

TELEPHONIE SUR IP(ToIP)

Auteur : Abdoulaye NDIAYE

Professeur : M Samuel OUYA

Année : 2019 – 2020

Partie 1 : Aspect non technique de la téléphonie sur IP	3
I Concept de base de la ToIP	3
II Les solutions en environnement de ToIP	3
1- Box ADSL Pro	3
2- Mini Standard et Compte SIP	3
3- Standard téléphonique IP (IPBX)	3
III Les grands fournisseurs de solutions de ToIP.....	4
1- Asterisk	4
2- Avaya	4
3- Cisco.....	4
4- Aastra.....	4
5- Freeswitch	4
Partie 2 : Mise en place d'une Plateforme de ToIP avec Asterisk	4
I Présentation de Asterisk	4
II Installation et configuration	5
2- Configuration de Asterisk	9
3- Installation et configuration des clients	12
III Mise en place de quelques services avec asterisk.....	14
1- Gestion de boite vocale	14
2- Gestion de la conférence.....	15
3- Gestion de la musique d'attente	16
4- Gestion du transfert d'appel	19
5- Gestion des packages d'appel	20
6- Gestion des files d'attente d'un centre d'appel.....	21
Partie 3 : Mise en place d'une plateforme de ToIP en environnement CISCO	25

I La téléphonie en environnement CISCO	25
II Configuration d'un CUCME ou CME	26
1- Configuration du CME comme contrôleur SCCP	26
2- Configuration du CME comme contrôleur SIP	29
III Asterisk comme contrôleur SCCP	32
1- Téléchargement et installation du module SCCP	32
2- Installation du serveur TFTP	33
3- Configuration de asterisk comme contrôleur SCCP	35
4- Configurations des téléphones et test d'appel	38
Partie 4 : Mise en place d'une Plateforme de ToIP avec FreeSWITCH.....	41
I Présentation de FreeSWITCH	41
II Installation de FreeSWITCH	41
III Configuration des téléphones SIP et test d'appel	46
IV Fonctionnalités de base de FreeSWITCH	48
1- Ajout de nouveau compte utilisateurs	48
2- Définition du plan de numérotation	51
3- Gestion des conférences	52
4- Gestion des files d'attente d'un centre d'appel.....	57
a) Gestion des files d'attentes	58
b) Gestion des agents.....	59
c) Gestion des membres d'un centre d'appel	60
d) Quelques commandes utiles de gestion de centre d'appels et Test	60
e) Définition des numéros de connexion et de déconnexion d'un agent.....	64
V Paramétrage et utilisation de Verto.....	66
1- Présentation de Verto	66
2- Architecture de Verto freeswitch.....	66
3- Configuration de webrtc et de verto	66
4- Test de connexion avec Google chrome	69
Partie 5 : Interconnexion de asterisk et FreeSWITCH	72
1- Gestion des appels entrants d'asterisk vers FreeSWITCH.....	72
2- Gestion des appels sortants de FreeSWITCH vers asterisk	73

Partie 1 : Aspect non technique de la téléphonie sur IP

I Concept de base de la ToIP

La téléphonie sur IP (en anglais, telephony over IP ou IP telephony) est un mode de téléphonie dans lequel la voix est numérisée puis acheminée par le protocole TCP/IP sous forme de paquet de données. Ce service de téléphonie est offert sur un réseau de télécommunications, public ou privé utilisant principalement le protocole IP.

La téléphonie IP définit l'utilisation de liens Internet pour acheminer des appels téléphoniques d'une personne à une autre. Un appel téléphonique de type IP diffère de la téléphonie conventionnelle (RTC) dans l'encodage de la voix. Cette technologie repose totalement sur un transport VoIP.

Dans le système traditionnel, la voix est encodée de façon analogique et numérique et transmise sur un réseau de commutation de circuit alors que dans le système IP, la voix est encodée en format numérique et mis en paquets sous format IP.

En fait, la téléphonie sur IP utilise la même méthode (processus) que pour la transmission de l'information sur le réseau Internet. C'est-à-dire une fois la voix formatée, on peut la transmettre sur un lien Internet commun ou encore l'envoyer sur des liens dédiés

II Les solutions en environnement de ToIP

1- Box ADSL Pro

C'est une solution viable pour une petite structure : elle permet le partage sur un seul équipement la « box » ADSL d'un accès Internet et d'une ligne de téléphone (voire deux selon l'opérateur). L'offre est très économique avec un seul abonnement comprenant un forfait téléphonique illimité et l'accès à Internet.

Mais attention ! En cas d'incident sur la ligne téléphonique support de l'ADSL, il n'y a aucune garantie de temps de rétablissement (GTR) et l'abonné peut rester plusieurs jours, voire plusieurs semaines, sans aucun service...

2- Mini Standard et Compte SIP

Le mini-standard se connecte via l'Internet à un opérateur VoIP auprès duquel vous souscrivez des abonnements SIP. Vous obtenez ainsi les possibilités classiques d'un standard téléphonique qui vous permet de distribuer plusieurs appels simultanés (typiquement 3 ou 4) vers un parc de téléphones sans fil DECT (typiquement 6 ou 8).

Même remarque que pour la solution précédente : la perte de l'accès à Internet signifie la perte du service téléphonique

3- Standard téléphonique IP (IPBX)

Dans l'industrie des télécommunications, on désigne par **PABX IP (PBX IP** ou encore **IPBX**; de l'anglais *Internet Protocol Private Branch eXchange*) un autocommutateur téléphonique privé utilisant le protocole internet (IP) pour gérer les appels téléphoniques d'une

entreprise, en interne sur son réseau local (LAN). Couplé à des technologies de voix sur IP, les communications téléphoniques peuvent ainsi être acheminées sur le réseau étendu (WAN) de l'entreprise

III Les grands fournisseurs de solutions de ToIP

1- Asterisk

Asterisk est un PABX-IP open source créée en 1999 par Mark Spencer fondateur de la société Digium. Asterisk est publié sous licence GPL. Il est aujourd'hui adopté sur une majorité de plateformes VoIP du marché

2- Avaya

Avaya est leader mondial dans les systèmes de communications d'entreprise. Les solutions de Communications Unifiées d'Avaya simplifient et rationalisent la multiplication des technologies auxquelles sont confrontées quotidiennement les entreprises

3- Cisco

Les solutions de Communications IP Cisco s'intègrent dans une approche système globale, qui associe les forces d'une infrastructure de données IP Cisco aux applications de Communications IP. Déployer une solution de Communications IP reposant entièrement sur une infrastructure Cisco permet de bénéficier de l'intégration fonctionnelle de ces applications IP avec l'infrastructure réseau IP sous-jacente

4- Aastra

Aastra Technologies Limited est un groupe spécialisé dans la communication d'entreprise. Basé à Concord en Ontario au Canada, Aastra développe et commercialise des solutions intégrées à destination des entreprises de toute taille. La gamme de logiciels basés sur les technologies traditionnelles et IP d'Aastra couvre les besoins en communication et collaboration des entreprises

5- Freeswitch

FreeSWITCH est un serveur d'applications gratuit et open-source pour la communication en temps réel, WebRTC, les télécommunications, la vidéo et le protocole Voice over Internet (VoIP). Multiplateforme, il fonctionne sous Linux, Windows, macOS et FreeBSD

[Partie 2 : Mise en place d'une Plateforme de ToIP avec Asterisk](#)

I Présentation de Asterisk

Asterisk est un autocommutateur téléphonique privé (PABX) libre et propriétaire (publié sous double licence GPLv2 ou commerciale⁶) pour systèmes GNU/Linux. Il permet, entre autres, la messagerie vocale, les files d'attente, les agents d'appels, les musiques d'attente et les mises en garde d'appels, la distribution des appels. Il est possible également d'ajouter l'utilisation des conférences par le biais de l'installation de modules supplémentaires et la recompilation des binaires.

Asterisk implémente les protocoles H.320, H.323 et SIP, ainsi qu'un protocole spécifique nommé IAX (Inter-Asterisk eXchange). Ce protocole IAX permet la communication entre deux serveurs Asterisk ainsi qu'entre client et serveur Asterisk. Asterisk peut également jouer le rôle de *registrar* et passerelle avec les réseaux publics (RTC, GSM, etc.) Asterisk est extensible par des scripts ou des modules en langage Perl, C, Python, PHP, et Ruby.

II Installation et configuration

1- Installation de Asterisk

Il y a deux façons d'installer **Asterisk** soit par la commande **apt-get install asterisk** soit par source c'est-à-dire installé la version tar de asterisk puis le compiler comme on le fera avec la version 16 de Asterisk

Après avoir téléchargé le fichier tar on le désarchive dans le dossier `/usr/local/src` qui un dossier de notre choix avec la commande tar

```
root@ubuntu-VirtualBox:/usr/local/src# tar -xvzf asterisk-16-current.tar.gz
asterisk-16.7.0/
asterisk-16.7.0/.cleancount
asterisk-16.7.0/.gitignore
asterisk-16.7.0/.gitreview
asterisk-16.7.0/.lastclean
asterisk-16.7.0/.version
```

Ensuite on se déplace dans le dossier `asterisk/contrib/scripts` pour installer les prérequis de asterisk avec le script **install_prereq**

```
root@ubuntu-VirtualBox:/usr/local/src# cd asterisk-16.7.0/contrib/scripts/
root@ubuntu-VirtualBox:/usr/local/src/asterisk-16.7.0/contrib/scripts# ls
addons                live_ast
agents.php            loadtest.tcl
astcli                lookup.agi
ast_coredumper        managerproxy.pl
asterisk.ldap-schema messages-expire.pl
asterisk.ldif         qview.pl
asterisk.logrotate   README.messages-expire
astgenkey             refcounter.py
astgenkey.8           reflocks.py
ast_grab_core         refstats.py
ast_logescalator     retrieve_extensions_from_mysql.pl
ast_loggrabber       retrieve_extensions_from_sql.pl
ast_tls_cert         retrieve_sip_conf_from_mysql.pl
astversion           safe_asterisk
autosupport          safe_asterisk.8
autosupport.8        safe_asterisk_restart
clang-scan-build     sip_nat_settings
dahdi_span_config_hook sipp-sendfax.xml
dbsep.cgi            sip_to_pjsip
file.convert.sh      spandspflow2pcap.log
get_ilbc_source.sh   spandspflow2pcap.py
get_mp3_source.sh    valgrind_compare
get_swagger_ui.sh    vmail.cgi
import-cdr-csv-mysql.pl voicemailpwcheck.py
install_prereq
```

Pour exécuter ce script on tape la commande `./install_prereq install` une fois l'installation terminée on a le message comme montre l'image ci-dessous

```
..
#####
## install completed successfully
#####
root@ubuntu-VirtualBox: /usr/local/src/asterisk-17.1.0/contrib/scripts#
```

Pour pouvoir jouer du mp3 on installe aussi le script `get_mp3_source.sh`

```
root@ubuntu-VirtualBox: /usr/local/src/asterisk-16.7.0/contrib/scripts# ./get_mp3_source.sh
```

On peut installer les paquets `libopus-dev libvpx-dev` pour la prise en charge des nouveaux codecs opus et vp8

Ensuite on revient dans le dossier source de asterisk `/usr/local/src/asterisk` et taper les commande suivante

✓ `./configure --with-jansson-bundled`

Si la configuration s'est bien passée vous aurez le logo de asterisk comme montre l'image si dessous

```

      .$$$$$$$$$$$$$$$$=..
     .7$7..           .7$$7:.
    .$$:.            ,7.7
   .7.             7$$$$ .$$77
  ..$$           $$$$$ .$$$7
 .7$   ?.   $$$$$   ?.   7$$$
 $.$.   .$$$7.  $$$$7  .7$$$   .$$$
 .777.   .$$$$$77$$$$77$$$$$7.   $$$
 $$$~   .7$$$$$$$$$$$$$7.   .$$$
 .$$7   .7$$$$$$$$7:   ?$$$
 $$$    ?7$$$$$$$$$$$I   .$$$7
 $$$   .7$$$$$$$$$$$$$$$   :$$$
 $$$   $$$$$$7$$$$$$$$$$$   .$$$
 $$$   $$$ 7$$$7  .$$$   .$$$
 $$$$   $$$$7   .$$$   .$$$
 7$$$7   7$$$7   7$$$
  $$$$$   $$$$
  $$$$7.   $$$$ (TM)
  $$$$$$.   .7$$$$$  $$
  $$$$$$$$$$7$$$$$$$$. $$$$$$
  $$$$$$$$$$$$$$.

```

```

configure: Package configured for:
configure: OS type : linux-gnu
configure: Host CPU : x86_64
configure: build-cpu:vendor:os: x86_64 : pc : linux-gnu :
```

✓ **make menuselect** pour choisir les fonctionnalités

On active la prise en charge du format mp3 pour pouvoir jouer de la musique sur asteris

```
Asterisk Module and Build Option Selection
Add-ons (See README-addons.txt)
Applications [ ] chan_mobile
Bridging Modules [*] chan_oo323
Call Detail Recording [*] format.mp3
Channel Event Logging [*] res_config_mysql
Channel Drivers --- Deprecated ---
Codec Translators [ ] app_mysql
Format Interpreters [ ] cdr_mysql
Dialplan Functions
PBX Modules
Resource Modules
```

On active aussi la prise en charge du codec audio opus

```
Asterisk Module and Build Option Selection
Add-ons (See README-addons.txt)
Applications [*] codec_a_mu
Bridging Modules [*] codec_adpcm
Call Detail Recording [*] codec_alaw
Channel Event Logging [*] codec_codec2
Channel Drivers XXX codec_dahdi
Codec Translators [*] codec_g722
Format Interpreters [*] codec_g726
Dialplan Functions [*] codec_gsm
PBX Modules [*] codec_ilbc
Resource Modules [*] codec_lpc10
Test Modules [*] codec_resample
Compiler Flags [*] codec_speex
Voicemail Build Options [*] codec_ulaw
Utilities --- External ---
AGI Samples [*] codec_opus
Core Sound Packages [ ] codec_silk
Music On Hold File Packages [ ] codec_siren7
Extras Sound Packages [ ] codec_siren14
[ ] codec_g729a
```

On active aussi le son en français comme montre les images ci-dessous

```
Asterisk Module and Build Option Selection
Add-ons (See README-addons.txt) [ ] CORE-SOUNDS-ES-ULAW
Applications [*] CORE-SOUNDS-ES-ALAW
Bridging Modules [ ] CORE-SOUNDS-ES-GSM
Call Detail Recording [ ] CORE-SOUNDS-ES-G729
Channel Event Logging [ ] CORE-SOUNDS-ES-G722
Channel Drivers [ ] CORE-SOUNDS-ES-SLN16
Codec Translators [ ] CORE-SOUNDS-ES-SIREN7
Format Interpreters [ ] CORE-SOUNDS-ES-SIREN14
Dialplan Functions [*] CORE-SOUNDS-FR-WAV
PBX Modules [*] CORE-SOUNDS-FR-ULAW
Resource Modules [*] CORE-SOUNDS-FR-ALAW
Test Modules [*] CORE-SOUNDS-FR-GSM
Compiler Flags [*] CORE-SOUNDS-FR-G729
Voicemail Build Options [*] CORE-SOUNDS-FR-G722
Utilities [*] CORE-SOUNDS-FR-SLN16
AGI Samples [*] CORE-SOUNDS-FR-SIREN7
Core Sound Packages [ ] CORE-SOUNDS-FR-SIREN14
Music On Hold File Packages [ ] CORE-SOUNDS-IT-WAV
Extras Sound Packages [ ] CORE-SOUNDS-IT-ULAW
[ ] CORE-SOUNDS-IT-ALAW
[ ] CORE-SOUNDS-IT-GSM
[ ] CORE-SOUNDS-IT-G729
Spanish, a-Law format
```

```

Asterisk Module and Build Option Selection

Add-ons (See README-addons.txt)      --- Core ---
Applications                          [*] MOH-OPSOUND-WAV
Bridging Modules                      [*] MOH-OPSOUND-ULAW
Call Detail Recording                 [*] MOH-OPSOUND-ALAW
Channel Event Logging                [*] MOH-OPSOUND-GSM
Channel Drivers                      [*] MOH-OPSOUND-G729
Codec Translators                    [*] MOH-OPSOUND-G722
Format Interpreters                  [*] MOH-OPSOUND-SLN16
Dialplan Functions                   [*] MOH-OPSOUND-SIREN7
PBX Modules                          [*] MOH-OPSOUND-SIREN14
Resource Modules
Test Modules
Compiler Flags
Voicemail Build Options
Utilities
AGI Samples
Core Sound Packages
Music On Hold File Packages
Extras Sound Packages

opsound.org Music On Hold Files, G.722.1C (Siren14) format

```

```

Asterisk Module and Build Option Selection

Add-ons (See README-addons.txt)      [ ] EXTRA-SOUNDS-EN-G722
Applications                          [ ] EXTRA-SOUNDS-EN-SLN16
Bridging Modules                      [ ] EXTRA-SOUNDS-EN-SIREN7
Call Detail Recording                 [ ] EXTRA-SOUNDS-EN-SIREN14
Channel Event Logging                [ ] EXTRA-SOUNDS-EN_GB-WAV
Channel Drivers                      [ ] EXTRA-SOUNDS-EN_GB-ULAW
Codec Translators                    [ ] EXTRA-SOUNDS-EN_GB-ALAW
Format Interpreters                  [ ] EXTRA-SOUNDS-EN_GB-GSM
Dialplan Functions                   [ ] EXTRA-SOUNDS-EN_GB-G729
PBX Modules                          [ ] EXTRA-SOUNDS-EN_GB-G722
Resource Modules                     [ ] EXTRA-SOUNDS-EN_GB-SLN16
Test Modules                         [ ] EXTRA-SOUNDS-EN_GB-SIREN7
Compiler Flags                       [ ] EXTRA-SOUNDS-EN_GB-SIREN14
Voicemail Build Options              [*] EXTRA-SOUNDS-FR-WAV
Utilities                            [*] EXTRA-SOUNDS-FR-ULAW
AGI Samples                          [*] EXTRA-SOUNDS-FR-ALAW
Core Sound Packages                  [*] EXTRA-SOUNDS-FR-GSM
Music On Hold File Packages          [*] EXTRA-SOUNDS-FR-G729
Extras Sound Packages                 [*] EXTRA-SOUNDS-FR-G722
                                      [*] EXTRA-SOUNDS-FR-SLN16
                                      [*] EXTRA-SOUNDS-FR-SIREN7
                                      [*] EXTRA-SOUNDS-FR-SIREN14

French, G.722.1C (Siren14) format

```

- ✓ **make** pour compiler asterisk
- ✓ **make install** pour installer les fonctionnalités
- ✓ **make samples** pour installer les fichiers de configurations
- ✓ **make config** pour générer les scripts de démarrage de asterisk

Maintenant l'installation de asterisk est terminée on tape la commande **asterisk** pour démarrer le serveur et pour entrer dans la console d'asterisk on tape la commande **asterisk -rvvvvvvvvv**

```

root@ubuntu-VirtualBox:~# asterisk -rvvvvvvvvvvvvvvvv
Asterisk 16.7.0, Copyright (C) 1999 - 2018, Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
Asterisk comes with ABSOLUTELY NO WARRANTY; type 'core show warranty' for details.
This is free software, with components licensed under the GNU General Public
License version 2 and other licenses; you are welcome to redistribute it under
certain conditions. Type 'core show license' for details.
=====
Connected to Asterisk 16.7.0 currently running on ubuntu-VirtualBox (pid = 4216)
ubuntu-VirtualBox*CLI>

```

2- Configuration de Asterisk

Après installation de asterisk ces fichiers de configuration se trouve dans **/etc/asterisk**

```

root@ubuntu-VirtualBox:~# cd /etc/asterisk/
root@ubuntu-VirtualBox:/etc/asterisk# ls
acl.conf                confbridge.conf        pjsip.conf
adsi.conf               config_test.conf       pjsip_notify.conf
agents.conf             console.conf           pjsip_wizard.conf
alarmreceiver.conf     dbsep.conf             queuerules.conf
alsa.conf              dnsmgr.conf            queues.conf
amd.conf               dsp.conf               res_config_mysql.conf
app_mysql.conf         dundi.conf             res_config_sqlite3.conf
app_skel.conf          enum.conf              res_config_sqlite.conf
ari.conf               extconfig.conf         res_corosync.conf
ast_debug_tools.conf  extensions.ael         res_curl.conf
asterisk.adsi          extensions.conf        res_fax.conf
asterisk.conf          extensions.lua         res_ldap.conf
calendar.conf          extensions_minivm.conf res_ldap.conf
ccss.conf              features.conf          res_odbc.conf
                                              resolver_unbound.conf

```

Nous allons faire un petit récapitulatif de quelques fichiers importants de asterisk dans le tableau suivant

Fichiers dans /etc/asterisk	significations
sip.conf	Fichier dans lequel, on crée des comptes ; dans sa section [general], on trouve les paramètres tels videosupport=yes permettant d'activer la visiophonie et le paramètre language qu'il faut mettre à fr pour activer le son en français
extensions.conf	C'est dans ce fichier qu'on crée les groupes d'appels appelés context, les numéros de téléphones et on fait les différentes actions
modules.conf	c'est ici qu'on peut charger ou décharger des modules qui activent ou désactivent des fonctionnalités telle que prise en charge ou non de sip
voicemail.conf	Fichier dans lequel, on crée les contextes de boîtes vocales et les boîtes vocales
musiconhold.conf	c'est dans ce fichier qu'on crée les classes de musique qu'on peut utiliser comme musique d'attente ou sonnerie
confbridge.conf	Fichier dans lequel on crée des conférences audio et vidéo

queues.conf	Fichier dans lequel on crée des files d'attente des centres d'appels
agents.conf	Fichier de création des agents répondant aux appels dans les centres d'appels
features.conf	Fichier ou on peut trouver des combinaisons de touches a faire pour utiliser certaines fonctionnalités d'asterisk telles que : enregistrements d'appels, fonction patron secrétaire, mise en attente d'appels(parking

Ensuite après avoir téléchargé les sons français on active la langue française dans le fichier sip.conf en décommentant le paramètre **language** et le mettre en **fr** au lieu de en

```

;
;mohsuggest=default
;
;parkinglot=plaza           ; Sets the default parking lot for call parking
                           ; This may also be set for individual users/peer
s
                           ; Parkinglots are configured in features.conf
language=fr                ; Default language setting for all users/peers
                           ; This may also be set for individual users/peer
s
;tonezone=se               ; Default tonezone for all users/peers
                           ; This may also be set for individual users/peer
s
;relaxdtmf=yes             ; Relax dtmf handling
;trustpid = no             ; If Remote-Party-ID should be trusted
;sendrpid = yes            ; If Remote-Party-ID should be sent (defaults to
no)

```

Maintenant que notre serveur est bien installé et parle la langue français on peut créer des comptes et des numéros de téléphone pour commencer à s'appeler.

La création de compte se fait dans le fichier **sip.conf** comme le montre l'image ci-dessous

[abdoulaye] c'est le nom du compte

defaultuser= abdoulaye c'est nom de l'utilisateur

secret=passer c'est le mot de passe de l'utilisateur

type=friend signifie que l'utilisateur est autorisé à appeler et recevoir

context=ToIP c'est le groupe d'appel de l'utilisateur

host=dynamic signifie que l'utilisateur peut appeler ou recevoir avec n'importe quelle adresse IP

Dans cette image on vient de créer des comptes aux utilisateurs Abdoulaye, pape saliou et ismaila

```

[abdoulaye]
defaultuser=abdoulaye
type=friend
secret=passer
host=dynamic
context=ToIP

[pape saliou]
defaultuser=pape saliou
type=friend
secret=passer
host=dynamic
context=ToIP

[ismaila]
defaultuser=ismaila
type=friend
secret=passer
host=dynamic
context=ToIP

```

Pour créer des numéros de téléphones aux groupe d'appel il faut aller dans le fichier **extensions.conf**. Voici la syntaxe pour créer des numéro dans ce fichier
exten=>numero,priorite,action

Parmi les actions on peut citer

- ✓ Dial () qui veut dire appeler
- ✓ Hangup qui veut dire raccrocher
- ✓ Answer() qui veut dire décrocher
- ✓ Voicemail() qui veut dire envoyer l'appelant vers une boite vocal
- ✓ VoicemailMain() qui permet à un utilisateur de consulter sa boite vocal

Nous allons affecter les numéros 1000,1001 et 1002 aux utilisateurs abdoulaye, pape saliou et ismaila avec les actons appeler et raccrocher

```

[ToIP]
exten =>1000,1,Dial(SIP/abdoulaye,40,tr)
exten =>1000,2,hangup

exten =>1001,1,Dial(SIP/pape saliou,40,tr)
exten =>1001,2,hangup

exten =>1002,1,Dial(SIP/ismaila,40,tr)
exten =>1000,2,hangup

```

On peut aussi activer la prise en charge des appels vidéo dans le fichier **sip.conf** en décommentant le paramètre **videosupport** et le mettre à **yes** comme suit

```

;compactheaders = yes          ; send compact sip headers.
;
videosupport=yes              ; Turn on support for SIP video. You need to turn
this                          ; on in this section to get any video support at
all.                           ;
                                ; You can turn it off on a per peer basis if the
general

```

3- Installation et configuration des clients

Les clients SIP permet aux utilisateurs de se connecter sur leurs comptes pour appeler et recevoir des appels. Parmi ces clients on peut citer MicroSip, X-LITE, Zoiper, CSipSimple.

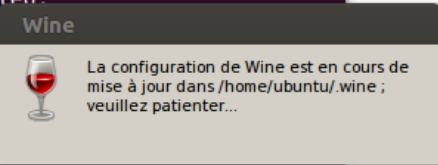
Nous allons installer MicroSip sous Linux et zoiper sous Android pour tester les appels. Nous allons aller sur le site : <https://www.microsip.org/downloads> pour télécharger le fichier exécutable de MricroSip. Ensuite on installe wine permettant d'installer des logiciels .exe sous Linux en installant les paquets **wine** et **wine-development**

Ensuite on installe MicroSip en ligne de commande en tant que simple utilisateur comme montre l'image ci-dessous

```

ubuntu@ubuntu-VirtualBox:~$ wine /home/ubuntu/Documents/MicroSIP-3.19.28.exe
wine: created the configuration directory '/home/ubuntu/.wine'
fixme:storage:create_storagefile Storage share mode not implemented.
err:mSCOREE:LoadLibraryShim error reading registry key for installroot
err:mSCOREE:LoadLibraryShim error reading registry key for installroot
err:mSCOREE:LoadLibraryShim error reading registry key for installroot
err:mSCOREE:LoadLibraryShim error reading registry key for installroot
fixme:storage:create_storagefile Storage share mode not implemented.
err:mhtml:check_version Unexpected version "", expected "Wine Gecko 2.21"
Could not load wine-gecko. HTML rendering will be disabled.
err:advpack:create_tmp_ini_file Write failed
err:advpack:create_tmp_ini_file Write failed
err:shell:SHGetFolderPathAndSubDirW Failed to create directory (x86)
fixme:setupapi:get_csidl_dir CSIDL 2a not found
err:shell:SHGetFolderPathAndSubDirW Failed to create directory (x86)

```



Après l'installation on peut configurer le compte de ismaila comme le montre l'image ci-dessous

Compte

Nom du compte: ismaila

Serveur SIP: 192.168.1.111

Proxy SIP: 192.168.1.111

Nom utilisateur*: ismaila

Domaine*: 192.168.1.111

Login: ismaila

Mot de passe: ●●●●●

[Afficher le mot de passe](#)

Afficher le nom:

Numéro de la boîte ...:

Préfixe d'appel:

Dial Plan:

Hide Caller ID

Chiffrement: Désactivé

Transport: Auto (UDP & TCP)

Adresse publique: Auto

Actualiser l'enregist...: 300 Garder en direct: 15

Afficher ma présence

Autoriser la ré-écriture de l'IP

ICE

Désactiver les temporisateurs de ses

Ensuite on se connecte sur le client android avec la compte de abdoulaye et vois que les deux utilisateurs peuvent s'appeler

```

== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
> 0x7fcfb0015380 -- Strict RTP learning after remote address set to: 192.
168.1.25:40006
-- Executing [1002@ToIP:1] Dial("SIP/abdoulaye-00000006", "SIP/ismaila,40,tr
") in new stack
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
-- Called SIP/ismaila
-- SIP/ismaila-00000007 is ringing
ubuntu-VirtualBox*CLI>

```

Appel entrant

abdoulaye

abdoulaye@192.168.1.111
Asterisk PBX 16.7.0

à: ismaila@192.168.1.26

close window to ignore call

III Mise en place de quelques services avec asterisk

1- Gestion de boite vocale

Le principe de boite vocal consiste à laisser un message vocal à un utilisateur lorsqu'il est injoignable. Pour sa mise en place sous asterisk il y'a deux actions à faire

- ✓ Créer les numéros de boites vocales dans le fichier **voicemail.conf** comme montre l'image ci-dessous

La syntaxe est la suivante : **numero de boite => mot de passe, nom user ,email user**

```
[context_bv]
1000 => 1234,abdoulaye
1001 => 2345,pape saliou
1002 => 3456,ismaila
```

- ✓ Affecter les boites aux utilisateurs dans le fichier **extensions.conf** avec l'action **Voicemail** comme priorité 2

```
[ToIP]
exten =>1000,1,Dial(SIP/abdoulaye,40,tr)
exten =>1000,2,Voicemail(1000@context_bv)
exten =>1000,3,hangup

exten =>1001,1,Dial(SIP/pape saliou,40,tr)
exten =>1001,2,Voicemail(1001@context_bv)
exten =>1001,3,hangup

exten =>1002,1,Dial(SIP/ismaila,40,tr)
exten =>1002,2,Voicemail(1002@context_bv)
exten =>1003,3,hangup
```

Maintenant l'utilisateur Ismaila qui appelle abdoulaye qui ne décroche pas au bout de 40 seconde le serveur lui communique de lui laisser un message vocal ensuite d'appuyer sur Diez(#) ou bien de raccrocher

```

-- Executing [1000@ToIP:1] Dial("SIP/ismaila-00000001", "SIP/abdoulaye,40,tr
") in new stack
== Using SIP RTP CoS mark 5
-- Called SIP/abdoulaye
[Apr 21 12:09:55] WARNING[1951]: chan_sip.c:4119 retrans_pkt: Retransmission tim
eout reached on transmission 723b47e5142af20330c3da0f569e6081@192.168.1.12:5060
for seqno 102 (Critical Request) -- See https://wiki.asterisk.org/wiki/display/A
ST/SIP+Retransmissions
Packet timed out after 31999ms with no response
[Apr 21 12:09:55] WARNING[1951]: chan_sip.c:4143 retrans_pkt: Hanging up call 72
3b47e5142af20330c3da0f569e6081@192.168.1.12:5060 - no reply to our critical pack
et (see https://wiki.asterisk.org/wiki/display/AST/SIP+Retransmissions).
== Everyone is busy/congested at this time (1:0/0/1)
-- Executing [1000@ToIP:2] VoiceMail("SIP/ismaila-00000001", "1000@context_b
v") in new stack
> 0x7f308800a1b0 -- Strict RTP learning after remote address set to: 192.
168.1.11:4000
-- <SIP/ismaila-00000001> Playing 'vm-intro.ulaw' (language 'fr')
> 0x7f308800a1b0 -- Strict RTP switching to RTP target address 192.168.1.
11:4000 as source
> 0x7f308800a1b0 -- Strict RTP learning complete - Locking on source addr
ess 192.168.1.11:4000
-- <SIP/ismaila-00000001> Playing 'beep.ulaw' (language 'fr')
-- Recording the message
-- x=0, open writing: /var/spool/asterisk/voicemail/context_bv/1000/tmp/rRW
tMi format: wav49, 0x7f3088038470

```

Ensuite nous allons définir un autre numéro qui va permettre à un utilisateur de consulter sa boîte vocale avec l'action VoicemailMain comme montre l'image suivante

```
exten =>602,1,VoicemailMain(@context_bv)
```

Maintenant l'utilisateur Abdoulaye peut composer le numéro 602 pour consulter son message vocal. Le serveur lui demande d'entrer son numéro de boîte vocal et son mot de passe. Ces informations ont été définies dans **voicemail.conf**

```

-- Executing [602@ToIP:1] VoiceMailMain("SIP/abdoulaye-00000003", "@context_
bv") in new stack
> 0x7f308800a1b0 -- Strict RTP learning after remote address set to: 192.
168.1.10:40002
> 0x7f308800a1b0 -- Strict RTP switching to RTP target address 192.168.1.
10:40002 as source
-- <SIP/abdoulaye-00000003> Playing 'vm-login.ulaw' (language 'fr')
> 0x7f308800a1b0 -- Strict RTP learning complete - Locking on source addr
ess 192.168.1.10:40002
-- <SIP/abdoulaye-00000003> Playing 'vm-password.ulaw' (language 'fr')
-- <SIP/abdoulaye-00000003> Playing 'vm-youhave.ulaw' (language 'fr')
-- <SIP/abdoulaye-00000003> Playing 'digits/1.ulaw' (language 'fr')
-- <SIP/abdoulaye-00000003> Playing 'vm-INBOX.ulaw' (language 'fr')
-- <SIP/abdoulaye-00000003> Playing 'vm-and.ulaw' (language 'fr')
-- <SIP/abdoulaye-00000003> Playing 'digits/1.ulaw' (language 'fr')
-- <SIP/abdoulaye-00000003> Playing 'vm-Old.ulaw' (language 'fr')
-- <SIP/abdoulaye-00000003> Playing 'vm-message.ulaw' (language 'fr')
-- <SIP/abdoulaye-00000003> Playing 'vm-onefor.ulaw' (language 'fr')

```

2- Gestion de la conférence

La conférence consiste à mettre en communications plus de trois personnes. Pour sa mise en place il y'a deux actions à faire

- ✓ Créer la conférence dans le fichier **confbridge.conf** comme le montre l'image suivante

```
[conference_asterisk]
type=bridge
video_mode=sfu
```

- ✓ Définir un numéro pour participer à la conférence dans le fichier **extensions.conf** avec l'action Confbridge comme montre l'image suivant

```
exten =>6000,1,Confbridge(conference_asterisk)
```

Maintenant les utilisateurs qui veulent participer à la conférence appellent le numéro 6000

```
-- Executing [6000@ToIP:1] ConfBridge("SIP/pape saliou-00000006", "conferenc
e_asterisk") in new stack
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
> 0x7fa7cc06c7a0 -- Strict RTP learning after remote address set to: 192.
168.1.25:40006
-- Executing [6000@ToIP:1] ConfBridge("SIP/abdoulaye-00000007", "conference_
asterisk") in new stack
> 0x7fa7cc07cd10 -- Strict RTP switching to RTP target address 192.168.1.
8:33790 as source
> 0x7fa7cc06c7a0 -- Strict RTP learning after remote address set to: 192.
168.1.25:40006
> 0x7fa7cc006300 -- Strict RTP switching to RTP target address 192.168.1.
8:33792 as source
> 0x7fa7cc06c7a0 -- Strict RTP switching to RTP target address 192.168.1.
25:40006 as source
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
> 0x7fa7cc031f00 -- Strict RTP learning after remote address set to: 192.
168.1.3:4006
-- Executing [6000@ToIP:1] ConfBridge("SIP/ismaila-00000008", "conference_as
terisk") in new stack
-- Channel CBAnn/conference_asterisk-00000002;2 joined 'softmix' base-bridge
<9ee6a6c3-1b1e-47c5-ad6b-b40c2599cf5b>
-- <SIP/abdoulaye-00000007> Playing 'conf-onlyperson.ulaw' (language 'fr')
-- <SIP/pape saliou-00000006> Playing 'confbridge-join.ulaw' (language 'fr')
```

3- Gestion de la musique d'attente

La musique d'attente est une des fonctionnalités les plus utilisées dans la téléphonie. Le principe est simple, faire patienter un utilisateur avec une musique plutôt qu'un blanc.

Pour sa mise en place il y'a deux action à faire

- ✓ Créer la classe de musique dans le fichier **musiconhold.conf** comme montre l'image suivante

```
[wally]
mode=custom
directory=/var/lib/asterisk/mohmp3
application=/usr/bin/mpg123 -q -r 8000 -f 8192 -b 2048 --mono -s
```

[wally] est le nom de notre classe de musique

Ensuite nous allons créer le dossier mohmp3 dans var/lib/asterisk ou nous allons mettre notre fichier mp3

```
root@ubuntu-VirtualBox:~# cd /var/lib/asterisk/
root@ubuntu-VirtualBox:/var/lib/asterisk# mkdir mohmp3
root@ubuntu-VirtualBox:/var/lib/asterisk# cd mohmp3/
root@ubuntu-VirtualBox:/var/lib/asterisk/mohmp3# cp /home/ubuntu/Musique/wally.mp3 .
root@ubuntu-VirtualBox:/var/lib/asterisk/mohmp3# ls
wally.mp3
```

Et nous allons installer le lecteur mp3 avec le paquet **mpg123**

```
root@ubuntu-VirtualBox:/etc/asterisk# apt-get install mpg123
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
mpg123 est déjà la version la plus récente (1.22.4-1ubuntu0.1).
0 mis à jour, 0 nouvellement installés, 0 à enlever et 65 non mis à jour.
```

- ✓ Définir un numéro dans **extensions.conf** pour écouter de la musique avec l'action **musicOnhold** comme le montre l'image suivante

```
exten =>3000,1,Answer()
exten =>3000,2,musicOnhold(wally)
```

La première action Answer() permet au serveur de décrocher avec de jouer la musique. Maintenant quand un utilisateur appelle le numéro 3000 le serveur lui fait écouter la musique de la classe wally

```
> 0x7f8184048960 -- Strict RTP learning after remote address set to: 192.168.1.25:40014
-- Executing [3000@ToIP:1] Answer("SIP/abdoulaye-00000005", "") in new stack
> 0x7f8184048960 -- Strict RTP learning after remote address set to: 192.168.1.25:40014
> 0x7f8184048960 -- Strict RTP switching to RTP target address 192.168.1.25:40014 as source
-- Executing [3000@ToIP:2] MusicOnHold("SIP/abdoulaye-00000005", "wally") in new stack
-- Started music on hold, class 'wally', on channel 'SIP/abdoulaye-00000005'
> 0x7f8184048960 -- Strict RTP learning complete - Locking on source address 192.168.1.25:40014
```

Il y'a aussi la possibilité de jouer de la musique en ligne (webradio). Comme tous à l'heure on crée la classe dans le fichier **musiconhold.conf**

```
[webradio]
mode=custom
application=/usr/bin/mpg123 -q -r 8000 -f 8192 -s --mono http://100radio-albi.ic
e.infomaniak.ch/100radio-albi-128.mp3
```

Le paramètre application spécifie le chemin du lecteur son et l'url du web radio

Ensuite on définit un numéro dans **extensions.conf**

```
exten =>3001,1,Answer()
exten =>3001,2,musicOnhold(webradio)
```

Maintenant les utilisateurs qui veulent écouter du web radio appellent le numéro 3001

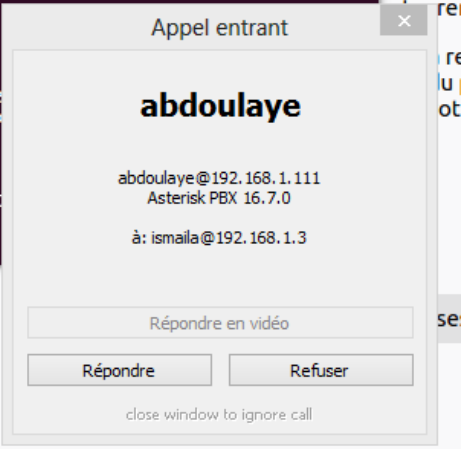
```
10071125:10002
-- Executing [3001@ToIP:1] Answer("SIP/abdoulaye-00000010", "") in new stack
> 0x7fa7cc070bd0 -- Strict RTP learning after remote address set to: 192.
168.1.25:40002
> 0x7fa7cc070bd0 -- Strict RTP switching to RTP target address 192.168.1.
25:40002 as source
-- Executing [3001@ToIP:2] MusicOnHold("SIP/abdoulaye-00000010", "webradio")
in new stack
-- Started music on hold, class 'webradio', on channel 'SIP/abdoulaye-000000
10'
> 0x7fa7cc070bd0 -- Strict RTP learning complete - Locking on source addr
ess 192.168.1.25:40002
```

Il y'a aussi la possibilité de personnaliser la sonnerie d'un utilisateur en choisissant la classe de musique qu'on veut lui faire jouer avec l'option **m** dans l'action **Dial** comme montre l'image ci-dessous

```
exten =>1002,1,Dial(SIP/ismaila,40,tm(wally))
exten =>1002,2,Voicemail(1002@context_bv)
exten =>1002,3,hangup
```

Maintenant quand on appelle l'utilisateur ismaila le serveur joue de la musique de la classe wally avant qu'ismaila décroche

```
10071125:10006
-- Executing [1002@ToIP:1] Dial("SIP/abdoulaye-00000013", "SIP/ismaila,40,tm(wally)") in new stack
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
-- Called SIP/ismaila
-- Started music on hold, class 'wally', on channel 'SIP/ismaila-00000013'
> 0x7fa7cc070bd0 -- Strict RTP switching to RTP target address 192.168.1.25:40006 as source
-- SIP/ismaila-00000014 is ringing
> 0x7fa7cc070bd0 -- Strict RTP learning complete
ess 192.168.1.25:40006
ubuntu-VirtualBox*CLI>
```



4- Gestion du transfert d'appel

Le principe du transfert d'appel est très simple. Comme son nom l'indique il s'agit de transférer un appel vers un autre poste. Il y'a deux façon de transfert d'appel :le transfert d'appel aveugle et supervisé

- ✓ Le transfert d'appel dit **aveugle** est le fait de transféré un appel directement à une autre personne
- ✓ Le transfert d'appel dit **supervisé** est le fait d'appeler le destinataire du transfert avant de lui transférer l'appel.

La configuration se fait dans le fichier **/etc/asterisk/features.conf** avec les paramètres **blindxfer** pour le transfert aveugle et **atxfer** pour le transfert supervisé

```
[featuremap]
blindxfer => ##                ; Blind transfer (default is #) -- Make sure to
set the T and/or t option in the Dial() or Queue() app call!
;disconnect => *0             ; Disconnect (default is *) -- Make sure to set
the H and/or h option in the Dial() or Queue() app call!
;automon => *1                 ; One Touch Record a.k.a. Touch Monitor -- Make
sure to set the W and/or w option in the Dial() or Queue() app call!
atxfer => **                    ; Attended transfer -- Make sure to set the T an
d/or t option in the Dial() or Queue() app call!
;parkcall => #72               ; Park call (one step parking) -- Make sure to
```

On vient de définir les touches ## pour les transferts aveugles et ** pour les transferts supervisés

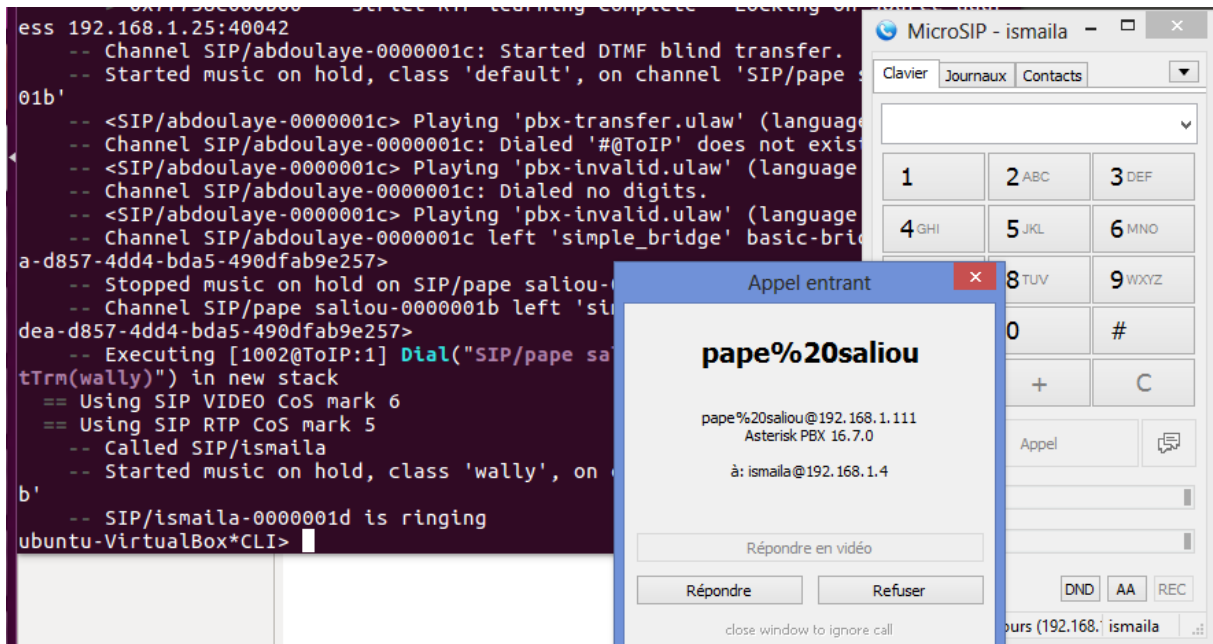
Ensuite nous allons ajouter les **T** et **t** à l'action Dial de nos utilisateurs. **t** pour autoriser l'appelé à transférer et **T** pour autoriser l'appelant à transférer

```
[ToIP]
exten =>1000,1,Dial(SIP/abdoulaye,40,tTrm(webradio))
exten =>1000,2,Voicemail(1000@context_bv)
exten =>1000,3,hangup

exten =>1001,1,Dial(SIP/pape saliou,40,tTrm(wally))
exten =>1001,2,Voicemail(1001@context_bv)
exten =>1001,3,hangup

exten =>1002,1,Dial(SIP/ismaïla,40,tTrm(wally))
exten =>1002,2,Voicemail(1002@context_bv)
exten =>1002,3,hangup
```

Maintenant, après avoir fait un reload d'Asterisk on peut tester avec pape saliou qui appelle l'utilisateur Abdoulaye au cours d'un appel Abdoulaye appui sur ## une voix lui dit « Transfert »,il appui le numero 1002 on peut voir que l'appel est transféré automatiquement à l'utilisateur Ismaila qui a le numéro 1002



5- Gestion des packages d'appel

Le package d'appel ou **Parking Call** est la possibilité de mettre en attente une personne sur un numéro de parking, c'est à dire que vous transférez cette personne sur un numéro qui lui est attribué, dès lors il vous suffit de rappeler ce numéro à partir de n'importe quel poste pour reprendre l'appel.

Asterisk a vingt emplacements (slots) réservés pour « packer » les appels, 701-720. Transférer les appels à « parker » en cliquant sur **##700**, Asterisk va automatiquement « parker » cet appel dans le premier emplacement libre et vous communiquer le numéro où vous pourrez récupérer la communication à partir d'un quelconque terminal

La configuration se fait aussi dans le fichier **features.conf** comme suit

```

parkext => 700
parkpos => 701-720
context => parkedcalls
parkingtime => 180

```

Ensuite il nous reste qu'à inclure le contexte **parkedcalls** dans le contexte de nos utilisateurs

```

[ToIP]
include => parkedcalls
exten =>1000,1,Dial(SIP/abdoulaye,40,tTrm(webradio))
exten =>1000,2,Voicemail(1000@context,bv)

```

Après avoir fait un reload de asterisk on peut tester avec l'utilisateur pape saliou entre en communication avec abdoulaye qui appui sur **##700** Le serveur met cet appel sur le slot 1 et lui communique le numéro de park qui est 701. Maintenant l'utilisateur se déplace sur un autre poste composé le numéro 701 pour récupérer l'appel

```

SCP: Linc 1000 not found.
-- Parking 'SIP/pape saliou-00000023' in 'default' at space 701
-- Channel SIP/pape saliou-00000023 joined 'holding_bridge' parking-bridge <
4604fda6-92b9-4d54-9e05-2a6e5be248eb>
-- Started music on hold, class 'default', on channel 'SIP/pape saliou-00000
023'
-- <SIP/abdoulaye-00000024> Playing 'digits/7.ulaw' (language 'fr')
-- <SIP/abdoulaye-00000024> Playing 'digits/0.ulaw' (language 'fr')
-- <SIP/abdoulaye-00000024> Playing 'digits/1.ulaw' (language 'fr')
-- Channel SIP/abdoulaye-00000024 left 'simple_bridge' basic-bridge <59a710b
a-09a7-42fa-8b1a-dabbd2a076c2>
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
> 0x7f759c01a230 -- Strict RTP learning after remote address set to: 192.
168.1.4:4030
-- Executing [701@ToIP:1] ParkedCall("SIP/ismaila-00000025", "default,701")
in new stack
> 0x7f759c01a230 -- Strict RTP learning after remote address set to: 192.
168.1.4:4030
-- Channel SIP/pape saliou-00000023 left 'holding_bridge' parking-bridge <46
04fda6-92b9-4d54-9e05-2a6e5be248eb>
-- Stopped music on hold on SIP/pape saliou-00000023

```

6- Gestion des files d'attente d'un centre d'appel

Asterisk permet de mettre en place des centres d'appel efficaces grâce à une gestion des agents et des files d'attente souple. Généralement un centre d'appel se compose :

- ✓ D'agents : qui prennent ou émettent des appels
- ✓ De files d'attente : qui distribuent les appels aux différents agents
- ✓ De statistiques : qui permettent d'avoir une visibilité sur appels effectués

Un agent est un compte qui va pouvoir se connecter, se déconnecter et se mettre en pause.

Une file d'attente (queue) est une "salle" virtuelle dans laquelle sont connectés des agents et qui attendent des appels. Quand un appel arrive, il est distribué suivant une "stratégie" à un ou plusieurs agents

La politique est définie à travers le paramètre **strategy**

strategy=ringall alors tous les téléphones des agents vont sonner en même temps à l'arrivée d'un appel

strategy=fewestcalls alors l'agent qui a reçu le moins sera celui à qui le prochain appel sera envoyé

La file d'attente se crée dans le fichier queues.conf comme montre l'image ci-dessous

```

[corona]
musicclass=default
strategy=fewestcalls
joinempty=yes
member=>Local/1002@ToIP

```

[corona] est le nom de la file d'attente

musicclass=default permet de jouer comme sonnerie la classe de musique par défaut

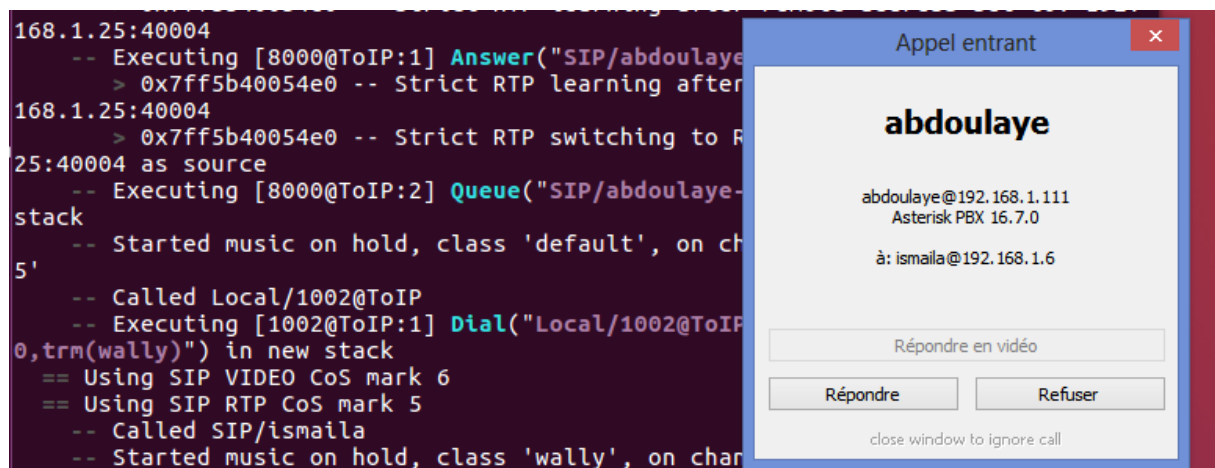
strategy=fewestcalls permet de faire sonner le téléphone de l'agent qui a reçu le moins d'appel

La **ligne member =>Local/1002@ToIP** permet d'ajouter le numéro 1002 du contexte ToIP comme agent de la file d'attente corona

Ensuite on définit un numéro pour joindre la file d'attente dans extensions.conf avec l'action Queue comme montre l'image ci-dessous

```
exten =>8000,1,Answer()  
exten =>8000,2,Queue(corona)
```

Maintenant quand un utilisateur compose le numéro 8000 il entre dans la file et le téléphone de l'utilisateur 1002 sonne puisqu'il est le seul dans la file pour répondre



The image shows a terminal window on the left and a graphical call window on the right. The terminal output shows the execution of the 'Answer()' and 'Queue()' actions, followed by the 'Dial()' action which connects the call to the 'Local/1002@ToIP' extension. The call window, titled 'Appel entrant', displays the caller's name 'abdoulaye', their phone number 'abdoulaye@192.168.1.111', and the Asterisk version 'Asterisk PBX 16.7.0'. It also shows the destination 'à: ismaila@192.168.1.6'. The window contains three buttons: 'Répondre en vidéo', 'Répondre', and 'Refuser', along with a 'close window to ignore call' link.

Il y'a la possibilité en tant que administrateur de voir les agents qui sont dans la file, d'ajouter des agents, de supprimer des agents dans la console de asterisk

Pour vérifier les informations par rapport aux agents on tape la commande **queue show nom de la file**

```
ubuntu-VirtualBox*CLI> queue show corona  
corona has 0 calls (max unlimited) in 'ringall' strategy (0s holdtime, 0s talktime), W:0, C:0, A:1, SL:0.0%, SL2:0.0% within 0s  
Members:  
Local/1002@ToIP (ringinuse enabled) (Not in use) has taken no calls yet  
No Callers
```

On peut voir que seul l'utilisateur 1002 se trouve dans la file

Pour ajouter un agent dans la file on tape la commande **queue add member SIP/nom user to nom de la file**

```

ubuntu-VirtualBox*CLI> queue add member SIP/mum to corona
Added interface 'SIP/mum' to queue 'corona'
ubuntu-VirtualBox*CLI> queue show corona
corona has 0 calls (max unlimited) in 'ringall' strategy (0s holdtime, 0s talktime), W:0, C:0, A:4, SL:0.0%, SL2:0.0% within 0s
Members:
  SIP/mum (ringinuse enabled) (dynamic) (Not in use) has taken no calls yet
  Local/1002@ToIP (ringinuse enabled) (Not in use) has taken no calls yet
No Callers

```

On vient d'ajouter l'utilisateur mum dans la file corona et si on refait un **queue show corona** on peut voir qu'on a deux agents maintenant sur la file

Pour enlever un agent dans la file on tape la commande **queue remove member SIP/nom user from nom de la file**

```

ubuntu-VirtualBox*CLI> queue remove member SIP/mum from corona
Removed interface SIP/mum from queue 'corona'
ubuntu-VirtualBox*CLI> queue show corona
corona has 0 calls (max unlimited) in 'ringall' strategy (1s holdtime, 13s talktime), W:0, C:1, A:6, SL:0.0%, SL2:0.0% within 0s
Members:
  Local/1002@ToIP (ringinuse enabled) (Not in use) has taken no calls yet
No Callers

```

On a enlevé l'utilisateur mum qu'on venait d'ajouter et vois que il a été bien supprimé

Il y'a aussi la possibilité de personnaliser la musique d'attente lorsque tous les agents de la file sont en communication. Par exemple l'utilisateur qui appelle on lui fait écouter de la webradio avant qu'un agent décroche. Pour cela on ajoute la priorité 2 comme montre l'image ci-dessous

```

exten =>8000,1,Answer()
exten =>8000,2,Set(CHANNEL(musicclass)=webradio)
exten =>8000,3,Queue(corona)

```

On peut voir bien que la musique webradio est jouée l'utilisateur qui veut rejoindre la file d'attente sur le numéro 8000

```

168.1.48:64546
-- Executing [8000@ToIP:1] Answer("SIP/pape saliou-00000000", "") in new stack
--> 0x7f385400afd0 -- Strict RTP switching to RTP target address 192.168.1.48:64546 as source
-- Executing [8000@ToIP:2] Set("SIP/pape saliou-00000000", "CHANNEL(musicclass)=webradio") in new stack
-- Executing [8000@ToIP:3] Queue("SIP/pape saliou-00000000", "corona") in new stack
-- Started music on hold, class 'webradio', on channel 'SIP/pape saliou-00000000'

```

En cas de télétravail on peut affecter des numéros aux agents dans extensions.conf quand ils appellent ce numéro on lui met dans la file avec l'action **AddQueueMember** comme montre l'image suivante

```

exten =>8001,1,AddQueueMember(corona)
exten =>8001,n,Playback(agent-loginok)
exten =>8001,n,hangup

```

L'action **Playback** permet de lui jouer un son « Vous êtes maintenant en ligne »

Maintenant quand un l'utilisateur pape saliou compose le numéro 8001 on voit qu'il est bien ajouté dans la file corona

```

> 0x7f385400a8d0 -- Strict RTP learning after remote address set to: 192.168.1.48:38156
-- Executing [8001@ToIP:1] AddQueueMember("SIP/pape saliou-00000008", "corona") in new stack
[Apr 27 18:50:43] NOTICE[11898][C-00000005]: app_queue.c:8077 aqm_exec: Added in interface 'SIP/pape saliou' to queue 'corona'
-- Executing [8001@ToIP:2] Playback("SIP/pape saliou-00000008", "agent-loginok") in new stack
> 0x7f385400a8d0 -- Strict RTP switching to RTP target address 192.168.1.48:38156 as source
-- <SIP/pape saliou-00000008> Playing 'agent-loginok.ulaw' (language 'fr')
-- Executing [8001@ToIP:3] Hangup("SIP/pape saliou-00000008", "") in new stack
== Spawn extension (ToIP, 8001, 3) exited non-zero on 'SIP/pape saliou-00000008'

```

```

ubuntu-VirtualBox*CLI> queue show corona
corona has 0 calls (max unlimited) in 'ringall' strategy (1s holdtime, 23s talktime), W:0, C:1, A:1, SL:0.0%, SL2:0.0% within 0s
Members:
SIP/pape saliou (ringinuse enabled) (dynamic) (Not in use) has taken no calls yet
Local/1002@ToIP (ringinuse enabled) (Not in use) has taken no calls yet
No Callers

```

On prévoit aussi un autre numéro pour quitter la file corona dans **extensions.conf** avec l'action **RemoveQueueMember** comme montre l'image ci-dessous

```

exten =>8002,1,RemoveQueueMember(corona)
exten =>8002,n,Playback(agent-loggedoff)
exten =>8002,n,hangup

```

Ici l'action **Playback** permet de jouer un son « vous n'êtes plus en ligne »

Quand l'utilisateur pape saliou compose le numéro 8002 on constate que qu'il a été enlevé de la file

```

-- Executing [8002@ToIP:1] RemoveQueueMember("SIP/pape saliou-00000009", "co
rona") in new stack
[Apr 27 19:01:30] NOTICE[11947][C-00000006]: app_queue.c:7990 rqm_exec: Removed
interface 'SIP/pape saliou' from queue 'corona'
-- Executing [8002@ToIP:2] Playback("SIP/pape saliou-00000009", "agent-logge
doff") in new stack
> 0x7f385400a8d0 -- Strict RTP switching to RTP target address 192.168.1.
48:38156 as source
-- <SIP/pape saliou-00000009> Playing 'agent-loggedoff.ulaw' (language 'fr')
-- Executing [8002@ToIP:3] Hangup("SIP/pape saliou-00000009", "") in new sta
ck
== Spawn extension (ToIP, 8002, 3) exited non-zero on 'SIP/pape saliou-0000000
9'

```

```

ubuntu-VirtualBox*CLI> queue show corona
corona has 0 calls (max unlimited) in 'ringall' strategy (0s holdtime, 0s talkti
me), W:0, C:0, A:1, SL:0.0%, SL2:0.0% within 0s
Members:
  Local/1002@ToIP (ringinuse enabled) (Not in use) has taken no calls yet
No Callers

```

Partie 3 : Mise en place d'une plateforme de ToIP en environnement CISCO

I La téléphonie en environnement CISCO

Chez Cisco, il y a deux grandes familles de serveurs centraux de TOIP :

- ✓ Le **Cisco Unified Communications Manager (CUCM)** qui est un système de communication basé sur IP intégrant la voix et la vidéo. Il fait partie d'un ensemble de produits de la gamme VoIP de Cisco
- ✓ Le **Cisco Unified Communications Manager Express (CUCME)** ou **CME** qui est un système de communication basé sur IP intégrant la voix et la vidéo et tournant sur des routeurs avec des IOS adaptés (On utilise l'IOS **c3745-ipvoice_ivs-mz.124-15,T7.bin** téléchargé sur le site <http://tfr.org/cisco/>)

Les équipements CISCO supportent aussi bien son protocole de signalisation SCCP que le protocole de signalisation standard SIP

SCCP est un protocole propriétaire Cisco utilisé pour la gestion des appels téléphoniques et de la signalisation. SCCP est supporté nativement dans le CUCM. SCCP a été beaucoup utilisé avec les téléphones d'anciennes générations car c'est un protocole très simple mais est de moins en moins utilisé avec les téléphones récents et laisse sa place à SIP. SCCP est exclusivement utilisé entre le CUCM et les téléphones IP

SIP est un protocole de signalisation standardisé par l'IETF. SIP est de plus en plus utilisé aujourd'hui dans le monde de la VoIP mais aussi dans la messagerie instantanée ou encore la visiophonie.

Tous les téléphones IP récents utilisent SIP pour s'enregistrer dans le CUCM. SIP offre beaucoup plus de fonctionnalités que SCCP

II Configuration d'un CUCME ou CME

Dans nos travaux nous allons utiliser l'**IOS c3745-ipvoice_ivs-mz.124-15,T7.bin** téléchargé sur le site <http://tfr.org/cisco/> et gns3 avec cet ios, en utilisant un cloud pour permettre à notre routeur CUCME de communiquer avec notre réseau physique.

Voici un tableau de comparaison des commandes CUME SCCP et SIP

Commandes CME SCCP	Commandes CME SIP	Signification
telephony-service	Voice register global	Activation de service toip
max-ephones	max-pool	Nbre maximal de téléphones à gerer
max-dn	max-dn	Nbre maximal de numéros de téléphones à gérer
ip source-address	source-address	Adresse ip du routeur et port d'écoute
ephone-dn	Voice register dn	Déclaration de numéro de téléphone avec son étiquette
ephone	Voice register pool	Déclaration de téléphone
Button n:m	Number n dn m	Affectation d'un numéro à un bouton d'un téléphone
Voice service voip	Autorisation de sip à sip	
allow-connection sip to sip sip registrar server	et déclaration comme serveur d'enregistrement sip	
Sip-ua authentication username toto password passer sip-server ipv4:192.168.0.250	Paramètre de connexion à un autre serveur Sip	

1- Configuration du CME comme contrôleur SCCP

Tous d'abord on fait configuration basique en donnant une adresse IP et un masque de réseau dans notre réseau local et vois bien que le routeur parviens à communiquer avec notre passerelle 192.168.1.1

```

R1#conf t
Enter configuration commands, one per line. End with CNTL/Z.
R1(config)#int f0/0
R1(config-if)#ip add 192.168.1.20 255.255.255.0
R1(config-if)#no sh
R1(config-if)#
*Mar 1 00:00:52.499: %LINK-3-UPDOWN: Interface FastEthernet0/0, changed state to up
*Mar 1 00:00:53.499: %LINEPROTO-5-UPDOWN: Line protocol on Interface FastEthernet0/0, changed state to up
R1(config-if)#do ping 192.168.1.1

Type escape sequence to abort.
Sending 5, 100-byte ICMP Echos to 192.168.1.1, timeout is 2 seconds:
!!!!
Success rate is 80 percent (4/5), round-trip min/avg/max = 8/9/12 ms

```

Ensuite on active le service SCCP en précisant le port 2000 et définir 5 téléphones et 5 numéros de téléphone aussi

```

R1(config)#telephony-service
R1(config-telephony)#ip so
R1(config-telephony)#ip source-address 192.168.1.20 port 2000
R1(config-telephony)#max-e
R1(config-telephony)#max-ephones 5
R1(config-telephony)#max-d
R1(config-telephony)#max-dn 5
R1(config-telephony)#cre
R1(config-telephony)#create cnf
R1(config-telephony)#create cnf-files
Creating CNF files
CNF-FILES: Clock is not set or synchronized, retaining old versionStamps
R1(config-telephony)#exit

```

Ensuite on crée les numéros 2001,2002 et 2003

```

R1(config)#ephone-dn 1
R1(config-ephone-dn)#nu
*Mar 1 00:04:18.279: %LINK-3-UPDOWN: Interface ephone_dsp DN 1.1, changed state to up
R1(config-ephone-dn)#num
R1(config-ephone-dn)#number 2001
R1(config-ephone-dn)#exit
R1(config)#ephone-dn 2
R1(config-ephone-dn)#exit
*Mar 1 00:04:32.755: %LINK-3-UPDOWN: Interface ephone_dsp DN 2.1, changed state to up
R1(config-ephone-dn)#number 2002
R1(config-ephone-dn)#exit
R1(config)#ephone-dn 3
R1(config-ephone-dn)#
*Mar 1 00:04:48.131: %LINK-3-UPDOWN: Interface ephone_dsp DN 3.1, changed state to up
R1(config-ephone-dn)#number 2003
R1(config-ephone-dn)#exit

```

On met enfin la commande **auto assign** qui va attribuer les numéros au téléphone dans l'ordre de connexion

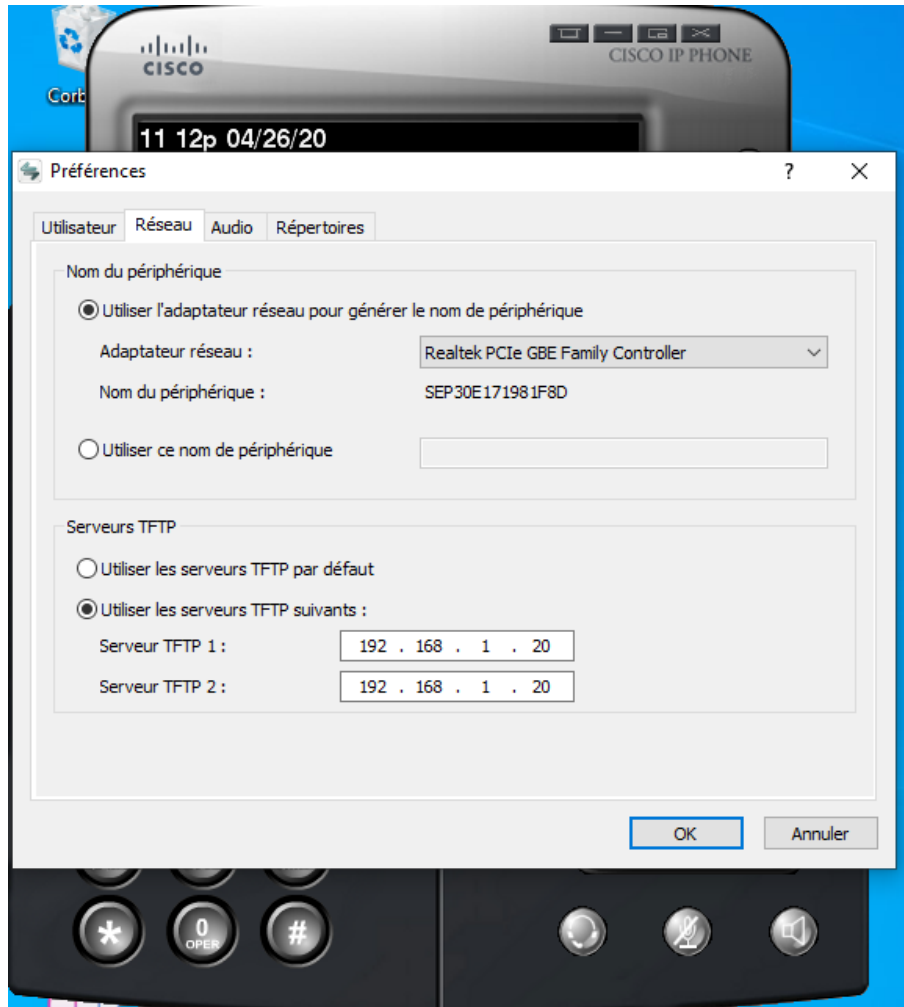
```

R1(config)#telephony-service
R1(config-telephony)#auto as
R1(config-telephony)#auto assign 1 t
R1(config-telephony)#auto assign 1 to 5
R1(config-telephony)#

```

Une fois les configurations terminées on peut utiliser un client SSCP comme **Cisco IP Communicator** pour le connecter au serveur. Une fois ce logiciel installé on peut passer à la configuration du client comme montre l'image ci-dessous

On va dans le menu du premier icône du téléphone puis présences et réseau on choisit utiliser les serveurs TFTP suivant on met l'adresse du routeur 192.168.1.20



Après validation on peut voir sur le routeur que le téléphone s'est bien enregistré avec l'adresse 192.168.1.11

```
R1(config-telephony)#
reseting 30E1.7198.1F8D
*Mar 1 00:21:09.003: %IPPHONE-6-REGISTER_NEW: ephone-1:SEP30E171981F8D IP:192.1
68.1.11 Socket:1 DeviceType:Phone has registered.
*Mar 1 00:21:09.495: %IPPHONE-6-UNREGISTER_NORMAL: ephone-1:SEP30E171981F8D IP:
192.168.1.11 Socket:1 DeviceType:Phone has unregistered normally.
R1(config-telephony)#
*Mar 1 00:21:52.415: %IPPHONE-6-REGISTER: ephone-1:SEP30E171981F8D IP:192.168.1
.11 Socket:1 DeviceType:Phone has registered.
R1(config-telephony)#
```

Le serveur lui a attribué le premier numéro 2001



2- Configuration du CME comme contrôleur SIP

Tous d'abord on active le protocole SIP sur le routeur en lui renseignant l'adresse IP 192.168.1.20 et le numéro de port 5060. On prévoit aussi 5 téléphones et 5 numéros de téléphones

```
R1(config)#voice register glo
R1(config)#voice register global
R1(config-register-global)#mode cme
R1(config-register-global)#sou
R1(config-register-global)#source-address 192.168.1.20 port 5060
R1(config-register-global)#max-dn 5
R1(config-register-global)#max-pool 5
R1(config-register-global)#auth
R1(config-register-global)#authenticate regi
R1(config-register-global)#authenticate register
R1(config-register-global)#time-format 24
R1(config-register-global)#date-format D/M/Y
R1(config-register-global)#tf
R1(config-register-global)#tftp-path flash:
R1(config-register-global)#cre
R1(config-register-global)#create pr
R1(config-register-global)#create profile
```

Ensuite on autorise du service SIP et active le routeur comme serveur d'enregistrement

```

R1(config)#voice service voi
R1(config)#voice service voip
R1(conf-voi-serv)#all
R1(conf-voi-serv)#allow-connections sip ti sip
^
% Invalid input detected at '^' marker.

R1(conf-voi-serv)#allow-connections sip to sip
R1(conf-voi-serv)#sip
R1(conf-serv-sip)#regi
R1(conf-serv-sip)#registrar ser
R1(conf-serv-sip)#registrar server
R1(conf-serv-sip)#exit

```

Ensuite on crée un numéro 1000 à l'utilisateur du nom d'abdoulaye ndiaye

```

R1(config)#voice register dn 1
R1(config-register-dn)#num
R1(config-register-dn)#number 1000
R1(config-register-dn)#name abdoulaye ndiaye
R1(config-register-dn)#label abou ToIP
R1(config-register-dn)#exit
R1(config)#

```

Ensuite on crée le compte SIP 1000 avec le mot de passe passer et active le codec audio g711

```

R1(config)#voice regi
R1(config)#voice register pool 1
R1(config-register-pool)#id mac 0000.0000.1000
R1(config-register-pool)#type 7965
R1(config-register-pool)#number 1 dn 1
R1(config-register-pool)#codec g711
R1(config-register-pool)#codec g711u
R1(config-register-pool)#codec g711ulaw
R1(config-register-pool)#usern
R1(config-register-pool)#username 1000 pass
R1(config-register-pool)#username 1000 password passer
R1(config-register-pool)#exit

```

Maintenant on peut utiliser un client SIP comme micropSIP pour se connecter avec le compte 1000 et le mot de passe passer

Compte X

Nom du compte

Serveur SIP ?

Proxy SIP ?

Nom utilisateur* ?

Domaine* ?

Login ?

Mot de passe ?
[Afficher le mot de passe](#)

Afficher le nom ?

Numéro de la boîte ... ?

Préfixe d'appel ?

Dial Plan ?

Hide Caller ID

Chiffrement ?

Transport ?

Adresse publique ?

Actualiser l'enregist... Garder en direct

Afficher ma présence ?

Autoriser la ré-écriture de l'IP ?

ICE ?

Désactiver les temporisateurs de ses ?

On peut voir sur l'interface du routeur que le téléphone SIP s'est bien enregistré avec l'adresse 192.168.1.11

```
R1(config-register pool)#
R1(config)#
*Mar 1 00:37:23.967: VOICE REGISTER POOL-1 has registered. Name:SEP00000000100
0 IP:192.168.1.11 DeviceType:Phone
```

On peut voir que les clients SCCP et SIP peuvent s'appeler



III Asterisk comme contrôleur SCCP

Asterisk support le Skinny Call Control Protocol (SCCP) depuis un certain nombre d'années et il vous faut uniquement le module SCCP pour que ça fonctionne. Si vous souhaitez utiliser un téléphone SCCP avec Asterisk. SCCP a été développé par Cisco, alors leurs téléphones sont les téléphones IP les plus communs utilisant ce protocole. Si vous possédez actuellement des téléphones Cisco, vous pourriez tenter de les utiliser en mode SIP avant d'essayer de les utiliser en mode SCCP avec Asterisk

1- Téléchargement et installation du module SCCP

Chan_SCCP est un pilote de canal de remplacement pour chan_skinny dans la bibliothèque de pilotes de canaux Asterisk. Il offre de meilleures performances, évolutivité, interopérabilité et fonctionnalités que chan_skinny ou chan_sip sur un téléphone compatible SCCP / Skinny

Pour son téléchargement on installe d'abord le paquet git en suite lance la commande suivante : **git clone https://github.com/chan-sccp/chan-sccp.git chan-sccp**

```

root@ubuntu-VirtualBox:~# cd /usr/local/src/
root@ubuntu-VirtualBox:/usr/local/src# ls
asterisk-16.7.0  asterisk-16-current.tar.gz
root@ubuntu-VirtualBox:/usr/local/src# git clone https://github.com/chan-sccp/chan-sccp.git chan-sccp
Clonage dans 'chan-sccp'...
remote: Enumerating objects: 55760, done.
remote: Total 55760 (delta 0), reused 0 (delta 0), pack-reused 55760
Réception d'objets: 100% (55760/55760), 36.99 MiB | 239.00 KiB/s, fait.
Résolution des deltas: 100% (41641/41641), fait.
Vérification de la connectivité... fait.
root@ubuntu-VirtualBox:/usr/local/src#

```



```

root@ubuntu-VirtualBox:~# apt-get install tftpd-hpa
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
tftpd-hpa est déjà la version la plus récente (5.2+20150808-1ubuntu1.16.04.1).
Le paquet suivant a été installé automatiquement et n'est plus nécessaire :
  libjson-c2:i386
Veuillez utiliser « sudo apt autoremove » pour le supprimer.
0 mis à jour, 0 nouvellement installés, 0 à enlever et 3 non mis à jour.

```

Après installation on peut voir que son fichier de configuration se trouve dans **/etc/default/tftpd-hpa** et son contenu est comme nous montre l'image suivante

```

# /etc/default/tftpd-hpa

TFTP_USERNAME="tftp"
TFTP_DIRECTORY="/var/lib/tftplibboot"
TFTP_ADDRESS=":69"
TFTP_OPTIONS="--secure"

~
~

```

On peut voir dans le paramètre **TFTP_DIRECTORY** que le dossier de tftp est **/var/lib/tftplibboot**. C'est dans ce dossier que nous allons copier les exemples de configuration des téléphones se trouvant dans **/chan-sccp/conf/tftp**

```

root@ubuntu-VirtualBox:/usr/local/src# cd chan-sccp/conf/tftp/
root@ubuntu-VirtualBox:/usr/local/src/chan-sccp/conf/tftp# ls
Desktops
properties.xml
SEP00000000000.cnf.xml_69xx_template
SEP00000000000.cnf.xml_791x_template
SEP00000000000.cnf.xml_7925_template
SEP00000000000.cnf.xml_792x_vendorConfig_entries.xml
SEP00000000000.cnf.xml_7937_template
SEP00000000000.cnf.xml_7940_7960_template
SEP00000000000.cnf.xml_796x_template
SEP00000000000.cnf.xml_797x_extended_template
SEP00000000000.cnf.xml_797x_template
SEP00000000000.cnf.xml_annotated
SEP00000000000.cnf.xml_ipv6_template
WLANDefault.xml
XMLDefault504G.cnf.xml
XMLDefault.cnf.xml
root@ubuntu-VirtualBox:/usr/local/src/chan-sccp/conf/tftp# cp -r * /var/lib/tftplibboot/
root@ubuntu-VirtualBox:/usr/local/src/chan-sccp/conf/tftp#

```

On peut bien voir que notre serveur tftp tourne sur le port 69

```

root@ubuntu-VirtualBox:~# vi /etc/default/tftpd-hpa
root@ubuntu-VirtualBox:~# netstat -anp | grep -w 69
udp        0      0 0.0.0.0:69          0.0.0.0:*
1210/in.tftpd
udp6      0      0 :::69              :::*
1210/in.tftpd

```

3- Configuration de asterisk comme contrôleur SCCP

La configuration va se faire dans deux fichiers **sccp.conf** et **extensions.conf**

✓ Dans **sccp.conf**

On définit d'abord les paramètres généraux du contrôleur dans la section **[general]**

```
musicclass=default
language=en
;deny=0.0.0.0/0.0.0.0
permit=0.0.0.0/0.0.0.0
;permit=internal ; 'inte
automatically converted to these private cidr address: ; 127.0.0.0/255
0.0.0.0/255.0.0.0, ; 172.0.0.0/255
192.168.0.0/255.255.0.0
;permit=192.168.1.0/255.255.255.0
localnet = 192.168.1.0/255.255.0.0
-ENTRY) All RFC 1918 addresses are local networks, example '192.168.1.0
```

On a juste commenté la ligne **deny** et changer par **permit** pour autoriser tous les adresses et le paramètre **localnet** ou on a mis l'adresse et le masque de notre réseau local

Ensuite on définit des comptes aux téléphones SCCP en leur attribuant des boutons dans leur propre section [SEPXXXXXXXXXXXXX]. On remplace les X par l'adresse mac de la machine ou se trouve le client SCCP

```
[SEP98541B8EFC24]
description = Phone Number One
addon = 7914
devicetype = 7960
park = off
button = speeddial,Helpdesk, 98112, 98112@hints ; Add SpeedDial to Hel
button = line, 98011,default ; Assign Line 98011 to
button = empty ; Assign an Empty Butt
button = line, 98012 ; Assign Line 98012 to
button = speeddial,Phone 2 Line 1, 98021, 98021@hints ; Add SpeedDial to Pho
e 'Ã©')
cfwdall = off
type = device
keepalive = 60
;tzoffset = +2
transfer = on
park = on
cfwdall = off
cfwdbusy = off
cfwdnoanswer = off
permit=0.0.0.0/0.0.0.0
;permit=192.168.1.5/255.255.255.255
dndFeature = on
dnd = off
directrtp=off
earlyrtp = progress
private = on
mwilamp = on
mwioncall = off
setvar=testvar=value
cfwdall = on
```

Pour notre cas l'adresse MAC du téléphone est **98 :54 :1B :8E :FC :24** il faut aussi changer le paramètre permit le mètre à 0.0.0.0/0.0.0.0 pour permettre au téléphone de connecter avec n'importe quelle adresse IP

Enfin on définit les lignes c'est-à-dire les numéros dans leurs propres sections **[numero]**

```
[98011]
id = 1000
type = line
pin = 1234
label = Phone 1 Line 1
description = Line 98011
mailbox = 1001
cid_name = abou
cid_num = 98011
accountcode=79011
callgroup=1,3-4
pickupgroup=1,3-5
directed_pickup = on
directed_pickup_context = ""
pickup_modeanswer = on
;amaflags =
context = ToIP
incominglimit = 2
transfer = on
vmnum = 600
meetme = on
meetmeopts = qxd
meetmenu = 700
trnsfvm = 1000
secondary_dialtone_digits = 9
secondary_dialtone_tone = Outside Dial Tone
musicclass=default
language=en
echocancel = on
silencesuppression = off
setvar=testvar2=my value
dnd = reject
parkinglot = myparkspace
```

On vient de définir le numéro 98011 en précisant son nom dans le paramètre **cid_name** et le contexte de nos utilisateurs qui **ToIP** dans le paramètre **context**. On fait de même pour créer la ligne 98012

- ✓ Dans **extensions.conf** on crée les numéros 98011 et 98012 avec l'action Dial pour lancer des appels dans le canal SCCP

```
exten => 98011,1,Dial(SCCP/98011,40,tr)
exten => 98012,1,Dial(SCCP/98012,40,tr)
```

Maintenant on passe à l'adaptation du fichier du téléphone SCCP. On va dans le dossier du serveur TFTP `/var/lib/tftpboot/`, on copie l'exemple `SEP0000000000.cnf.xml_annotated` et on le nomme selon l'adresse mac du téléphone sccp qui est `98 :54 :1B :8E :FC :24`

```
root@ubuntu-VirtualBox:~# cd /var/lib/tftpboot/
root@ubuntu-VirtualBox:/var/lib/tftpboot# cp SEP0000000000.cnf.xml_annotated SEP
98541B8EFC24.cnf.xml
root@ubuntu-VirtualBox:/var/lib/tftpboot# vi SEP98541B8EFC24.cnf.xml
root@ubuntu-VirtualBox:/var/lib/tftpboot# vi XMLDefault.cnf.xml
```

Ensuite on édite le fichier `SEP98541B8EFC24.cnf.xml` qu'on vient de créer et on adapte l'adresse IP du serveur asterisk dans la balise `<processNodeName>`

```
<callManager>
  <name>Asterisk 1</name>
  <description>Secondary Asterisk Server</description>
  <ports>
    <ethernetPhonePort>2000</ethernetPhonePort>
    <!--
    <sipPort>5060</sipPort>
    <securedSipPort>5061</securedSipPort>
    <mgcpPorts>
      <listen>2427</listen>
      <keepAlive>2428</keepAlive>
    </mgcpPorts>
    -->
  </ports>
  <processNodeName>192.168.1.111</processNodeName>
  <!--
    'processNodeName' is the ip address of your asterisk server.
  -->
</callManager>
```

On édite aussi le fichier `XMLDefault.cnf.xml` qui est le fichier de configuration par défaut de tous les téléphones pour adapter l'adresse du serveur asterisk

```
<Default>
  <callManagerGroup>
    <members>
      <member priority="0">
        <callManager>
          <ports>
            <ethernetPhonePort>2000</ethernetPhonePort>
          </ports>
          <processNodeName>192.168.1.111</processNodeName>
        </callManager>
      </member>
      <member priority="1">
        <callManager>
          <ports>
            <ethernetPhonePort>2000</ethernetPhonePort>
          </ports>
          <processNodeName>10.0.0.2</processNodeName>
        </callManager>
      </member>
    </members>
```

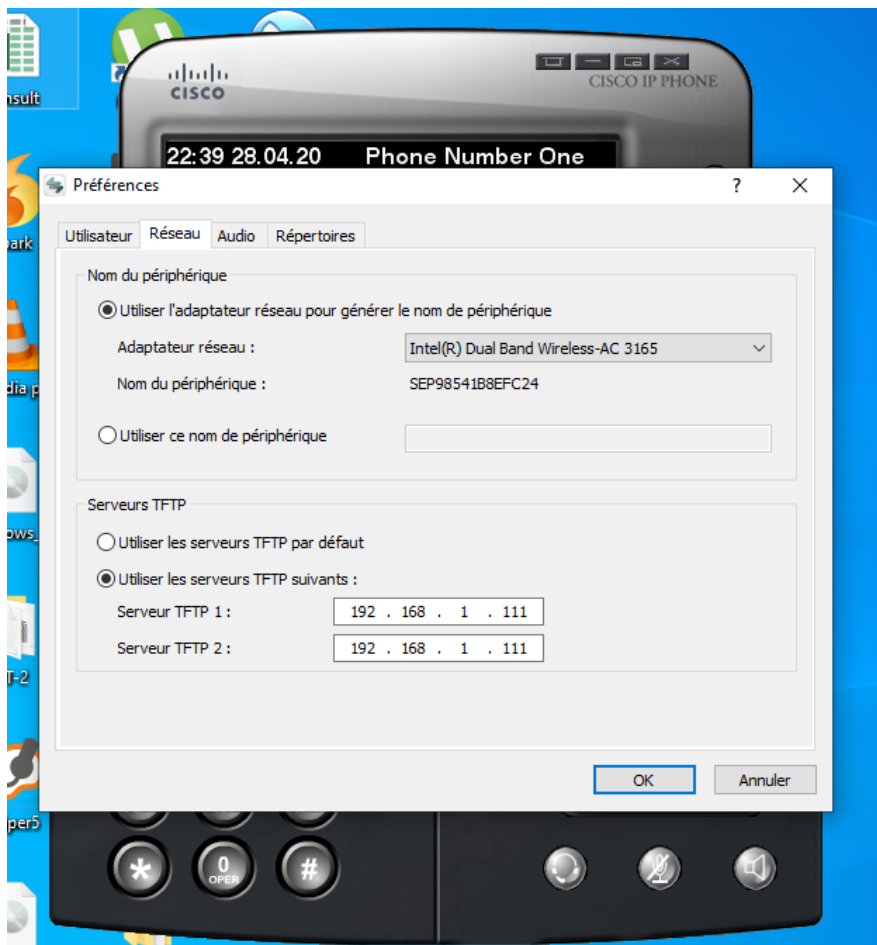
4- Configurations des téléphones et test d'appel

Pour les téléphones SCCP nous allons utiliser les logiciels Cisco IP Communicator qu'on peut télécharger sur ce lien

<https://download.workbooth.com/installers/CiscoIPCommunicatorSetup.exe>

Après téléchargement pour installer le logiciel il faut exécuter le fichier en tant que administrateur puis suivre la procédure d'installation

Pour la configuration on va dans menu puis dans préférence on met l'adresse du serveur TFTP ou doit télécharger son fichier de configuration



Ensuite quand on clique sur ok on peut voir que le téléphone s'est connecté et voit boutons qu'on l'avait attribué



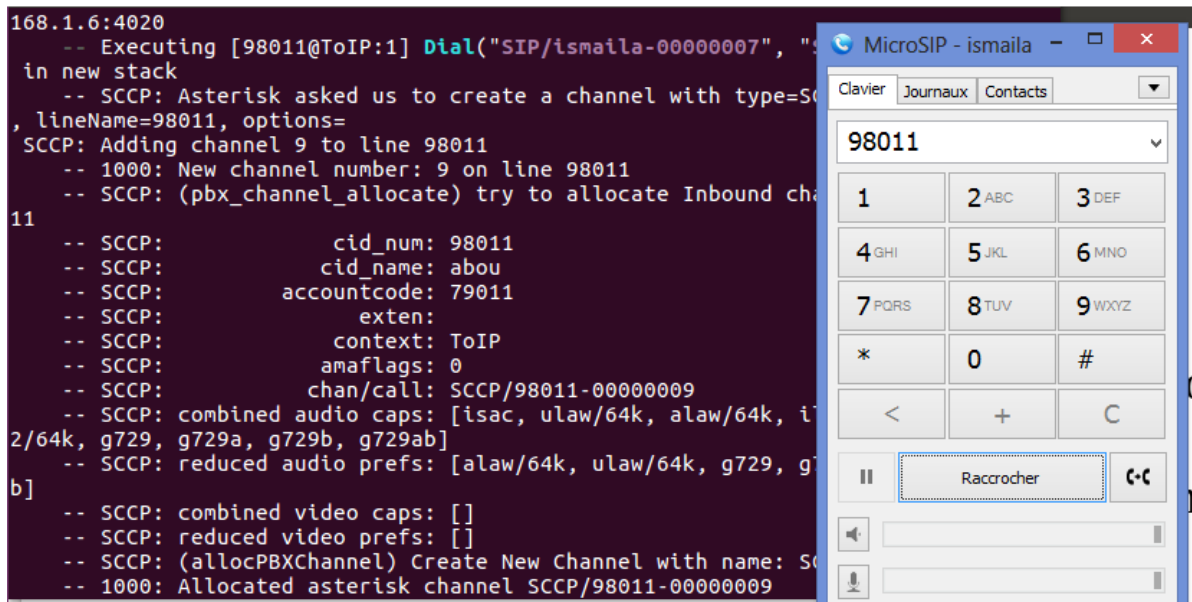
On peut voir sur l'interface de asterisk que le téléphone s'enregistre et les comptes SIP ismaila et mum

```

-- SEP98541B8EFC24: Getting Database Settings...
-- SEP98541B8EFC24: Update last number dialed to 1002.
-- SEP98541B8EFC24: Set softkeyset to ONHOOK(0) on line 0 and call 0
-- SEP98541B8EFC24: validKeyMask 4294967295
> SEP98541B8EFC24: clear message stack 1
-- SEP98541B8EFC24: Clear the status prompt on line 0 and callid 0
-- SEP98541B8EFC24: ( sccp_device.c:2138:sccp_dev_check_displayprompt ) sccp
_dev_displayprompt '0913' for line 0 (0)
-- SEP98541B8EFC24: Display prompt on line 0, callid 0, timeout 0
-- SEP98541B8EFC24: Set softkeyset to ONHOOK(0) on line 0 and call 0
-- SEP98541B8EFC24: validKeyMask 4294967295
-- SEP98541B8EFC24: Post registration process... done!
-- SEP98541B8EFC24: ( sccp_actions.c:257:sccp_handle_message ) sccp_dev_disp
laynotify 'Asterisk 0913' (5)
-- SEP98541B8EFC24: Display notify timeout 5
-- SEP98541B8EFC24: Display notify with timeout 5
> Saved useragent "MicroSIP/3.19.28" for peer mum
> Saved useragent "MicroSIP/3.19.28" for peer ismaila

```

Maintenant on peut tester depuis le compte de Ismaila appelé le client SCCP sur le numéro 98011



Le téléphone SCCP sonne



On peut tester aussi l'autre sens c'est-à-dire l'appel depuis le client SCCP vers le client SIP



Partie 4 : Mise en place d'une Plateforme de ToIP avec FreeSWITCH

I Présentation de FreeSWITCH

FreeSWITCH est une pile de télécommunications définie par logiciel permettant la transformation numérique de commutateurs de télécommunications propriétaires vers une implémentation logicielle polyvalente qui s'exécute sur n'importe quel matériel de base. Du Raspberry PI au serveur multi cœur, FreeSWITCH peut libérer le potentiel de télécommunications de n'importe quel appareil

FreeSWITCH peut gérer les communications vocales, vidéo et textuelles à partir d'un réseau IP (VoIP) et du RTPC (c'est-à-dire des lignes fixes régulières). FreeSWITCH prend en charge tous les protocoles VoIP populaires ainsi que l'interface avec les PRI

II Installation de FreeSWITCH

Avant de commencer l'installation de FreeSWITCH il faut d'abord installer ses prérequis comme ceux-ci :

```
apt-get install --yes build-essential pkg-config uuid-dev zlib1g-dev libjpeg-dev  
libsqlite3-dev libcurl4-openssl-dev libpcres3-dev libspeexdsp-dev libldns-dev  
libedit-dev libtiff5-dev yasm libopus-dev libsndfile1-dev unzip libsmpp34-dev  
git
```

Après installation des prérequis on se déplace dans un dossier de notre choix (/usr/src/) et télécharger la version zip de FreeSWITCH avec la commande suivante

wget <https://files.freeswitch.org/freeswitch-releases/freeswitch-1.8.6.zip>

```
root@ubuntu-VirtualBox:~# cd /usr/src/
root@ubuntu-VirtualBox:/usr/src# wget https://files.freeswitch.org/freeswitch-releases/freeswitch-1.8.6.zip
--2020-04-29 20:01:27-- https://files.freeswitch.org/freeswitch-releases/freeswitch-1.8.6.zip
Résolution de files.freeswitch.org (files.freeswitch.org)... 190.102.98.174, 2803:d000:fffe::174
Connexion à files.freeswitch.org (files.freeswitch.org)|190.102.98.174|:443... connecté.
requête HTTP transmise, en attente de la réponse... 200 OK
Taille : 60290902 (57M) [application/zip]
Enregistre : «freeswitch-1.8.6.zip»

freeswitch-1.8.6.zip 100%[=====] 57,50M 245KB/s ds 4m 24s
2020-04-29 20:05:54 (223 KB/s) - «freeswitch-1.8.6.zip» enregistré [60290902/60290902]
```

Ensuite on décompresse le fichier avec la commande **unzip freeswitch-1.8.6.zip**

Ensuite on passe à l'activation et désactivation de module dans fichier **modules.conf** de **/usr/local/freeswitch**. Dans ce fichier On fait des choix des applications, **des** codecs, **des** endpoints qui sont des terminaisons d'appels, du gestionnaire d'évènements comme montre les images suivantes

On peut activer ou désactiver des applications selon nos besoins

```
#applications/mod_bert
#applications/mod_blacklist
applications/mod_callcenter
applications/mod_cidlookup
#applications/mod_cluechoo
applications/mod_commands
applications/mod_conference
applications/mod_curl
#applications/mod_cv
applications/mod_db
#applications/mod_directory
#applications/mod_distributor
applications/mod_dptools
#applications/mod_easyroute
applications/mod_enum
```

On peut aussi activer ou désactiver les codecs

```
#codecs/mod_com_g729
#codecs/mod_dahdi_codec
codecs/mod_g723_1
codecs/mod_g729
codecs/mod_h26x
#codecs/mod_ilbc
#codecs/mod_isac
#codecs/mod_mp4v
codecs/mod_opus
#codecs/mod_sangoma_codec
```

On peut aussi activer ou désactiver les endpoints

```
#endpoints/mod_khomp
endpoints/mod_loopback
#endpoints/mod_opal
#endpoints/mod_portaudio
endpoints/mod_rtc
#endpoints/mod_rtmp
endpoints/mod_skinny
endpoints/mod_sofia
endpoints/mod_verto
#event_handlers/mod_amqp
```

On peut aussi activer ou désactiver les logs et les langues

```
#languages/mod_yaml
loggers/mod_console
#loggers/mod_graylog2
loggers/mod_logfile
loggers/mod_syslog
#loggers/mod_raven
#say/mod_say_de
say/mod_say_en
#say/mod_say_es
#say/mod_say_es_ar
#say/mod_say_fa
say/mod_say_fr
```

A la fin du fichier il y'a les modules xml

```
#timers/mod_timerfd
xml_int/mod_xml_cdr
#xml_int/mod_xml_curl
#xml_int/mod_xml_ldap
#xml_int/mod_xml_radius
xml_int/mod_xml_rpc
xml_int/mod_xml_scgi

#../../libs/freetdm/mod_freetdm
```

Après avoir choisi les modules on reste toujours dans le dossier **/usr /local/freeswitch/** et tape la commande **./configure**

A la fin de la configuration d'installation on obtient des informations très importantes concernant la localisation future des fichiers de configuration comme montre l'image ci-dessous


```
+ FreeSWITCH has been successfully installed. +
+
+ Install sounds: +
+ (uhd-sounds includes hd-sounds, sounds) +
+ (hd-sounds includes sounds) +
+ ----- +
+ make cd-sounds-install +
+ make cd-moh-install +
+
+ make uhd-sounds-install +
+ make uhd-moh-install +
+
+ make hd-sounds-install +
+ make hd-moh-install +
+
+ make sounds-install +
+ make moh-install +
+
+ Install non english sounds: +
+ replace XX with language +
+ (ru : Russian) +
+ (fr : French) +
+ ----- +
+ make cd-sounds-XX-install +
+ make uhd-sounds-XX-install +
+ make hd-sounds-XX-install +
+ make sounds-XX-install +
+
+ Upgrade to latest: +
+ ----- +
+ make current +
+
+ Rebuild all: +
+ ----- +
+ make sure +
+
```

Ensuite On installe les fichiers sons en français par les 4 commandes suivantes :

- ✓ **make cd-sounds-fr-install**
- ✓ **make uhd-sounds-fr-install**
- ✓ **make hd-sounds-fr-install**
- ✓ **make sounds-fr-install**

On installe les exemples fichiers de configuration par la commande: **make samples**

Maintenant que l'installation est terminer on doit se familiarise aux répertoires de configuration de FreeSWITCH qui est **/usr/local/freeswitch**

```
root@ubuntu-VirtualBox:/usr/src/freeswitch-1.8.6# cd /usr/local/freeswitch/
root@ubuntu-VirtualBox:/usr/local/freeswitch# ls
bin  db  grammar  images  lib  mod  run  sounds
conf  fonts  htdocs  include  log  recordings  scripts
root@ubuntu-VirtualBox:/usr/local/freeswitch#
```

Pour démarrer freeswitch on tape la commande **./bin/freeswitch -nc** dans le dossier **/usr/local/freeswitch** et pour accède à l'interface de freeswitch en lançant la commande **./bin/fs_cli** toujours dans le même dossier

```

root@ubuntu-VirtualBox:/usr/local/freeswitch# ./bin/fs_cli
=====
                FSCLI
=====
Anthony Minessale II, Ken Rice,
Michael Jerris, Travis Cross
FreeSWITCH (http://www.freeswitch.org)
Paypal Donations Appreciated: paypal@freeswitch.org
Brought to you by ClueCon http://www.cluecon.com/
=====

                ClueCon.com
=====

Type /help <enter> to see a list of commands

[This app Best viewed at 160x60 or more..]
+OK log level [7]
freeswitch@ubuntu-VirtualBox>

```

Une fois dans la console freeswitch on peut taper la commande /help pour voir les commande disponibles

```

freeswitch@ubuntu-VirtualBox> /help
Command          Description
-----
/help            Help
/exit, /quit, /bye, ...  Exit the program.
/event, /noevents, /nixevent  Event commands.
/log, /nolog      Log commands.
/uuid            Filter logs for a single call uuid
/filter          Filter commands.
/logfilter        Filter Log for a single string.
/debug [0-7]     Set debug level.

freeswitch@ubuntu-VirtualBox>

```

III Configuration des téléphones SIP et test d'appel

Après installation de freeswitch des comptes SIP sont créés par défaut et on peut les voir dans le dossier **/usr/local/freeswitch/conf/directory/default/**

```

root@ubuntu-VirtualBox:~# cd /usr/local/freeswitch/conf/directory/default/
root@ubuntu-VirtualBox:/usr/local/freeswitch/conf/directory/default# ls
1000.xml  1004.xml  1008.xml  1012.xml  1016.xml  brian.xml
1001.xml  1005.xml  1009.xml  1013.xml  1017.xml  default.xml
1002.xml  1006.xml  1010.xml  1014.xml  1018.xml  example.com.xml
1003.xml  1007.xml  1011.xml  1015.xml  1019.xml  skinny-example.xml

```

On voit qu'il y a 20 comptes SIP 1000 à 1019 et un compte SCCP

Le mot de passe de ces comptes est par défaut **1234** et fixé par défaut dans le fichier **/usr/local/freeswitch/conf/vars.xml**

Maintenant on paramètre deux téléphones SIP avec les comptes 1005 et 1010 pour tester les appels

Le numéro 1005 avec microsip

Compte

Nom du compte: 1005

Serveur SIP: 192.168.1.111

Proxy SIP: 192.168.1.111

Nom utilisateur *: 1005

Domaine *: 192.168.1.111

Login: 1005

Mot de passe: ••••

Afficher le mot de passe

Afficher le nom

Numéro de la boîte ...

Préfixe d'appel

Dial Plan

Hide Caller ID

Chiffrement: Désactivé

Transport: Auto (UDP & TCP)

Adresse publique: Auto

Actualiser l'enregist...: 300

Garder en direct: 15

Afficher ma présence

Autoriser la ré-écriture de l'IP

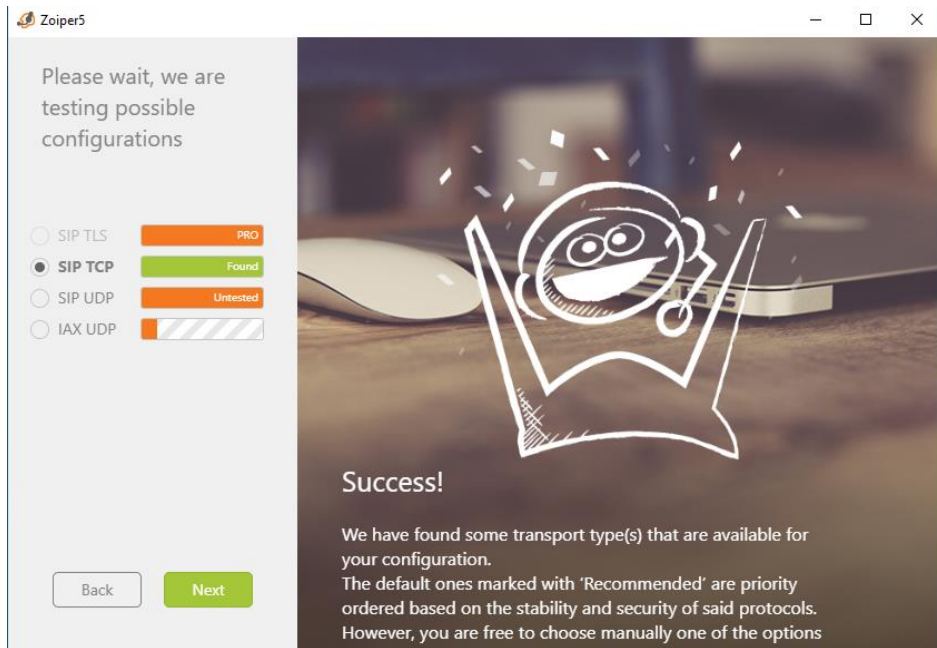
ICE

Désactiver les temporisateurs de ses

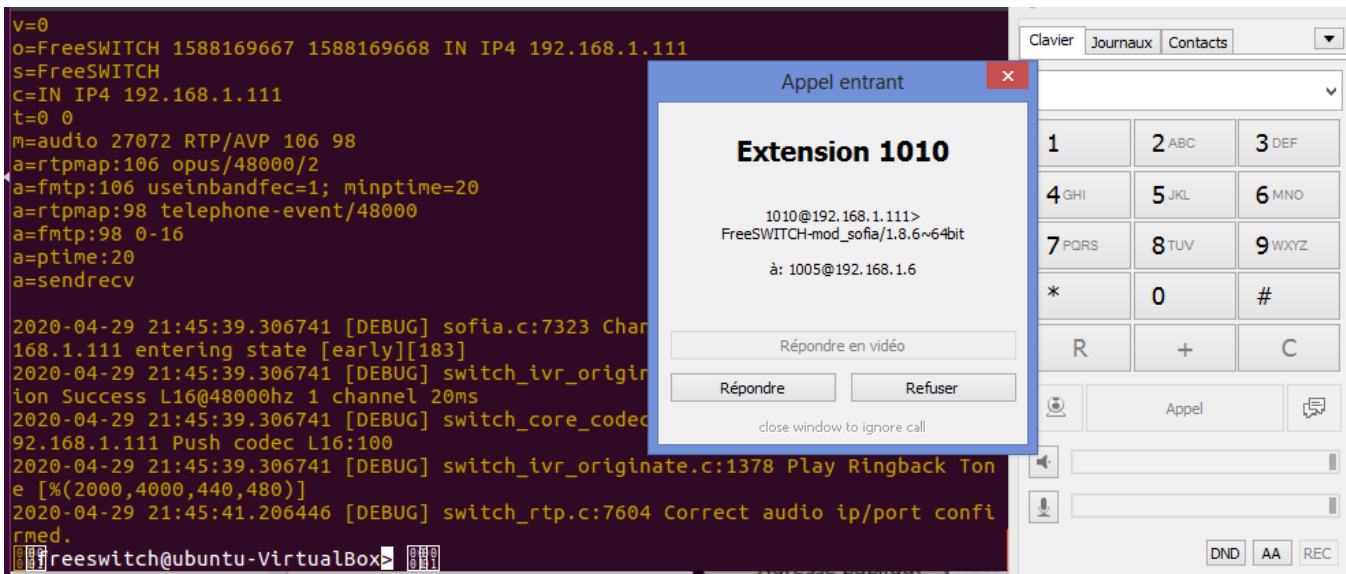
Sauvegarder Annuler

Le numéro 1010 avec zoiper

On peut voir que le téléphone a bien vu le serveur SIP en UDP



Maintenant que les deux téléphones sont bien connectés avec les comptes par défaut on peut tester les appels et on voit que les appels passent



IV Fonctionnalités de base de FreeSWITCH

1- Ajout de nouveau compte utilisateurs

Comme on avait pu voir après installation on n'a des comptes SIP créer par défaut dans le dossier `/usr/local/freeswitch/conf/directory/default/`. On a la possibilité de créer d'autre compte avec le script `add_user` dans le dossier dezippé de FreeSWITCH `/usr/src/freeswitch-1.8.6/scripts/perl`

```

root@ubuntu-VirtualBox:~# cd /usr/src/freeswitch-1.8.6/scripts/perl/
root@ubuntu-VirtualBox:/usr/src/freeswitch-1.8.6/scripts/perl# ls
add_user          fsconsole.pl      POE-Filter-FSSocket
analyze-debug-alloc.pl  fslog             puppet.cgi
blacklist.pl      fs.pl             randomize-passwords.pl
call.cgi          ftc_dnc_register.pl  rpc.cgi
cnam.cgi          g729_activate     sendmail
combineconf.pl    honeypot.pl       snom-pnpd.pl
dhcp-inform.pl    mkgws.pl          sock.pl
extract_log_range.pl  multicast         tag.pl
fax.cgi           mwi_event.pl      timezone-gen.pl
f-off-friendly-scanner.pl  originate.pl     tonegen
FreeSWITCH        perltidyrc        wplookup.cgi

```

On peut créer un utilisateur 1021 avec ce script comme montre l'image ci-dessous

```

root@ubuntu-VirtualBox:/usr/src/freeswitch-1.8.6/scripts/perl# ./add_user 1021

Added 1021 in file /usr/local/freeswitch/conf/directory/default/1021.xml

Operation complete. 1 user added.
Be sure to reloadxml.

If CPAN module Regexp::Assemble were installed this program would be able to suggest a regex for your new users.

```

Après création on peut voir que le fichier 1021.xml est créé dans le répertoire **/usr/local/freeswitch/conf/directory/default/**

```

root@ubuntu-VirtualBox:/usr/src/freeswitch-1.8.6/scripts/perl# cd /usr/local/freeswitch/conf/directory/default/
root@ubuntu-VirtualBox:/usr/local/freeswitch/conf/directory/default# ls
1000.xml  1005.xml  1010.xml  1015.xml  1021.xml
1001.xml  1006.xml  1011.xml  1016.xml  brian.xml
1002.xml  1007.xml  1012.xml  1017.xml  default.xml
1003.xml  1008.xml  1013.xml  1018.xml  example.com.xml
1004.xml  1009.xml  1014.xml  1019.xml  skinny-example.xml

```

Et on peut voir le contenu du fichier avec la commande cat

```

root@ubuntu-VirtualBox:/usr/local/freeswitch/conf/directory/default# cat 1021.xml
<include>
  <user id="1021">
    <params>
      <param name="password" value="${default_password}"/>
      <param name="vm-password" value="1021"/>
    </params>
    <variables>
      <variable name="toll_allow" value="domestic,international,local"/>
      <variable name="accountcode" value="1021"/>
      <variable name="user_context" value="default"/>
      <variable name="effective_caller_id_name" value="Extension 1021"/>
      <variable name="effective_caller_id_number" value="1021"/>
      <variable name="outbound_caller_id_name" value="${outbound_caller_name}"/>
    >
      <variable name="outbound_caller_id_number" value="${outbound_caller_id}"/>
    >
      <variable name="callgroup" value="techsupport"/>
    </variables>
  </user>
</include>

```

On peut se connecter avec MicroSIP sur le compte qu'on vient de créer

The screenshot shows a window titled 'Compte' with the following fields and values:

- Nom du compte: 1021
- Serveur SIP: 192.168.1.111
- Proxy SIP: 192.168.1.111
- Nom utilisateur *: 1021
- Domaine *: 192.168.1.111
- Login: 1021
- Mot de passe: [masked]
- Afficher le nom: [empty]
- Numéro de la boîte ...: [empty]
- Préfixe d'appel: [empty]
- Dial Plan: [empty]
- Hide Caller ID:
- Chiffrement: Désactivé
- Transport: Auto (UDP & TCP)
- Adresse publique: Auto
- Actualiser l'enregist...: 300
- Garder en direct: 15
- Afficher ma présence:
- Autoriser la ré-écriture de l'IP:
- ICE:
- Désactiver les temporisateurs de ses:

Buttons at the bottom: [x], Sauvegarder, Annuler.

On voit sur la console de freeswitch que cet utilisateur s'est bien connecté avec l'adresse 192.168.1.20 avec la commande **show registrations**

```
freeswitch@ubuntu-VirtualBox> show registrations
reg_user,realn,token,url,expires,network_ip,network_port,network_proto,hostname,
metadata
1021,192.168.1.111,3bb805896b5f46fa9b62b02cd3f83830,sofia/internal/sip:1021@192.
168.1.20:60208;ob,1588445972,192.168.1.20,60208,udp,ubuntu-VirtualBox,
1 total.
```

Il y'a aussi la possibilité de créer en ensemble de compte avec ce scripte avec l'action **--users** comme montre l'image ci-dessous

```
root@ubuntu-VirtualBox:/usr/src/freeswitch-1.8.6/scripts/perl# ./add_user --user
s=1022-1025

Added 1022 in file /usr/local/freeswitch/conf/directory/default/1022.xml
Added 1023 in file /usr/local/freeswitch/conf/directory/default/1023.xml
Added 1024 in file /usr/local/freeswitch/conf/directory/default/1024.xml
Added 1025 in file /usr/local/freeswitch/conf/directory/default/1025.xml

Operation complete. 4 users added.
Be sure to reloadxml.

If CPAN module Regexp::Assemble were installed this program would be able to sug
gest a regex for your new users.
```

On vient de créer les utilisateurs 1022, 1023, 1024,1025

2- Définition du plan de numérotation

En général, les plans de numérotation sont utilisés pour acheminer un appel vers un point de terminaison, qui peut être un poste traditionnel, une messagerie vocale, un menu de réponse vocale interactive (IVR) ou une autre application compatible. Les plans de numérotation sont extrêmement flexibles.

Dans FreeSWITCH le plan de numérotation est défini dans fichier **/usr/local/freeswitch/conf/dialplan/default.xml** et dans la section `<extension name="Local_Extension">` on peut voir que les numéros sont définis comme suit :

```
<condition field="destination_number" expression="^(10[01][0-9])$">
```

C'est-à-dire les numéros commençant de 1000 à 1019 ce sont les numéros de compte par défaut de FreeSWITCH

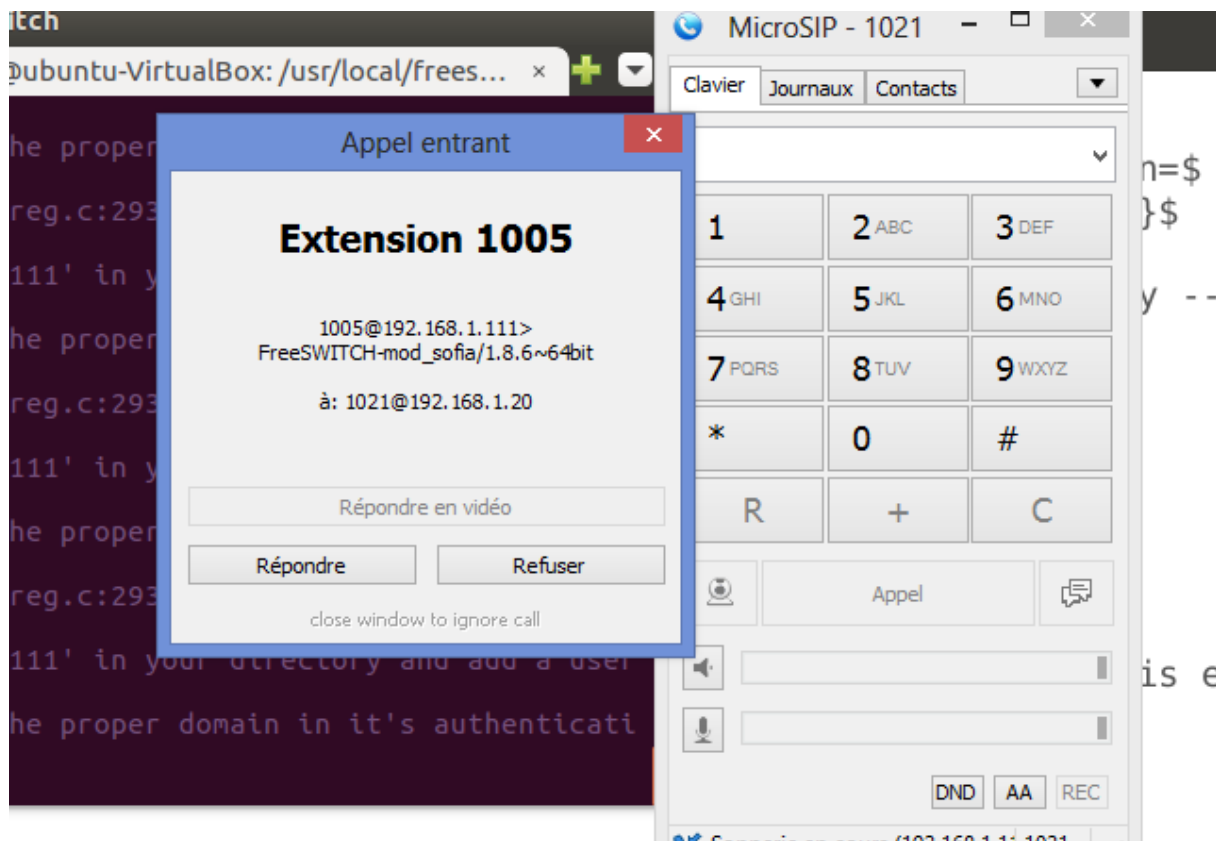
On définit des numéros pour des comptes qu'on vient de créer en en ajoutant **2** comme suit

```
<condition field="destination_number" expression="^(10[012][0-9])$">
```

```
-->
<extension name="Local_Extension">
  <condition field="destination_number" expression="^(10[012][0-9])$">
    <action application="export" data="dialed_extension=$1"/>
    <!-- bind_meta_app can have these args <key> [a|b|ab] [a|b|o|s] <app> --
  </condition>
</extension>
```

C'est pour dire tous numéros commençant de 1000 à 1029

Maintenant on peut voir les comptes qu'on avait créé peuvent appeler et recevoir d'appel



3- Gestion des conférences

mod_conference fournit un service de passerelle de conférence entrante et sortante pour FreeSWITCH™. Il peut traiter plusieurs débits binaires, charger divers profils qui spécifient les commandes DTMF, jouer des sons et des tonalités rapides, et de nombreuses autres fonctions.

Les conférences créées dans le plan de numérotation utilisent des profils définis dans **/usr/local/freeswitch/conf/autoload_configs/conference.conf.xml**. Dans ce fichier plusieurs profils de conférence sont définis et les numéros pour attribués à ces conférences sont défini dans le fichier **/usr/local/freeswitch/conf/dialplan/default.xml**. Voici le un tableau récapitulatif des numéros de conférence selon le taux d'échantillonnage

Numéros de conférence	taux d'échantillonnage de la conférence
3000 - 30099	8 KHz
3100 - 3199	16 KHz
3200 - 3299	32 KHz
3300 - 3399	48 KHz

Nous pouvons regarder certain profil dans le fichier **/usr/local/freeswitch/conf/autoload_configs/conference.conf.xml**

- ✓ Le profil wideband

```
<profile name="wideband">
  <param name="domain" value="${domain}"/>
  <param name="rate" value="16000"/>
  <param name="interval" value="20"/>
  <param name="energy-level" value="100"/>
  <!-- <param name="sound-prefix" value="${sound_prefix}"/> -->
  <param name="muted-sound" value="conference/conf-muted.wav"/>
```

- ✓ Le profil ultrawideband

```
<profile name="ultrawideband">
  <param name="domain" value="${domain}"/>
  <param name="rate" value="32000"/>
  <param name="interval" value="20"/>
  <param name="energy-level" value="100"/>
  <!-- <param name="sound-prefix" value="${sound_prefix}"/> -->
  <param name="muted-sound" value="conference/conf-muted.wav"/>
  <param name="unmuted-sound" value="conference/conf-unmuted.wav"/>
```

- ✓ Le profil muc-video-stereo-720

```

<profile name="video-mcu-stereo-720">
  <param name="domain" value="${domain}"/>
  <param name="rate" value="48000"/>
  <param name="channels" value="2"/>
  <param name="interval" value="20"/>
  <param name="energy-level" value="200"/>
  <!-- <param name="tts-engine" value="flite"/> -->
  <!-- <param name="tts-voice" value="kal16"/> -->
  <param name="muted-sound" value="conference/conf-muted.wav"/>
  <param name="unmuted-sound" value="conference/conf-unmuted.wav"/>

```

On peut regarder aussi la définition du plan de numérotation des conférences dans fichier `/usr/local/freeswitch/conf/dialplan/default.xml`

- ✓ Tous numéros commençant par 30 suivant de 2 chiffres pour accéder à la conférence du profile default

```

-->
<extension name="nb_conferences">
  <condition field="destination_number" expression="^(30\d{2})$">
    <action application="answer"/>
    <action application="conference" data="$1-${domain_name}@default"/>
  </condition>
</extension>

```

- ✓ Tous numéros commençant par 31 suivant de 2 chiffres pour accéder à la conférence du profile wideband

```

<extension name="wb_conferences">
  <condition field="destination_number" expression="^(31\d{2})$">
    <action application="answer"/>
    <action application="conference" data="$1-${domain_name}@wideband"/>
  </condition>
</extension>

```

- ✓ Tous numéros commençant par 35 suivant de 2 chiffres pour accéder à la conférence du profile video-mcu-stereo

```

<extension name="cdquality_conferences">
  <condition field="destination_number" expression="^(35\d{2})$">
    <action application="answer"/>
    <action application="conference" data="$1@video-mcu-stereo"/>
  </condition>
</extension>

```

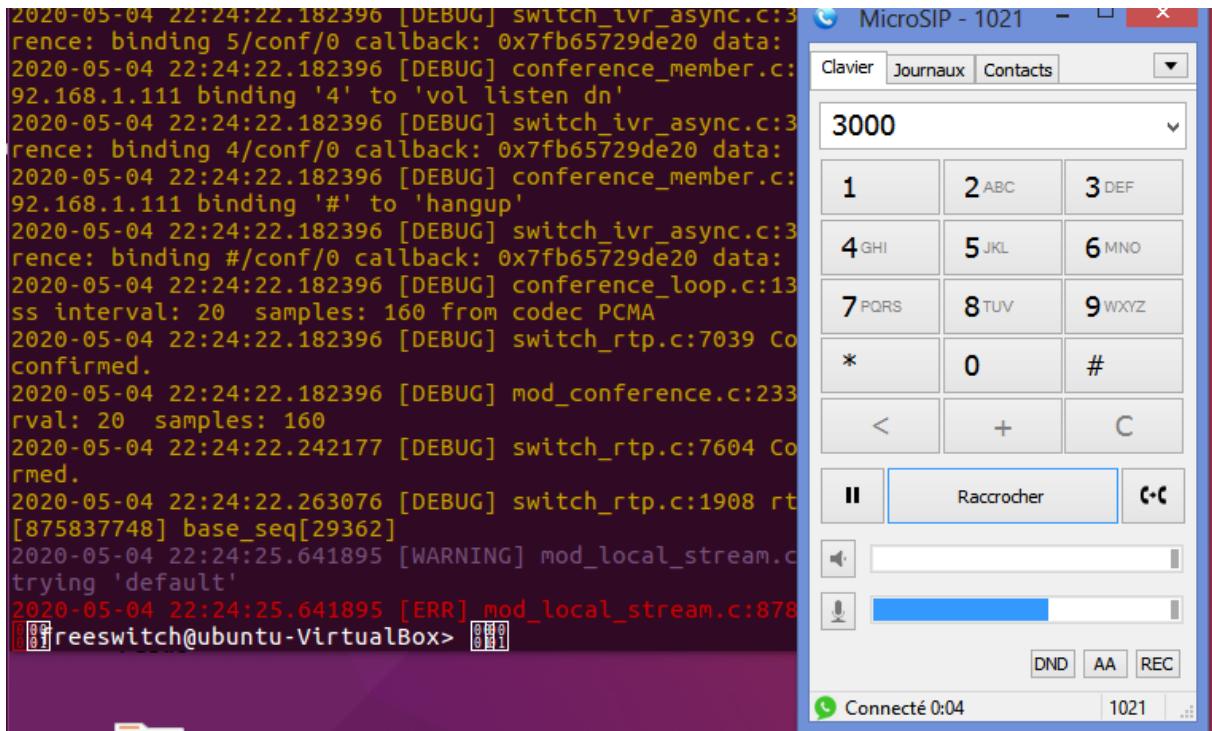
- ✓ Tous numéros commençant par 36 suivant de 2 chiffres pour accéder à la conférence du profile video-mcu-stereo-720

```

<extension name="cdquality_conferences_720">
  <condition field="destination_number" expression="^(36\d{2})$">
    <action application="answer"/>
    <action application="conference" data="$1@video-mcu-stereo-720"/>
  </condition>
</extension>

```

On peut tester par exemple une conférence en composant le numéro 3000 et constate qu'il nous que vous êtes le seul participant de cette conférence

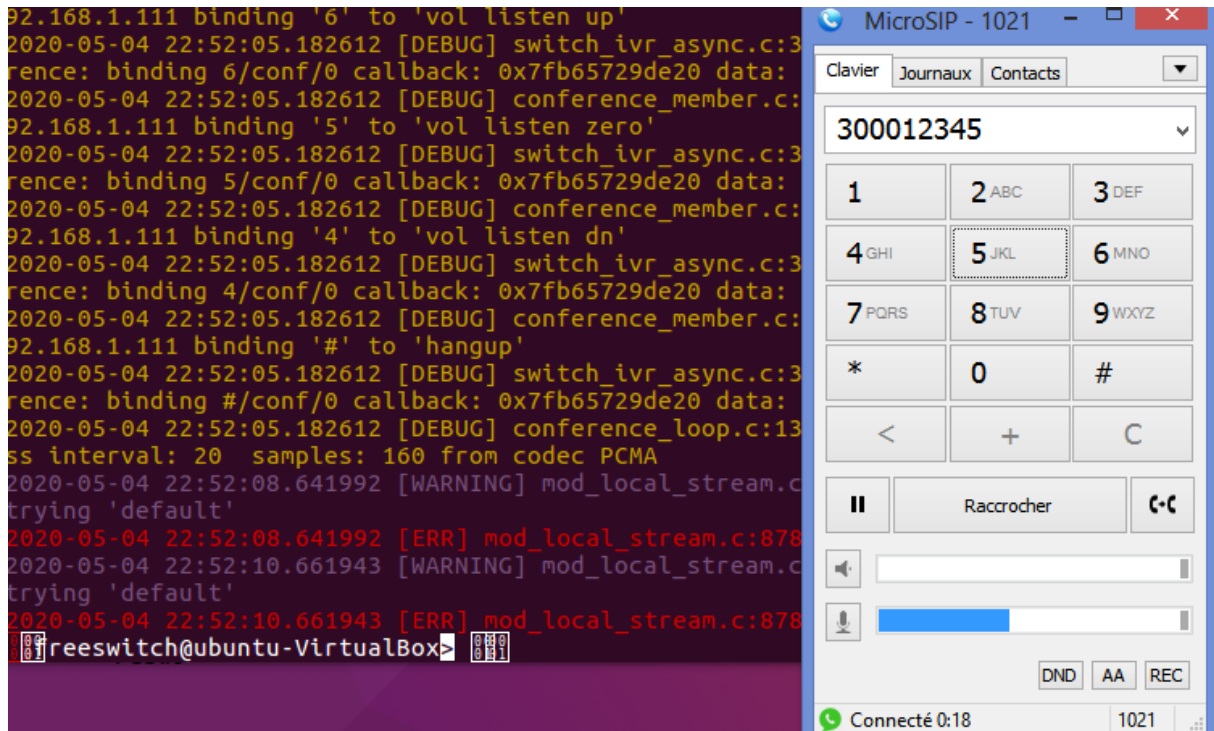


Sur les autres téléphones si on compose le même numéro 3000 et on entre en communication avec les autres participants

Il y'a la possibilité d'ajouter un code PIN pour accéder à la conférence. Pour cela on ajoute dans le profil de la conférence la ligne suivante **<param name="pin" value="12345" />**

```
<!-- Conference pin -->  
<param name="pin" value="12345" />  
</param name="pin" value="12345" />
```

Maintenant quand l'utilisateur compose le numéro 3000 pour accéder à la conférence on lui demande d'entrer le code de la conférence on met le code qui était 12345

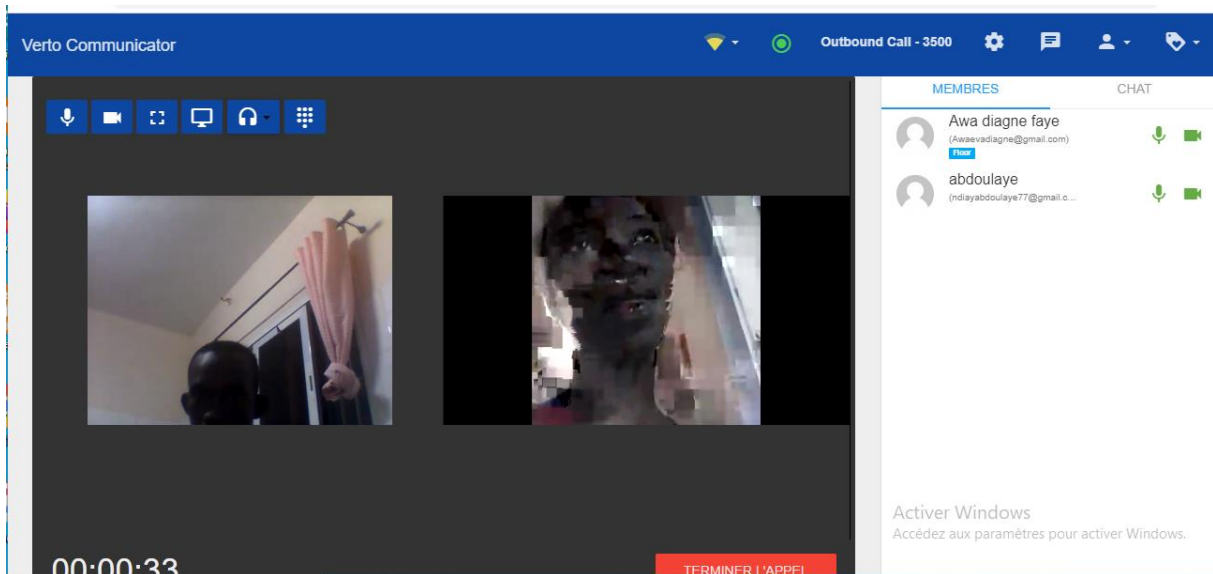


Après avoir mis le code on peut voir sur la console du serveur que le code est bon (Success)

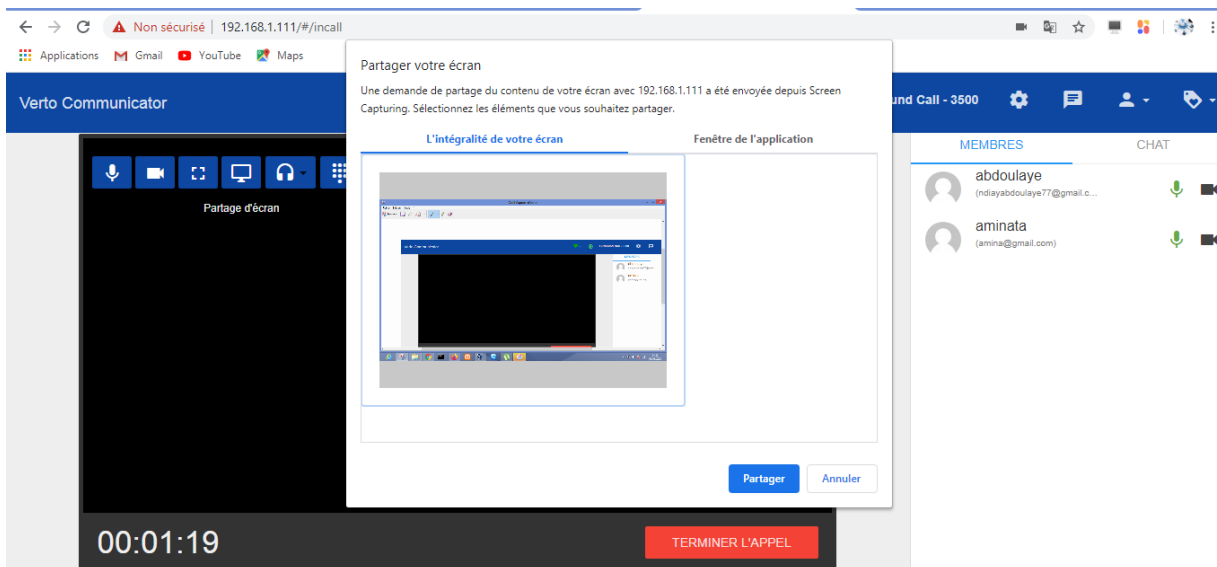


On peut aussi tester la conférence du profil **video-mcu-stereo-720** avec le numéro 3500 sur le navigateur avec Verto.

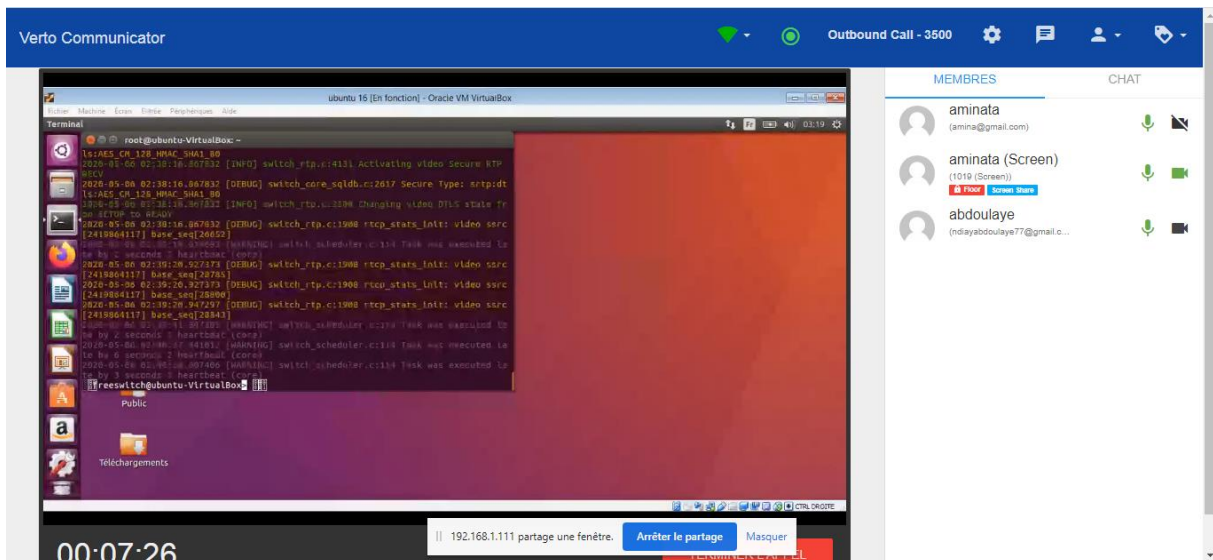
Quand on se connecte sur deux navigateurs en tapant <https://192.168.1.1111> on voit peut que ces deux utilisateurs entrent en communications en composant les numéros 3500



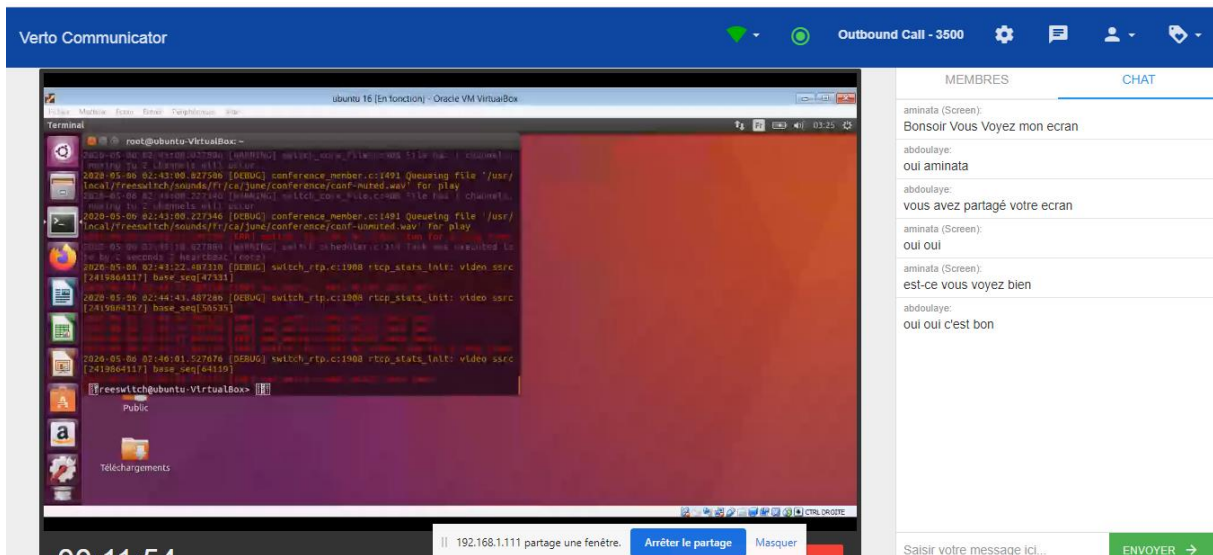
L'interface de Verto nous donne aussi la possibilité de partage d'écran après avoir installé certains plugins. Après avoir installé ces prérequis on peut cliquer sur l'icône de partage d'écran et il nous demande de choisir la fenêtre qu'on veut partager



Après avoir choisi la fenêtre à partager on peut voir que notre écran est partagé aux autres participants de la conférence



On peut aussi voir que Verto nous la possibilité de faire du chat entre les participant de la conférence



4- Gestion des files d'attente d'un centre d'appel

FreeSWITCH dispose un module **mod_callcenter** qui est une application de mise en file d'attente des appels entrants qui peut être utilisée pour les besoins du centre d'appels **il y'a trois concepts autour des centres de contact avec FreeSWITCH**

- ✓ **file d'attente** appelée *queue*
- ✓ Agent
- ✓ *tiers qui sont les membres d'une file d'attente*

Dans FreeSWITCH les file d'attente, les agents et les membres d'une file sont tous gères dans le même fichier /usr/local/freeswitch/conf/autoload_configs/callcenter.conf.xml

Mais tous d'abord nous allons activer le module callcenter dans le fichier /usr/local/freeswitch/conf/autoload_configs/modules.conf.xml comme suit

```

<!-- Third party modules -->
<!--<load module="mod_nibblebill"/>-->
<load module="mod_callcenter"/>

```

a) Gestion des files d'attentes

Comme on avait vu en haut les files d'attente sont gérées dans le fichier /usr/local/freeswitch/conf/autoload_configs/callcenter.conf.xml

Nous pouvons voir dans ce fichier un exemple de file d'attente créé dans les balise `<queues>` `</queues>` comme suit :

```

<queues>
  <queue name="support@default">
    <param name="strategy" value="longest-idle-agent"/>
    <param name="moh-sound" value="${hold_music}"/>
    <!--<param name="record-template" value="${recordings_dir}/${strftime(%Y-%m-%d-%H-%M-%S)}.${destination_number}.${caller_id_number}.${uuid}.wav"/>-->
    <param name="time-base-score" value="system"/>
    <param name="max-wait-time" value="0"/>
    <param name="max-wait-time-with-no-agent" value="0"/>
    <param name="max-wait-time-with-no-agent-time-reached" value="5"/>
    <param name="tier-rules-apply" value="false"/>
    <param name="tier-rule-wait-second" value="300"/>
    <param name="tier-rule-wait-multiply-level" value="true"/>
    <param name="tier-rule-no-agent-no-wait" value="false"/>
    <param name="discard-abandoned-after" value="60"/>
    <param name="abandoned-resume-allowed" value="false"/>
  </queue>

```

Le nom de la file d'attente est spécifié dans les balises `<queue>`/`</queue>` ensuite il y'a les paramètres et leurs valeurs. Voici une liste de quelques paramètres d'une file d'attente et leurs significations

Strategy : La stratégie définit la façon dont les appels sont répartis dans une file d'attente

moh-sound : définit la music d'attente

tier-rule-wait-second : définit le temps en secondes pendant lequel un appelant doit attendre avant de passer au niveau suivant.

tier-rule-wait-multiply-level: Peut-être vrai ou faux. Si Faux, alors une fois que la règle de niveau d'attente-seconde est passée, l'appelant est proposé à tous les niveaux dans l'ordre (niveau / position). Si la valeur est True, la seconde de règle de niveau d'attente sera multipliée par le niveau de niveau et l'appelant devra attendre à chaque niveau de règle de seconde d'attente de niveau avant de passer au niveau suivant

discard-abandoned-after :définit le nombre de secondes avant que nous supprimions complètement un membre abandonné de la file d'attente

max-wait-time-with-no-agent : définit la durée pendant laquelle la file d'attente doit être vide (sans agents connectés, en appel ou non) avant de déconnecter tous les membres

max-wait-time-with-no-agent-time-reached :définit la durée après laquelle le délai d'attente maximal sans agent est atteint pour rejeter le nouvel appelant

ring-progressively-delay : définit le délai à attendre avant de commencer l'appel à l'agent suivant lors de l'utilisation de la stratégie de file d'attente de «sonnerie progressive».

max-wait-time : définit le délai avant de quitter l'application callcenter SI le membre n'a pas reçu de réponse d'un agent

Pour aller loin dans les stratégies de gestion des files d'attente consulter le tableau suivant :

stratégie	Signification
ring-all	Tous les téléphones des agents disponibles sonnent
longest-idle-agent	Sonne l'agent qui est resté inactif le plus longtemps en tenant compte du niveau
round-robin	Fait sonner l'agent en tour de rôle mais souvenez-vous du dernier agent essayé.
top-down	Fait sonner l'agent en position d'ordre à partir de 1 pour chaque membre
agent-with-least-talk-time	Fait sonner l'agent avec le moins de temps de conversation.
agent-with-fewest-calls	Fait sonner l'agent avec le moins d'appels.
sequentially-by-agent-order	Sonne les agents séquentiellement par niveau et ordre
random	Sonner les agents dans un ordre aléatoire
ring-progressively	Fait sonner les agents de la même manière que de haut en bas, mais en gardant les membres précédents qui sonnent (cela conduit à la fin à tout sonner).

b) Gestion des agents

Les agents sont aussi gérés dans le fichier

/usr/local/freeswitch/conf/autoload_configs/callcenter.conf.xml entre les balises <agents><agents>. Voici une liste de quelques options de création d'un agent et leurs significations :

Type : 2 types, 'callback' et 'uuid-standby'. Callback tentera d'atteindre l'agent via la valeur des champs de contact. uuid-standby tentera de relier l'appel directement à l'aide de l'agent uuid.

Contact : Une simple chaîne de numérotation peut être insérée ici, comme: user / 1000 @ default. Si vous utilisez verto: \$ {verto_contact (1000 @ default)}

Status : Définissez le statut actuel d'un agent soit disponible, en pause ou indisponible

max-no-answer : Si l'agent ne répond pas aux appels ce nombre de fois, son statut passe automatiquement à *En pause*.

wrap-up-time : Temps d'attente avant de remettre l'agent dans la file d'attente disponible pour recevoir un autre appel, pour lui permettre de terminer les notes ou d'autres tâches

reject-delay-time : Si l'agent appuie sur le bouton de rejet de son téléphone, attendez cette durée définie

busy-delay-time : Si l'agent est sur *Ne pas déranger*, attendez cette heure définie avant de le réessayer.

no-

answer-delay-time : Si l'agent ne répond pas à l'appel, attendez cette heure définie avant de le réessayer

truncate-

agents-on-load : Si défini sur **true**, nous supprimerons tous les agents lors du chargement du module. **false** par défaut.

truncate-tiers-

on-load Si défini sur **true**, nous supprimerons tous les niveaux lors du chargement du module. **false** par défaut.

On va créer deux agents ce fichiers comme montre l'image ci-dessous

```
<agents>
<agent name="1000@default" type="callback" contact="[leg_timeout=10]user/1000@default" status="Available" max-no-answer="3" wrap-up-time="10" reject-delay-time="10" busy-delay-time="60" />
</agents>
<agents>
<agent name="1021@default" type="callback" contact="[leg_timeout=10]user/1021@default" status="Available" max-no-answer="3" wrap-up-time="10" reject-delay-time="10" busy-delay-time="60" />
</agents>
<tiers>
```

c) Gestion des membres d'un centre d'appel

Les membres d'un centre d'appel sont gérés dans le fichier

/usr/local/freeswitch/conf/autoload_configs/callcenter.conf.xml dans les balises

<tiers></tiers> avec les options suivantes

agent : définit le nom de l'agent

queue : définit le nom de la file d'attente

level : définit le niveau de l'agent

position : définit la position de l'agent

Nous allons ajouter les agents 1000 et 1005 dans la file d'attente **support@default** comme montre l'image ci-dessous

```
<tiers>
<!-- If no level or position is provided, they will default to 1. You should do this to keep db value on restart. -->
<tier agent="1000@default" queue="support@default" level="1" position="1"/>
<tier agent="1021@default" queue="support@default" level="1" position="2"/>
</tiers>
```

Maintenant il nous reste qu'à définir le numéro de la file d'attente dans le fichier

/usr/local/freeswitch/conf/dialplan/default.xml comme suit

```
<extension name="centreatappel">
<condition field="destination_number" expression="^(10000)$">
<action application="answer"/>
<action application="callcenter" data="support@default"/>
</condition>
</extension>
```

Maintenant un utilisateur doit composer le numéro 10000 pour entrer dans la file

support@default

d) Quelques commandes utiles de gestion de centre d'appels et Test

Il y'a la possibilité de gérer les centre d'appels sur la console de FreeSWITCH avec la commande **callcenter_config**. La syntaxe de la commande est la suivante

callcenter_config <target> <args>

Voici quelques commandes de gestions de centre d'appel

```
callcenter_config agent add [name] [type] |
callcenter_config agent del [name] |
callcenter_config agent reload [name] |
callcenter_config agent set status [agent_name] [status] |
callcenter_config agent set state [agent_name] [state] |
callcenter_config agent set contact [agent_name] [contact] |
callcenter_config agent set ready_time [agent_name] [wait till epoch] |
callcenter_config agent set reject_delay_time [agent_name] [wait second] |
callcenter_config agent set busy_delay_time [agent_name] [wait second] |
callcenter_config agent set no_answer_delay_time [agent_name] [wait second] |
callcenter_config agent get status [agent_name] |
callcenter_config agent get state [agent_name] |
callcenter_config agent get uuid [agent_name] |
callcenter_config agent list [[agent_name]] |
callcenter_config tier add [queue_name] [agent_name] [[level]] [[position]] |
callcenter_config tier set state [queue_name] [agent_name] [state] |
callcenter_config tier set level [queue_name] [agent_name] [level] |
callcenter_config tier set position [queue_name] [agent_name] [position] |
callcenter_config tier del [queue_name] [agent_name] |
callcenter_config tier reload [queue_name] [agent_name] |
callcenter_config tier list |
callcenter_config queue load [queue_name] |
callcenter_config queue unload [queue_name] |
callcenter_config queue reload [queue_name] |
callcenter_config queue list |
callcenter_config queue list agents [queue_name] [status] [state] |
callcenter_config queue list members [queue_name] |
callcenter_config queue list tiers [queue_name] |
callcenter_config queue count |
callcenter_config queue count agents [queue_name] [status] [state] |
callcenter_config queue count members [queue_name] |
callcenter_config queue count tiers [queue_name]
```

Nous allons juste utiliser certaines commandes pour quelques actions de base

- ✓ Afficher la liste des agents de votre freeswitch avec la commande

callcenter_config agent list

```
freeswitch@ubuntu-VirtualBox> callcenter_config agent list
name|system|uuid|type|contact|status|state|max_no_answer|wrap_up_time|reject_delay_time|busy_delay_time|no_answer_delay_time|last_bridge_start|last_bridge_end|last_offered_call|last_status_change|no_answer_count|calls_answered|talk_time|ready_time|external_calls_count
1021|single_box||Callback|user/1021|Available|Waiting|0|0|0|0|0|0|0|0|1589139045|1588804607|0|0|0|0|0
1000|single_box||Callback|user/1000|Available|Waiting|0|0|0|0|0|0|0|0|1588804600|0|0|0|0|0
1000@default|single_box||callback|[leg_timeout=10]user/1000@default|On Break|Waiting|3|10|10|60|0|0|0|1589136922|1589136922|3|0|0|0|0
+OK
```

- ✓ Supprimer un agent agents par exemple l'agent 1000 avec la commande **callcenter_config agent del 1000**

```
freeswitch@ubuntu-VirtualBox> callcenter_config agent del 1000
+OK
2020-05-10 19:43:09.516284 [DEBUG] mod_callcenter.c:935 Deleted Agent 1000
freeswitch@ubuntu-VirtualBox> callcenter_config agent list
name|system|uuid|type|contact|status|state|max_no_answer|wrap_up_time|reject_delay_time|busy_delay_time|no_answer_delay_time|last_bridge_start|last_bridge_end|last_offered_call|last_status_change|no_answer_count|calls_answered|talk_time|ready_time|external_calls_count
1021|single_box||Callback|user/1021|Available|Waiting|0|0|0|0|0|0|0|0|1589139045|1588804607|0|0|0|0|0
1000@default|single_box||callback|[leg_timeout=10]user/1000@default|On Break|Waiting|3|10|10|60|0|0|0|1589136922|1589136922|3|0|0|0|0
+OK
```

On peut bien voir que l'agent 1000 a été supprimé

- ✓ Ajouter un agent 1010 avec son type Callback avec la commande **callcenter_config agent add 1010 Callback**

```
freeswitch@ubuntu-VirtualBox> callcenter_config agent add 1010 Callback
+OK
2020-05-10 19:49:49.276377 [DEBUG] mod_callcenter.c:907 Adding Agent 1010 with type Callback with default status Logged Out
```

- ✓ Regarder le statut de l'agent 1010 avec la commande **callcenter_config agent get status 1010**

```
freeswitch@ubuntu-VirtualBox> callcenter_config agent get status 1010
Logged Out
2020-05-10 19:56:39.656656 [DEBUG] mod_callcenter.c:990 Get Info Agent 1010 status = Logged Out
```

On peut voir le statut de l'agent 1010 est Logged Out c'est à dire déconnecter

- ✓ Changer le status de l'agent 1010 à Available c'est-à-dire disponible ou connecter avec la commande **callcenter_config agent set status 1010 Available**

```
freeswitch@ubuntu-VirtualBox> callcenter_config agent set status 1010 Available
+OK
2020-05-10 20:00:47.097120 [DEBUG] mod_callcenter.c:1179 Updated Agent 1010 set status = Available
```

- ✓ Ajouter le numero 1010 à l'agent 1010 avec la commande **callcenter_config agent set contact 1010 user/1010**

```
freeswitch@ubuntu-VirtualBox> callcenter_config agent set contact 1010 user/1010
+OK

2020-05-10 21:05:30.836268 [DEBUG] mod_callcenter.c:1179 Updated Agent 1010 set co
ntact = user/1010
```

- ✓ Afficher la liste des files d'attente avec la commande **callcenter_config queue list**

```
freeswitch@ubuntu-VirtualBox> callcenter_config queue list
name|strategy|moh_sound|time_base_score|tier_rules_apply|tier_rule_wait_second|t
ier_rule_wait_multiply_level|tier_rule_no_agent_no_wait|discard_abandoned_after|
abandoned_resume_allowed|max_wait_time|max_wait_time_with_no_agent|max_wait_time
_with_no_agent_time_reached|record_template|calls_answered|calls_abandoned|ring_
progressively_delay|skip_agents_with_external_calls|agent_no_answer_status
support@default|top-down|local_stream://moh|system|false|300|true|false|60|false
|0|0|5||0|6|0|true|On Break
+OK
```

On peut voir qu'il y'a que la file **support@default** avec leur différents paramètres

- ✓ Afficher les membres des files d'attente avec la commande **callcenter_config tier list**

```
freeswitch@ubuntu-VirtualBox> callcenter_config tier list
queue|agent|state|level|position
support@default|1000@default|No Answer|1|1
support@default|1021|Ready|1|1
+OK
```

On peut voir que y'a les agents 1000@default et 1021 dans la file support@default

- ✓ Supprimer l'agent 1021 dans la file support@default avec la commande **callcenter_config tier support@default 1021**

```
freeswitch@ubuntu-VirtualBox> callcenter_config tier del support@default 1021
+OK

2020-05-10 20:14:57.176226 [DEBUG] mod_callcenter.c:1310 Deleted tier Agent 1021
in Queue support@default
```

- ✓ Ajouter l'agent 1010 en niveau 1 et position 1 dans la file support@default avec la **callcenter_config tier support@default 1010 1 1**

```
freeswitch@ubuntu-VirtualBox> callcenter_config tier add support@default 1010 1 1
+OK

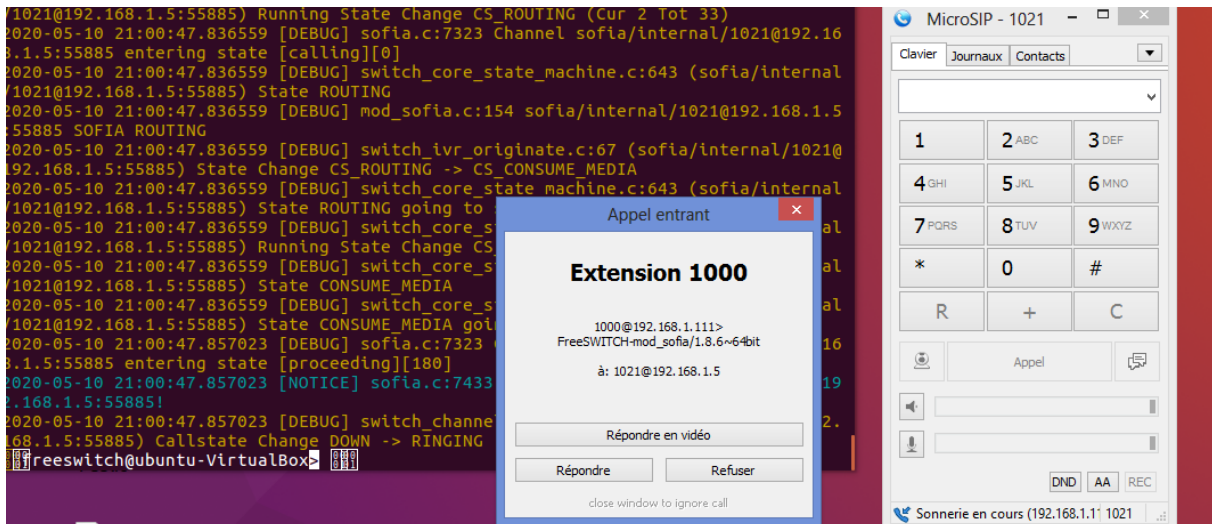
2020-05-10 20:23:05.476378 [DEBUG] mod_callcenter.c:1220 Adding Tier on Queue supp
ort@default for Agent 1010, level 1, position 1
```

- ✓ On peut ajouter ensuite l'agent 1021 en niveau 1 et position 2 1 dans la file support@default avec la **callcenter_config tier support@default 1021 1 2**

```
freeswitch@ubuntu-VirtualBox> callcenter_config tier add support@default 1021 1 2
+OK

2020-05-10 20:55:03.896196 [DEBUG] mod_callcenter.c:1220 Adding Tier on Queue supp
ort@default for Agent 1021, level 1, position 2
```

Maintenant on peut tester le numéro 10000 pour entrer dans la file d'attente on voit que le téléphone de l'agent 1021 sonne



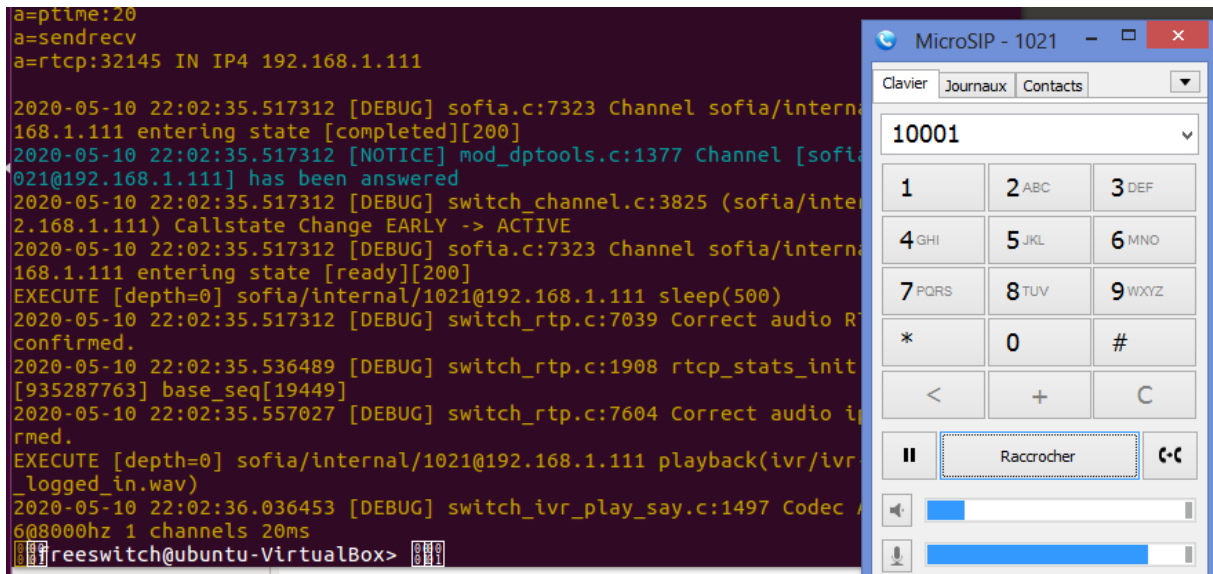
e) Définition des numéros de connexion et de déconnexion d'un agent

En cas télétravail nous allons définir des numéros aux agents pour se connecter et se déconnecter. Par exemple quand l'agent appelle le numéro 10001 on lui met automatiquement dans la file d'attente et quand il appelle le numéro 10002 on lui retire de la file.

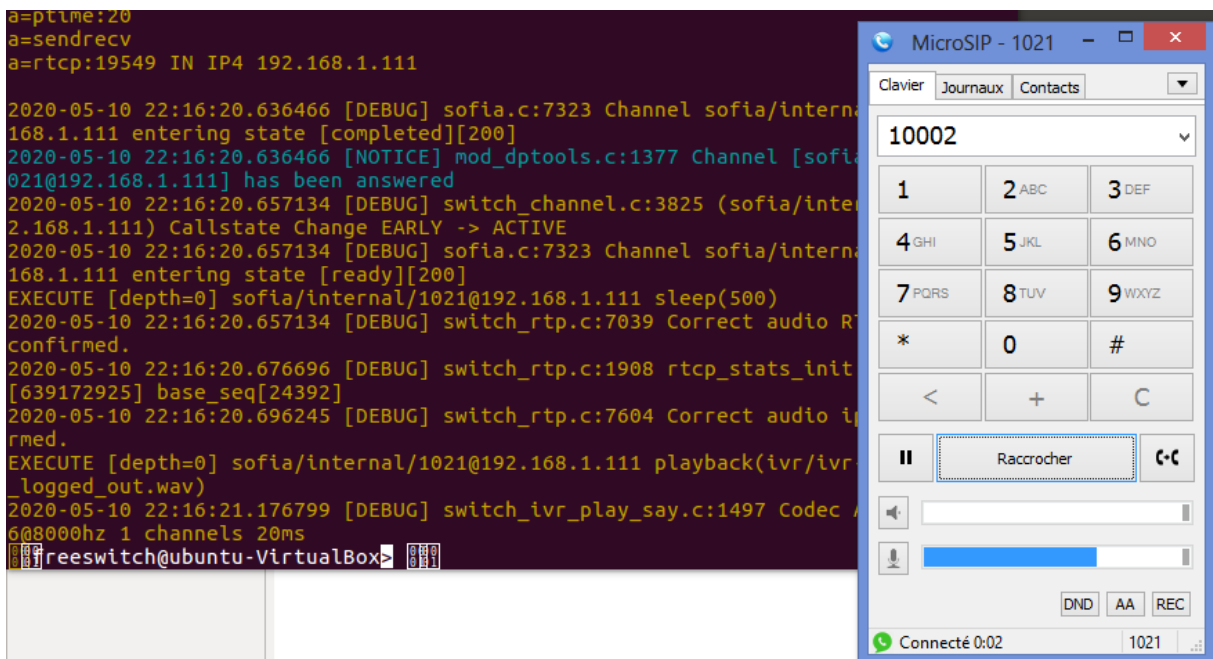
Pour cela on va créer un fichier **agentlog.xml** dans le dossier **/usr/local/freeswitch/conf/dialplan/default/** et définir les extensions 10001 et 10002 comme suit :

```
<include>
<extension name="agent_login">
<condition field="destination_number" expression="^10001$">
<action application="set" data="res=${callcenter_config(agent set status ${caller_id_number}
'Available')}" />
<action application="answer" data=""/>
<action application="sleep" data="500"/>
<action application="playback" data="ivr/ivr-you_are_now_logged_in.wav"/>
<action application="hangup" data=""/>
</condition>
</extension>
<extension name="agent_logoff">
<condition field="destination_number" expression="^10002$">
<action application="set" data="res=${callcenter_config(agent set status ${caller_id_number}
'Logged Out')}" />
<action application="answer" data=""/>
<action application="sleep" data="500"/>
<action application="playback" data="ivr/ivr-you_are_now_logged_out.wav"/>
<action application="hangup" data=""/>
</condition>
</extension>
</include>
```

Maintenant si agent appelle le numéro 10001 le serveur lui dans la file et lui envoie un son « vous êtes maintenant connectés »



En ce moment l'utilisateur 1021 pourra recevoir les appels de la file d'attente. Ensuite pour se déconnecter il compose le numéro 10002 on lui dit que « vous êtes maintenant déconnecter »



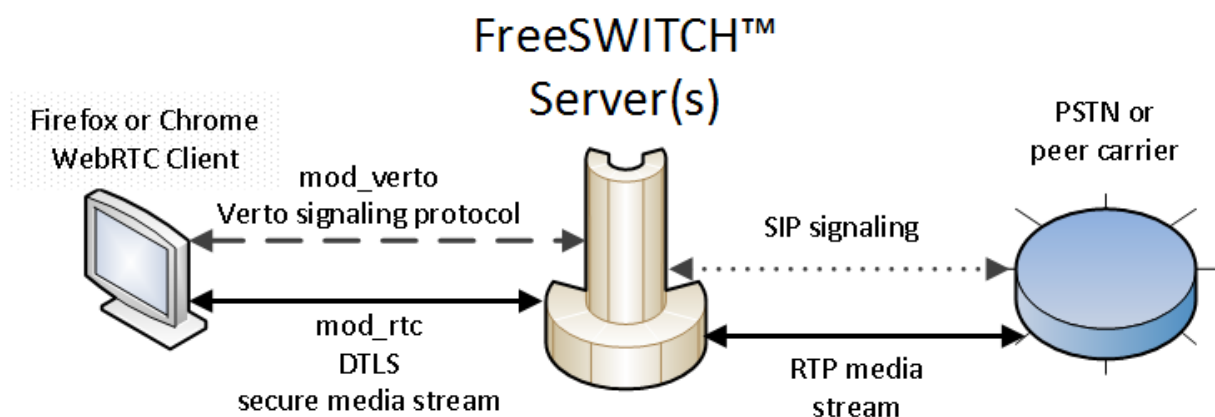
V Paramétrage et utilisation de Verto

1- Présentation de Verto

Verto (VER-to) RTC est un point de terminaison FreeSWITCH qui implémente un sous-ensemble d'une connexion JSON-RPC conçu pour être utilisé sur des sockets Web sécurisés. L'objectif initial est WebRTC pour simplifier le codage et la mise en œuvre des appels des navigateurs Web et des appareils vers FreeSWITCH. Cela permet à un navigateur Web ou à un autre client WebRTC de lancer un appel à l'aide de Verto dans une installation FreeSWITCH puis de le transférer vers le RTPC à l'aide de SIP, SS7 ou d'un autre protocole pris en charge. Cela déplace FreeSWITCH plus loin sur la pointe de la technologie de communication en temps réel tout en maintenant l'interopérabilité avec SIP et d'autres protocoles hérités

2- Architecture de Verto freeswitch

La figure ci-après montre que l'activation des 2 modules de freeswitch, que sont **mod_rtc** et **mod_verto**, permettant à un navigateur d'utilisateur le protocole de signalisation verto pour négocier les paramètres de session media et utilise ensuite le webRTC pour le transport des media. Entre FreeSWITCH et un transporteur SIP connecté au réseau des opérateurs, la signalisation verto est convertie par FreeSWITCH en SIP alors que les flux media sont convertis en RTP. Notons que le transfert des media se faisant de manière sécurisée entre deux entités webRTC, il est important d'utiliser une connexion https



3- Configuration de webrtc et de verto

On commence par activer la connexion sécurisée ssl aussi sur le profile interne qu'externe **/usr/local/freeswitch/conf/vars.xml** comme montre l'imace ci-dessous

```

<!-- Internal SIP Profile -->
<X-PRE-PROCESS cmd="set" data="internal_auth_calls=true"/>
<X-PRE-PROCESS cmd="set" data="internal_sip_port=5060"/>
<X-PRE-PROCESS cmd="set" data="internal_tls_port=5061"/>
<X-PRE-PROCESS cmd="set" data="internal_ssl_enable=true"/>

<!-- External SIP Profile -->
<X-PRE-PROCESS cmd="set" data="external_auth_calls=false"/>
<X-PRE-PROCESS cmd="set" data="external_sip_port=5080"/>
<X-PRE-PROCESS cmd="set" data="external_tls_port=5081"/>
<X-PRE-PROCESS cmd="set" data="external_ssl_enable=true"/>

```

Par défaut ces deux paramètres étaient à false on les a mis à **true**

On profite pour activer la langue française dans ce même fichier comme montre l'image ci-dessous

```

<X-PRE-PROCESS cmd="set" data="sound_prefix=${sounds_dir}/fr/ca/june"/>
<!-- <X-PRE-PROCESS cmd="set" data="sound_prefix=${sounds_dir}/en/us/callie"/> -->
<!--<Z-PRE-PROCESS cmd="set" data="sound_prefix=${sounds_dir}/en/us/allison"/> -->

```

On active verto sur les comptes des utilisateurs en ajoutant les lignes suivantes dans le fichier `/usr/local/freeswitch/conf/directory/default.xml`

```

<param name="jsonrpc-allowed-event-channels" value="demo,conference,presence"/>
<param name="verto-context" value="public"/>
<param name="verto-dialplan" value="XML"/>

```

```

<include>
<!-- the domain or ip (the right hand side of the @ in the addr-->
<domain name="${domain}">
  <params>
    <param name="dial-string" value="{^:sip_invite_domain=${dialed_domain};presence_id=${dialed_user}@dialed_user}@${dialed_domain}};${verto_contact(${dialed_user}@${dialed_domain})}"/>
    <!-- These are required for Verto to function properly -->
    <param name="jsonrpc-allowed-methods" value="verto"/>
    <param name="jsonrpc-allowed-event-channels" value="demo,conference,presence"/>
    <param name="verto-context" value="public"/>
    <param name="verto-dialplan" value="XML"/>
    <!-- <param name="jsonrpc-allowed-event-channels" value="demo,conference,presence"/> -->
  </params>
</domain>

```

Verto étant une interface web sécurisée, il convient d'installer un serveur web tel qu'apache et le sécuriser par ssl en installant le paquet **apache2** et activer le module ssl

```

root@ubuntu-VirtualBox:/usr/local/freeswitch/conf# apt-get install apache2
Lecture des listes de paquets... Fait
Construction de l'arbre des dépendances
Lecture des informations d'état... Fait
apache2 est déjà la version la plus récente (2.4.18-2ubuntu3.14).
Les paquets suivants ont été installés automatiquement et ne sont plus nécessaires :

```

Ensuite on installe les anciennes versions de **npm** avec les commandes suivantes

- ✓ **apt remove --purge nodejs npm**
- ✓ **apt clean**
- ✓ **apt autoclean**
- ✓ **apt install -f**
- ✓ **apt autoremove**

Ensuite on installe les prérequis de verto avec les commandes suivantes

- ✓ **apt install curl nodejs npm**
- ✓ **curl -sL https://deb.nodesource.com/setup_10.x | sudo -E bash -**

Maintenant on peut passer à l'installation proprement dite de verto communicator. On se déplace dans le dossier **/usr/src/freeswitch-1.8.6/html5/verto/verto_communicator** et installe verto avec les commandes suivantes

- ✓ **npm install -g grunt grunt-cli bower**
- ✓ **npm install**
- ✓ **bower --allow-root install**
- ✓ **grunt build --force**

Après installation, dossier **dist** sera créé et on copie ce dossier dans le documentroot (**/var/www/html**) du serveur apache en le nommant **vc**

```
root@ubuntu-VirtualBox:/usr/src/freeswitch-1.8.6/html5/verto/verto_communicator#
ls
bower_components  dist          node_modules   README.md
bower.json        Gruntfile.js package.json    src
debian8-install.sh js            package-lock.json
root@ubuntu-VirtualBox:/usr/src/freeswitch-1.8.6/html5/verto/verto_communicator#
cp -r dist/ /var/www/html/vc
root@ubuntu-VirtualBox:/usr/src/freeswitch-1.8.6/html5/verto/verto_communicator#
```

Ensuite on active le module **ssl** de apache avec la commande **a2enmod ssl** ensuite redémarrer le serveur apache pour passer à la configuration de du site ssl par défaut

```
root@ubuntu-VirtualBox:/etc/apache2/mods-enabled# a2enmod ssl
Considering dependency setenvif for ssl:
Module setenvif already enabled
Considering dependency mime for ssl:
Module mime already enabled
Considering dependency socache_shmcb for ssl:
Enabling module socache_shmcb.
Enabling module ssl.
See /usr/share/doc/apache2/README.Debian.gz on how to configure SSL and create self-signed certificates.
To activate the new configuration, you need to run:
  service apache2 restart
root@ubuntu-VirtualBox:/etc/apache2/mods-enabled# service apache2 restart
```

Maintenant on peut configurer le site ssl par défaut dans le fichier **/etc/apache2/sites-available/default-ssl.conf** en ajoutant les deux lignes suivantes pour utiliser le certificat de FreeSWITCH

SSLCertificateFile /usr/local/freeswitch/certs/wss.pem
SSLCertificateKeyFile /usr/local/freeswitch/certs/wss.pem

```
<VirtualHost _default_:443>
    ServerAdmin webmaster@localhost
    DocumentRoot /var/www/html/vc
    SSLCertificateFile /usr/local/freeswitch/certs/wss.pem
    SSLCertificateKeyFile /usr/local/freeswitch/certs/wss.pem

    # Available loglevels: trace8, ..., trace1, debug, info, notice,
warn,
```

Après il faut désactiver les certificats par défaut en mettant les deux lignes suivantes en commentaire

SSLCertificateFile SSLCertificateKeyFile

Ensuite on active le site par la commande `a2ensite default-ssl.conf` et recharger le serveur apache

```
root@ubuntu-VirtualBox:/etc/apache2/sites-available# a2ensite default-ssl.conf
Enabling site default-ssl.
To activate the new configuration, you need to run:
  service apache2 reload
root@ubuntu-VirtualBox:/etc/apache2/sites-available# service apache2 reload
```

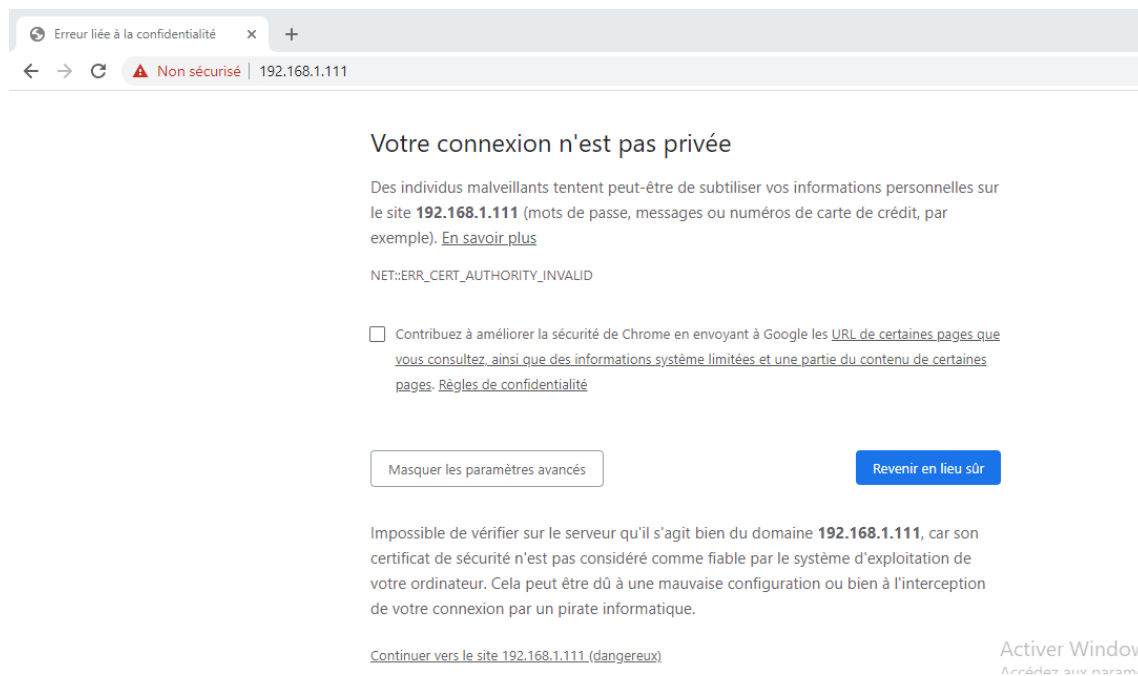
Maintenant Il nous faut redémarrer freeswitch, on affiche le numéro de processus actuel de freeswitch et on arrête ce processus par la commande `kill -9`

```
root@ubuntu-VirtualBox:/etc/apache2/sites-available# netstat -anp | grep -w 5060
tcp        0      0 192.168.1.111:5060  0.0.0.0:*          LISTEN
2665/freeswitch
tcp6       0      0 :::5060             :::*              LISTEN
2665/freeswitch
udp        0      0 192.168.1.111:5060  0.0.0.0:*          LISTEN
2665/freeswitch
udp6       0      0 :::5060             :::*              LISTEN
2665/freeswitch
root@ubuntu-VirtualBox:/etc/apache2/sites-available# kill -9 2665
```

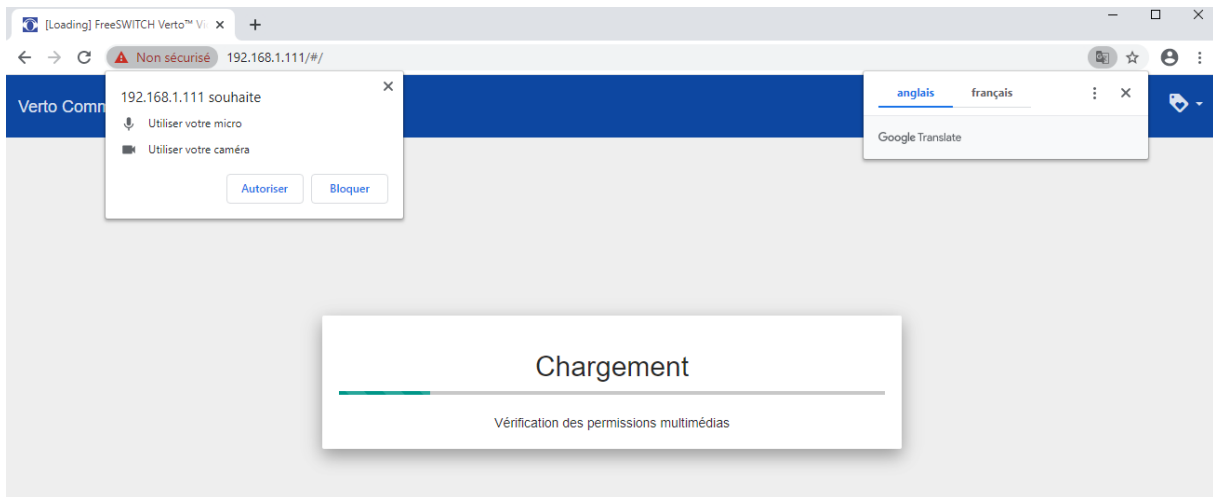
Ensuite on peut redémarrer FreeSWITCH, allez sur un navigateur pour tester la connexion

4- Test de connexion avec Google chrome

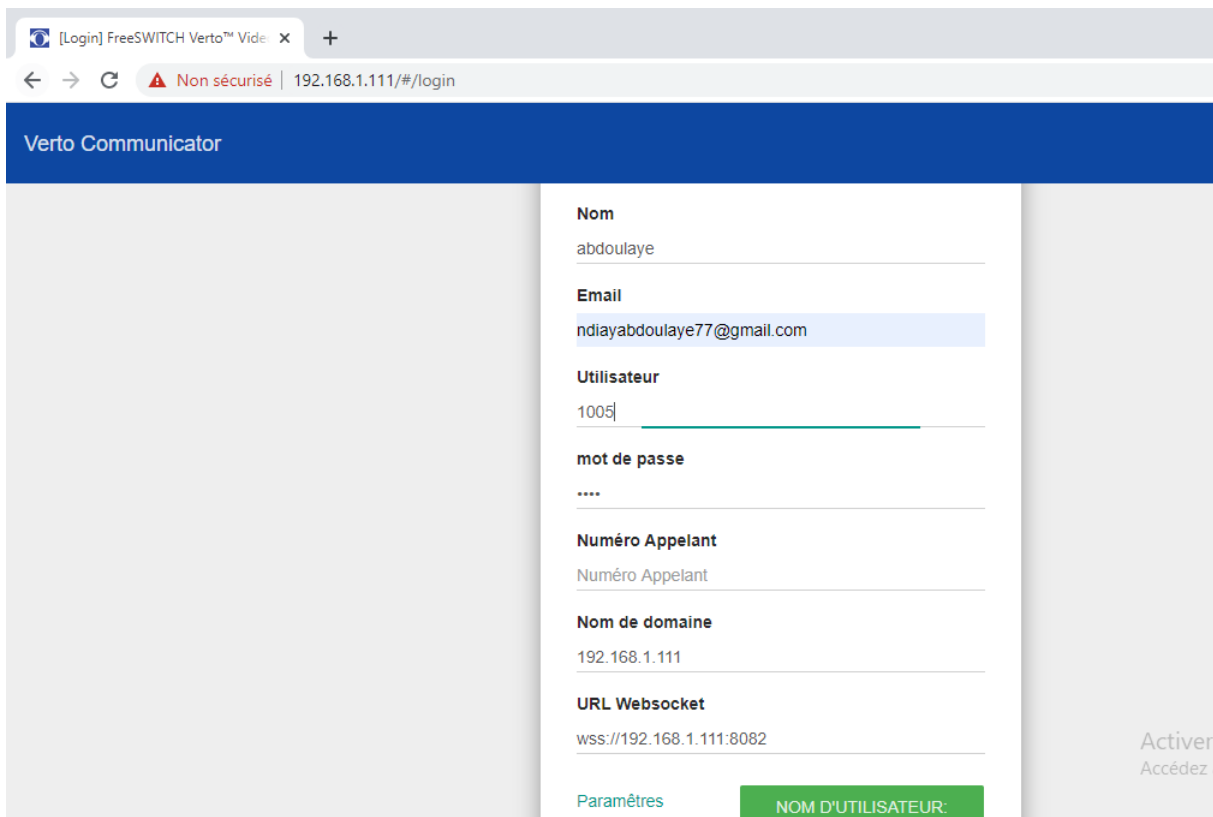
Une fois dans le navigateur on tape <https://192.168.1.11> et on voit qu'on a comme montre l'image ci-dessous pour nous dire que la connexion dangereuse



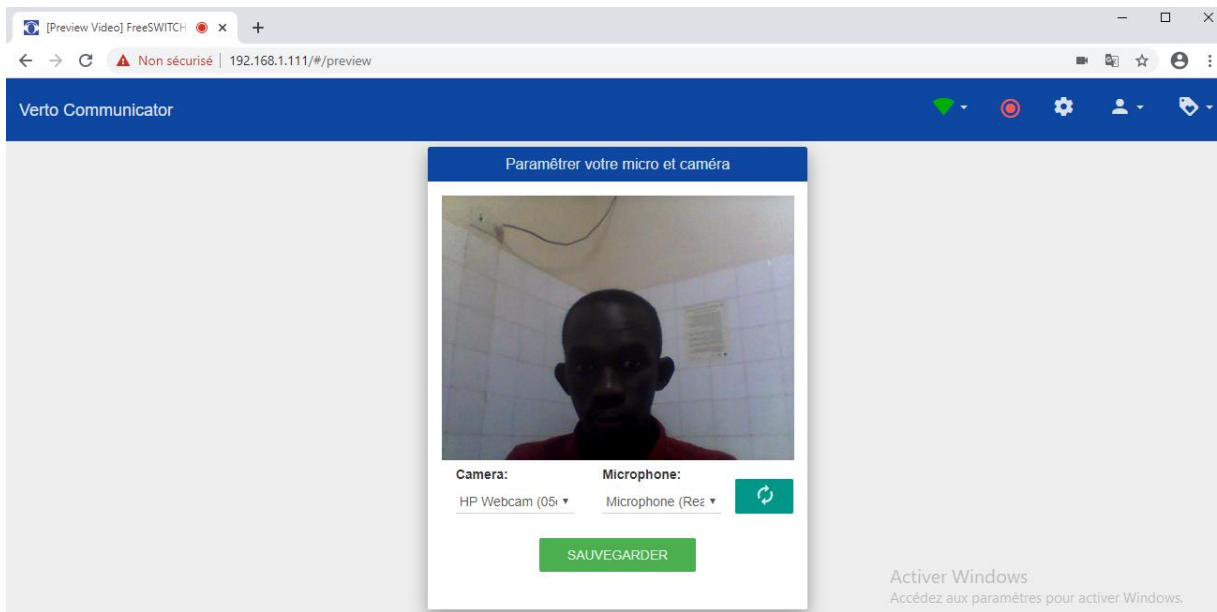
C'est tout simplement que le navigateur ne reconnait pas le certificat, on clique sur continué et on peut voir qu'on est sur l'interface de verto



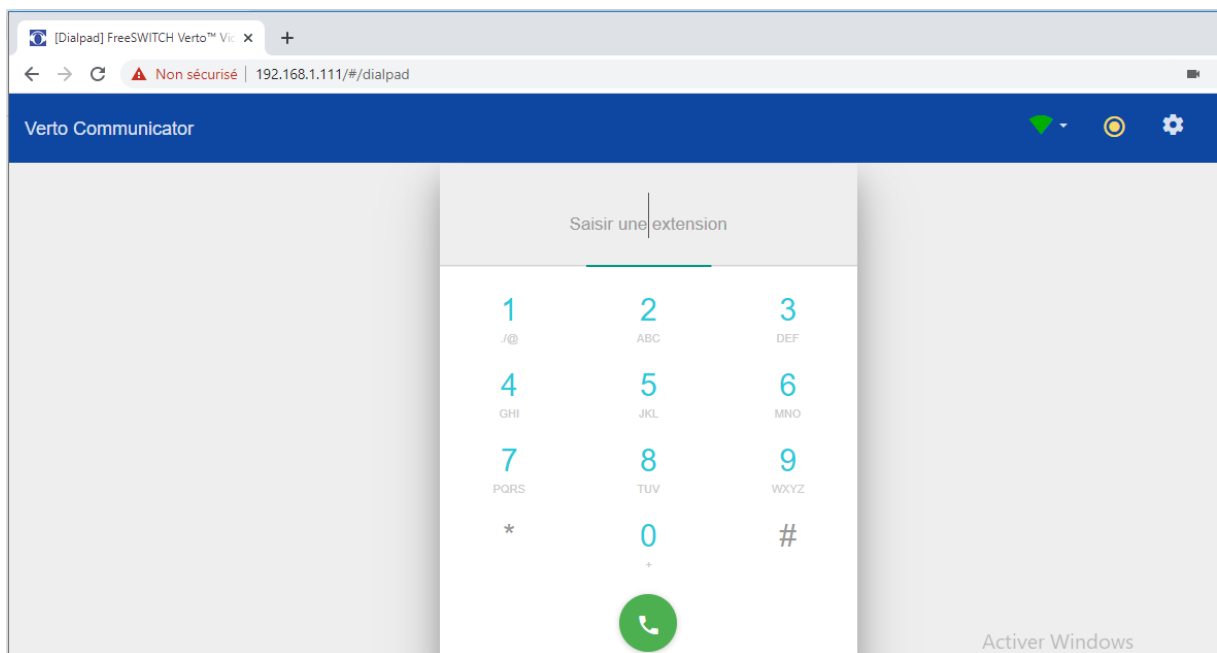
Ensuite on remplit le nom et l'email et on change les paramètres de l'utilisateur pour choisir un autre numéro et on se connecte



Une fois que l'utilisateur s'est connecté on lui demande de choisir son micro et son camera



Ensuite on sauvegarde et on voit qu'on s'est bien connecté et on peut commencer et appeler



Au moment de la connexion on peut voir sur l'interface de freeswitch qu'un utilisateur s'est connecté avec l'adresse 192.168.1.63

```

2020-05-01 20:16:32.224189 [DEBUG] mod_verto.c:4262 192.168.1.63:49859 Client Connect from 192.168.1.63:49859 accepted
2020-05-01 20:16:32.224189 [DEBUG] mod_verto.c:2015 192.168.1.63:49859 Starting client thread.
2020-05-01 20:16:32.542584 [DEBUG] mod_verto.c:1308 192.168.1.63:49859 re-connecting session 657e146b-4911-91ca-7b09-19371d4a73a6
2020-05-01 20:16:32.562584 [DEBUG] mod_verto.c:1057 auth using username & password
  
```

Partie 5 : Interconnexion de asterisk et FreeSWITCH

Puisque asterisk FreeSWITCH écoutent sur le même port qui 5060 nous allons changer celle de FreeSWITCH pour que les deux serveurs tournent en même temps

Pour cela on va dans le fichier `/usr/local/freeswitch/conf/vars.xml` mettre le port de FreeSWITCH à 5070 comme montre l'image ci-dessous

```
<!-- Internal SIP Profile -->
<X-PRE-PROCESS cmd="set" data="internal_auth_calls=true"/>
<X-PRE-PROCESS cmd="set" data="internal_sip_port=5070"/>
<X-PRE-PROCESS cmd="set" data="internal_tls_port=5071"/>
<X-PRE-PROCESS cmd="set" data="internal_ssl_enable=true"/>
```

Ensuite redémarre freeswitch et on peut qu'il écoute sur le port 5070

```
root@ubuntu-VirtualBox:/usr/local/freeswitch# ./bin/freeswitch -nc
root@ubuntu-VirtualBox:/usr/local/freeswitch# 5330 Backgrounding.

root@ubuntu-VirtualBox:/usr/local/freeswitch# netstat -anp | grep -w 5070
tcp        0      0 192.168.1.111:5070      0.0.0.0:*                LISTEN
5330/freeswitch
tcp6       0      0 :::1:5070                :::*                       LISTEN
5330/freeswitch
udp        0      0 192.168.1.111:5070      0.0.0.0:*
5330/freeswitch
udp6       0      0 :::1:5070                :::*
5330/freeswitch
```

Maintenant pour s'enregistrer, il faut ajouter derrière les adresses du serveur:5070 au niveau des téléphones SIP.

1- Gestion des appels entrants d'asterisk vers FreeSWITCH

On définit d'abord le plan de numérotation des appels entrants dans le fichier `/usr/local/freeswitch/conf/dialplan/public/00_inbound_did.xml` comme montre l'image ci-dessous

```
<extension name="public_did">
  <condition field="destination_number" expression="^(1021)$">
    <!--
      If you're hosting multiple domains you will want to set the
      target_domain on these calls so they hit the proper domain after you
      transfer the caller into the default context.

      ${domain} is the default domain set from vars.xml but you can set it
      to any domain you have setup in your user directory.
    -->
    <action application="set" data="domain_name=${domain}"/>
    <!-- This example maps the DID 5551212 to ring 1000 in the default context
  -->
    <action application="transfer" data="1021 XML default"/>
  </condition>
```

Ensuite on va dans `/etc/asterisk/extensions.conf` de asterisk pour définir le numéro vers laquelle on peut joindre l'utilisateur 1021 de FreeSWITCH avec la ligne suivante

exten => _01XXX,1,Dial(SIP/\${EXTEN:1}@freeswitch,40,tr)

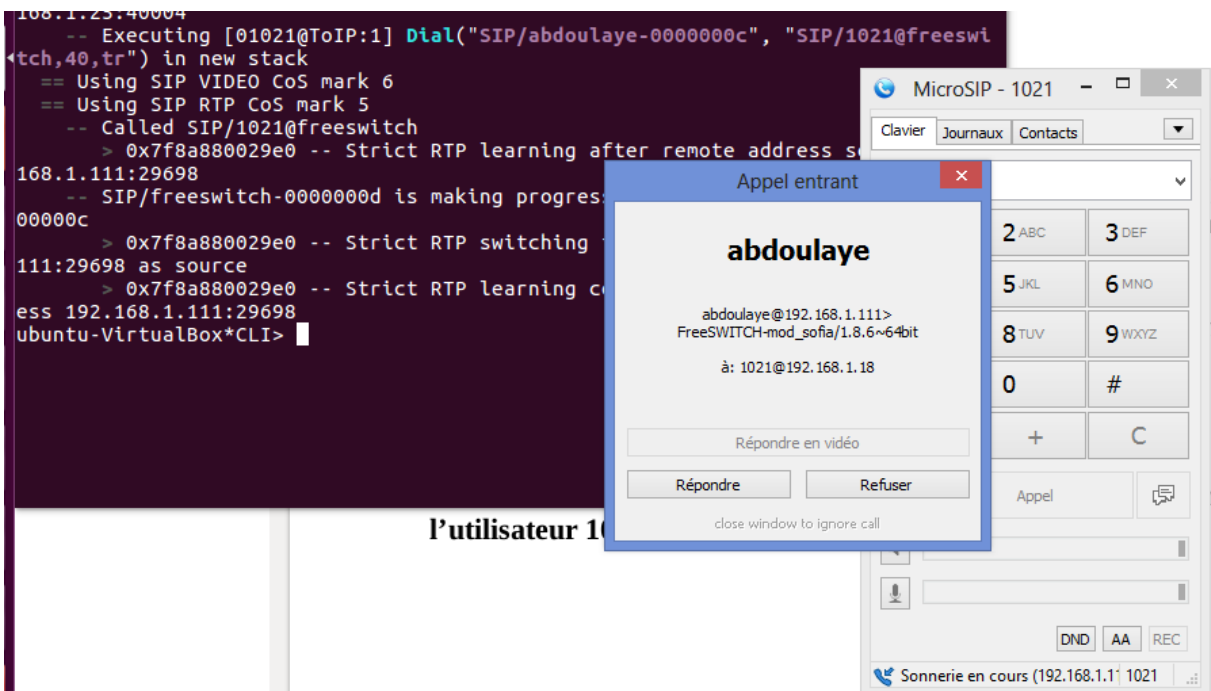
Cette extension veut dire tous numéro commençant par 01 suivi de 3 autres chiffres envoie l'appel vers le compte FreeSWITCH. Elle doit être définie dans le contexte de nos utilisateurs comme montre l'image ci-dessous

```
[ToIP]
exten => _01XXX,1,Dial(SIP/${EXTEN:1}@freeswitch,40,tr)
exten =>1000,1,Dial(SIP/abdoulaye,40,trim(webradio))
exten =>1000,2,Voicemail(1000@context_bv)
exten =>1000,3,hangup
```

Ensuite on crée un compte à FreeSWITCH dans le fichier /etc/asterisk/sip.conf

```
[freeswitch]
defaultuser=freeswitch
type=friend
secret=passer
host=dynamic
context=ToIP
allow=h263,h264,vp8
```

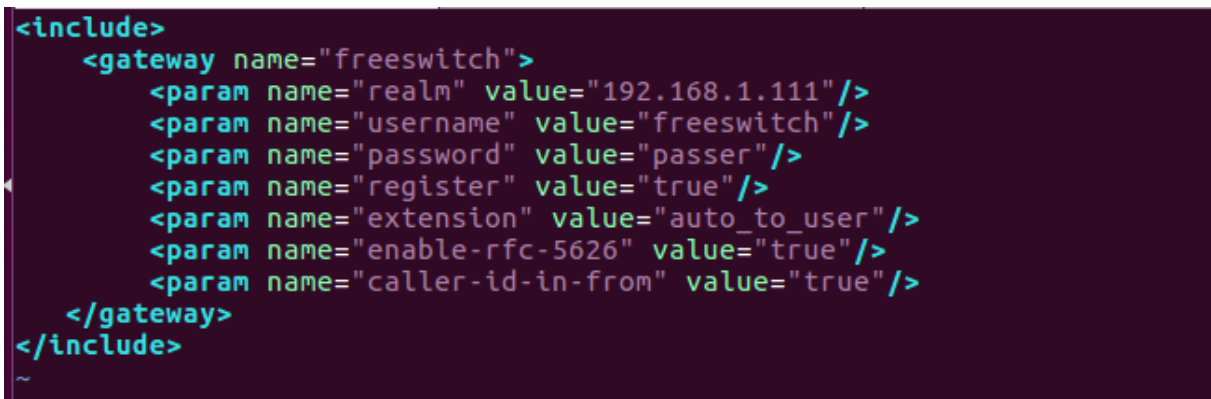
Maintenant on peut aller sur l'interface de **asterisk** recharger le serveur et sur le téléphone de Abdoulaye, on compose 01021 alors asterisk enlève le 0 et envoie l'appel vers FreeSWITCH sur le numéro 1021 comme montre l'image suivante



2- Gestion des appels sortants de FreeSWITCH vers asterisk

Tous d'abord on dans le dossier `/usr/local/freeswitch/conf/sip_profiles/external/` on crée un fichier `freeswitch.conf`. Dans ce fichier on met les instructions suivantes en changeant l'adresse par votre adresse IP

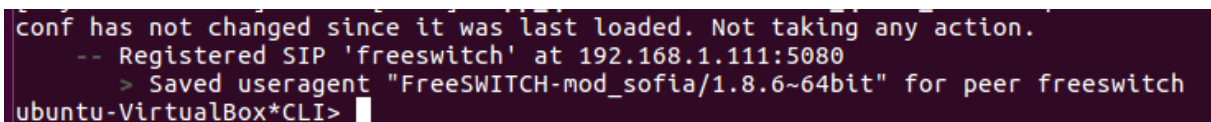
```
<include>
<gateway name="freeswitch">
<param name="realm" value="192.168.1.111"/>
<param name="username" value="freeswitch"/>
<param name="password" value="passer"/>
<param name="register" value="true"/>
<param name="extension" value="auto_to_user"/>
<param name="enable-rfc-5626" value="true"/>
<param name="caller-id-in-from" value="true"/>
</gateway>
</include>
```



```
<include>
  <gateway name="freeswitch">
    <param name="realm" value="192.168.1.111"/>
    <param name="username" value="freeswitch"/>
    <param name="password" value="passer"/>
    <param name="register" value="true"/>
    <param name="extension" value="auto_to_user"/>
    <param name="enable-rfc-5626" value="true"/>
    <param name="caller-id-in-from" value="true"/>
  </gateway>
</include>
```

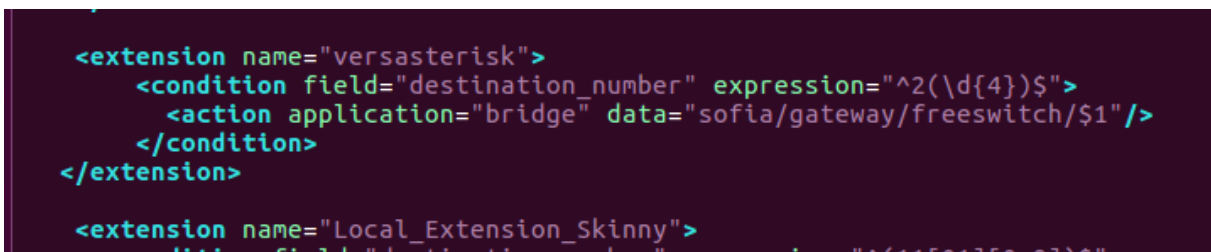
Puis sur l'interface de FreeSWITCH, on recharge le module sofia qui gere sip pour que FreeSWITCH se connecte à asterisk par la commande `reload mod_sofia`

Automatiquement on voit sur la console de asterisk que freeswitch s'enregistre sur asterisk



```
conf has not changed since it was last loaded. Not taking any action.
-- Registered SIP 'freeswitch' at 192.168.1.111:5080
> Saved useragent "FreeSWITCH-mod_sofia/1.8.6~64bit" for peer freeswitch
ubuntu-VirtualBox*CLI>
```

Maintenant on définit un plan de numérotation sur freeswitch pour dire comment acheminer les appels vers asterisk dans le fichier `/usr/local/freeswitch/conf/dialplan/default.xml` juste avant l'extension `Local_Extension_Skinny` comme montre l'image ci-dessous



```
<extension name="versasterisk">
  <condition field="destination_number" expression="^2(\d{4})$">
    <action application="bridge" data="sofia/gateway/freeswitch/$1"/>
  </condition>
</extension>

<extension name="Local_Extension_Skinny">
  <condition field="destination_number" expression="^([1][01][0-9])$">
```

Pour dire que tout numéro de 5 chiffres commençant par 2 doit être envoyé vers la passerelle FreeSWITCH qui correspond à asterisk

Maintenant on peut tester avec l'utilisateur 1021 de FreeSWITCH compose le prefix 2+1000 et entre en communication avec l'utilisateur abdoulaye de d'asterisk

The image shows a terminal window on the left and a MicroSIP client window on the right. The terminal output is as follows:

```
168.1.111:26688
-- Executing [1000@ToIP:1] Dial("SIP/freeswitch-0000000e", "SIP/abdoulaye,40
,term(webradio)") in new stack
== Using SIP VIDEO CoS mark 6
== Using SIP RTP CoS mark 5
-- Called SIP/abdoulaye
-- Started music on hold, class 'webradio', on chann
00e'
> 0x7f8a7c062100 -- Strict RTP switching to RTP t
111:26688 as source
-- SIP/abdoulaye-0000000f is ringing
-- SIP/abdoulaye-0000000f is ringing
> 0x7f8a7c062100 -- Strict RTP learning complete
ess 192.168.1.111:26688
ubuntu-VirtualBox*CLI>
```

The MicroSIP window is titled "MicroSIP - 1021" and shows a dial pad with the number "21000" entered. The dial pad includes buttons for digits 1-9, *, 0, #, and function keys like "Raccrocher" (Hangup), "DND", "AA", and "REC". The status bar at the bottom indicates "Session en cours (192.168.1.111 1021)".

l'utilisateur 1000 d'as