti si rimanda alla documentazione ufficiale di Asterisk.

- **Playback:** questa applicazione fa ascoltare al chiamante un file audio passato come argomento (solo il nome del file senza indicarne l'estensione). Se non viene specificato il percorso completo del file bensì solo il nome, Asterisk cerca il file in */var/lib/asterisk/sounds*
- **Background:** L'applicazione Background è simile a Playback ma ha un uso differente: dopo aver fatto ascoltare un file audio all'utente l'applicazione si mette in attesa che l'utente digiti un'estensione sulla tastiera del proprio telefono.

```
exten => 123,2,Background(enter-ext-of-person)
exten => 100,1,Dial(SIP/100)
exten => 101,1,Dial(SIP/101)
```

Per una lista completa delle possibili applicazioni si rimanda alla documentazione ufficiale.

3.4 Pattern Matching

Per poter semplificare il dialplan, soprattutto quando le estensioni da gestire sono molte è conveniente utilizzare il pattern matching.

- **_** : ogni volta che si intende sfruttare il pattern matching è necessario inserire l'underscore davanti al numero (estensione) che si intende "catturare"
- **X** : rappresenta qualsiasi numero intero da 0 a 9
- **N** : rappresenta qualsiasi numero intero da 2 a 9
- **Z** : rappresenta qualsiasi numero intero da 1 a 9
- **[35-8]**: il match avviene sui numeri 3,5,6,7,8
- **.** : il pattern matching avviene su qualsiasi stringa alfanumerica.

Esempio

```
[chiama]
exten => 100,1,Dial(SIP/100)
exten => 101,1,Dial(SIP/101)
.
.
.
exten => 200,1,Dial(SIP/200)
```

Per gestire 200 estensioni senza utilizzare il pattern matching è necessario specificare 200 righe all'interno del dialplan. Usando il pattern matching:

```
[chiama]
exten => _[12]XX,1,Dial(SIP/${EXTEN})
```

La variabile `EXTEN` è una variabile di canale (visibile solo all'interno del canale di comunicazione che si crea tra i due interlocutori) e assume il valore dell'estensione chiamata.

3.5 Utilizzo dei contesti Macro

Il comando Macro permette di organizzare il dialplan in maniera più semplice, permettendo a contesti differenti di condividere le stesse estensioni, riducendo le ripetizioni all'interno del dialplan.

```
Macro(nome, arg1, arg2, . . . , argN)
```

All'interno della macro è possibile ricavare l'utente chiamato (estensione), il contesto e la priorità rispettivamente con `${MACRO_EXTEN}`, `${MACRO_CONTEXT}` e `${MACRO_PRIORITY}`.

Più in generale è possibile ricavare gli argomenti passati a una macro con: `${ARG1}`, `${ARG2}` . . . `${ARGN}`.

L'unico fattore limitante all'interno di una macro è l'utilizzo dell'estensione s.

Esempio

```
[context_1]
exten => 222,1,Dial(SIP/222,10)
```

```
exten => 222,2,Playback(vm-nobodyavail)
```

```
[context_2]  
exten => 333,1,Dial(SIP/333,10)  
exten => 333,2,Playback(vm-nobodyavail)
```

Usando il comando macro :

```
[context_1]  
exten => 222,1,Macro(chiama,SIP/222)
```

```
[context_2]  
exten => 333,1,Macro(chiama,SIP/333)
```

```
[Macro-chiama]  
exten => s,1,Dial(${ARG1},10)  
exten => s,2,Playback(vm-nobodyavail)
```

3.6 Voicemail (segreteria telefonica)

Asterisk permette l'utilizzo delle voicemail all'interno del dialplan. La configurazione avviene mediante il file `/etc/asterisk/voicemail.conf`. Questo file, suddiviso in contesti, contiene i dettagli riguardanti le voicemail degli utenti:

```
extension_number => voicemail_password,username,user_email
```

- `extension_number`: numero di telefono per cui è attiva la segreteria telefonica
- `voicemail_password`: password numerica della voicemail
- `username`: l'utente della mailbox
- `user_email`: indirizzo email dell'utente. Ogni volta che un chiamante lascia un messaggio nella voicemail asterisk manda un messaggio di notifica a questo indirizzo.

Ad esempio:

```
[default]  
100 => 6262,Nancy, nancy@acme-widgets.com
```

Dopo aver configurato la voicemail si deve permetterne l'accesso all'interno del dialplan

```
[esempio]
exten => 100,1,Dial(SIP/100,10)
exten => 100,2,VoiceMail(u100@default)
```

Con la priorità 2 del contesto “esempio” si specifica che se l'utente non è disponibile entro 10 secondi si passi la chiamata all'applicazione voicemail a cui si passa il parametro “u100@default” per forzarla a eseguire un messaggio di “utente non disponibile” (“u” sta per “unavailable”), a chiedere di lasciare un messaggio vocale, a memorizzare tale messaggio nella casella vocale dell'utente 100 del contesto "default" (cioè appunto [“100@default”](#)).

L'utente 100 deve poter telefonare ad Asterisk per vedere se ha messaggi nella propria casella; occorre quindi impostare un numero di telefono apposito nel dialplan.

```
[voicemail]
exten => 500,1,VoicemailMain()
```

3.7 Asterisk e NAT

La maggior parte dei protocolli pensati per il VOIP (come il protocollo SIP) non sono stati pensati per lavorare con il Network Address Translation.

Il problema più frequente, riguarda il “one way audio”: al momento del setup della chiamata sia il telefono con IP pubblico che il telefono all'interno della rete privata squillano ma al momento della conversazione solo l'utente con il telefono con IP pubblico sentirà la voce dell'interlocutore.

Il problema è descritto dalla figura 3: la comunicazione comincia dall'host A, interno alla NAT, con un messaggio SIP specificando al suo interno: “sono A, aspetto il media della chiamata all'indirizzo 192.168.3.5”. Allo stesso modo B risponde con un messaggio SIP (il messaggio 2) “sono B aspetto il media della chiamata all'indirizzo 203.34.5.23”. Quando i due host cominciano a scambiarsi messaggi RTP, contententi la parte multimediale della comunicazione, tutti i messaggi di B rivolti ad A non vengono recapitati con successo a causa di un indirizzo privato specificato nei messaggi SIP di A. Al contrario, B ha un indirizzo pubblico, quindi A riesce a spedire con successo i suoi messaggi RTP.

In sostanza, l'utente che risponde al telefono con IP pubblico sarà in grado di ascoltare la voce dell'interlocutore, mentre l'utente con indirizzo privato non riuscirà ad ascoltare la voce del suo interlocutore.

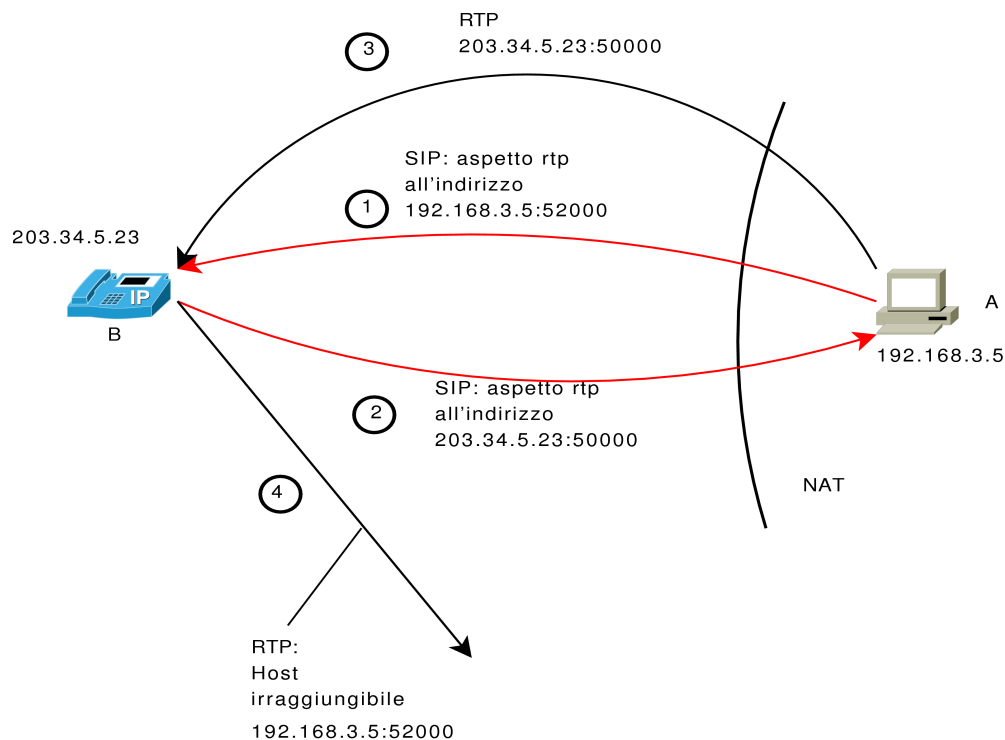


Figura 3. Scenario che determina il problema "one way audio".

Per ovviare al problema esistono delle configurazioni particolari di asterisk che variano in base alla configurazione delle rete sulla quale è posto il server stesso:

- **Asterisk e chiamante dietro una NAT, destinatario con indirizzo pubblico**

In questo caso il problema può essere evitato utilizzando il port forwarding: sul firewall viene aperta la porta utilizzata per la segnalazione della chiamata (di default 5060, nel caso di SIP) e tutte le porte che asterisk utilizza per il protocollo RTP (definite all'interno del file /etc/asterisk/rtp.conf) infine vanno definiti all'interno del file sip.conf i parametri "localnet" ed "externip" nella sezione general:

```
[general]
```

```
.
.
.
```

```
localnet=192.168.1.0/8 ; La rete privata dove asterisk è connesso
externip=147.34.56.23 ; L'indirizzo pubblico che la rete privata usa
; per comunicare con l'esterno
```

In sip.conf, per ogni telefono che è su una rete privata, va aggiunto il parametro:

```
nat=yes ; Il telefono lavora dietro una nat
```

In questo modo si risolve il problema “one way audio”: ogni volta che un telefono con IP privato intende iniziare una comunicazione verso l’esterno contatta il server asterisk, asterisk riconosce che è un dispositivo nattedo (all’interno della sua configurazione è configurato “nat=yes”) e quindi cambia l’indirizzo privato, specificato all’interno dei messaggi SIP del dispositivo, con il parametro externip. In questo modo ogni messaggio SIP che raggiungerà l’esterno conterrà un indirizzo valido (l’indirizzo del nat-router) che sarà utilizzato dal dispositivo con IP pubblico per mandare i suoi pacchetti RTP.

- **Asterisk e chiamante dietro una rete NATtata, destinatario dietro un’altra rete NATtata**

In questo caso si deve usare un mediatore, come ad esempio un altro server asterisk, interno alla seconda rete NATtata che abbia il compito di dirigere il traffico verso il server asterisk interno alla prima nat. Il protocollo IAX in questo caso risulta essere il più efficiente per connettere i due asterisk: utilizza una sola porta (la 4069) sia per i dati riguardanti la segnalazione sia per i pacchetti riguardanti la chiamata, che si traduce in un numero minore di porte da aprire sui firewall e una maggiore sicurezza per la le reti private.

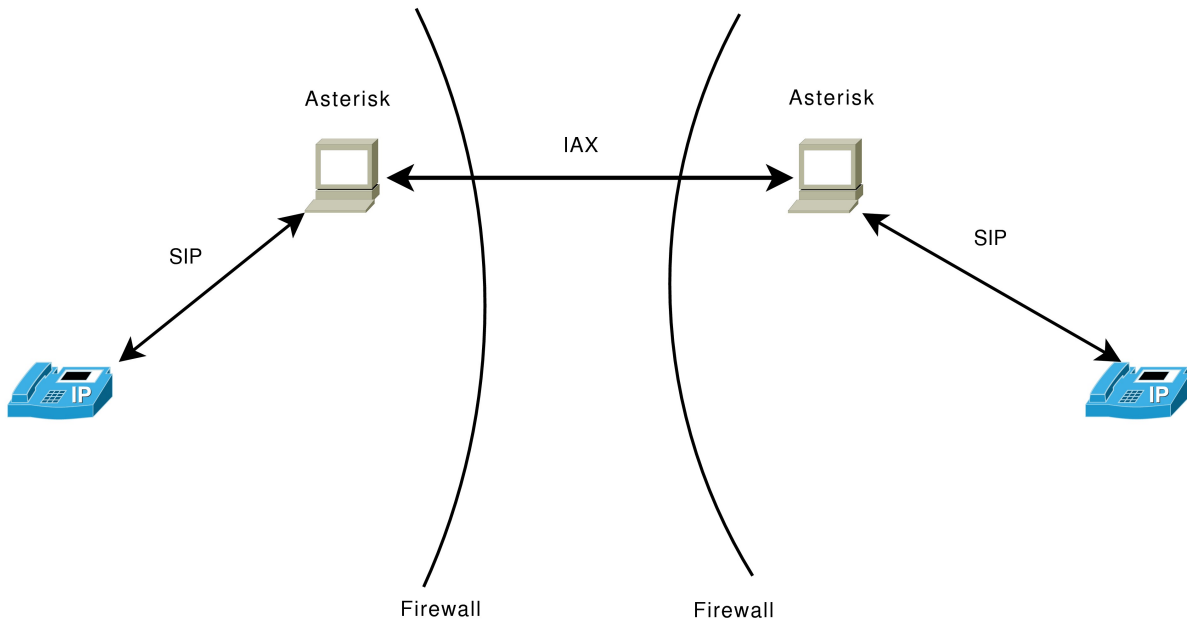


Figura 4: Asterisk in una NAT, peers in un’altra NAT

- **Asterisk con IP pubblico e telefoni dietro una rete NATtata:**

In questo caso i telefoni devono riuscire a ricavare l’indirizzo pubblico usato dal proprio nat-router, così da poterlo includere nei messaggi che vengono inviati verso

il server asterisk.

Una possibile soluzione è l'utilizzo del server STUN che permette al device all'interno della rete privata di conoscere il tipo di nat nel quale è posto e l'indirizzo pubblico con il quale può comunicare con l'esterno. Il protocollo STUN è definito dalla norma RFC 3489.

4. Esercitazione

4.1 Creazione degli utenti

Dopo aver installato Asterisk come da istruzioni nelle sezioni 2 e 3, ed aver preso dimestichezza con i comandi.

Aprire il file sip.conf con i permessi di amministratore

```
> sudo gedit /etc/asterisk/sip.conf
```

oppure > sudo nautilus

Modificare il file sip.conf, aggiungendovi **in fondo al file** le seguenti righe:

```
[1234]
type=friend
username=1234
secret=youpasswd
host=dynamic
context=internal
qualify=yes
nat=yes
```

```
[1000]
type=friend
username=1000
secret=youpasswd
host=dynamic
context=internal
qualify=yes
nat=yes
```

4.2 Dialplan

Modificare ora il file extensions.conf aggiungendovi **in fondo al file** le seguenti righe:

```
[internal]
exten => 1234,1,Dial(SIP/1234)
exten => 1234,2,Hangup()
exten => 1000,1,Dial(SIP/1000)
exten => 1000,2,Hangup()
```

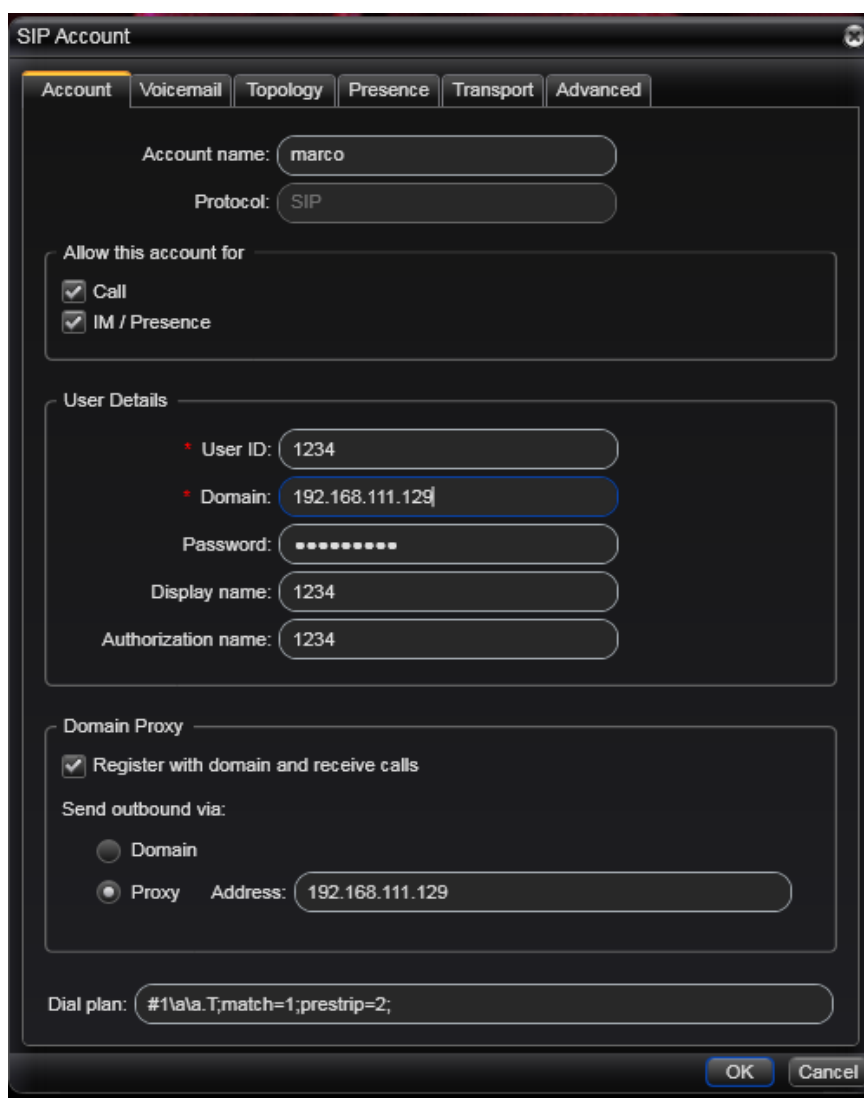
4.3 Configurazione dei softphones

Occorre ora fare una distinzione in base a quale sistema operativo host si sta utilizzando.

Utenti Windows e MAC:

Installare Xlite <http://www.counterpath.com/x-lite-download.html>

Configurazione X Lite su Windows: cliccare "SoftPhone" sul menù in alto a sinistra e successivamente "Account Settings". Si raggiunge la scheda SIP account di figura 5. Si impostino ora i dati in base all'utente come da istruzioni di seguito :



The image shows a screenshot of the 'SIP Account' configuration window in X-Lite. The window has a dark theme and several tabs: 'Account', 'Voicemail', 'Topology', 'Presence', 'Transport', and 'Advanced'. The 'Account' tab is selected. The configuration fields are as follows:

- Account name: marco
- Protocol: SIP
- Allow this account for:
 - Call
 - IM / Presence
- User Details:
 - User ID: 1234
 - Domain: 192.168.111.129
 - Password: [masked]
 - Display name: 1234
 - Authorization name: 1234
- Domain Proxy:
 - Register with domain and receive calls
 - Send outbound via:
 - Domain
 - Proxy Address: 192.168.111.129
- Dial plan: #1\va.T;match=1;prestrip=2;

At the bottom right, there are 'OK' and 'Cancel' buttons.

Figura 5. Configurazione X Lite

Account name: inserire una stringa a piacere

User ID, Display Name e Authorization Name: Il nome (numero) dato alla sezione riguardante l'utente, in questo caso è 1234, quello definito all'interno del file sip.conf.

Domain: L'indirizzo del Server Asterisk, un modo per trovarlo facilmente su Ubuntu è tasto destro su NetworkManager e poi click su Connection information da cui si legge l'IP address della macchina virtuale.

Password: quella decisa nel file sip.conf.

Proxy: nuovamente l'IP già digitato per Domain.

Utenti Linux :

Sia sull'host sia sul guest installare Zoiper : http://www.zoiper.com/download_list.php .

Il softphone Zoiper si avvia cliccando sul file Zoiper scaricato e decompresso, apparirà la finestra come in figura 6.

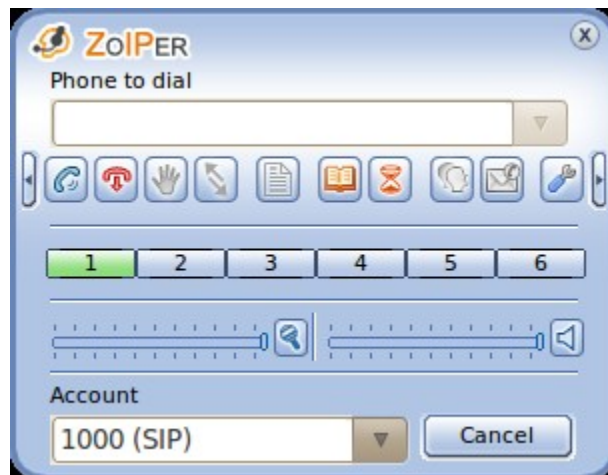


Figura 6. Zoiper

Dopo aver installato Zoiper sia sulla macchina di partenza che sulla macchina virtuale, assegnare due account diversi come da file sip.conf. Cliccando su Options si accede alla finestra delle impostazioni, si scelga Add new SIP account (Figura 7).

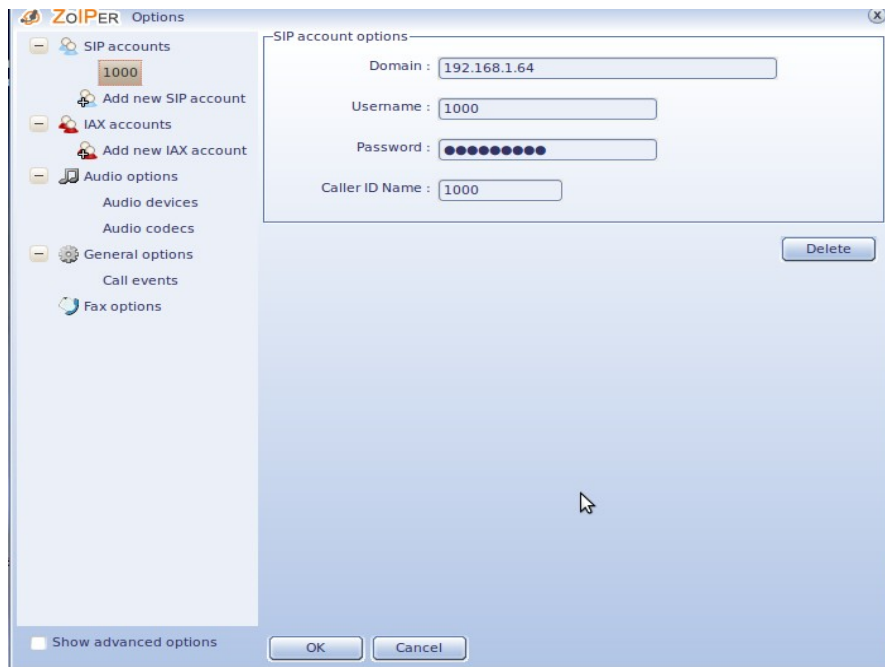


Figura 7 Opzioni SIP.

Username: Il nome (numero) dato alla sezione riguardante l'utente, in questo caso è 1000, quello definito all'interno del file sip.conf.

Domain: L'indirizzo del Server Asterisk, un modo per trovarlo facilmente su ubuntu è tasto destro su NetworkManager e poi click su Connection information da cui si legge l'IP address della macchina virtuale.

Password: quella decisa nel file sip.conf.

Infine cliccare su Register . Eseguendo l'operazione per entrambe le macchine (virtuale e non), siamo ora in grado di chiamare un Softphone con l'altro (Figura 8).

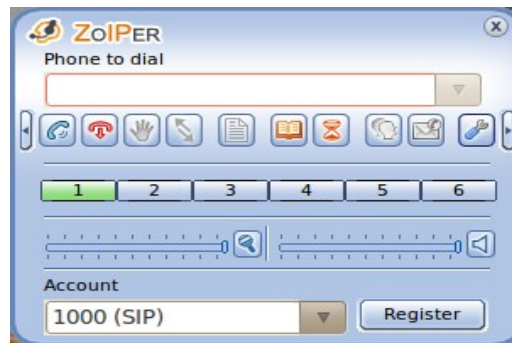


Figura 8 Utilizzo dell'account 1000.

La figura 9 riproduce i due softphone installati uno sulla macchina fisica e l'altro sulla macchina virtuale, insieme al server asterisk, gli indirizzi IP sono puramente indicativi, e non rispecchiano le configurazioni delle macchine durante l'esercitazione.

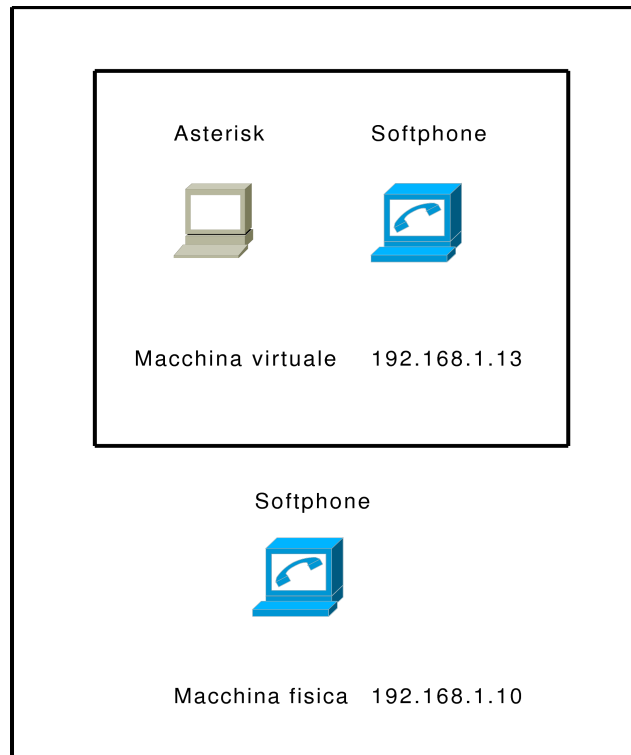


Figura 9. Posizione dei softphone rispetto ad Asterisk.

A questo punto riavviamo asterisk (come super utente) per rendere effettive le modifiche compiute; si usa il comando **core restart now**.

Si può ad ogni modifica del file sip.conf aggiornare Asterisk con il comando **sip reload** e ad ogni modifica del Dialplan aggiornarlo con il comando **dialplan reload**.

Connettiamoci alla console CLI, dove si possono vedere tutte le operazioni che compie asterisk durante le nostre chiamate.

```
>sudo asterisk -vvvr
```

A questo punto con uno dei comandi:

```
sip show users  
sip show peers
```

si vedranno i due telefoni registrati sul server asterisk.

4.4 Dialplan avanzati

I due telefoni registrati possono comunicare tra di loro, questo è possibile perchè abbiamo specificato ad asterisk come gestire le chiamate in ingresso e in uscita dai softphones, avendo creato precedentemente un dialplan. Come primo Dialplan abbiamo permesso semplicemente che i due telefoni possano comunicare tra loro: all'interno del file `/etc/asterisk/extensions.conf`

```
[internal]
exten => 1234,1,Dial(SIP/1234)
exten => 1000,1,Dial(SIP/1000)
```

Il context si chiama `internal` come definito nel file `sip.conf`. La prima riga permette di chiamare l'utente 1234 digitando 1234 sulla tastiera del softphone, la seconda riga permette di chiamare l'utente 1000 digitando 1000, tutto attraverso l'applicazione `Dial`.

Dalla console CLI di Asterisk verifichiamo se riusciamo a raggiungere i softphone mediante il dialplan appena scritto:

```
*CLI> console dial 1000@internal
```

Questo comando permette di eseguire un'estensione all'interno di un contesto, a questo punto il softphone con numero 1000 dovrebbe squillare. Dato che entrambi i softphone sono gestiti dal contesto `internal` è possibile chiamare l'estensione 1000 dall'estensione 1234 e viceversa.

Per rendere più robusto il dialplan possiamo gestire i casi in cui l'utente chiamato non risponda entro un determinato tempo, utilizzando l'applicazione `playback` che permette di rispondere al chiamante con un messaggio vocale:

```
[internal]
exten => 1234,1,Dial(SIP/1234,5)
exten => 1234,2,Playback(vm-nobodyavail)
exten => 100,3,Hangup()

exten => 1000,1,Dial(SIP/1000,5)
exten => 1000,2,Playback(vm-nobodyavail)
exten => 1000,3,Hangup()
```

Il secondo argomento dell'applicazione `Dial` permette di far squillare il telefono per 5 secondi, dopo questo tempo si rinuncia alla chiamata. Le priorità 2 e 3 permettono di gestire il caso in cui l'utente non risponda alla chiamata entro 5 secondi mandando il messaggio `vm-nobodyavail` e di seguito chiudendo la chiamata tramite l'applicazione

hangup.

Se si vogliono utilizzare le voci in italiano presenti nella cartella /usr/share/asterisk/sounds/it/ occorre modificare il file extension.conf utilizzando la funzione Set (CHANNEL(language)=it)

```
[internal]
exten => 1000,1,Set(CHANNEL(language)=it)
exten => 1000,2,Dial(SIP/1000,5)
exten => 1000,3,Playback(vm-nobodyavail)
exten => 1000,4,Hangup()
```

4.5 Asterisk e GoogleTalk

Asterisk include la funzionalità di potersi connettere, come client, al server GoogleTalk e in questo modo ricevere e effettuare chiamate:

- Asterisk-GTalk
- Gtalk-Asterisk

Prima di iniziare si controlla che il modulo res_jabber sia stato installato. Dalla console di Asterisk, si scrive il comando:

```
CLI> help jabber
jabber reload Reload Jabber configuration
jabber set debug {on|off} Enable/Disable Jabber debug
jabber show buddies Show buddy lists of our clients
jabber show connected Show state of clients and components
jabber test Shows roster, but is generally used for mog's
debugging.
```

Se compaiono questi messaggi, il modulo è caricato e funzionante.

Per le procedure spiegate in seguito useremo il ";" per commentare i comandi.

Si crea un nuovo utente nella pagina di google (in questo esempio l'utente creato è marco@gmail.com) .

Per configurare GoogleTalk in Asterisk, bisogna modificare due files:

- jabber.conf
- gtalk.conf

Si comincia con jabber.conf:

```
gedit /etc/asterisk/jabber.conf
```



```

[general]
debug=yes      ; si attiva il debug
autoprune=yes  ; rimuove automaticamente gli utenti che ne fanno richiesta dalla lista
                ; di amici
autoregister=yes ; aggiunge automaticamente nuovi utenti che ne fanno richiesta
                ; alla lista di amici
[gtalk]
                ; etichetta che identifica questa connessione
type=client    ; Asterisk si collegherà come un utente
serverhost=talk.google.com ; nome del server GoogleTalk
username = marco@gmail.com ; utente che registreremo in GoogleTalk
secret = password ; password associata all'utente
port=5222     ; porta di connessione a GoogleTalk
usetls=yes   ; attiva il protocollo TLS
usesasl=yes  ; attiva il protocollo SASL
buddy=amico_1@gmail.com ; a partire da questa linea possiamo definire una lista di
                ; amici predefiniti (uno per linea)
buddy=amico_2@gmail.com
buddy=amico_3@gmail.com
statusmessage=Asterisk Server ; messaggio di stato associato all'utente
timeout=100 ; tempo massimo disponibile per connettersi al server di GoogleTalk

```

È possibile configurare in questo file più di un utente GoogleTalk; per farlo si configura un altro blocco simile al precedente avendo cura di cambiare l'etichetta [gtalk] con un altro nome.

Configurazione di gtalk.conf:

```
gedit /etc/asterisk/gtalk.conf
```

```

[general] ; inizia la parte generale
context=google-in ; contesto dove entreranno le chiamate provenienti da utenti
                ; GoogleTalk
bindaddr=IPasterisk ; indirizzo IP dove si accetteranno le chiamate per questo
                ; canale (IP pubblico di asterisk)
allowguest=yes ; si permettono chiamate da utenti GoogleTalk non presenti
                ; nella lista degli amici
[guest]

```

```

disallow=all
allow=ulaw
context=google-in ; configurazione per le chiamate di utenti GoogleTalk ospiti (non
                  ; presenti nella lista di amici)
[amico_1]          ; in questo blocco si definisce il primo amico inserito in
                  ; jabber.conf con il parametro buddy. Si ripete il blocco per gli
                  ; altri due utenti definiti:
username=amico_1@gmail.com
disallow=all
allow=ulaw
connection=gtalk
context=google-in

[amico_2]
username=amico_2@gmail.com
disallow=all
allow=ulaw
connection=gtalk
context=google-in
[amico_3]
username=amico_3@gmail.com
disallow=all
allow=ulaw
connection=gtalk
context=google-in

```

Si salvano le modifiche.

Se si vuole, si possono differenziare i contesti delle chiamate in entrata provenienti da GoogleTalk a seconda che siano di utenti presenti nella lista di amici o utenti ospiti.

Per farlo in guest si mette un contesto diverso da quello configurato per amici.

Per ogni utente “amico” configurato è stato definito un parametro (connection). Se si osserva bene il suo valore (gtalk) è quello presente in jabber.conf come etichetta di inizio della configurazione dell'utente marco@gmail.com

Terminata la configurazione dei due files, si modifica il dialplan in modo da configurare le chiamate in entrata e in uscita usando il canale Asterisk chan_gtalk:

```
gedit /etc/asterisk/extensions.conf
```

Per le chiamate entranti si configura il contesto google-in (come definito in gtalk.conf) dopo il contesto esterne, e si aggiungono le seguenti linee:

```
[google-in]
exten => s,1,NoOp( Call from Gtalk )
exten => s,n,Dial(SIP/1000,30)
exten => s,n,GotoIf("${DIALSTATUS}" = "BUSY"?busy:unavail)
exten => s,n(unavail),Voicemail(1000@default,u)
exten => s,n,Hangup()
exten => s,n(busy),VoiceMail(1000@default,b)
exten => s,n,Hangup()
```

Tutte le chiamate entranti da GoogleTalk si inoltrano all'interno 1000. Se entro 30 secondi nessuno risponde, la chiamate si trasferiscono alla segreteria telefonica dello stesso interno.

Per le chiamate in uscita (verso i tre utenti-amici configurati in gtalk.conf) nel contesto interne si aggiungono le seguenti linee:

```
exten => 10000,1,Dial(gtalk/gtalk/amico_1@gmail.com)
exten => 10000,n,Hangup
exten => 10001,1,Dial(gtalk/gtalk/amico_2@gmail.com)
exten => 10001,n,Hangup
exten => 10002,1,Dial(gtalk/gtalk/amico_3@gmail.com)
exten => 10002,n,Hangup
```

Digitando il numero 10000 chiameremo all'utente amico_1@gmail.com, digitando il numero 10001 l'utente amico_2@gmail.com e digitando il numero 10002 l'utente amico_3@gmail.com.

Nei parametri di Dial() il primo parametro indica il protocollo usato per effettuare la chiamata, in questo caso il canale chan_gtalk, il secondo parametro indica il nome dell'etichetta definita in jabber.conf per l'utente marco@gmail.com; il terzo l'utente GoogleTalk da chiamare.

Si aggiorna la configurazione di Asterisk, riavviandolo.

Apriamo il client GoogleTalk e vedremo il nuovo utente presente nella lista di amici.