

# CHAPITRE 3: INSTALLATION ET CONFIGURATION : GNUGK ET ASTERISK

## 3.1. INSTALLATION DU GNU GATEKEEPER

Dans une architecture H.323, le Gatekeeper peut être comparé à un DNS. Il permet d'enregistrer les adresses des clients H.323 et effectue les résolutions entre une adresse H.323 et une adresse IP. Nous allons dans ce billet voir comment installer et configurer un très bon Gatekeeper open-source: GNU Gatekeeper.

### 3.1.1. INSTALLATION MANUELLE

Pour l'installation manuelle du gatekeeper, il faudrait télécharger les fichiers suivants :

- pwlib\_1.5.2.ta.r.gz [http://www.OpenH323.org/bin/pwlib\\_1.5.2.tar.gz](http://www.OpenH323.org/bin/pwlib_1.5.2.tar.gz)
  - OpenH323\_1.12.2.tar.gz [http://www.OpenH323.org/bin/OpenH323\\_1.12.2.tar.gz](http://www.OpenH323.org/bin/OpenH323_1.12.2.tar.gz)
  - gnugk-2.2.1-2.tgz <http://prdownloads.sourceforge.net/OpenH323gk/gnugk-2.2.1-2.tgz?download>
- ❖ Compilation du Pwlib
  - ✓ tar xvzf pwlib\_1.5.2.tar.gz
  - ✓ cd pwlib
  - ✓ pwd (pour récupérer le chemin absolu du répertoire courant PWLIBDIR)
  - ✓ export LD\_LIBRARY\_PATH=PWLIBDIR/lib
  - ✓ ./configure
  - ✓ make
- ❖ Compilation de OpenH323
  - ✓ tar xvzf OpenH323\_1.12.2.tar.gz
  - ✓ cd OpenH323
  - ✓ pwd (pour récupérer le chemin absolu du répertoire courant OPENH323LIBDIR)

- ✓ export  
LD\_LIBRARY\_PATH=\$LD\_LIBRARY\_PATH":OPENH323LIBDIR/lib
- "
- ✓ ./configure
- ✓ make

❖ Compilation et installation de Gnugk

- ✓ tar xvzf gnugk-2.2.1-2.tgz
- ✓ cd gnugk-2.2.1
- ✓ ./configure
- ✓ make opt
- ✓ make addpasswd (nous servira à ajouter des comptes pour l'enregistrement des clients)
- ✓ cp obj\_linux\_x86\_r/gnugk /usr/sbin (facultatif mais fait pour un confort d'utilisation)
- ✓ cp obj\_linux\_x86\_r/addpasswd /usr/sbin (facultatif mais fait pour un confort d'utilisation)
- ✓ touch /etc/gatekeeper.ini (on crée le fichier qui servira de fichier configuration de gnugk)

❖ Lancement du gnugk

- ✓ ./gnugk -c /etc/gnugk.ini -ttt (on appelle le fichier de configuration)

### 3.1.2. INSTALLATION AUTOMATIQUE

Pour installer automatiquement gnugk sous ubuntu il faut faire :

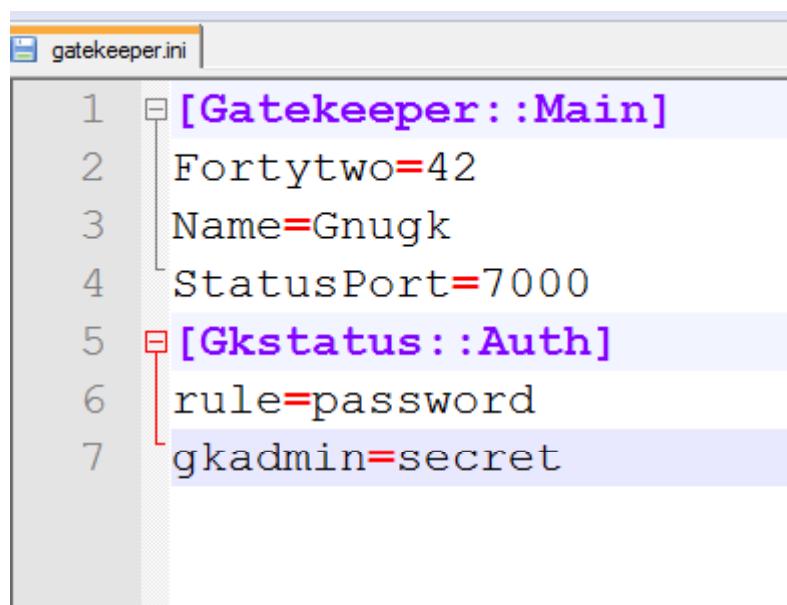
```
# apt-get install gnugk
```

Les options sont réunies dans le fichier /etc/gatekeeper.ini. On commence par sauvegarder le fichier par défaut :

```
# sudo mv /etc/gatekeeper.ini /etc/gatekeeper.ini.default
```

Ensuite on édite un nouveau fichier avec nos options:

```
sudo gedit /etc/gatekeeper.ini
```



**Figure 3.1. Capture de la configuration du gatekeeper.ini**

Nous allons lancer le daemon GNUGk en mode foreground:

```
# gnugk -tt -c /etc/gatekeeper.ini
```

Il est possible de se connecter à distance à GNU Gatekeeper en utilisant un telnet sur le port 7000:

```
# telnet 192.168.1.106 7000
...
?
All Registrations
RCF|192.168.1.108:1720|popof: h323_ID|terminal |1420_endp
RCF|192.168.1.110:1720|popof: h323_ID|terminal |4393_endp
RCF|192.168.1.109:1720|popof: h323_ID|terminal |4593_endp
Number Of endpoints: 2
qui t
```

### 3.2. INSTALLATION D'ASTERISK

Pour commencer, il faudrait créer un dossier dans lequel vous allez télécharger les sources :

```
mkdir ~/usr/src/asterisk
```

Et placez-vous dans ce dossier :

```
cd ~/usr/src/asterisk
```

Télécharger les sources à partir de <http://www.asterisk.org/downloads> ou utilisez le lien direct <http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-1.8.5.0.tar.gz> :

```
wget http://downloads.asterisk.org/pub/telephony/asterisk/releases/asterisk-1.8.5.0.tar.gz
```

dé-gzippez le fichier téléchargé :

```
gunzip asterisk-1.8.5.0.tar.gz
```

dé-tarrez le fichier téléchargé :

```
tar xvf asterisk-1.8.5.0.tar
```

Cela va créer un dossier "asterisk-1.8.5.0", placez vous dans ce dossier :

```
cd asterisk-1.8.5.0
```

**Préparez les sources pour la compilation**

```
(sudo) ./configure
```

(La commande "sudo" permet de prendre les priviléges de "root" ou super utilisateur sur un système Ubuntu. si vous êtes déjà connecté en tant que "root" alors la commande "sudo" n'est pas nécessaire.)

(Cela peut prendre plusieurs minutes), cela va vérifier si il y a des dépendances manquantes, et va préparer les sources ;

S'il y a un problème de dépendances ou autre, l'utilitaire "configure" vous indiquera laquelle. Malheureusement, ayant déjà installé Asterisk, je ne peux plus simuler ce message d'erreur.

Les packages suivants ont été nécessaire durant cette phase pour la version 1.8.9.2

```
# apt-get install libncurses5-dev libxml2-dev build-essential libnewt-dev
```

Sélectionner les modules optionnels

```
(sudo) make menuselect
```

Avertissement : la fenêtre de votre terminal doit être au moins de 80 x 27, sinon un message comme celui ci s'affichera :

```
make[2]: quittant le répertoire « /home/andrei66/asterisk/asterisk-1.8.5.0/menuselect »  
Terminal must be at least 80 x 27.  
menuselect changes NOT saved!  
make[1]: quittant le répertoire « /home/andrei66/asterisk/asterisk-1.8.5.0 »
```

Si tout va bien, vous aurez un menu comme ceci :

```
*****  
Asterisk Module and Build Option Selection  
*****  
  
Press 'h' for help.  
  
---> Add-ons (See README-addons.txt)  
      Applications  
      Bridging Modules  
      Call Detail Recording  
      Channel Event Logging  
      Channel Drivers  
      Codec Translators  
      Format Interpreters  
      Dialplan Functions  
      PBX Modules  
      Resource Modules  
      Test Modules
```

```
Compiler Flags
Voicemail Build Options
Utilities
AGI Samples
Module Embedding
Core Sound Packages
Music On Hold File Packages
Extras Sound Packages
```

Utilisez les flèches du clavier pour faire Haut/Bas, la flèche gauche pour retourner au menu précédent, la flèche droite pour entrer dans un menu...

Appuyez sur -> (flèche droite) pour entrer dans le menu "Add-ons" :

```
*****
Asterisk Module and Build Option Selection
*****
Press ' h' for help.

XXX app_mysql
[ ] app_saycountpl
XXX cdr_mysql
XXX chan_mobile
[*] chan_ooh323
[*] format_mp3
XXX res_config_mysql
```

Utilisez les flèches Haut/Bas pour sélectionner une case, et Espace pour cocher la case ; en bas, vous avez une description de l'élément sélectionné.

Cochez "format\_mp3" pour le support des "music on hold" au format MP3.

Cocher "chan\_ooh323" pour pouvoir interconnecter asterisk et le gnugk

Ensuite, retournez au menu précédent avec la flèche gauche :

```
*****
Asterisk Module and Build Option Selection
*****  
Press 'h' for help.  
---> Add-ons (See README-addons.txt)  
      Applications  
      Bridging Modules  
      Call Detail Recording  
      Channel Event Logging  
      Channel Drivers  
      Codec Translators  
      Format Interpreters  
      Dialplan Functions  
      PBX Modules  
      Resource Modules  
      Test Modules  
      Compiler Flags  
      Voicemail Build Options  
      Utilities  
      AGI Samples  
      Module Embedding  
      Core Sound Packages  
      Music On Hold File Packages  
      Extras Sound Packages
```

Ensuite, parcourez les autres catégories en utilisant le même principe décrit plus haut, puis appuyez sur X pour sauvegarder les changements, ou appuyez sur Q pour quitter sans sauvegarder.

(Optionnel) Installer "mpg123"

**AVERTISSEMENT** : si vous avez sélectionné plus haut "format\_mp3", alors avant de passer à l'étape suivante, installez le paquet "mpg123" :

```
(sudo) apt-get install mpg123
```

(la commande ci-dessus est valable pour un système basé sur Debian ; Ubuntu, etc...)

Compiler Asterisk

Ensuite, faite :

```
(sudo) make
```

(Cela peut être long, même très long, de l'ordre de 15 minutes)

Encore une fois, si vous avez sélectionné l'addon "format\_mp3" comme ci dessus, il faudra avoir le paquet subversion et télécharger le code source grâce à un script déjà existant :

```
# apt-get install subversion  
$ bash contrib/scripts/get_mp3_source.sh
```

A la fin, si tout est bon, alors vous avez un message comme ça :

```
cptl s. o tdd. o term. o test. o threadstorage. o timi ng. o transl ate. o udptl . o ul aw. o  
util s. o versi on. o xml . o xml doc. o edi tline/l i bedi t. a db1-ast/l i bdb1. a ->  
asterisk  
Building Documentation For: channels pbx apps codecs formats cdr cel bridges  
funcs tests main res addons  
----- Asterisk Build Complete -----  
+ Asterisk has successfully been buil t, and +  
+ can be installed by running: +  
+ +  
+ make install +  
+----- +
```

## Installer Asterisk

Comme ce message l'indique, faites :

```
(sudo) make instal l
```

Pour installer Asterisk (peut prendre quelques minutes).

Une fois qu'Asterisk est installé, vous pouvez faire :

```
(sudo) make sampl es
```

pour installer des exemples de fichiers de configuration.

## Conseils

Faites une copie de sauvegarde des exemples de fichiers de configuration (si vous les avez installés) en faisant :

```
(sudo) mkdir /etc/asterisk/sampl es  
(sudo) cp /etc/asterisk/* /etc/asterisk/sampl es
```

Configurer Asterisk pour qu'il démarre automatiquement au démarrage de l'ordinateur :

```
(sudo) make config
```

## Commandes utiles

---

Démarrer Asterisk :

```
(sudo) /etc/init.d/asterisk start
```

Arrêter Asterisk :

```
(sudo) /etc/init.d/asterisk stop
```

Afficher si Asterisk est en cours d'exécution :

```
(sudo) /etc/init.d/asterisk status
```

Redémarrer Asterisk :

```
(sudo) /etc/init.d/asterisk restart
```

Recharger les fichiers de configuration d'Asterisk :

```
(sudo) /etc/init.d/asterisk reload
```

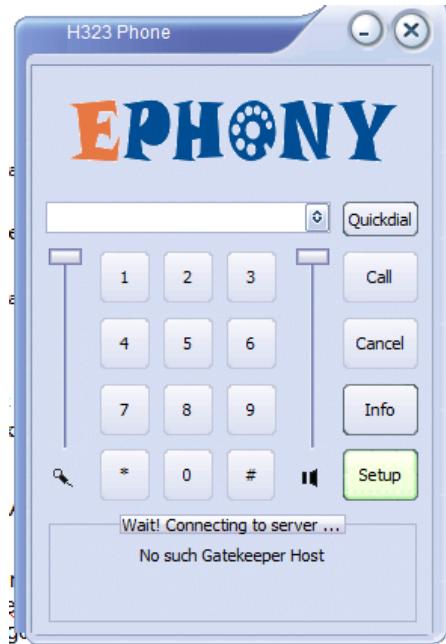
C'est tout, Asterisk est installé. Maintenant nous pouvons passer à l'étape suivante pour la configuration d'Asterisk et du Gnu Gatekeeper.

### 3.3. CONFIGURATION DES CLIENTS H323

Pour configurer un client pour qu'il puisse communiquer avec le protocole H323, il faudrait disposer d'un soft phone compatible avec le protocole H323. Pour télécharger un soft phone gratuit il faut le télécharger à l'adresse suivante :

- Ekiga : <http://ekiga.org/download-ekiga-binaries-or-source-code>
- H323Phone : <http://www.miaphone.com/download>

Dans notre cas nous avons utilisé le H323 Phone. Une fois téléchargée et l'installation terminée, vous aurez une interface comme ceci



**Figure 3.2. Softphone H.323**

Après cette interface, vous devez cliquer sur setup pour la configuration du soft phone



**Figure 3.3. Configuration du Softphone H.323**

Appuyer sur appliquer puis sur ok et vous pouvez émettre et recevoir des appels avec votre soft phone H323. Pour retrouver l'historique des appels émis et reçus depuis votre soft phone, allez sur cette interface et vous trouverez le chemin de vos traces

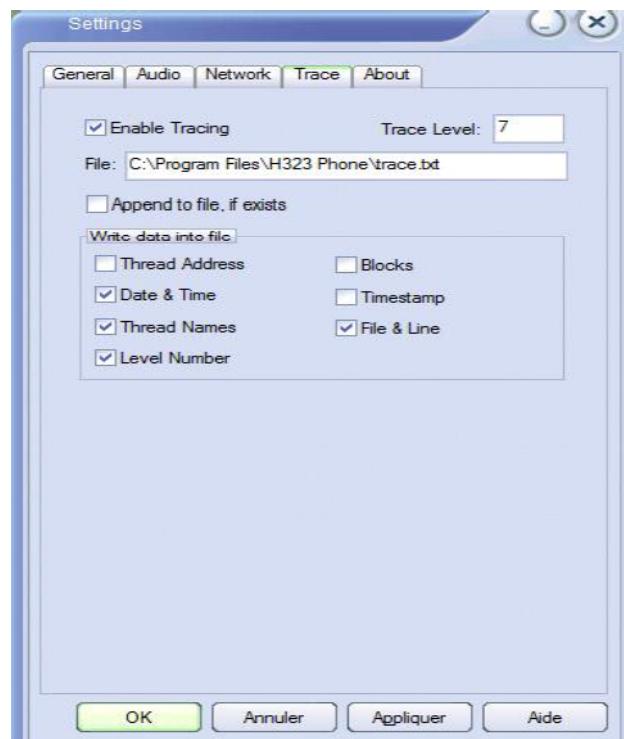


Figure 3.4. Configuration avancée Softphone H.323

### 3.4. INTERCONNEXION DU GNU GATEKEEPER AVEC ASTERISK

Nous allons d'abord commencer Par configurer le GnuGk. Pour cela nous allons éditer le fichier gatekeeper.ini.

```
(sudo) gedit /etc/gatekeeper.ini
```

Et vous aurez cette fenêtre qui s'affiche:

The screenshot shows a text editor window with the title 'gatekeeper.ini'. The code is color-coded: blue for numbers, red for some keywords, and green for comments. The configuration file contains two sections: [Main] and [RoutedMode]. The [Main] section includes parameters like Fortytwo=42, TimeToLive=300, rule=allow, and port=1720. The [RoutedMode] section includes parameters like GKRouted=1, H245Routed=0, RemoveH245AddressOnTunneling=0, AcceptNeighborsCalls=1, AcceptUnregisteredCalls=0, SupportNATEdEndpoints=1, and DropCallsByReleaseComplete=1.

```
1 [Main]
2 Fortytwo=42
3 ;Ce paramètre est utilisé pour tester
4 ;la présence du fichier de configuration.
5 ;Si il n'est pas trouvé, un avertissement est émis.
6 ; Assurez-vous qu'il soit dans tous vos fichiers de configuration.
7 TimeToLive=300;
8 ;L'enregistrement d'un terminal avec un gatekeeper peut avoir une
9 ;durée de vie limitée. Le gatekeeper précise la durée de l'enregistrement
10 ;d'un terminal en incluant un champ TimeToLive. Ce paramètre
11 ;de configuration indique la minuterie en secondes du temps à vivre
12 ;avant que l'enregistrement se termine. Pour désactiver cette
13 ;fonction, lui donner la valeur -1.
14 rule=allow;// Autoriser toutes les connections des téléphones
15 port=1720
16 [RoutedMode]
17 GKRouted=1
18 H245Routed=0
19 RemoveH245AddressOnTunneling=0
20 AcceptNeighborsCalls=1
21 AcceptUnregisteredCalls=0
22 SupportNATEdEndpoints=1
23 DropCallsByReleaseComplete=1
```

**Figure 3.5. configuration du gatekeeper pour son utilisation**

Une fois le gatekeeper configurée, nous allons sur Asterisk pour modifier certains fichiers pour qu'ils puissent communiquer :

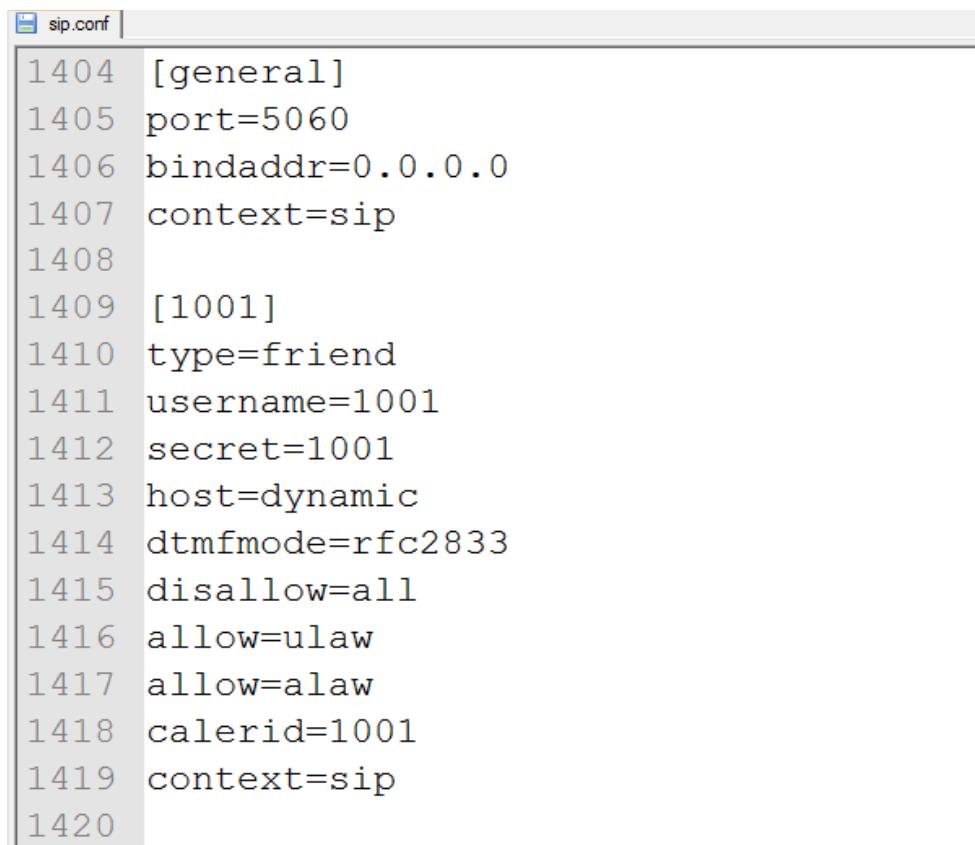
```
(sudo) cd /etc/asterisk/
      cp chan_h323.conf ooh323.conf
      gedit ooh323.conf
```

```
ooh323.conf
176 [general]
177 port=1720
178 bindaddr=0.0.0.0
179 gatekeeper=192.168.1.102;adresse IP du gatekeeper
180 gateway=yes
181 h323id=Asterisk
182 e164=500;tous les appels commençant par 500 sont achéminés sur asterisk
183 callerid=500
184 faststart=yes
185 h245tunneling=yes
186 context=sip
187 disallow=all
188 allow=ulaw
189 allow=alaw
190 allow=gsm
191 dtmfmode=rfc2833
192 language=fr
193
194 [5000]
195 IP=192.168.5.104;adresse IP du user H323
196 type=friend
197 context=sip
198 port=1720
199 e164=5000
200 dtmfmode=inband
```

**Figure 3.6. Capture de la configuration de ooh323.conf**

Une fois le gnugk configure, nous allons configurer des utilisateurs SIP et pour cela il faut éditer le fichier sip.conf:

```
(sudo) gedit sip.conf
```



The screenshot shows a text editor window titled "sip.conf". The content of the file is as follows:

```
1404 [general]
1405 port=5060
1406 bindaddr=0.0.0.0
1407 context=sip
1408
1409 [1001]
1410 type=friend
1411 username=1001
1412 secret=1001
1413 host=dynamic
1414 dtmfmode=rfc2833
1415 disallow=all
1416 allow=ulaw
1417 allow=alaw
1418 calerid=1001
1419 context=sip
1420
```

**Figure 3.7. Capture de la configuration de sip.conf**

Ensuite on édite le fichier extensions.conf :

```
sudo) gedit extensions.conf
```

```
extensions.conf
826 ; "core show application <command>" will show details of how you
827 ; use that particular application in this file, the dial plan.
828 ; "core show functions" will list all dialplan functions
829 ; "core show function <COMMAND>" will show you more information about
830 ; one function. Remember that function names are UPPER CASE.
831
832 [general]
833 static=yes
834 writeprotect=yes
835
836
837 [default]
838
839 exten => s,1,Wait,1
840 exten => s,2,Answer
841 exten => s,3,Playback(tt-monkeys)
842 exten => s,4,Hangup
843
844 [sip]
845 exten => 1001,1,Dial(SIP/1001, 25, tr)
846
847 exten => 500,1,Dial(OOH323/500, 25, tr)
848
849
```

**Figure 3.8. Capture de la configuration de extensions.conf**

Puis on charge les fichiers dans asterisk et on redemarre asterisk et le gnuGK :

```
sudo) /etc/init.d/asterisk reload
sudo) /etc/init.d/asterisk restart
sudo) /etc/init.d/gnugk start
```

Et nos deux utilisateurs peuvent s'appeler mutuellement avec leur numéro respectif.

### 3.5. ETUDE DES CALL FLOW DES INTERCONNEXIONS

#### 3.5.1. CALL FLOW H323-H323

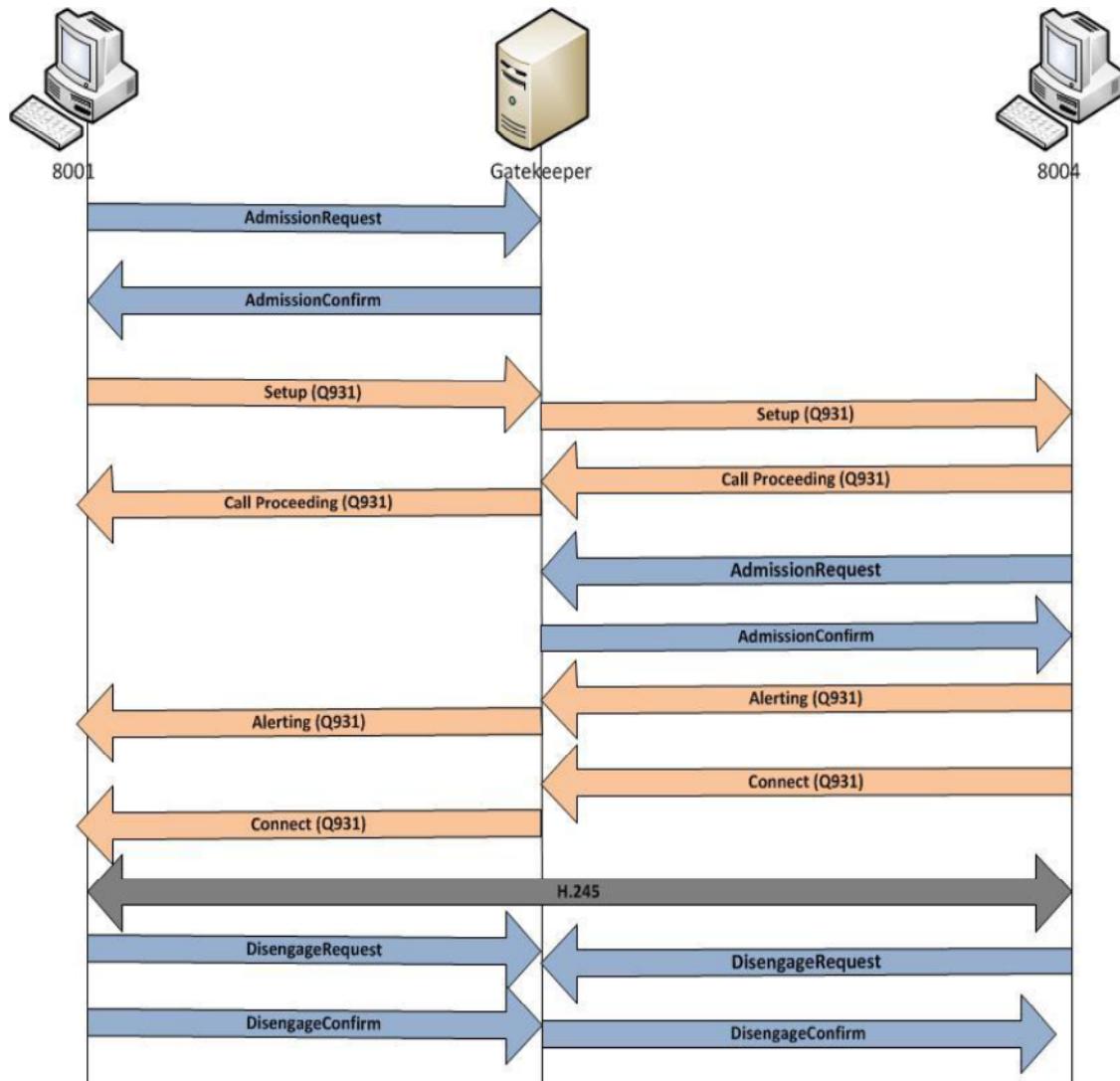


Figure 3.9. Diagramme de communication entre les clients H.323

No.	Time	Source	Destination	Protocol	Length	Info
26	3.817153	192.168.80.1	192.168.80.128	H.225.0	219	RAS: infoRequestResponse
25	3.805614	192.168.80.1	192.168.80.128	H.225.0	224	RAS: infoRequestResponse
223	115.284352	192.168.80.1	192.168.80.128	H.225.0	144	RAS: disengageRequest
209	115.178173	192.168.80.1	192.168.80.128	H.225.0	144	RAS: disengageRequest
224	115.286114	192.168.80.128	192.168.80.1	H.225.0	45	RAS: disengageConfirm
210	115.180807	192.168.80.128	192.168.80.1	H.225.0	45	RAS: disengageConfirm
21	3.751679	192.168.80.1	192.168.80.128	H.225.0	253	RAS: admissionRequest
7	3.706457	192.168.80.1	192.168.80.128	H.225.0	164	RAS: admissionRequest
22	3.752411	192.168.80.128	192.168.80.1	H.225.0	82	RAS: admissionConfirm
8	3.708392	192.168.80.128	192.168.80.1	H.225.0	83	RAS: admissionConfirm
17	3.740404	192.168.80.128	192.168.80.1	H.225.0	274	CS: setup
12	3.719907	192.168.80.1	192.168.80.128	H.225.0	209	CS: setup
214	115.188046	192.168.80.1	192.168.80.128	H.225.0	103	CS: releaseComplete
207	115.177052	192.168.80.128	192.168.80.1	H.225.0	103	CS: releaseComplete
205	115.175955	192.168.80.1	192.168.80.128	H.225.0	115	CS: releaseComplete
28	3.947654	192.168.80.128	192.168.80.1	H.225.0	190	CS: connect
23	3.794978	192.168.80.1	192.168.80.128	H.225.0	202	CS: connect
20	3.744328	192.168.80.128	192.168.80.1	H.225.0	164	CS: callProceeding
18	3.742139	192.168.80.1	192.168.80.128	H.225.0	176	CS: callProceeding

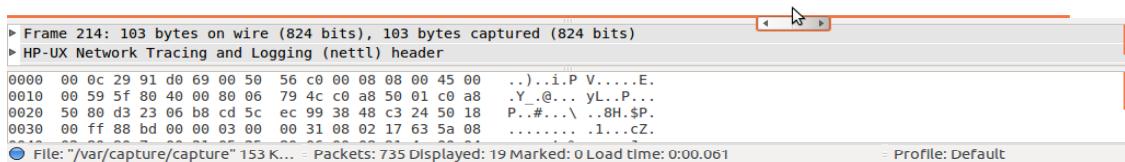


Figure 3.10. Capture des messages échangés entre les clients H.323

Les messages RAS :

**Capture message RAS : GatekeeperRequest (GRQ)** : Message envoyé par un endpoint à la recherche de son Gatekeeper :

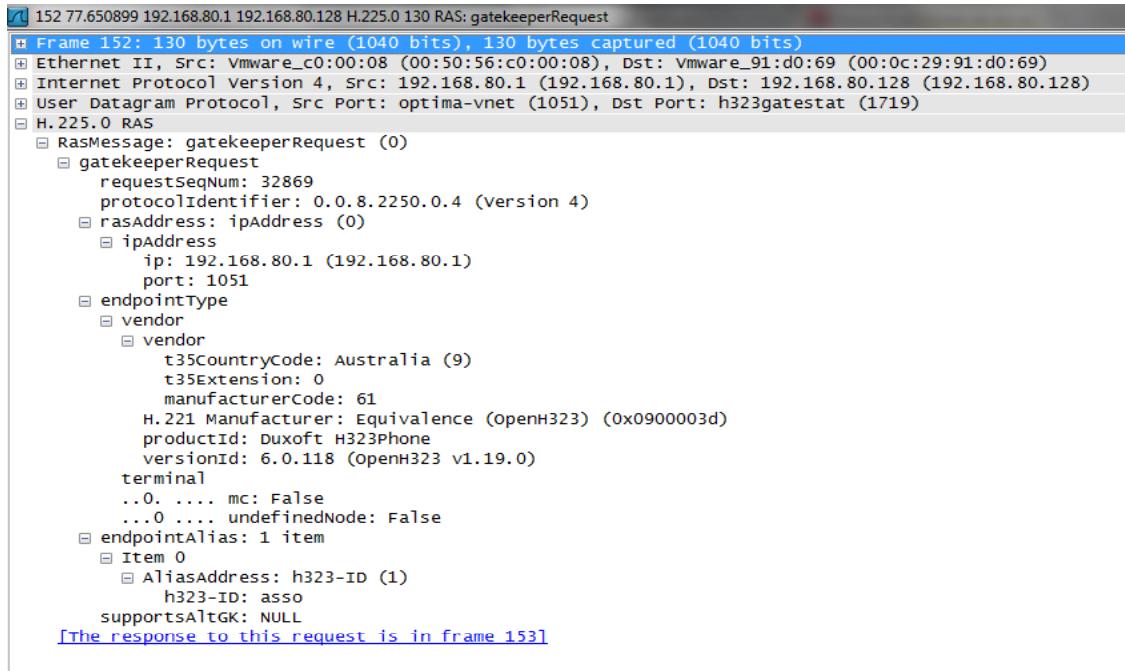


Figure 3.11. capture du détail de GatekeeperRequest

Adresse de l'EP qui cherche le GK : 192.168.80.1

Port : 1051

**Capture message RAS : GatekeeperConfirm (GCF) :** Retourné par le Gatekeeper pour informer l'endpoint qu'il sera son Gatekeeper

The screenshot shows a network capture in Wireshark. The selected frame is 153, which is a RAS message. The details pane shows the following structure:

- Frame 153: 81 bytes on wire (648 bits), 81 bytes captured (648 bits)
- Ethernet II, Src: VMware\_91:d0:69 (00:0c:29:91:d0:69), Dst: VMware\_c0:00:08 (00:50:56:c0:00:08)
- Internet Protocol Version 4, Src: 192.168.80.128 (192.168.80.128), Dst: 192.168.80.1 (192.168.80.1)
- User Datagram Protocol, Src Port: h323gatestat (1719), Dst Port: optima-vnet (1051)
- H.225.0 RAS
  - RasMessage: gatekeeperConfirm (1)
    - gatekeeperConfirm
      - requestSeqNum: 32869
      - protocolIdentifier: 0.0.8.2250.0.4 (version 4)
      - gatekeeperIdentifier: OpenH323GK
    - rasAddress: ipAddress (0)
      - ipAddress
        - ip: 192.168.80.128 (192.168.80.128)
        - port: 1719

[This is a response to a request in frame 152]  
[RAS Service Response Time: 0.000576000 seconds]

Figure 3.12. capture du détail de GatekeeperConfirm

**Capture message RAS : RegistrationRequest (RRQ) :** Permet à l'endpoint de s'enregistrer auprès de son Gatekeeper

The screenshot shows a network capture in Wireshark. The selected frame is 154, which is a RAS message. The details pane shows the following structure:

- Frame 154: 243 bytes on wire (1944 bits), 243 bytes captured (1944 bits)
- Ethernet II, Src: VMware\_c0:00:08 (00:50:56:c0:00:08), Dst: VMware\_91:d0:69 (00:0c:29:91:d0:69)
- Internet Protocol Version 4, src: 192.168.80.1 (192.168.80.1), dst: 192.168.80.128 (192.168.80.128)
- User Datagram Protocol, Src Port: optima-vnet (1051), Dst Port: h323gatestat (1719)
- H.225.0 RAS
  - RasMessage: registrationRequest (3)
    - registrationRequest
      - requestSeqNum: 32870
      - protocolIdentifier: 0.0.8.2250.0.4 (version 4)
      - 1..... discoveryComplete: True
    - callSignalAddress: 3 items
      - item 0
        - TransportAddress: ipAddress (0)
          - ipAddress
            - ip: 192.168.80.1 (192.168.80.1)
            - port: 1720
        - item 1
          - TransportAddress: ipAddress (0)
            - ipAddress
              - ip: 10.0.2.1 (10.0.2.1)
              - port: 1720
          - item 2
            - TransportAddress: ipAddress (0)
              - ipAddress
                - ip: 192.168.0.13 (192.168.0.13)
                - port: 1720
        - rasAddress: 1 item
          - item 0
            - TransportAddress: ipAddress (0)
              - ipAddress
                - ip: 192.168.80.1 (192.168.80.1)
                - port: 1051
        - terminalType
        - terminalAlias: 1 item
          - item 0
            - AliasAddress: h323-ID (1)
              - h323-ID: asso
        - gatekeeperIdentifier: openH323GK
        - endpointVendor
          - vendor
            - t35CountryCode: Australia (9)
            - t35Extension: 0
            - manufacturerCode: 61
        - H.221 Manufacturer: Equivalence (OpenH323) (0x0900003d)
        - productId: duxoft H323Phone
        - versionId: 6.0.118 (OpenH323 v1.19.0)
        - timeToLive: 60

Figure 3.13. capture du détail de RegistrationRequest

- L'EP envoie une requête d'enregistrement (RRQ) vers le port (1719) RAS du GK
- @IP: 192.168.80.1 Port: 1720 : utiliser par l'EP pour la signalisation
- @IP: 192.168.80.1 Port: 1051 : l'adresse sur lesquels doivent parvenir les réponses RAS
- L'alias utilisé pour la requête est : asso
- Ces informations permettent au GK de mettre à jour ses tables de correspondance

**Captures des Messages RAS : RegistrationConfirm (RCF) :** Renvoyé par le Gatekeeper pour indiquer à l'endpoint qu'il est bien enregistré

```

1 155 77.671775 192.168.80.128 192.168.80.1 H.225.0 135 RAS: registrationConfirm
Frame 155: 135 bytes on wire (1080 bits), 135 bytes captured (1080 bits)
Ethernet II, Src: VMware_91:d0:69 (00:0c:29:91:d0:69), Dst: VMware_c0:00:08 (00:50:56:c0:00:08)
Internet Protocol Version 4, Src: 192.168.80.128 (192.168.80.128), Dst: 192.168.80.1 (192.168.80.1)
User Datagram Protocol, Src Port: h323gatestat (1719), Dst Port: optima-vnet (1051)
H.225.0 RAS
  RasMessage: registrationConfirm (4)
    registrationConfirm
      requestSeqNum: 32870
      protocolIdentifier: 0.0.8.2250.0.4 (version 4)
    nonStandardData
      nonStandardIdentifier: h221NonStandard (1)
        h221NonStandard
          t35CountryCode: Poland (138)
          t35Extension: 2
          manufacturerCode: 2
          H.221 Manufacturer: Unknown (0x8a020002)
          data: 5 octets
        Data (5 bytes)
          Data: 4e6f4e4154
          [Length: 5]
    callSignalAddress: 1 item
      Item 0
        TransportAddress: ipAddress (0)
          ipAddress
            ip: 192.168.80.128 (192.168.80.128)
            port: 1720
    terminalAlias: 1 item
      Item 0
        AliasAddress: h323-ID (1)
          h323-ID: asso
        gatekeeperIdentifier: openH323GK
        endpointIdentifier: 3937_endp
        0... .... willRespondToIRR: False
        0... .... maintainConnection: False
    serviceControl: 1 item
  [This is a response to a request in frame 154]
[RAS Service Response Time: 0.002574000 seconds]

```

**Figure 3.14. capture du détail de RegistrationConfirm**

- @IP: 192.168.80.128 port:1720 : L'adresse et le port utilisé par le GK pour la signalisation
- EndpointIdentifier 3937\_endp Numéro d'identification unique au terminal assigné par le GK qu'il utilisera dans chacune des transactions entre lui et l'endpoint
- Time to live 60 s: durée de vie de l'enregistrement

**Capture message RAS:AdmissionRequest (ARQ) :** Envoyé par un endpoint pour demander l'autorisation à son Gatekeeper de participer à un appel

```

67 26.851146 192.168.80.1 192.168.80.128 H.225.0 154 RAS: admissionRequest
Frame 67: 154 bytes on wire (1232 bits), 154 bytes captured (1232 bits)
Ethernet II, Src: VMware_c0:00:08 (00:50:56:c0:00:08), Dst: VMware_91:d0:69 (00:0c:29:91:d0:69)
Internet Protocol Version 4, Src: 192.168.80.1 (192.168.80.1), Dst: 192.168.80.128 (192.168.80.128)
User Datagram Protocol, Src Port: boinc-client (1043), Dst Port: h323gatestat (1719)
H.225.0 RAS
RasMessage: admissionRequest (9)
  admissionRequest
    requestSeqNum: 51432
    callType: pointToPoint (0)
      pointToPoint: NULL
      endpointIdentifier: 3937_endp
    destinationInfo: 1 item
      Item 0
        DestinationInfo item: dialleddigits (0)
          dialleddigits: 8004
    srcInfo: 1 item
      Item 0
        AliasAddress: h323-ID (1)
          h323-ID: 8001
      bandwidth: 200000
      callReferenceValue: 22840
      conferenceID: 50a758db-b508-1910-93fc-0800270090ac
      activeMC: False
      answerCall: False
      canMapAlias: True
    callIdentifier
      guid: 50a758db-b508-1910-93fb-0800270090ac
      gatekeeperIdentifier: OpenH323GK
      willSupplyUIEs: True
      canMapSrcAlias: False
The response to this request is in frame 68

```

**Figure 3.15. capture du detail de AdmissionRequest**

- L'EP envoie une requête de permission d'appeler (ARQ) vers le port RAS du GK où il est enregistré (le port UDP 1719). Cette requête comprend :
  - le type d'appel: Point to point
  - Identifiant unique de l'appelant: endpointIdentifier: 3937-endp
  - N°E 164 du correspondant à appeler: 8004
  - N°E 164 de la source: 8001
  - N° unique qui permettra d'identifier l'appelant et l'appelé: CallReferenceValue 22840
  - La bande passante bidirectionnelle que l'EP souhaite disposer: bandwidth : 200000 bits/s

**Capture message RAS: AdmissionConfirm (ACF) : Retourné par le Gatekeeper à l'endpoint pour lui confirmer son admission**

```

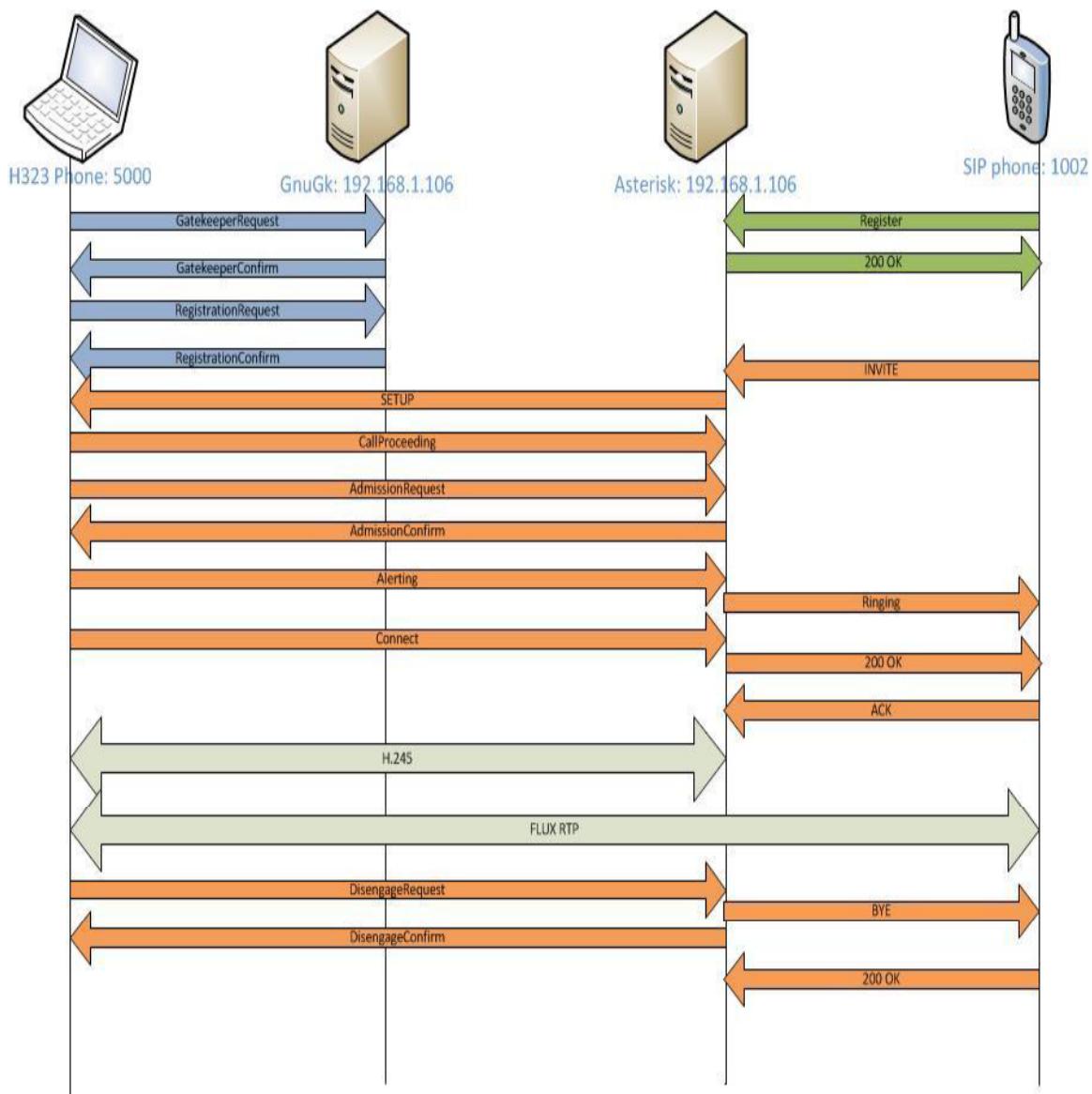
326 178.384268 192.168.80.128 192.168.80.1 H.225.0 83 RAS: admissionConfirm
Frame 326: 83 bytes on wire (664 bits), 83 bytes captured (664 bits)
Ethernet II, Src: vmware_91:d0:69 (00:0c:29:91:d0:69), Dst: vmware_c0:00:08 (00:50:56:c0:00:08)
Internet Protocol Version 4, Src: 192.168.80.128 (192.168.80.128), Dst: 192.168.80.1 (192.168.80.1)
User Datagram Protocol, Src Port: h323gatestat (1719), Dst Port: optima-vnet (1051)
H.225.0 RAS
  RasMessage: admissionConfirm (10)
    requestSeqNum: 32871
    bandwidth: 200000
    callModel: gatekeeperRouted (1)
      gatekeeperRouted: NULL
    destCallsignalAddress: ipAddress (0)
      ipAddress
        ip: 192.168.80.128 (192.168.80.128)
        port: 1720
      iirFrequency: 120
      0... .... willRespondToIRR: False
    uuiesRequested
      .0... .... setup: False
      .0... .... callProceeding: False
      ...0 .... connect: False
      .... 0... .... alerting: False
      .... 0... .... information: False
      .... 0... .... releasecomplete: False
      .... 0... .... facility: False
      0... .... progress: False
      .0... .... empty: False
      0... .... status: False
      0... .... statusInquiry: False
      0... .... setupAcknowledge: False
      0... .... notify: False
    serviceControl: 1 item
      Item 0
        ServiceControlsession
          sessionid: 0
          contents: callcreditServiceControl (3)
            callCreditServiceControl
              callStartingPoint: connect (1)
                connect: NULL
              reason: refresh (1)
                refresh: NULL
  [This is a response to a request in frame 325]
[RAS Service Response Time: 0.000858000 seconds]

```

**Figure 3.16. capture du detail de AdmissionConfirm**

- Le GK répond par un message d'acceptation (ACF) ou de rejet (ARJ) vers le port RAS de l'EP. Cette réponse comprend :
  - Bandwidth: 200000 bits/s bande passante maximum allouée pour cet appel
  - Mode d'appel: gatekeeperrouted: appel routé
  - l'adresse et le port H.225-Q.931 (le port TCP 1720) où envoyer l'invitation d'appel: dans notre cas c'est l'adresse du GK car on est en mode routé: 192.168.80.128
  - Fréquence des messages d'information (IIR) à envoyer au GK: iirfrequency: 120Hz

### 3.5.2. CALL FLOW H323-SIP



**Figure 3.17. Diagramme de communication entre les clients SIP et H.323**

Filter:	h225		Expression...	Clear	Apply	Save
No.	Time	Source	Destination	Protocol	Length	Info
749	194.444117	192.168.1.104	192.168.1.106	H.225.C	160	RAS: gatekeeperRequest
750	194.444760	192.168.1.106	192.168.1.104	H.225.C	81	RAS: gatekeeperConfirm
751	194.489688	192.168.1.104	192.168.1.106	H.225.C	259	RAS: registrationRequest
752	194.495414	192.168.1.106	192.168.1.104	H.225.C	158	RAS: registrationConfirm
808	200.060967	192.168.1.106	192.168.1.104	H.225.C	319	CS: setup OpenLogicalchannel
809	200.140641	192.168.1.104	192.168.1.106	H.225.C	178	CS: callProceeding
811	200.156799	192.168.1.104	192.168.1.106	H.225.C	248	RAS: admissionRequest
812	200.158847	192.168.1.106	192.168.1.104	H.225.C	83	RAS: admissionConfirm
813	200.281542	192.168.1.104	192.168.1.106	H.225.C	178	CS: alerting
816	200.283063	192.168.1.106	192.168.1.104	H.225.C	248	terminalCapabilitySet cs: facility
819	200.452918	192.168.1.104	192.168.1.106	H.225.C	232	terminalCapabilitySetAck terminalCapabilityset cs: empty
820	200.454535	192.168.1.106	192.168.1.104	H.225.C	133	terminalCapabilitysetAck cs: facility
821	200.523536	192.168.1.104	192.168.1.106	H.225.C	259	RAS: infoRequestResponse
823	200.662285	192.168.1.106	192.168.1.104	H.225.C	137	masterSlaveDetermination cs: facility
824	200.676907	192.168.1.104	192.168.1.106	H.225.C	94	masterSlaveDeterminationAck cs: empty
825	200.677364	192.168.1.106	192.168.1.104	H.225.C	132	masterSlaveDeterminationAck cs: facility
827	200.887234	192.168.1.106	192.168.1.104	H.225.C	149	openLogicalChannel CS: facility (g711U)
832	201.225598	192.168.1.104	192.168.1.106	H.225.C	118	openLogicalChannelAck CS: empty
845	206.708378	192.168.1.104	192.168.1.106	H.225.C	202	CS: connect
847	206.708457	192.168.1.104	192.168.1.106	H.225.C	202	[TCP Retransmission] CS: connect
849	206.708486	192.168.1.104	192.168.1.106	H.225.C	112	openLogicalChannel CS: empty (g711U)
851	206.709539	192.168.1.106	192.168.1.104	H.225.C	153	openLogicalChannelAck CS: facility
893	207.242147	192.168.1.106	192.168.1.104	H.225.C	153	openLogicalChannelAck [TCP Retransmission] CS: facility
895	207.246333	192.168.1.106	192.168.1.104	H.225.C	149	openLogicalChannel CS: facility (g711U)
913	207.760339	192.168.1.104	192.168.1.106	H.225.C	118	openLogicalChannelAck CS: empty
2316	214.945286	192.168.1.104	192.168.1.106	H.225.C	120	endSessionCommand CS: releaseComplete
2333	215.048803	192.168.1.104	192.168.1.106	H.225.C	149	RAS: disengageRequest
2334	215.054693	192.168.1.106	192.168.1.104	H.225.C	45	RAS: disengageConfirm

**Figure 3.18. capture des messages échangés entre les clients SIP et H.323**

**Capture message RAS:AdmissionRequest (ARQ) :** Envoyé par l'endpoint qui a le numéro 5000 avec l'adresse IP 192.168.1.104 pour demander l'autorisation à son Gatekeeper (qui est ici Asterisk) de participer à un appel

```

# Frame 811: 248 bytes on wire (1984 bits), 248 bytes captured (1984 bits)
# Ethernet II, Src: Arcadyan_e3:eb:15 (88:25:2c:e3:eb:15), Dst: HonHaiPr_11:30:f8 (4c:0f:6e:11:30:f8)
# Internet Protocol Version 4, Src: 192.168.1.104 (192.168.1.104), Dst: 192.168.1.106 (192.168.1.106)
# User Datagram Protocol, Src Port: 64340 (64340), Dst Port: h323gatestat (1719)
# H.225.0 RAS
  □ RasMessage: admissionRequest (9)
    □ admissionRequest
      requestSeqNum: 47056
    □ callType: pointToPoint (0)
      pointToPoint: NULL
      endpointIdentifier: 3473_endp
    □ destinationInfo: 3 items
      □ Item 0
        □ DestinationInfo item: h323-ID (1)
          h323-ID: user
      □ Item 1
        □ DestinationInfo item: dialleddigits (0)
          dialledDigits: 5000
      □ Item 2
        □ DestinationInfo item: dialleddigits (0)
          dialledDigits: 5000
    □ destCallSignalAddress: ipAddress (0)
      □ ipAddress
        ip: 192.168.1.104 (192.168.1.104)
        port: 1720
    □ srcInfo: 1 item
      □ Item 0
        □ AliasAddress: h323-ID (1)
          h323-ID: asterisk (1002, 1002) [192.168.1.106]
    □ srcCallSignalAddress: ipAddress (0)
      □ ipAddress
        ip: 192.168.1.106 (192.168.1.106)
        port: 12030
    bandwidth: 200000
    callReferenceValue: 17
    conferenceID: 33343536-3738-393a-3b3c-3d3e3f404142
    0... .... activeMC: False
    1... .... answerCall: True
    1... .... canMapAlias: True
  □ callIdentifier
    guid: 6f6f6833-3233-632d-5283-08000000006a
    gatekeeperIdentifier: OpenH323GK
    1... .... willSupplyUUIEs: True
    0... .... canMapSrcAlias: False
  [The response to this request is in frame 812]

```

**Figure 3.19. capture du détail de AdmissionRequest 2**

- L'utilisateur envoie une requête de permission d'appeler (ARQ) vers Asterisk.  
Cette requête comprend :
  - le type d'appel: Point to point
  - Identifiant unique de l'appelant: endpointIdentifier: 3473\_endp
  - N° du correspondant à appeler: 1002
  - N°E 164 de la source: 5000
  - N° unique qui permettra d'identifier l'appelant et l'appelé: CallReferenceValue 17
  - La bande passante bidirectionnelle que l'EP souhaite disposer: bandwidth : 200000 bits/s

**Capture message RAS: AdmissionConfirm (ACF)** : Retourné par Asterisk à l'utilisateur 5000 pour lui confirmer son admission à l'appel

```

# Frame 812: 83 bytes on wire (664 bits), 83 bytes captured (664 bits)
# Ethernet II, Src: HonHaiPr_11:30:f8 (4c:0f:6e:11:30:f8), Dst: Arcadyan_e3:eb:15 (88:25:2c:e3:eb:15)
# Internet Protocol Version 4, Src: 192.168.1.106 (192.168.1.106), Dst: 192.168.1.104 (192.168.1.104)
# User Datagram Protocol, Src Port: h323gatestat (1719), Dst Port: 64340 (64340)
# H.225.0 RAS
  □ RasMessage: admissionConfirm (10)
    □ admissionConfirm
      requestSeqnum: 47056
      bandwidth: 200000
      callModel: direct (0)
        direct: NULL
      destCallSignalAddress: ipAddress (0)
        ipAddress
          ip: 0.0.0.0 (0.0.0.0)
          port: 0
      irrFrequency: 120
      0... .... willRespondToIIRR: False
      □ uuiesRequested
        .0. .... setup: False
        ..0. .... callProceeding: False
        ...0 .... connect: False
        ....0... alerting: False
        ....0.. information: False
        ....0.. releaseComplete: False
        ....0 facility: False
        0... .... progress: False
        .0.... empty: False
        0... .... status: False
        0... .... statusInquiry: False
        0... .... setupAcknowledge: False
        0... .... notify: False
      □ serviceControl: 1 item
        □ Item 0
          □ ServiceControlSession
            sessionId: 0
            □ contents: callcreditServiceControl (3)
              □ callcreditServiceControl
                □ callStartingPoint: connect (1)
                  connect: NULL
                □ reason: refresh (1)
                  refresh: NULL
  [This is a response to a request in frame 811]
  [RAS Service Response Time: 0.002048000 seconds]

```

**Figure 3.20. capture du détail de AdmissionConfirm 2**

- Asterisk répond par un message d'acceptation (ACF) vers le port RAS de l'EP.  
Cette réponse comprend :
  - Bandwidth: 200000 bits/s bande passante maximum allouée pour cet appel
  - Mode d'appel: direct: appel direct
  - Fréquence des messages d'information (IIR) à envoyer au GK: iirfrequency: 120Hz

**Message RAS : InfoRequestResponse (IIR)** : Retourné soit sur demande, soit périodiquement par un endpoint à son Gatekeeper pour l'informer de son état (désactivé, dérangement, etc)

```

# Frame 821: 259 bytes on wire (2072 bits), 259 bytes captured (2072 bits)
# Ethernet II, Src: Arcadyan_e3:eb:15 (88:25:2c:e3:eb:15), Dst: HonHaiPr_11:30:f8 (4c:0f:6e:11:30:f8)
# Internet Protocol Version 4, Src: 192.168.1.104 (192.168.1.104), Dst: 192.168.1.106 (192.168.1.106)
# User Datagram Protocol, Src Port: 64340 (64340), Dst Port: h323gatestat (1719)
# H.225.0 RAS
  □ RasMessage: infoRequestResponse (22)
    □ infoRequestResponse
      requestSeqNum: 47057
    □ endpointType
      □ vendor
        □ vendor
          t35CountryCode: Australia (9)
          t35Extension: 0
          manufacturerCode: 61
          H.221 Manufacturer: Equivalence (OpenH323) (0x0900003d)
          productId: Duxoft H323Phone
          versionId: 6.0.118 (OpenH323 v1.19.0)
        terminal
        ..0. .... mc: False
        ...0 .... undefinedNode: False
      endpointIdentifier: 3473_endp
    □ rasAddress: ipAddress (0)
      □ ipAddress
        ip: 192.168.1.104 (192.168.1.104)
        port: 64340
    □ callSignalAddress: 1 item
      □ Item 0
        □ TransportAddress: ipAddress (0)
          □ ipAddress
            ip: 192.168.1.104 (192.168.1.104)
            port: 1720
    □ endpointAlias: 3 items
      □ Item 0
        □ AliasAddress: h323-ID (1)
          h323-ID: user
      □ Item 1
        □ AliasAddress: h323-ID (1)
          h323-ID: e164:5000
      □ Item 2
        □ AliasAddress: h323-ID (1)
          h323-ID: 5000
    □ perCallInfo: 1 item
      0... .... needResponse: False
      1... .... unsolicited: True

```

**Figure 3.21. capture du détail de InfoRequestResponse**