

3. REALIZZAZIONE DEL TEST SUL SERVER ASTERISK

In questo capitolo viene illustrata la realizzazione del test sul PBX Asterisk.

Inizialmente, come prova, è stata installata una macchina virtuale VmWare sul pc portatile utilizzato per stilare questa tesi; sulla macchina virtuale è stato installato il sistema operativo Linux CentOS 2.4 kernel 2.6.9-34.EL e quindi i pacchetti di Asterisk.

Per avviare una semplice comunicazione tra il server Asterisk ed un computer, sfruttando semplicemente la rete locale, senza l'installazione di alcuna scheda ISDN, è stato installato il softphone Siphone, sia sul portatile che ospita la macchina virtuale che su altri Pc collegati tramite rete locale.

A questo punto, modificando il dialplan, è stata resa possibile la comunicazione tra i due softphone, sfruttando la connessione IP tra i due computer.

L'operazione è stata ripetuta sia sfruttando il protocollo SIP, che quello H.323.

Successivamente è stata effettuata una videochiamata via SIP col softphone Kapanga.

A questo punto è stato installato Asterisk sul server del laboratorio LIT della facoltà, e ripetuti gli stessi passaggi, collegando anche un telefono IP alla rete Ethernet del laboratorio.

» **3.1 Installazione di Asterisk su Vmware**

Collegandosi al sito <http://www.vmware.com/download/> è possibile, previa registrazione, ottenere il download di una versione trial della macchina virtuale Vmware, sulla quale installeremo il sistema operativo Linux necessario per far girare il server Asterisk.

Una volta ultimata la procedura d'installazione del sistema operativo, è possibile installare Asterisk.



Fig. 22 – schermata iniziale della macchina virtuale Centos

Nei paragrafi seguenti, partendo innanzitutto dai requisiti necessari, si illustrerà la procedura seguita per la compilazione e l'installazione del gateway Asterisk.

» **3.1.1 Requisiti necessari**

I tre 'pacchetti' principali usati da Asterisk sono:

- Il programma Asterisk principale (*asterisk*);
- I driver di Zapata Telephony (*zaptel*);
- Le librerie PRI (*libpri*).

Nella fattispecie sono state utilizzate la versione 1.2.12 del pacchetto asterisk, la versione 1.2.9.1 del pacchetto zaptel e quella 1.2.3 del pacchetto libri: tutti i pacchetti sono compressi (estensione .tar.gz) scaricabili dall'home page di www.asterisk.org, oppure da <http://ftp.digium.com/pub/asterisk>.

Se occorre lavorare esclusivamente su rete VoIP, l'unico pacchetto di cui si ha effettivamente bisogno è il primo; i driver zaptel sono richiesti se si deve utilizzare hardware analogico o digitale. La libreria libri è opzionale, a meno che si voglia utilizzare delle interfacce ISDN PRI e si vuole preservare il quantitativo di RAM non utilizzata.

Un altro pacchetto che si può utilizzare è *asterisk-sounds*, che aggiunge altri suoni oltre ai numerosi di default presenti nella distribuzione asterisk principale.

Prima di procedere alla compilazione di Asterisk occorre installare (se già non è presente) il *compilatore GCC* (consigliata la versione 3.x e successive), e inoltre i pacchetti bison and bison-devel, ncurses and ncurses-devel, zlib and zlib-devel, openssl and openssl-devel.

» **3.1.2 Download, compilazione, avvio**

Una volta installato Linux, eseguito il riavvio del sistema e fatto il login, si è aperto il terminale ed eseguito le seguenti istruzioni:

- ➔ `su -` # se non si è fatto il login come root, occorre lanciare una shell per il superutente:
- ➔ `cd /usr/src/` # entra nella directory /usr/src;
- ➔ `mkdir asterisk` # crea una cartella di nome asterisk;
- ➔ `wget`
`http://ftp.digium.com/pub/asterisk/releases/asterisk-`

- 1.2.12.1.tar.gz # fa il download dell'ultima realease del pacchetto asterisk
- tar -zxf asterisk-1.2.12.1.tar.gz # decomprime il pacchetto
- mv asterisk-1.2.12.1 asterisk # rinomina la cartella
- wget
http://ftp.digium.com/pub/zaptel/releases/zaptel-1.2.9.1.tar.gz # fa il download dell'ultima realease del pacchetto zaptel
- tar -zxf zaptel-1.2.9.1.tar.gz # decomprime il pacchetto
- mv zaptel-1.2.9.1 zaptel # rinomina la cartella
- wget
http://ftp.digium.com/pub/libpri/releases/libpri-1.2.3.tar.gz # fa il download dell'ultima realease del pacchetto libpri
- tar -zxf libpri-1.2.3.tar.gz # decomprime il pacchetto
- mv zaptel-1.2.9.1 zaptel # rinomina la cartella

A questo punto si procede con la compilazione, procedendo per ordine con l'installazione del pacchetto zaptel, poi libpri, ed infine asterisk:

- cd zaptel # entra nella cartella creata dopo la decompressione
- make clean # cancellazione delle precedenti installazioni
- make linux26 # occorre specificarlo, poiché su CentOS 4.x si usa il kernel-2.6
- make install # effettua l'installazione

Se si utilizza (come nel nostro caso) il kernel 2.6.9-34.EL, e la compilazione viene effettuata con errori, occorre modificare il file `spinlock.h`, contenuto in

`/usr/src/kernels/2.6.9-34.EL-x86_64/include/linux/`

occorre cercare la riga

```
#define DEFINE_RWLOCK(x) rw_lock_t x = RW_LOCK_UNLOCKED
```

e sostituirla con

```
#define DEFINE_RWLOCK(x) rwlock_t x = RW_LOCK_UNLOCKED
```

e riprocedere con la compilazione di `zaptel`.

Procedura simile con il pacchetto `libpri`:

- `cd..`
- `cd libri # entra nella cartella libri creata dopo la decompressione`
- `make clean # cancellazione delle precedenti installazioni`
- `make`
- `make install # effettua l'installazione`

e infine con `asterisk`:

- `cd..`
- `cd asterisk # entra nella cartella asterisk creata dopo la decompressione`
- `make clean # cancellazione delle precedenti installazioni`
- `make`
- `make install # effettua l'installazione`
- `make samples # installa gli esempi`

Al termine della procedura appare questa schermata, a indicare la corretta installazione:

```
+----- Asterisk Installation Complete -----+
+
+   YOU MUST READ THE SECURITY DOCUMENT   +
+
+ Asterisk has successfully been installed. +
+ If you would like to install the sample +
+ configuration files (overwriting any    +
+ existing config files), run:           +
+
+               make samples              +
+
+----- or -----+
+
+ You can go ahead and install the asterisk +
+ program documentation now or later run:  +
+
+               make progdocs             +
+
+ **Note** This requires that you have   +
+ doxygen installed on your local system  +
+-----+

```

A questo punto è possibile procedere con l'avvio di Asterisk:

- ➔ `cd /usr/sbin/ # accesso alla directory dove vi sono i file binari di Asterisk`
- ➔ `asterisk -vvvgc`

A tal proposito, occorre dare il significato di alcune tra le opzioni più importanti dell'ultima istruzione:

- `c` : **Console**. Permette la connessione al CLI (Command Line Interface);
- `v` : **Verbosità**. Usato per specificare l'ammontare delle uscite per il debug del CLI;
- `g` : **Core dump**. Utilizzato in caso di crash inaspettati di Asterisk, porta alla creazione di file che saranno ispezionati con `gdb`;

- `r` : **Remote**. Usato per connettersi in remoto ad un altro processo Asterisk in esecuzione;
- `rx` : **Restart now**. Usato in combinazione con `-r` permette di digitare ed eseguire un comando da console senza essere connessi ad essa.

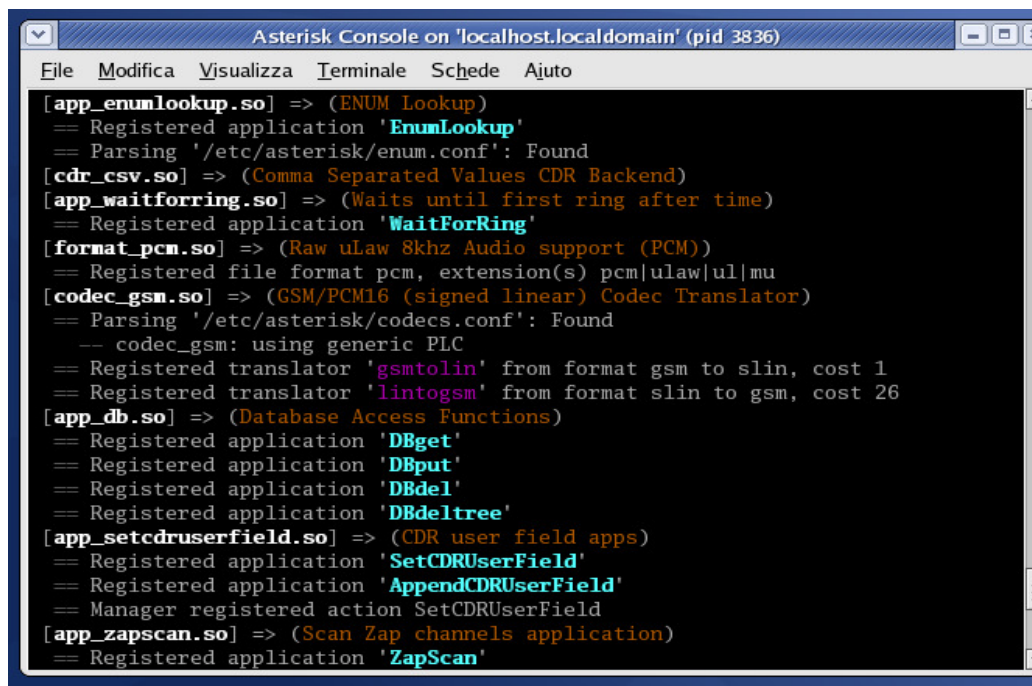
Dunque noi con

→ `asterisk -vvvgc`

abbiamo imposto il livello di verbosità a 3, e imposto il rilascio del file del core in caso di crash del sistema.

Per fermare immediatamente l'esecuzione di Asterisk, basta digitare da console la seguente riga:

→ `stop now`



```
Asterisk Console on 'localhost.localdomain' (pid 3836)
File Modifica Visualizza Terminale Schede Ajuto
[app_enumlookup.so] => (ENUM Lookup)
= Registered application 'EnumLookup'
= Parsing '/etc/asterisk/enum.conf': Found
[cdr_csv.so] => (Comma Separated Values CDR Backend)
[app_waitforring.so] => (Waits until first ring after time)
= Registered application 'WaitForRing'
[format_pcm.so] => (Raw uLaw 8khz Audio support (PCM))
= Registered file format pcm, extension(s) pcm|ulaw|ul|mu
[codec_gsm.so] => (GSM/PCM16 (signed linear) Codec Translator)
= Parsing '/etc/asterisk/codecs.conf': Found
-- codec_gsm: using generic PLC
= Registered translator 'gsmtoalin' from format gsm to slin, cost 1
= Registered translator 'lintogsm' from format slin to gsm, cost 26
[app_db.so] => (Database Access Functions)
= Registered application 'DBget'
= Registered application 'DBput'
= Registered application 'DBdel'
= Registered application 'DBdeltree'
[app_setcdruserfield.so] => (CDR user field apps)
= Registered application 'SetCDRUserField'
= Registered application 'AppendCDRUserField'
= Manager registered action SetCDRUserField
[app_zapscan.so] => (Scan Zap channels application)
= Registered application 'ZapScan'
```

Fig. 23 – console di Asterisk

» **3.2 Sjphone**

Il softphone Sjphone è un programma software che consente di effettuare e ricevere chiamate VoIP basate sul protocollo SIP. E' disponibile gratuitamente sul sito del produttore SJ Labs Inc. (<http://www.sjlabs.com>) ed una volta installato si presenta con una interfaccia grafica simile a quella di un telefono tradizionale.



Fig. 24 – interfaccia di Sjphone

Ecco in dettaglio le sue caratteristiche:

SIP and H.323 signalling support

PC Platforms: MS Windows 98SE/ME/2K/XP/2003 or Linux, Mac OS X

PDA Platforms: Intel xScale PDA with PocketPC 2002/2003/2003SE, Windows CE.NET 4.0/4.2/5.0, or Windows Mobile 5.0 Pocket PC. Softphone optimized for Intel XScale® CPU

Proved interoperability with most VOIP services, IP-PBXs, softphones and Gateways

Advanced multi-media softphone engine provides superb audio quality

Multiple service profiles support
Minimum codec set {G.711 A/u; GSM, Speex, iLBC}
Codecs of choice G.729 and more
Upgrade notification
Integration with most USB phones
Changeable skins
CallTo and SIP URL support
Automatic user registration
Dynamic Provisioning at login and On-the-Fly
Configuration
Remote management and support
Account balance display
Localization
Telefonia
Telephony Feature Set {DND, Mute, Flash, Redial, Three-way calls}
Business Feature Set {Forward, Transfer, Hold, MWI, Voice Mail, Ad-Hoc conferencing}
Push-to-Talk (coming soon)
Call logs
Contact lists
Hot keys and Speed-dial
Changeable ring-tones and dial-tones
Interoperabilità
SIP support
H.323 support
Proven interoperability with most softswitches, IP-PBXs, IP-Centrex
Proven interoperability with VoIP gateways and ATA boxes
Multiple DTMF options: inband, RFC2833, SIP INFO, H.245, Q931
Automatic NAT detection and traversal
Security
Symmetric and Asymmetric RTP stream

Integrazione con altre applicazioni
ActiveX webdialer for calls on webpage
IM&P integration
Neighbourhood discovery
Outlook/Outlook Express integration
ILS (Internet Locator Server) directory support
Directory integration
Softphone SDK
Major USB handsets and headsets support
Bluetooth headsets support
Key mapping for PDAs
Porting to new platform

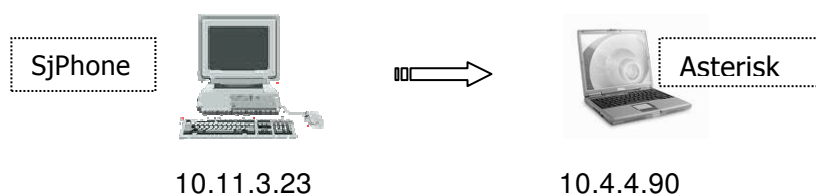
Audio e Video
Echo Cancellor (AEC)
VAD
Adaptive Jitter Buffer for better sound
Audio codecs G711A, G711u, GSM 6.10, G723.1, iLBC,
Speex (G729, G729a)
Video codecs H.261, H263 (in preparazione)
Codec preference management
Video conference
Video mute
Frame rate and Size (CIF/QCIF) tune (in preparazione)

In particolare il software Sjphone è considerato un ottimo strumento di telelavoro, consentendo di effettuare e ricevere chiamate da sedi remote, persino da casa, con estrema semplicità. In questo modo le aziende possono permettere ad esempio al lavoratore di svolgere i propri compiti direttamente da casa, oppure possono godere dei benefici della tecnologia VoIP anche da sedi remote dove non è presente una infrastruttura di rete completa, con innumerevoli vantaggi in termini di risparmio di risorse e di abbattimento dei costi.

» **3.3 Telefonata attraverso il protocollo SIP**

→ Chiamata da Siphone verso Asterisk

Per poter effettuare la chiamata dal softphone verso il PBX Asterisk, sfruttando SIP come protocollo di comunicazione, occorre innanzitutto configurare il dialplan di Asterisk: nei paragrafi che seguono si userà come esempio 10.11.4.90 pari all'IP di Asterisk installato sulla macchina virtuale e 10.11.3.23 pari a quella del pc con installato SjPhone, da dove partirà la chiamata.



Partiamo dalla modifica del file sip.conf in modo da effettuare una nuova registrazione di utente e quindi accettare chiamate in ingresso:

```
[gallo] ; abbiamo creato una nuova registrazione di utente  
        chiamato 'gallo'
```

```
type=friend ; questo significa che l'utente può fare e ricevere  
             chiamate (per le sole chiamate in uscita si usa type=user, per  
             le sole chiamate in ingresso si usa type=peer)
```

```
host=dynamic ; specifica che il tipo di IP sarà dinamico.
```

All'occorrenza è possibile configurare anche un IP statico.

```
dtmfmode=rfc2833 ; Il Dual-tone multi-frequency (in sigla DTMF),  
detto anche multifrequenza, è un sistema per codificare codici  
numerici sotto forma di segnali sonori in banda audio (usando  
'inband') o fuori banda..
```

```
username=gallo ;
```

```
secret=gallopass ; password per l'utente gallo, necessaria per  
l'autenticazione in Asterisk
```

```
context=tesi ; Serve per definire il contesto per l'utente (nella  
fattispecie l'abbiamo chiamata tesi)
```

A questo punto passiamo al file extensions.conf, dove creeremo il contesto 'tesi' indicando il numero associato all'utente 'gallo':

```
[tesi]  
exten => 121,1,Dial(SIP/gallo)  
exten => gallo,1,goto(121,1) ; In questo modo sarà  
                             possibile chiamare anche digitando semplicemente il testo  
                             'gallo'  
  
include => demo
```

Ovviamente si può ripetere la procedura per ogni nuovo utente che si vuole registrare, e per controllare direttamente da console il dialplan appena configurato si può digitare 'show dialplan'.

Vediamo ora come configurare SJPhone (la versione utilizzata è per Linux, in inglese, un po' più spartana rispetto alla corrispondente versione per Windows) :

1. cliccare su Phone – preferences , poi aprire la linguetta 'Profiles' e quindi 'New'
2. creare un nuovo profilo chiamato 'asterisk', facendo attenzione a impostare come tipo di chiamata 'Calls through SIP proxy'
3. nella scheda 'SIP Proxy' inseriamo l'IP del server Asterisk, indicando come porta la 5060.

Per il resto si possono lasciare le impostazioni di default.

Cliccando sul bottone 'initialize', verranno richiesti account e password, e si dovrà inserire 'gallo' e 'gallopass', rispettivamente.

Facendo ciò appare sulla console di Asterisk la seguente istruzione:

```
Registered SIP 'gallo' at 10.11.3.23 port 5060 expires 120
```

a confermare l'avvenuta registrazione del nuovo utente Sjphone presso Asterisk.

A questo punto può essere effettuata la chiamata da Sjphone verso il pc con Asterisk, in uno dei seguenti modi:

- inserendo l'indirizzo IP (nel nostro caso 10.11.4.90);
- digitando protocollo:indirizzo IP (nel nostro esempio SIP:10.11.4.90);
- digitando numero@indirizzo IP (121@10.11.4.90);
- scrivendo protocollo:numero@indirizzo IP (SIP:121@10.11.4.90);
- scrivendo numero@indirizzo IP;user=username (121@10.11.4.90;user=gallo);

- scrivendo protocollo:numero@indirizzo IP;user=username
(SIP:121@10.11.4.90;user=gallo);

Ma soprattutto è possibile chiamare, dato che abbiamo specificato nella configurazione di Siphone l'indirizzo IP del server Asterisk, digitando semplicemente il nome utente ('gallo', nel nostro caso) o il numero associato 121.

Nella console di Asterisk appaiono le seguenti istruzioni:

```
-- Executing Wait("SIP/10.11.3.23-08ec5618", "1") in new stack
-- Executing Answer("SIP/10.11.3.23-08ec5618", "") in new stack
-- Executing Set("SIP/10.11.3.23-08ec5618", "TIMEOUT(digit)=5") in new stack
-- Digit timeout set to 5
-- Executing Set("SIP/10.11.3.23-08ec5618", "TIMEOUT(response)=10") in new s
tack
-- Response timeout set to 10
-- Executing BackGround("SIP/10.11.3.23-08ec5618", "demo-congrats") in new s
tack
-- Playing 'demo-congrats' (language 'en')
```

a dimostrare che la chiamata è arrivata ed è stata accettata.

➔ Chiamata da Asterisk verso Siphone

Per effettuare una telefonata da Asterisk verso il computer con Siphone (eventualmente anche con un Siphone versione Windows sul computer che 'ospita' la macchina virtuale), occorre agire sul dialplan per creare un secondo utente (per esempio gallo2), assegnandogli un numero (per esempio 123), l'ip statico della macchina da chiamare (host=10.11.3.23), oppure dandogli come già fatto host=dynamic e effettuando la registrazione da Siphone con 'initialize'. Tutto questo avendo poi l'accortezza di includere in extensions.conf, nel contesto 'local', il contesto da noi creato, ovvero 'tesi'.

```
[local]
....
include => tesi
```

A questo punto, dopo un riavvio, dalla console di Asterisk basterà digitare 'dial gallo2', oppure 'dial 123' e nel corrispondente SjPhone apparirà la connessione con Asterisk.

→ **Chiamata da Sjphone verso Sjphone differente**

Chiaramente sarà possibile effettuare tra due utenti Sjphone differenti che usano computer diversi: le chiamate saranno gestite dal server Asterisk.

La configurazione è simile a quella fin qui vista: si creano due utenti nel dialplan, si configurano i rispettivi softphone sui due computer e si riavvia Asterisk.

» **3.4 Telefonata attraverso il protocollo H.323**

Se si vuole sfruttare come protocollo l'H.323, non occorre aggiungere nulla circa la configurazione di Sjphone, se non l'indicazione, ovviamente, al momento di creare un nuovo profilo, di effettuare la chiamata attraverso un gateway o un gatekeeper H.323.

La parte più onerosa nella procedura spetta alla configurazione di Asterisk.

Cominciamo dunque a vedere le caratteristiche salienti di un gateway e un gatekeeper, per poi passare ad illustrare l'installazione delle librerie per il supporto H.323, ed infine la configurazione dei Softphone utilizzati per la chiamata.

» **3.4.1 Gateway e Gatekeeper**

Il **gateway** è l'elemento della rete necessario per connettere reti di tipo diverso e traduce sia i protocolli di segnalazione che quelli per il flusso dei dati multimediali.

Molto utili nelle comunicazioni phone - to - phone, phone - to - PC, PC - to - phone, questi dispositivi permettono a chiunque abbia un telefono tradizionale a commutazione di circuito di godere dei vantaggi del VoIP: in sintesi, essi convertono le comunicazioni dalla rete PSTN in un formato adatto per la rete a commutazione di pacchetto, e viceversa.

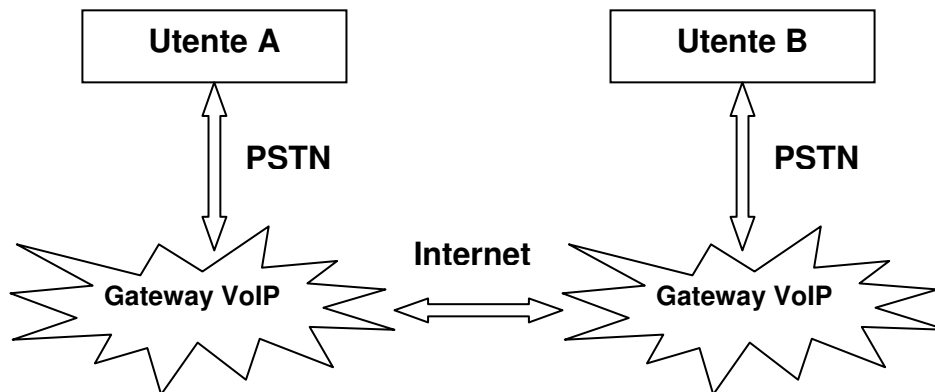


fig. 25 – Connessione phone - to - phone

La figura 24 descrive come lavora una connessione phone - to - phone tra l'utente A e l'utente remoto B.

L'utente A si connetterà al gateway VoIP interno alla propria rete PSTN locale. Per poter operare, infatti, tali gateway sono visti dal lato PSTN come un semplice numero telefonico. Il gateway risponde richiedendo il numero del telefono dell'utente B, e consulterà le proprie tabelle di instradamento per localizzare un altro gateway VoIP locale all'utente B. Quest'ultimo gateway, instaurerà la connessione chiamando telefonicamente B all'interno del suo PSTN.

I **gatekeepers** controllano l'accesso alla rete, accettando o rifiutando le chiamate e verificando la larghezza di banda di una chiamata. Facilitano inoltre la risoluzione degli indirizzi, convertendo gli indirizzi di posta elettronica negli indirizzi di rete appropriati.

Sono dispositivi logicamente separati dalle altre strutture presenti in rete, come terminali, gateway ecc., ma possono essere fisicamente implementati in ciascuna di esse.

I servizi che un gatekeeper deve garantire sono:

- Traduzione degli indirizzi, tramite una 'translation table' aggiornata dai Registration Messages.
- Controllo di ammissione, per autorizzare l'accesso alla LAN .
- Gestione della larghezza di banda dei canali di comunicazione ad esso connessi.
- Gestione della Zona e delle sue entità.

Altre funzionalità non obbligatorie sono:

- Segnalazione del controllo di chiamata e smistamento tra Endpoints
- Autorizzazioni alla chiamata per escludere/restringere l'accesso a/dai certi terminali, gateways ecc.

Un gatekeeper, insomma e' un apparato, dotato di indirizzo TCP/IP, attraverso il quale e' possibile accedere a diversi destinatari utilizzando opportuni codici, senza doverli contattare direttamente.

» **3.4.2 H323 e oh323**

Per Asterisk esistono essenzialmente due implementazioni di questo canale: h323 e oh323. Il primo, che agisce solo da gateway è incluso nella distribuzione sorgente di Asterisk in /channels/h323.

Asterisk-oh323, invece, agisce anche da gatekeeper ed è la prima implementazione in assoluto per Asterisk del canale H.323 ed è sviluppata da InAccess Networks.

Confrontando le due implementazioni si può vedere che oh323 è più stabile e affidabile di h323, nonostante le migliori prestazioni di quest'ultimo.

H323 utilizza la stack RTP di Asterisk e non ha un jitter buffer, mentre oh323 utilizza lo sviluppo di OpenH323 sia per lo stack RTP/RTCP che per l'implementazione di un jitter buffer adattativo.

» **3.4.3 Telefonata con Siphone via h323**

Proviamo ad effettuare la chiamata da Siphone sfruttando il supporto h.323 già contenuto di default in Asterisk.

Nella cartella /usr/src/asterisk/channels/h323 le prime righe del file *readme* indicano le versioni delle librerie pwlib e openh323 da scaricare da <http://sourceforge.net/projects/openh323> e installare.

In particolare si è utilizzato:

- Openh323 v 1.17.1
- Pwlib v 1.9.0

Una volta scaricati e decompressi i pacchetti (in una cartella a piacere, per esempio /usr/src/asterisk), occorre seguire questi passi:

→ compilazione PWLib

```
cd pwlib
./configure
make clean
make opt && install
```

→ compilazione OH323

```
cd..
cdopenh323
make clean
make opt && install
```

A questo punto occorre modificare con l'istruzione 'vi makefile' il file Makefile contenuto nella directory /usr/src/asterisk/channels/h323, e sostituire alla riga che inizia con 'OPENH323DIR=' l'effettivo percorso che punta alla cartella openh323.

Per evitare errori di compilazione di Asterisk occorre poi intervenire sulle seguenti variabili d'ambiente:

```
PWLIBDIR=/usr/src/asterisk/pwlib
export PWLIBDIR
OPENH323DIR=/usr/src/asterisk/openh323
export OPENH323DIR
LD_LIBRARY_PATH=$PWLIBDIR/lib:$OPENH323DIR/lib
export LD_LIBRARY_PATH
```

che possono essere inserite in /etc/profile, per evitare di riscriverle da shell ogni qualvolta occorre ricompilare.

Dopo una precauzionale disconnessione e un nuovo login, eseguiamo la ricompilazione di Asterisk digitando da terminale 'make' e 'make install' dalla directory principale di Asterisk.

La presenza di nuovi file nella cartella /channel/h323 e del file chan_h323.so in /usr/lib/asterisk/modules confermano il successo dell'operazione.

Ora passiamo alla configurazione del file h323.conf contenuto in /usr/src/asterisk/channels/h323 (i commenti di default sono stati omessi):

```
[general]
port=1720
bindaddr=0.0.0.0
disallow=all
allow=gsm
allow=g729
allow=ulaw
allow=ulaw
dtmfmode=rfc2833
gatekeeper = DISABLE
AllowGKRouted = yes
context=default
context=tesih323

[galloh323]
type=friend
callerid=("workstation3" <300>)
user=galloh323
host=dynamic
context=tesih323

[gallo2h323]
type=friend
user=gallo2h323
host=dynamic
context=tesih323
```

Per quanto riguarda il file extensions.conf abbiamo aggiunto il contesto:

```
[tesih323]
exten => 300,1,Dial(H323/${EXTEN}@gallioh323)
exten => 201,1,Dial(H323/${EXTEN}@gallio2h323)
include => demo
```

Passando infine alla configurazione del softphone, si crea un nuovo profilo indicando l'opzione 'Calls through H.323 Gateway' e indicando nel campo 'Gateway address' l'indirizzo di Asterisk e la porta 1720.

» **3.5 Videochiamata con Asterisk**

In Asterisk il supporto per la videochiamata è garantito attraverso canali SIP o IAX.

I canali H323 (chan_h323, chan_oh323, chan_ooh323), invece, al momento non permettono la videochiamata.

Per effettuare la videochiamata, dato che in Siphone non è ancora stata implementata tale funzionalità, si è utilizzato un altro Softphone: Kapanga, scaricabile, previa registrazione gratuita, dal sito www.kapanga.net.

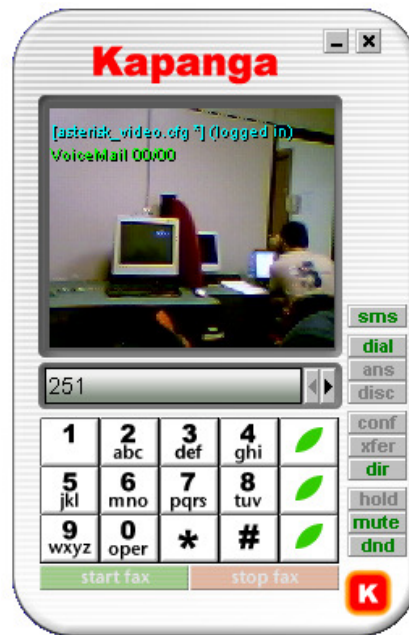


Fig. 26 – schermata del softphone Kapanga

Kapanga è un softphone SIP capace di trasmettere voce, fax e video over IP.

Inoltre Kapanga offre molte altre funzioni, come la registrazione di Voce in MP3 e videoconferenze nel formato MPEG, ricezione automatica di fax, misura in tempo reale della QoS (MOS, Mean Opinion Score) e molto altro.

Di seguito l'elenco dei codec supportati:

- Voce: G.711 (A and m-law), G.722.1, G.723.1, G.726, G.728, G.729A/B/D/E, AMR, GSM 6.10/EFR, iLBC, Speex
- Video: H.263/RFC 2190, H.263/RFC 2429, H.264/RFC 3984, MPEG-1, MPEG-2, Theora
- Fax: T.38 (MH, MR, MMR and ECM)

Per effettuare la videochiamata sfruttando SIP, occorre modificare i file sip.conf e extensions.conf nel dialplan, registrarsi e configurare i client Kapanga sui due pc interessati alla videocomunicazione.

Cominciamo da sip.conf.

Nella sezione [general] occorre togliere il commento dalla seguente riga:

```
videosupport=yes
```

Poi creiamo un nuovo contesto:

```
[videogallo]  
type=friend  
username=videogallo  
secret=pass  
host=dynamic  
context=video
```

```
canreinvite=no;
```

Quando SIP dà avvio alla chiamata, il messaggio INVITE contiene le informazioni su dove inviare gli stream. Asterisk usa se stesso come terminale per gli streams quando imposta la chiamata e, una volta che questa è accettata, invia un altro INVITE ai client con le informazioni necessarie affinché i due clienti possano inviarsi direttamente tra di loro i media streams. Con questa opzione si evita la ripetizione del messaggio 'INVITE'.

```
disallow=all  
allow=ulaw  
allow=alaw  
allow=speex  
allow=gsm  
allow=h261  
allow=h263  
allow=h263p
```

Per far dialogare tra loro due utenti si creerà un altro contesto di nome 'videogallo2'.

Nel file extensions.conf si creerà il contesto [video]:

```
[video]
exten => 250,1,Dial(SIP/videogallo)
exten => videogallo,1,Dial(SIP/videogallo)
exten => 251,1,Dial(SIP/videogallo2)
exten => videogallo2,1,Dial(SIP/videogallo2)
include => demo
```

Passiamo alla configurazione del softphone. Premendo il tasto 'K' – *Configuration – Call control setting* basterà, per avviare una semplice videochiamata, impostare i parametri di connessione settati sul server.

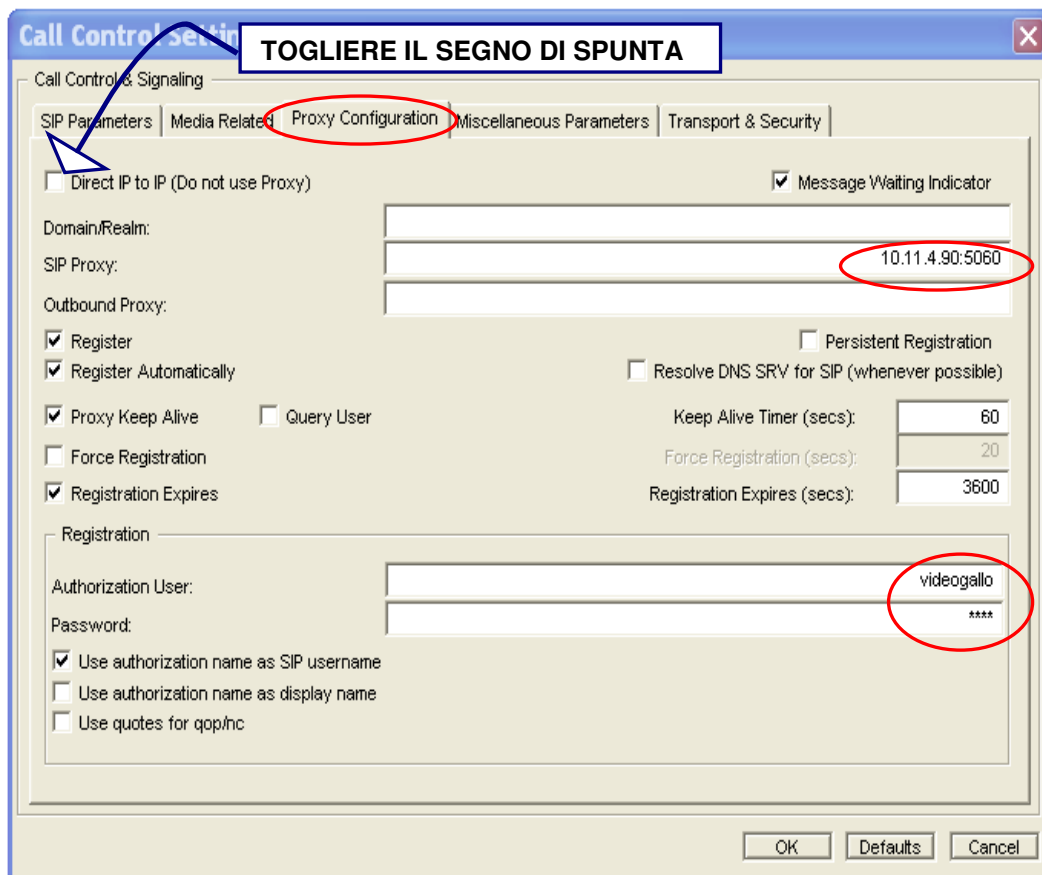


Fig. 27 – esempio configurazione di Kapanga

A questo punto, dopo il collegamento via usb di una webcam alle due postazioni e un riavvio del programma, sarà possibile dar vita alla videochiamata.

Kapanga permette di settare alcuni parametri avanzati, come i codec audio video che si vogliono utilizzare e, per ogni codec, la dimensione in millisecondi dei pacchetti, il rate, la risoluzione, ecc.

La parte sistemistica della tesi si conclude con la ripetizione del test, installando e configurando il dialplan di Asterisk sul proxy voip.lit.ing.unicas.it del laboratorio LIT (lit.ing.unicas.it) della facoltà d'Ingegneria. E' stato inoltre collegato, configurato e registrato un telefono IP SIP/H.323, e testato per chiamate da e verso softphone installati su PC collegati alla rete Ethernet del LIT stesso, sempre indirizzando e facendo controllare le chiamate dal centralino Asterisk.

Conclusioni e sviluppi futuri

Nella tesi, dopo un'analisi del Voice Over IP e dei principali protocolli utilizzati nella comunicazione su reti a commutazione di pacchetto, è stato configurato un semplice servizio per il Voice Over IP attraverso il gateway PBX Open Source Asterisk.

In particolare, dopo le fasi di installazione e compilazione dei pacchetti necessari al corretto funzionamento del gateway, sono stati modificati, rispetto alla configurazione di default già presente in Asterisk al momento della sua installazione, alcuni file di testo con estensione .conf facenti parte del dialplan, allo scopo di creare nuovi utenti in grado di registrarsi al centralino telefonico, il quale provvederà a gestire chiamate e videochiamate da e verso softphone e telefoni IP.

Nei prossimi lavori di tesi sarà possibile continuare su questa strada, affrontando problematiche sempre più complesse, come argomenti relativi alla sicurezza e all'autenticazione in Asterisk (distinzione di permessi di accesso a seconda della tipologia di utenti, riconoscimento di utenti 'guest' impossibilitati a effettuare chiamate su linee esterne), la configurazione dinamica del centralino (e in particolare l'inserimento automatico nel dialplan di nuovi utenti) attraverso LDAP e Active Directory, arrivando a completare e integrare, attraverso configurazioni del dialplan sempre più specifiche e dettagliate e attraverso l'inserimento di nuove e funzionali schede ISDN, il già presente sistema telefonico attivato nel laboratorio LIT della facoltà d'Ingegneria.

L'obiettivo è mettere su un robusto sistema per facilitare le videocomunicazioni fra dottorandi, studenti e insegnanti, allo scopo di migliorare la cooperazione e l'attività didattica d'ateneo.

