Esercitazione di laboratorio su Asterisk

8 gennaio 2010

Indice

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

2 Installazione

Asterisk viene distribuito in due versioni differenti chiamate Stable (serie 1.2.XX) e Head (serie 1.4.XX). Il ramo Head è usato dagli sviluppatori per testare nuove funzionalità e correggere possibili bug. Per un uso differente da questo è consigliabile l'installazione di una versione del ramo Stable.

2.1 Dipendenze

In una distribuzione linux standard tutte le dipendenze per l'utilizzo di asterisk con telefoni SIP sono già soddisfatte, Esistono comunque delle dipendenze opzionali:

- Zaptel: Contiene i Driver Zapata, necessari per l'utilizzo di telefoni Digium con asterisk.
- Libpri: Libreria utile nel caso si vogliano utilizzare interfacce telefoniche come ISDN.

2.1.1 Zaptel

L'interfaccia Zaptel (http://www.asterisk.org/downloads) è un modulo del kernel che permette la comunicazione tra il driver dell'hardware e il modulo Zapata in asterisk. Grazie a questo concetto è possibile modificare il device driver senza fare nessun cambiamento all'interno dei sorgenti di asterisk. Per installare Zaptel è necessario, nel caso si utilizzi un kernel della serie 2.4, creare un link simbolico alla cartella contentente i sorgenti del kernel [1]

```
ln -s /usr/src/‘uname -r‘ /usr/src/linux
```

a questo punto si può scaricare il pacchetto Zaptel.

Si scompatta il pacchetto scaricato mediante il comando:

```
tar -xzvf zapata-XXX.tar.gz
```

[1] Nel caso si utilizzi un kernel della serie 2.6 non è più necessario

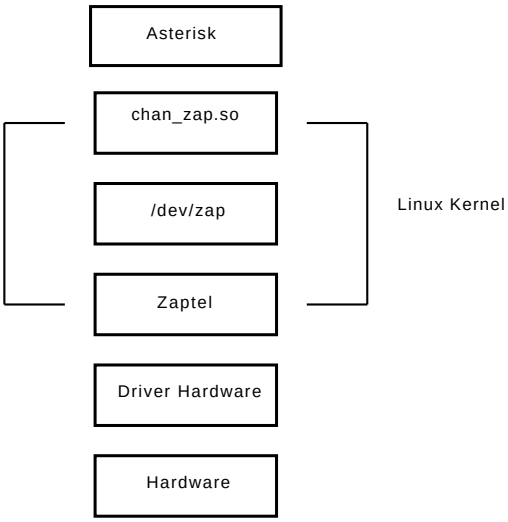


Figura 1: Interazione tra layers e asterisk

e dalla cartella ottenuta si eseguono i comandi di compilazione:

```
make
make install
make config
```

I primi due comandi permettono rispettivamente la compilazione e l'installazione dei drivers. L'ultimo comando consente di installare gli script di startup permettendo che il modulo Zaptel sia caricato all'avvio del sistema (in alternativa si può usare il comando modprobe per il caricamento manuale).

2.1.2 Libpri

L'installazione di Libpri (http://www.asterisk.org/downloads) segue gli stessi passi degli Zapata drivers. I sorgenti si scompattano mediante il comando:

```
tar -xzvf libpri-XXX.tag.gz
```

Con la sequenza di comandi sotto descritta si installa la libreria:

```
make
make install
```

2.2 Asterisk

La distribuzione di Asterisk può essere recuperata all'indirizzo:

http://www.asterisk.org/downloads

Dopo aver scompattato i sorgenti mediante

```
tar -xzvf asterisk-1.2XX.tar.gz
```

si compilano i binari dalla directory ottenuta mediante i comandi:

```
make
make install
make samples
make webvmail
make config
```

- make: consente di compilare il codice sorgente.
- make install: installa asterisk, necessita dei permessi di amministratore.
- make samples: installa nella cartella */etc/asterisk* dei file di configurazione d'esempio.
- webvmail: installa tutti i moduli necessari all'utilizzo della voicemail
- make config: installa gli script di avvio in */etc/init.d* permettendo al server asterisk di andare in esecuzione automaticamente all'avvio del pc. Se la distribuzione linux che si utilizza non fa uso della directory /etc/rc.d/init.d si possono trovare degli script di avvio automatico all'interno della distribuzione di asterisk in *contrib/init.d*

NOTA: esistono già molti pacchetti precompilati per molte distribuzioni Linux.

A questo punto, dopo aver ottenuto i permessi di root, si può avviare Asterik e connettersi alla sua console CLI con il comando:

```
/usr/sbin/asterisk -vvvc
```

Se invece esiste già una istanza di Asterisk in esecuzione, per aprire la console occorre usare il comando:

```
/usr/sbin/asterisk -vvvr
```

Per una dettagliata spiegazione dei comandi di avvio di asterisk si consiglia di digitare

```
/usr/sbin/asterisk -help
```

oppure consultare le pagine del manuale.

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 sip che utilizzano il centralino.
- extensions.conf: questo file di configurazione contiene il dialplan, specifica le operazioni che un utente può compiere dopo essersi registrato con il centralino.
- iax.conf: file di configurazione dove vengono specificate le caratteristiche degli utenti 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 "shortcut" per la definizione contemporanea di uno user e un peer.

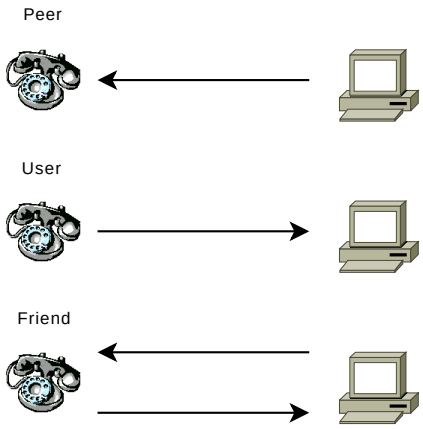


Figura 2: Gli utenti in Asterisk

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. 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 interfaccie 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:

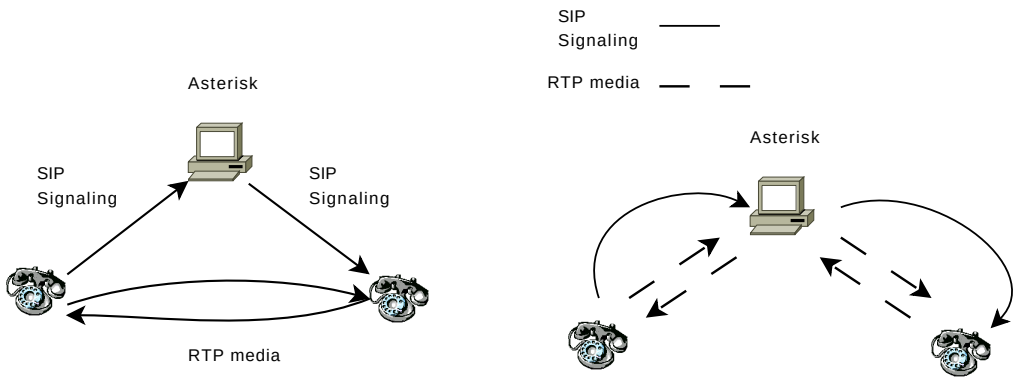
```
[user]     ; Può essere anche un numero.
type=friend
secret=password
qualify=yes
host=dynamic
canreinvite=no
context=internal
```

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 2.

- 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 consideati raggiungibili e qundi 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 reinvito.

  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.



(a) `canreinvite=yes` (b) `canreinvite=no`

Figura 3: L'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 altri contesti, a meno che non sia specificato con il comando include.
Il nome di un contesto viene specificato inserendolo all'interno di parentesi quadre. Esitono 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(VARIABILE)}
  ```

3.3.1 Regole

Le regole che andranno a formare i dialplan di un contesto possono essere pensate come qualsiasi entità raggiungibile mediante il server asterisk (un utente, un altro server asterisk, un numero PSTN ecc.)

Una regola ha la seguente forma:

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

- nome: è l'identificativo di un terminale SIP (stringa o classico numero di telefono).
- priorità: è un numero intero che identifica il "passo" dell'estensione. Ogni estensione 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 dell'estesione 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 l'utente digita il 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. 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 [2]. Allora può essere utile, ogni volta che dalla linea PSTN viene digitato

[2] I numeri cosiddetti "geografici" cioè raggiungibili dal pubblico devo essere acquistati da un operatore telefonico pubblico.

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 inoltri 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/estensione. 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 senza dichiarane 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, sopratutto 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
- [45 – 8]: il match avviene sui numeri 4,5,6,7,8
- . : il pattern matching avvine 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 speficicare 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'estensione chiamata, 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

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:

```
[default]
extensions_number => voicemail_password,username,user_email
```

- 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.

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 di “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()
;telefonare al 500 per vedere i messaggi in casella
```

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.

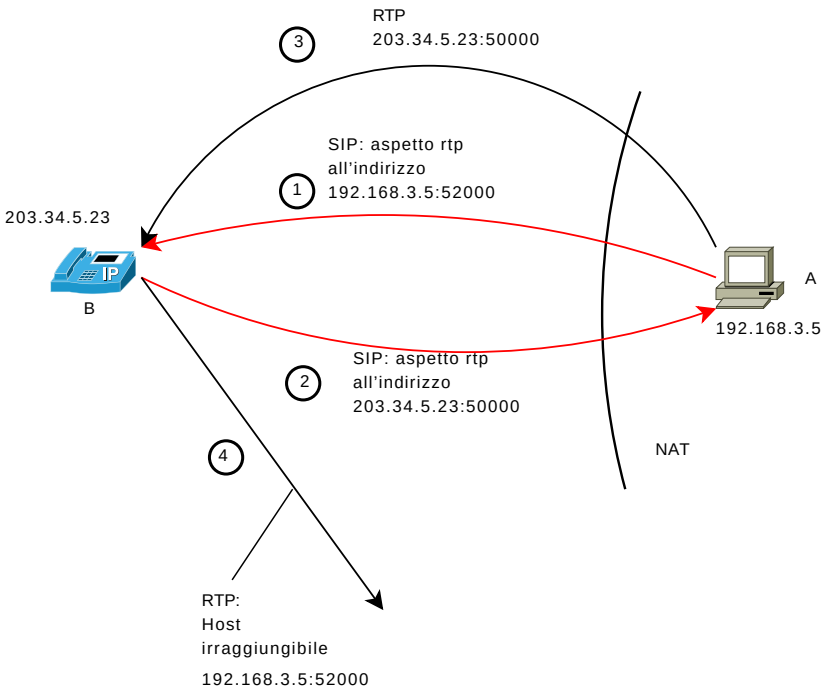


Figura 4: one way audio

Il problema è descritto dalla figura 4: 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 media 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.

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 problma "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 nattato (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.

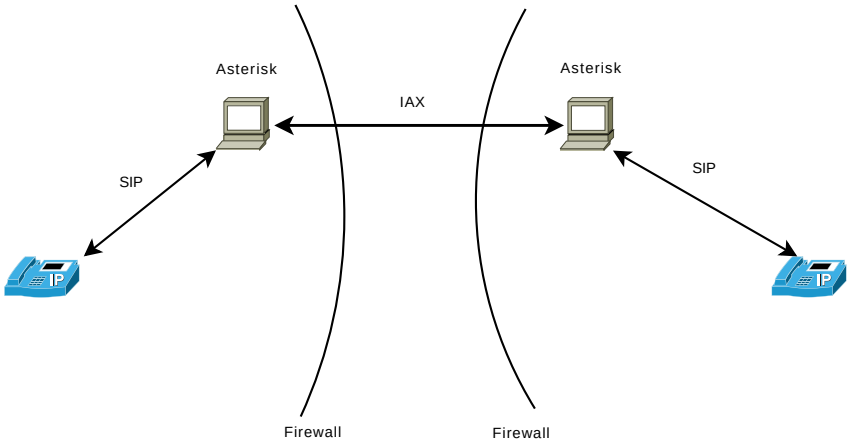


Figura 5: Asterisk in una NAT, peers in un'altra NAT

- **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, cosi 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

Dopo aver chiarito i concetti fondamentali di asterisk passiamo alla parte di esercitazione.

Useremo X-Lite, un softphone da configurare con asterisk.

4.1 Configurazione softphone

L'ultima versione del softphone X-Lite è scaricabile dall'indirizzo:

http://www.counterpath.com/xlitedownload.html.

Configurazione su Windows Una volta conclusa l'installazione si passa alla fase di configurazione:

- click bottone di configurazione e cliccare sulla voce "sip account settings"

- quindi fare click su add per aggiungere i dati relativi ad un nuovo account

- impostare i dati in base all'utente:

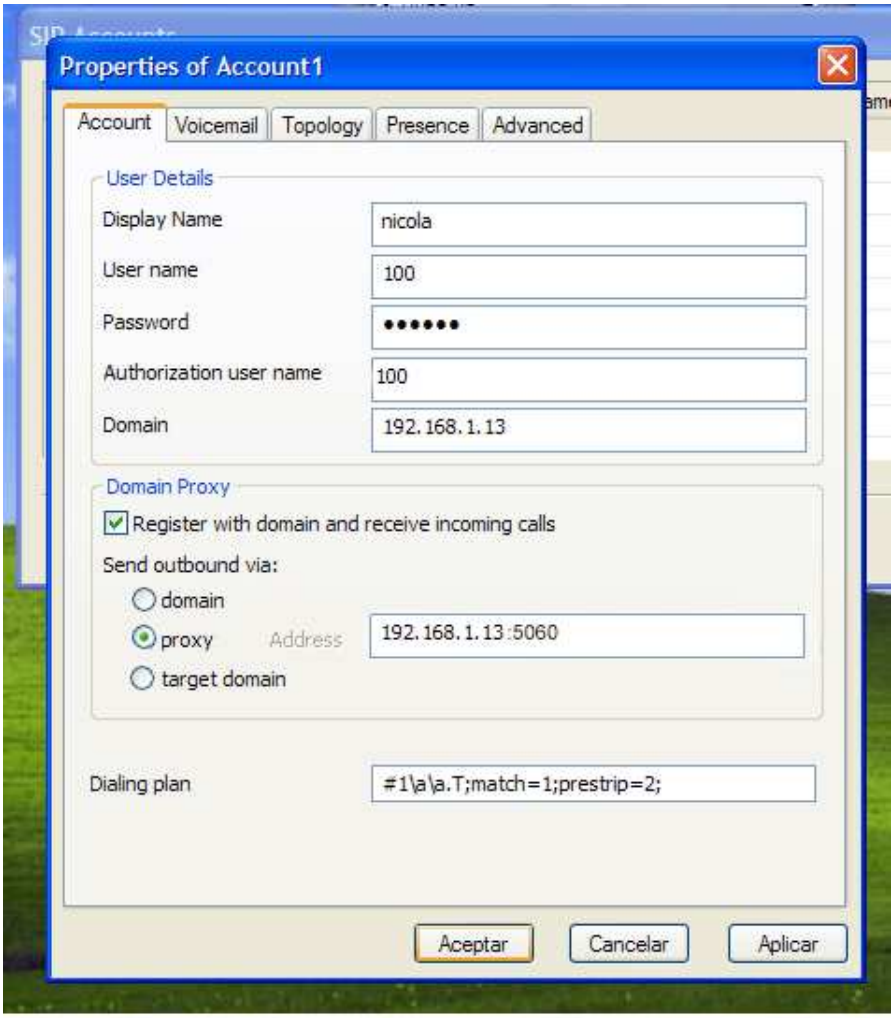


Figura 6: Configurazione X-Lite su windows

  - Dislplay name: il nome visualizzato sul display del telefono
  - Username: il nome (numero) dato alla sezione riguardante l'utente, in questo caso 100.
  - Password: la password definita all'interno del file sip.conf
  - Authorization user name: lo username definito all'interno del file sip.conf, usato per l'autenticazione, in questo caso 100.
  - Domain: l'indirizzo del server asterisk, sulla quale si registrerà il softphone.

– SIP Proxy: Proxy sul quale si registrerà il softphone: 192.168.1.13:5060 è necessario specificare anche la porta che userà il softphone per la registrazione.

I restanti parametri sono opzionali e possono essere tralasciati.

Conclusi questi passi ricordiamo di accettare le midifiche e spuntare la casella relativa a questo account per poterlo utilizzare. Riavviamo X-Lite, a questo punto, se tutto è andato a buon fine dovremmo poter vedere sullo schermo del softphone la scritta "ready":

Per vericare l'avvenuta registrazione su asterisk possiamo digitare dalla console CLI del centralino:

```
localhost*CLI> sip show users
```

Ottenendo la lista degli utenti registrati sul centralino.

Configurazione su Linux La versione di X-Lite per linux è più spartana, ma si configura con le stesse opzioni della versione per windows. Per poter avviare X-Lite su linux è necessario installare la libreria libstdc++ installabile dal gestore di pacchetti della distribuzione linux.

Non c'è bisogno di alcuna installazione basta eseguire il file binario:

```
./xtensoftphone
```

Cliccando sul bottone sulla sinistra del tasto verde si apre il menù di configurazione, scegliere quindi "System settings" e successivamente "SIP proxy". Doppio click su "Default" , click su "Enable" cosi da poter utilizzare il profilo e di seguito modificare le voci:

- Display Name: Il nome visualizzato sul display, nell'esempio Andrea.
- Username: Il nome dato alla sezione dell'utente andrea all'interno del file sip.conf (nel nostro caso il numero) : 101
- Authorizationo User: Lo username specificato nel file sip.conf, nome con cui il softphone si autenticherà sul centralino: 101.
- Password: la password dell'utente.
- Domain/Realm: l'indirizzo del server asterisk : 192.168.1.13
- SIP Proxy: Proxy sul quale si registrerà il softphone, nel nostro caso sempre asterisk: 192.168.1.13:5060 è necessario specificare anche la porta che userà il softphone per la registrazione.

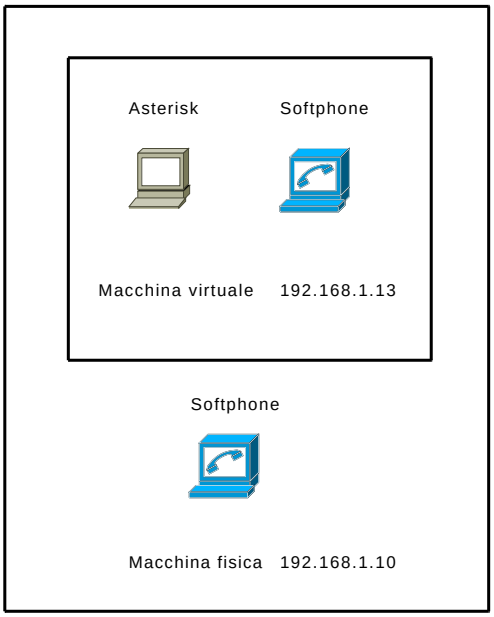


Figura 7: Indirizzi utilizzati

4.2 Esempi di chiamate

Negli esempi che seguono si tenterà di creare un sistema telefonico minimale tra due softphone e asterisk utilizzando il protocollo SIP come protocollo di segnalazione.

I due softphone saranno installati uno sulla macchina windows e l'altro sulla macchina virtuale, insieme al server asterisk. Gli indirizzi utilizzati sono in figura 7.

Rinominare i file *sip.conf, extensions.conf* e *iax.conf* installati automaticamente da asterisk nella cartella /etc/asterisk.

4.2.1 Creazione degli utenti

Come primo passo si devono creare i due utenti, uno per ogni softphone, all'interno dei file di configurazione di asterisk in base a quanto indicato nella sezione 3.2.

Editare quindi */etc/asterisk/sip.conf*. Riportare la sezione general come definito nella sezione 3.2

```
[100]                   ; Il numero telefonico dell'utente
type=friend             ; L'utente nicola può fare e ricevere chiamate
username=100            ; Il nome utente con il quale nicola si registrerà
secret=password         ; La password dell'utente
host=dynamic            ; L'ip di nicola sarà dinamico
context=internal        ; Il context che gestisce nicola in extensions.conf
```

Creare l'utente Andrea, che risponde all'estensione 101, gestito dal contesto internal.

Per una più precisa definizione dei parametri si vedano le sezioni 2 e 3.2.

Riavviamo asterisk (come super utente) per rendere effettive le modifiche compiute:

```
/etc/rc.d/init.d/asterisk restart
```

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

```
/usr/sbin/asterisk -vvvr
```

L'opzione "vvv" specificata da riga di comando imposta la verbosità della console a livello 3, la r permette di connettersi ad una istanza di asterisk già in esecuzione. A questo punto con il comando:

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

si dovrebbero vedere i due telefoni registrati sul server asterisk.

4.2.2 Dialplan

I due telefoni registrati non possono ancora comunicare tra di loro, si deve specificare ad asterisk come gestire le chiamate in ingresso e in uscita dai softphone, si deve quindi creare un dialplan. Come primo Dialplan permettiamo semplicemente che i due telefoni possano comunicare tra loro: all'interno del file */etc/asterisk/extensions.conf*

```
[internal]
exten => 100,1,Dial(SIP/100)
exten => 101,1,Dial(SIP/101)
```

Il context si chiama internal come definito nel file sip.conf. La prima riga permette di chiamare Nicola digitando 100 sulla tastiera del softphone, la seconda riga permette di chiamare Andrea digitando 101, tutto attraverso l'applicazione Dial. Dalla console CLI di asterisk verifichiamo se riusciamo a raggiungere i softphone mediante il dialplan appena scritto:

```
localhost*CLI> dial 100@internal
```

permette di eseguire un'estensione all'interno di un contesto, a questo punto il softphone con numero 100 dovrebbe squillare. Dato che entrambi i softphone sono gestiti dal contesto internal è possibile chiamare l'estensione 100 dall'estensione 101 e viceversa.

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

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

exten => 101,1,Dial(SIP/101,5)
exten => 101,2,Playback(vm-nobodyavail)
exten => 101,3,Hangup()
```

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

Asterisk può prendere decisioni sulla modalità di gestione delle chiamate in base a parametri esterni indipendenti dallo stato degli utenti, come ad esempio l'orario: Supponiamo di voler permettere che sia possibile effettuare chiamate solo in un determinato periodo del giorno, dovremmo usare l'applicazione GotoIfTime() che permette di compiere decisioni in base all'orario in cui avviene la chiamata.

```
[internal]
exten => 100,1,GotoIfTime(09:00-18.00,mon-fri,*,*?4)
exten => 100,2,Playback(vm-isunvail)
exten => 100,3,Hangup()
exten => 100,4,Dial(SIP/100,5)
exten => 100,5,Playback(vm-nobodyavail)
exten => 100,6,Hangup()

exten => 101,1,GotoIfTime(09:00-18.00,mon-fri,*,*?4)
exten => 101,2,Playback(vm-isunvail)
exten => 101,3,Hangup()
exten => 101,4,Dial(SIP/101,5)
exten => 101,5,Playback(vm-nobodyavail)
exten => 101,6,Hangup()
```

La prima riga del contesto permette di fare il check sul tempo:

- 09.00-18.00 è l'orario in cui sono consentite le chiamate
- il secondo argomento rappresentano i giorni della settimana, dal lunedì al venerdi
- il terzo argomento rappresentano i giorni del mese (in questo caso qualsiasi giorno del mese)
- l'ultimo asterisco rappresenta i mesi (in questo caso qualsiasi mese dell'anno).

Dopo il punto di domanda possono essere specificate, opzionalmente, l'estensione, il contesto e la priorità che viene eseguita se il check è andato a buon fine (in questo caso solo la priorità). Nel caso in cui la chiamata sia effettuata in un orario differente da quello specificato allora viene eseguita l'estensione con priorità 2 altrimenti si salta alla 4. Tutte le applicazioni Goto permettono di compiere salti all'estensione desiderata all'interno del dialplan rompendo cosi il flusso sequenziale nell'esecuzione delle estensioni. Per maggiori informazioni riguardo le applicazioni Goto si rimanda alla documentazione ufficiale.

4.2.3 Connessione tra due server asterisk

E' possibile connettere due o più server asterisk permettendo che le estensioni di un server siano raggiungibili dall'altro e viceversa.
L'approccio più semplice è quello di utilizzare il protocollo IAX (Inter Asterisk eXchange).
E' necessario configurare entrambi i server asterisk come "friend" nei file di configurazione dell'altro, in maniera tale che entrambi possano fare e ricevere chiamate.

Modifichiamo, su entrambi i server, il file */etc/asterisk/iax.conf* dove vengono specificati i parametri per il protocollo IAX:

Nella configurazione del server A

```
[general]
register => serverA:pass@192.168.1.20  ; registra il server A presso il server B

[serverB]
type = friend        ; server B può fare e ricevere chiamate presso A
auth = md5           ; il tipo di autenticazione
username = serverB   ; username usato per l'autenticazione
host = dynamic       ; l'ip del server B lo impostiamo a dinamico
secret = pass        ; la password per l'autenticazione
context = incoming   ; il context che gestirà il server B
                     ; all'interno di extensions.conf
```

```
[serverA-outbound]
type = friend        ; server A può fare e ricevere chiamate presso B
auth = md5           ; il tipo di autenticazione
username = serverA   ; username usato per l'autenticazione
host = 192.168.1.20  ; ip del server B dove si intende chiamare
secret = pass        ; la password per l'autenticazione
context = incoming   ; il context che gestirà il server A
                     ; all'interno di extensions.conf
```

La sezione serverB contiene i dati di autenticazione per la registrazione del server B presso il server A. La sezione serverA-outbound contiene i dati per l'autenticazione attraverso l'applicazione Dial all'interno del dialplan, necessaria per chiamare estensioni esterne.

Nella configurazione del server B

```
[general]
register => serverB:pass@192.168.1.13 ; registra il server B presso il server A

[serverA]
type = friend        ; server A può fare e ricevere chiamate presso B
username = serverA   ; username usato per l'autenticazione
auth = md5           ; il tipo di autenticazione
host = dynamic       ; l'ip del server A lo impostiamo a dinamico
secret = pass        ; la password per l'autenticazione
context = incoming   ; il context che gestirà il server A
                     ; all'interno di extensions.conf


[serverB-outbound]
type = friend        ; server B può fare e ricevere chiamate presso A
auth = md5           ; il tipo di autenticazione
username = serverB   ; username usato per l'autenticazione
host = 192.168.1.13  ; ip del server A dove si intende chiamare
secret = pass        ; la password per l'autenticazione
context = incoming   ; il context che gestirà il server B
                     ; all'interno di extensions.conf
```

La sezione serverA contiene i dati di autenticazione per la registrazione del server A presso il server B. La sezione serverB-outbound contiene i dati per l'autenticazione attraverso l'applicazione Dial all'interno del dialplan, necessaria per chiamare estensioni esterne.

A questo punto, dopo aver riavviato i server asterisk dovremmo poter vedere dalla console CLI:

```
Registered IAX2 'serverB' (AUTHENTICATED) at 192.168.1.13:4569
Registered IAX2 to '192.168.1.13', who sees us as 192.168.1.20:4569
```

questo ci assicura che il serverA ha registrato il serverB e che il serverA è registrato presso B. Per un'ulteriore conferma si può digitare dalla console CLI:

```
localhost*CLI> iax2 show registry
```

Il quale dovrebbe restituirci come stato "Registered". Se cosi non fosse verificare che le impostazioni in iax.conf siano corrette.

NB: da ora in avanti verrà considerata solo la configurazione del server A, il server B segue la stessa configurazione.

Ora non resta che modificare il dialplan: dobbiamo poter riconoscere quando un utente intende chiamare un'estensione dell'altro server, quindi supponiamo di anteporre uno zero ogni volta che si intende chiamare un estensione "esterna". Inseriamo un altro contesto all'interno del Dialplan che ci permetta di chiamare il server esterno:

```
[external]
exten => 0100,1,Dial(IAX2/serverA-outbound/100)
exten => 0101,1,Dial(IAX2/serverA-outboud/101)
```

La sintassi per il comando dial in questo caso cambia:

- Il protocollo utilizzato è IAX2.
- Il secondo argomento ci permette di identificarci per la chiamata presso il server B, serverA-outbound è la sezione in iax.conf che contiene i dati di autenticazine per le chiamate esterne.
- Il terzo argomento è l'estensione da chiamare.

I telefoni sono però gestiti dall'estensione internal, per ora quindi non possiamo ancora digitare 0100 e chiamare l'estensione 100 del server B. Dobbiamo aggiungere all'estensione internal l'estensione external dando la possibilità ai telefoni di effettuare anche chiamate esterne aggiungiamo quindi alla fine del contesto internal il comando:

```
include => external
```

Il comando include include il contesto external nel contesto internal, ora i telefoni hanno accesso anche alle estensioni del contesto external.

In tutti i Dialplan, per una questione di sicurezza [3], è buona regola mantenere separati i contesti che permettono di eseguire chiamate da quelli che permettono di riceverle, quindi inseriamo un contesto [incoming] che consente di ricevere chiamate dall'esterno:

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

exten => 101,1,Dial(SIP/101,5)
exten => 101,2,Playback(vm-nobodyavail)
exten => 101,3,Hangup()
```

[3] Si legga il file SECURITY all'interno della cartella dei sorgenti di asterisk

Ora si è in grado di fare chiamate al server B sul quale siamo registrati mediante il contesto external (incluso in internal) e ricevere chiamate dal server B che si è registrato presso A e gestito tramite il contesto incoming.

5 Esercizi

1. Riscrivere i contesti specificati usando il pattern matching.
2. Organizzare i contesti internal ed external con l'applicazione Macro.
3. Aggiungere la funzionalità di voicemail ad ogni utente registrato con asterisk
4. Organizzare i contesti creati per la gestione della voicemail con l'applicazione Macro.