

– TP-Projet 1 –

Serveur ToIP Asterisk

Introduction

Nous allons passer plusieurs séances de TP au montage d'un réseau téléphonique privé full-IP, incluant un serveur ToIP, des hardphones et softphones IP... Contrairement au déploiement d'une solution propriétaire « clé en main » (Aastra, Alcatel, Siemens...), nous allons aujourd'hui déployer une **solution** totalement **libre** : le **serveur de téléphonie ToIP Asterisk**, à monter sur une base d'**OS Linux**.

Ce TP-projet pourra notamment nous permettre :

- d'installer **Asterisk** en maitrisant les différentes étapes de construction de l'exécutable
- de mettre en service des **hardphones IP** et **softphones IP**
- d'effectuer une **analyse des trames** générées par le serveur et les postes
- d'installer des **serveurs DHCP et TFTP pour la mise à jour et la configuration automatique des postes**
- de mettre en place des **appels visio**, de développer des services (groupements de postes, messagerie, renvois, serveur vocal, visioconférences,...)
- de connecter le réseau privé à un **opérateur ITSP** (Internet Telephony Services Provider) par **trunk SIP**
- d'installer une solution de **messagerie unifiée** (avec serveurs SMTP-IMAP)
- de mettre en service une **base de données** pour le stockage des données, un serveur de **journalisation (listing)** et **taxation des appels...**

Ce qu'il resterait à faire pour avoir une solution de téléphonie privée « complète » serait :

- de gérer la **QoS** en mettant en place des **VLAN data et voix séparés** sur les switches
- de mettre en place un cluster haute disponibilité (redondance des serveurs) pour la continuité de services en cas de problème matériel sur le serveur
- de déployer une solution de **supervision des équipements et serveurs**, ...

Bref, l'installation d'un réseau téléphonique privé full-IP est sans limite, et permet de parcourir l'ensemble des domaines de compétences des réseaux informatiques.

Afin que le réseau que vous allez monter et gérer soit de taille conséquente, nous allons avoir recours à la solution de virtualisation VirtualBox. Vos machines appartiendront à un sous-réseau propre, et les différents réseaux seront interconnectés entre eux en utilisant la structure switches-routeur de la salle de TP.

NB : vous aurez un **compte-rendu de TP à déposer sur Moodle**, maximum 7 jours après la dernière séance de ce projet. **Ce compte-rendu de projet devra décrire efficacement les différentes étapes que vous aurez effectuées** pour remplir le cahier des charges demandé ; **votre compte-rendu doit permettre à un novice d'effectuer et comprendre ce que vous avez réalisé**. Toutes les **captures d'écran** explicites seront les bienvenues.

I. Configuration des paramètres réseau

Nous allons placer chacun de vos postes client (Dell T1700), dans un sous-réseau indépendant ; chaque poste de travail étant placé dans un VLAN différent, vous serez mieux isolé et protégé des éventuelles erreurs de config de serveurs des autres postes de travail.

- Copier en local (bureau) l'énoncé du TP-projet car l'accès au serveur sera momentanément indisponible.
- Connecter votre PC sur le switch 2 ou 3 conformément au tableau suivant :

N° poste de travail	Sous-réseau	Switch associé à votre poste de travail	N° ports associés à votre poste de travail
1	192.168.11.0 / [°] 24	SW 2	1 à 6
2	192.168.12.0 / [°] 24	SW 2	7 à 12
3	192.168.13.0 / [°] 24	SW 2	13 à 18
4	192.168.14.0 / [°] 24	SW 2	19 à 24
5	192.168.15.0 / [°] 24	SW 3	1 à 6
6	192.168.16.0 / [°] 24	SW 3	7 à 12
7	192.168.17.0 / [°] 24	SW 3	13 à 18

- Configurez votre carte réseau en statique avec les paramètres suivants : 192.168.1*i*.20*i* avec *i* correspondant à votre n° de poste ; préfixe /24 et passerelle = 192.168.1*i*.254 ; DNS : 192.168.1.1 et 10.2.40.230
*Attention ! : Si Windows vous demande de choisir le type de réseau, bien spécifier que c'est un réseau sûr de type « **réseau de bureau** » (ou « **réseau domestique** »), afin qu'il ne sorte pas la grosse artillerie pour son pare-feu, ce qui aurait alors pour conséquence de rendre difficile le dialogue avec d'autres machines par la suite...*

II. VM utilisées pour le projet

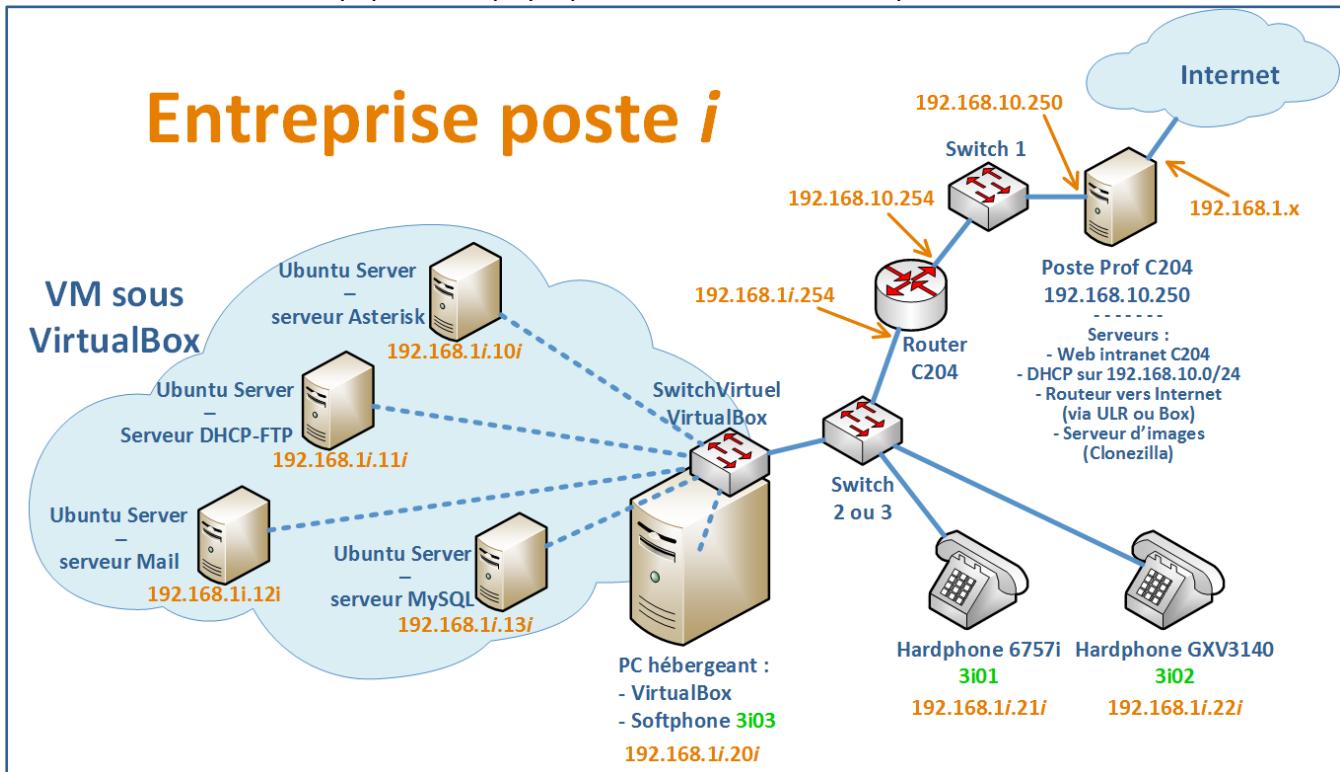
Nous allons créer un réseau d'entreprise composé des équipements suivants :

- 1 VM sous OS Linux embarquant un serveur de téléphonie Asterisk, qui permettra d'administrer le réseau téléphonique privé de l'entreprise
- 1 hardphone SIP Mitel-Aastra 6757i
- 1 hardphone visio SIP Grandstream GXV 3140
- 1 softphone sur votre PC physique Dell Precision T1700
- d'autres VM sous OS Linux hébergeant divers services selon les besoins (SMTP, IMAP, MySQL, DHCP, TFTP...).

La connexion entre les équipements virtuels et les équipements physiques se fera à l'aide d'un **switch virtuel** ; pour cela, il faudra configurer la **connexion des VM en mode « pont » (bridge)**. Les machines virtuelles et les équipements physiques seront ainsi sur le même sous-réseau, les VMs auront donc la même passerelle et le(s) même(s) DNS que la machine physique.

II.1. Schéma logique du réseau d'entreprise

L'entreprise de votre poste de travail *i* appartient au réseau **192.168.1*i*.0 / 24**. Les différentes @ IP à affecter aux équipements physiques ou virtuels sont indiquées ici :



Nous allons utiliser au total 4 VMs :

- VM « **M4205-Asterisk-Server** » : hébergera le serveur ToIP Asterisk, ainsi que quelques services complémentaires (serveur de temps NTP, ...) ; c'est la VM principale de notre projet, le cœur de notre réseau téléphonique
- VM « **M4205-DHCP-FTP-Server** » : hébergera un serveur DHCP pour votre sous-réseau 192.168.1*i*.0/24, ainsi qu'un serveur FTP pour que les téléphones IP puissent uploader les fichiers nécessaires à leur configuration
- VM « **M4205-Mail-Server** » : hébergera un serveur SMTP et IMAP permettant d'offrir un service de messagerie électronique au serveur ToIP et aux abonnés
- VM « **M4205-MySQL-Server** » : sera exploitée à la fin de ce projet pour centraliser toutes les données utiles à notre réseau téléphonique privé dans une base de données.

II.2. Plate-forme de virtualisation VirtualBox

- Avant tout, vérifier que vous êtes bien logés sous le compte Windows « **RT2-20i** » correspondant à votre groupe de TP (password : « **20i** »).
- Lancer VirtualBox. Vous devez avoir les 4 VM listées précédemment :
- Vérifier que toutes vos VMs ont leur configuration réseau en mode « **pont** » (bridge) afin qu'elles puissent faire partie du même sous-réseau 192.168.1*i*.0/24 que votre PC et les autres équipements physiques que nous connecterons ultérieurement.

III. Configuration du serveur ToIP Asterisk

Nous allons commencer par effectuer une configuration basique de la machine devant héberger le serveur ToIP Asterisk.

III.1. Configuration des paramètres réseau

- Démarrer la VM **M4205-Asterisk-Server**, s'identifier avec « **user/rtrt** ».
- Passer en mode administrateur :
`sudo -s`

Nous allons configurer la carte réseau virtuelle de façon permanente, afin de ne pas avoir à la re-paramétrer à chaque démarrage (à la volée...) : pour configurer manuellement la carte réseau, effectuer les manipulations suivantes :

- Consulter la configuration réseau avec « **ifconfig** ». Quelles sont interfaces actives ? Si l'interface virtuelle eth0 n'apparaît pas, activer eth0 avec l'instruction suivante, puis réafficher la config réseau de la VM :
`ifconfig eth0 up`
(si besoin, exécuter `/etc/init.d/networking restart`)

- Editer le fichier de config « **/etc/network/interfaces** » à l'aide de l'utilitaire « **nano** », et rajouter les lignes suivantes, avec les paramètres IPv4 adéquats, conformément à la topologie logique (cf schéma) :

```
...
auto eth0
iface eth0 inet static
    address adresse_IPv4_de_votre_VM_Asterisk-Server
    netmask 255.255.255.0
    gateway adresse_IPv4_de_votre_gateway
    dns-nameservers 192.168.1.1 10.2.40.230
```

- Utiliser les commandes suivantes pour redémarrer l'interface afin qu'elle prenne en compte les nouveaux paramètres définis dans `/etc/network/interfaces` :

```
ifdown eth0
ifup eth0
```

- Consulter à nouveau la configuration réseau ainsi que la table de routage IP pour vérifier que le paramétrage réseau est correct :

```
ifconfig
route -n
```

- Effectuer des tests de ping vers :

- votre machine physique, votre gateway,
- une autre machine (physique ou VM) de la salle,
- un site web sur internet

*NB : Normalement, tout doit être fonctionnel... (penser à vérifier l'activation de la **découverte réseau** sur l'OS Windows de votre PC fixe...)*

III.2. Installation du serveur Asterisk

Nous allons maintenant installer le serveur ToIP Asterisk en personnalisant notre installation. Pour ce faire, nous allons télécharger les paquets nécessaires, sélectionner les options à installer souhaitées, compiler et créer l'exécutable pour l'installation d'Asterisk.

- Mettre à jour la base de connaissance de paquets existants pour Ubuntu server :
`apt-get update`
- Installer les paquets suivants :
`apt-get install libsrtplib0-dev srtp-utils libxml2-dev openssl libssl-dev g++`
`apt-get install build-essential libncurses5-dev uuid-dev libjansson-dev`
`apt-get install sqlite3 libsqlite3-dev`
- Télécharger l'archive suivante :
`wget http://downloads.asterisk.org/pub/telephony/asterisk/asterisk-13-current.tar.gz`
- Visualiser dans quel dossier vous êtes actuellement positionné et lister son contenu pour vérifier que l'archive téléchargée soit bien là. 
- Extraire l'archive grâce à la commande ci-dessous puis lister à nouveau le contenu du dossier pour connaître le nom de l'archive récupérée (et donc le n° de la version...) :
`tar zxvf asterisk-13-current.tar.gz`
- Configurer, compiler et installer Asterisk :

<code>cd asterisk-13.xxxx/</code>	(Remplacer xxxx par votre n° de version)
<code>./configure</code>	(Ne passer à la suite que si aucun warning n'apparaît et si vous voyez le gros astérisque *)
<code>make menuconfig</code>	(ou <code>make menuselect</code>)

Vérifier que :

 - dans « Resource Modules », « res_srtp » est activé
 - dans « Channel Drivers », le protocole SIP (« chan_sip ») est activé
 - dans « Core Sound Packages » et « Extras Sound Packages », les bibliothèques de musiques et sons préenregistrés en français sont activés pour les codecs WAV, ALAW, ULAWS, GSM, G729, G722 (ainsi que les musiques d'attente (Music On Hold) si vous voulez...)

Sortir avec Echap puis S pour sauvegarder)

<code>make</code>	(compilation)
<code>make install</code>	(installation)
<code>make samples</code>	(écriture de commentaires dans les fichiers de configuration pour présenter des exemples de config)
<code>make config</code>	(pour ajouter démarrage du démon Asterisk /etc/init.d/asterisk dans /etc/rc0.d/, et /etc/rc5.d/)

L'installation personnalisée du serveur Asterisk est maintenant terminée.

III.3. Démarrage du serveur Asterisk

Avec l'installation réalisée, le serveur de téléphonie Asterisk démarrera automatiquement à la mise sous tension de la machine hébergeant le serveur. Nous n'aurons donc pas à le démarrer, il sera déjà en service. Mais là, si nous ne voulons pas redémarrer la VM, nous devons le lancer manuellement ; et nous allons en profiter pour ouvrir une interface de commande pour superviser Asterisk :

- Depuis le prompt du serveur, exécuter la commande suivante pour à la fois démarrer le serveur Asterisk et passer en mode d'interface de commande **CLI (Command Line Interface)** d'Asterisk :

```
asterisk -vvvvc
```

NB1 : les « v » donnent le niveau de verbose = la quantité d'informations renvoyées par le serveur concernant son fonctionnement

NB2 : pour simplement passer en mode CLI d'Asterisk quand le serveur Asterisk est déjà actif, il faut utiliser la commande :

```
asterisk -vvvvr
```

NB3 : pour relancer le serveur Asterisk afin qu'il recharge ses fichiers de configuration (suite à une modif de config par exemple), entrer dans la fenêtre CLI la commande « reload ».

NB4 : pour afficher tous les processus contenant « asterisk », puis tuer un processus, entrer dans un terminal Linux :

```
ps aux | grep asterisk  
kill -9 n°_du_processus_à_tuer
```

- Visualiser les terminaux SIP enregistrés sur le serveur Asterisk en composant dans l'interface CLI (*NB : pour l'instant, aucun poste ne doit apparaître*) :

```
sip show peers
```

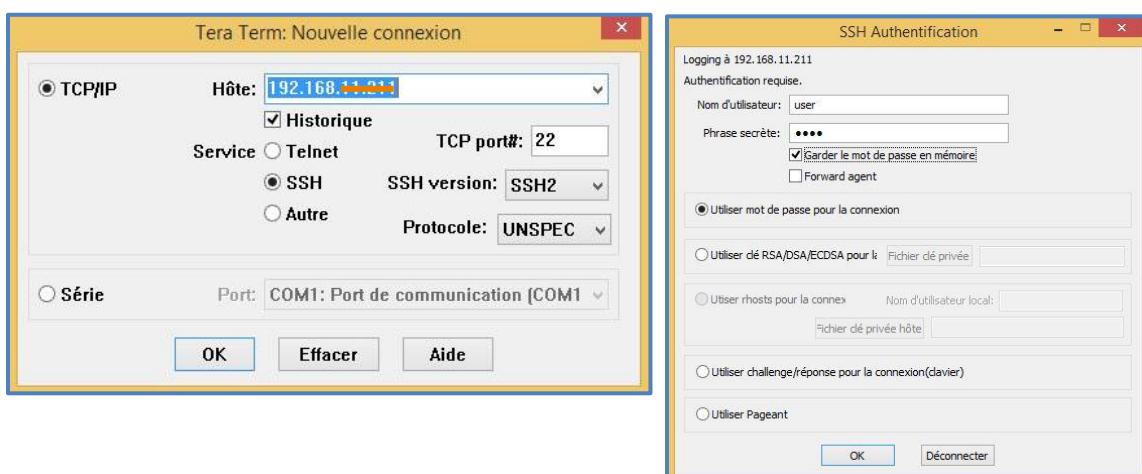
- Effectuer enfin un « Ctrl + C » pour sortir du CLI et arrêter Asterisk.

Il n'est actuellement pas possible, sur ce serveur Ubuntu, de pouvoir à la fois avoir accès en permanence à l'interface CLI d'Asterisk et à la fois à l'invite de commande root du serveur. (L'utilisation des Ctrl-Alt-F4 et Ctrl-Alt-F1 ne sera pas très pratique non plus pour tout visualiser en permanence...) Pour cela, il est plus pratique d'ouvrir plusieurs sessions distantes en liaison SSH. Nous allons donc installer un serveur (deamon) SSH :

- Installer un serveur SSH (paquet « openssh-server ») sur la VM Asterisk-Server :

```
apt-get install openssh-server
```

- Sur le PC physique (Windows 8.1), ouvrir Tera Term, et lancer une session cliente SSH vers la VM Asterisk-Server, en utilisant les identifiants du compte admin de la VM :



Vous êtes alors connecté à distance à la VM Asterisk-Server, en connexion sécurisée.

- Ouvrir en parallèle une 2^{ème} session cliente SSH vers la VM, passer en mode admin, puis lancer Asterisk.

Il va ainsi être très pratique d'avoir en permanence 2 terminaux qui ont la main sur :

- la machine Asterisk-Server (prompt user@asterisk-server ou root@asterisk-server)
- le serveur Asterisk (prompt *CLI>)

```

192.168.11.211:22 - user@asterisk-server: ~ VT
Fichier Edition Configuration Contrôle Fenêtre(W) Aide
Welcome to Ubuntu 14.04 LTS (GNU/Linux 3.13.0-24-generic x86_64)
* Documentation: https://help.ubuntu.com/
System information as of Fri May 9 21:50:18 CEST 2014
System load: 0.14 Processes: 218
Usage of /: 7.4% of 29.40GB Users logged in: 1
Memory usage: 11% IP address for eth0: 192.168.11.211
Swap usage: 0%
Graph this data and manage this system at:
https://landscape.canonical.com/
21 packages can be updated.
7 updates are security updates.

*** Le système doit être redémarré ***
Last login: Fri May 9 21:50:19 2014 from 192.168.11.201
user@asterisk-server: ~ $ 

```

Avant d'installer des téléphones IP, nous allons juste effectuer quelques premières modifications sur des fichiers de configuration :

- Editer le fichier « /etc/asterisk/extensions.conf » et chercher en dessous du contexte « [default] » la ligne :
include => demo
et la remplacer par :
include => plan-num-prive
- Toujours dans ce fichier « extensions.conf », ajouter les lignes de code suivantes **à la fin du fichier** pour configurer le numéro d'abonnement 3999 :
[plan-num-prive]
exten => 3999,1,Set(CHANNEL(language)=fr)
exten => 3999,n,BackGround(install-congrats)
- Télécharger le fichier install-congrats.gsm sur le serveur web intranet de la salle :
wget http://192.168.10.250/annexes/ressources_annexes/install-congrats.gsm
- Déplacer ce fichier depuis votre dossier actuel vers le dossier /var/lib/asterisk/sounds/fr/ (dossier à créer si besoin).

C'est tout pour le moment concernant la configuration du serveur Asterisk.

Maintenant, intéressons-nous à la mise en service de téléphones ToIP...

IV. Mise en service de clients ToIP

Nous allons utiliser dans notre réseau téléphonique privé des téléphones IP utilisant le **protocole de signalisation SIP** (**Session Initiation Protocol**) libre (gratuit et ouvert), utilisé pour signalisation d'appels, de décrochage et raccrochage...

IV.1. Mise en service du hardphone Aastra 6757i

Le hardphone 6757i (ou 57i) sait gérer le protocole SIP, il est donc compatible avec notre serveur Asterisk. Nous lui affecterons le **n° d'abonné « 3i01 »**. Sa couche logicielle étant orientée pour naturellement se connecter sur un IPBX Aastra, sa configuration sera un peu plus complexe que d'autres produits, mais nous y arriverons ;-)



- Alimenter le 6757i. Pendant le démarrage, appuyer sur skip lorsque cela vous est proposé pour accélérer le démarrage.
- Appuyer sur la touche Outils (clé) ; aller dans le menu « 5° Administrateur », rentrer le password « 22222 » ; aller dans le menu « 4 Reset usine » (ou « Factory default ») afin d'effectuer un reset en configuration usine du téléphone.
- Une fois le téléphone redémarré, retourner dans le menu « 5 Admin Menu », « 3 Network Settings », « 1 DHCP Settings », et désactiver le mode dynamique DHCP. Redémarrer à nouveau le téléphone.
- Une fois le téléphone redémarré, retourner dans « 5 Admin Menu », « 3 Network Settings », « 2 IP Address », et noter l'@ IP affectée au poste ; que pouvez-vous dire sur ce type d'@ IPv4 ? 

Sur la dernière version de firmware, il n'est pas possible de modifier l'@ IP directement depuis le poste (bogue ??) ; pour modifier les paramètres IP du 6757i, nous allons alors utiliser temporairement notre PC fixe, en le plaçant dans le même sous-réseau que le 6757i :

- Affecter temporairement à la carte Ethernet de votre PC une @ IPv4 fixe compatible avec l'@ actuelle du 6757i.
- Connecter le port LAN du téléphone directement sur le port RJ45 de la carte Ethernet du PC. Ouvrir un navigateur web, composer comme URL l'@ IPv4 du 6757i, et utiliser les identifiants « admin / 22222 » pour vous connecter sur le portail web embarqué du 6757i.
- Dans l'onglet « Network », régler les paramètres réseau suivants :
 - @ IP : *adresse_IP_de_votre_6757i*
 - masque : 255.255.255.0
 - passerelle : *adresse_IP_de_votre_gateway*
 - DNS primaire et secondaire : 192.168.1.1Cliquer sur « Save Settings » en bas de la page.
- Fermer le navigateur web, réaffecter au PC ses bons paramètres IP, puis reconnecter le PC et le 6757i chacun sur port du switch appartenant à votre VLAN.

- Dans un navigateur web, composer l'@ IP affectée à votre 6757i (identifiants inchangés). Dans le menu « **Ligne 1** » (profil de la ligne par défaut), préciser :
 - Nom d'écran : « poste 3i01 » (ou ce que vous voulez, ce n'est pas important...)
 - Phone Number, Caller ID et Authentication Name : « poste3i01 » (*c'est ce nom qui sera utilisé pour déclarer le poste dans Asterisk*)
 - Mot de passe : rtrt
 - Serveur proxy et Serveur registrar: @_IP_asterisk-server / port 5060
- Cliquer enfin en bas de la page sur « Enregistrement des paramètres »
- Redémarrer le téléphone.

Nous avons maintenant terminé la configuration du client, il nous reste maintenant à effectuer la configuration côté serveur.

Le poste SIP tente de s'enregistrer à intervalles réguliers. Pour déclarer son existence au serveur Asterisk, il faut modifier un des fichiers de configuration, « /etc/asterisk/sip.conf » :

- Ouvrir le fichier /etc/asterisk/sip.conf et ajouter à la fin les lignes suivantes :
;le « **contexte** » [poste3i01] va définir les paramètres de l'abonné poste3i01 :


```
[poste3i01]
type=friend
secret=rtrt
host=dynamic
context=plan-num-prive ; (default ?)
disallow=all
allow=alaw
allow=ulaw
allow=gsm
dtmfmode=inband ; et pas rfc2833 car sinon les caractères composés sur
                  ; le 6757i sont interprétés dupliqués sur Asterisk (ex :
                  ; 1234 peut devenir 11223344)
                  ; mais pour les autres postes, nous mettrons rfc2833
canreinvite = yes ; permet à 2 postes de communiquer directement
                    ; ensemble (flux RTP notamment), une fois l'appel lancé par le serveur
language=fr
```

- Identifier à quoi correspondent les différents paramètres indiqués ci-dessus.
- Redémarrer Asterisk afin que les modifications faites soient prises en compte.
- Redémarrer le 6757i afin qu'il tente de s'accrocher sur le serveur Asterisk (appui sur touche Outils (clé), puis menu « 6 Restart phone »).
- Vérifier avec l'instruction suivante que l'ajout du 6757i a été pris en compte : son @ IP doit maintenant apparaître dans le champ « Host » (au lieu de « Unspecified ») :


```
sip show peers
```
- Composer le « 3999 » sur le téléphone. Entendez-vous un message de bienvenue ?



Maintenant que **le poste SIP est accroché sur le serveur**, il ne reste qu'à configurer le **plan de numérotation** du serveur afin de définir le n° de l'abonné et le comportement à avoir lorsqu'est composé ce n° d'abonné :

- Editer « extensions.conf » et ajouter à la fin du fichier, après les 3 lignes de code suivantes, la ligne en gras, permettant de configurer le n° 3*i*01 pour l'abonné 6757i :


```
[plan-num-prive]
exten => 3999,1,Set(CHANNEL(language)=fr)
exten => 3999,n,BackGround(install-congrats)
exten => 3i01,1,Dial(SIP/poste3i01,10)
```
- Quel est le rôle et la syntaxe de cette instruction rajoutée ? 
- Recharger Asterisk puis tester le plan de numérotation depuis la console CLI en utilisant la commande « console dial » suivie du numéro du poste créé :


```
console dial 3i01
```

 Si vous avez tout bien configuré, le poste 6757i doit alors sonner.

IV.2. Mise en service du visiophone GXV3140

Ce visiophone sait également gérer le protocole SIP, il est donc compatible avec notre serveur Asterisk. En plus d'un téléphone classique, il faudra également nous intéresser, ultérieurement, aux codecs vidéo en plus des codecs audio si nous souhaitons exploiter les capacités vidéo de ce poste. Nous lui affecterons le **n° d'abonné « 3*i*02 »**.



- Alimenter et connecter le GXV3140 à un port du switch appartenant à votre VLAN.
- Aller dans « Menu > Settings > Maintenance > Upgrade » pour effectuer un reset usine (« Full Reset »).
- Une fois redémarré, affecter les paramètres réseau dans « Menu > Settings > Network > Connection » :
 - configuration IPv4 statique (non DHCP)
 - @ IP : *adresse_IP_de_votre_GXV3140* ; masque : 255.255.255.0 ; passerelle : *adresse_IP_de_votre_gateway*
- Redémarrer le téléphone (« Reboot » dans « Menu > Settings > Maintenance > Upgrade »), puis accéder depuis un navigateur au portail web embarqué dans le GXV3140 ; utiliser les identifiants « admin / admin ».
- Dans les onglets « Account1 » et « Account3 », désactiver les comptes 1 et 3 (ne pas oublier de sauver...)
- Dans l'onglet « Account2 », configurer le profil 2 comme ceci :
 - Account Name : poste3*i*02
 - SIP Server : *adresse_IP_asterisk_server*
 - SIP User ID : poste3*i*02

- Authenticate ID : poste3i02
- Authenticate Password : rtrt
- Voice Mail UserID : (rien)
- Name : Poste de l'abonné 3i02
- User ID is phone number : (non coché)

Effectuons maintenant côté serveur une configuration similaire à celle effectuée pour le téléphone précédent :

- Expliciter les changements que vous devez réaliser dans le fichier « sip.conf » afin d'inscrire le GXV3140 sur votre réseau téléphonique privé ; NB : le « dtmfmode » devra être configuré en « rfc2833 ».
Effectuer ces changements.
- Redémarrer Asterisk afin que les modifications faites soient prises en compte, puis redémarrer le GXV3140.
- Vérifier que l'ajout du téléphone a été pris en compte par le serveur Asterisk.
- Composer le « 3999 » sur le téléphone. Entendez-vous un message de bienvenue ?

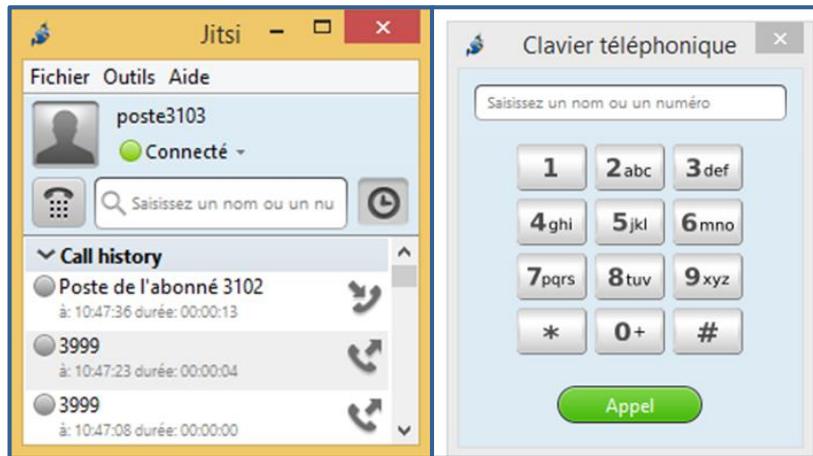
Il ne reste qu'à configurer le plan de numérotation du serveur afin de définir le n° d'abonné 3i02 au GXV3140 :

- En vous inspirant de ce qui a été fait pour le téléphone précédent, expliciter les changements que vous devez effectuer dans le fichier « extensions.conf » pour affecter le n° d'abonné « 3i02 » à ce poste.
- Recharger Asterisk puis tester le plan de numérotation depuis la console en utilisant la commande « console dial » suivie du numéro du poste créé :
`console dial 3i02`
Joignez-vous bien le GXV3140 ?
- Tester dans les 2 sens les appels entre les abonnés 3i01 et 3i02 pour vérifier que le fonctionnement soit correct.

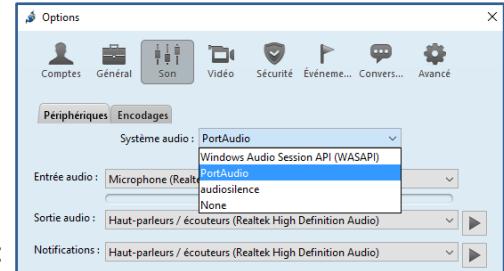
IV.3. Mise en service d'un softphone

Il est aussi possible de connecter sur le réseau téléphonique privé des téléphones logiciels, appelés **softphones**. Nous allons donc installer un softphone sur votre PC physique. Différents softphones existent, notamment **X-Lite** ou encore **Jitsi**. Ce dernier a l'avantage de permettre les communications vidéo, alors que X-Lite ne le propose pas dans sa version gratuite. Nous allons ainsi choisir d'installer le softphone gratuit **Jitsi**, projet communautaire LGPL de jitsi.org :

- Télécharger le logiciel **Jitsi** sur le site de Jitsi (jitsi.org) et l'installer sur votre PC physique (sous OS Windows 8.1)



- Pour configurer correctement Jitsi avec le n° d'abonné 3*i*03 : fermer la fenêtre « S'identifier », et dans la fenêtre principale, aller dans « Fichier > Ajouter un nouveau compte » et choisir le protocole Réseau « SIP » ; cliquer sur « Avancé », et remplir les champs suivants :
 - Identifiant SIP : poste3*i*03
 - Mot de passe : rrtt
 - Nom affiché : poste 3*i*03
 - Registrar : *adresse_IP_asterisk_server*
 - Port : 5060
 - Nom d'autorisation : poste3*i*03
- Configurer les paramètres audio comme ci-contre :



Déclarons maintenant le softphone sur le serveur :

- Expliciter les changements à réaliser dans « sip.conf » afin d'inscrire le softphone sur votre réseau téléphonique privé ; (« dtmfmode » configuré en « rfc2833 »). Effectuer ces changements.
- Redémarrer Asterisk puis redémarrer le softphone. Vérifier que l'ajout du softphone a été pris en compte par le serveur Asterisk.
- Brancher un casque, et composer le « 3999 » sur le softphone. Entendez-vous un message de bienvenue ? (si besoin aller dans les options du softphone pour être certain que le casque et micro sont bien configurés).

Configurons le plan de numérotation pour lui affecter le n° d'abonné 3*i*03 :

- Modifier « extensions.conf » pour affecter le n° d'abonné « 3*i*03 » à ce poste.
- Recharger Asterisk puis tester le plan de numérotation depuis la console en utilisant la commande « console dial » suivie du numéro du poste créé :


```
console dial 3i03
```
- Tester dans les 2 sens les appels entre les abonnés 3*i*01, 3*i*02 et 3*i*03 pour vérifier que le fonctionnement soit correct.

Nous avons maintenant installé suffisamment de téléphones IP pour pouvoir explorer plus loin le potentiel d'Asterisk.

V. Serveur de temps NTP

Afin de synchroniser l'horloge des téléphones, nous allons installer un **serveur de temps** sur la VM exécutant Asterisk.

- Donner la définition de **NTP** et les fonctionnalités d'un serveur NTP.
- Sur la VM hébergeant Asterisk, mettre à jour la liste des paquets et installer NTP :

```
apt-get update
apt-get install ntp
```
- Ouvrir le fichier `/etc/ntp.conf` permettant de configurer le serveur NTP, et modifier la liste des serveurs de temps afin que ceux servant de référence soient :

```
server 0.fr.pool.ntp.org
server 1.fr.pool.ntp.org
server 2.fr.pool.ntp.org
server 3.fr.pool.ntp.org
```
- Mettre en commentaires la ligne suivante :

```
server ntp.ubuntu.com
```
- Limiter l'accès au serveur NTP (port 123) uniquement aux clients NTP de votre LAN :

```
restrict 192.168.1.0 mask 255.255.255.0
```
- Pour diagnostiquer le bon état de notre serveur, nous allons l'interroger pour connaître les serveurs qui ont servi de source de temps, et leur niveau dans la strate des serveurs permettant la définition au plus juste du temps :

```
ntpq -p
```
- Vérifier le temps (date – heure) de votre OS
`date`

NB : si l'heure n'est pas correcte, redémarrer le service NTP et réactiver l'interface `eth0` (ou au pire redémarrer la VM), car la recherche de l'heure exacte se fait normalement à l'activation d'une carte réseau.

Nous allons configurer nos hardphones pour se servir du serveur NTP de votre VM comme référence de temps ; ainsi tous les postes seront synchronisés :

- Pour le **6757i** : appuyer sur la clé, puis > 2-Préférences > 6-Time and Date :> 4-Time : Zone : Bruxelles => appuyer sur Done> 5-Time Server 1 : *adresse_IP_asterisk_server* => DoneSortir des menus : le temps doit maintenant être correctement affiché sur le poste.
- Pour le **GXV3140** : > Menu > Settings > Time :NTP server : *adresse_IP_asterisk_server* / Time Zone : GMT+01 => appuyer sur Save et sortir.L'heure doit être affichée correctement sur l'écran.

VI. Observations des trames générées (pas le temps...)

Avant de « jouer » avec Asterisk, nous allons observer le trafic généré par les téléphones et le serveur, et notamment nous intéresser à la sécurité des communications audio... Nous allons observer les trames émises par les différentes entités lors d'une communication téléphonique IP. Nous allons notamment voir qu'il existe trois phases distinctes, faisant appel à des protocoles différents :

- la **phase de mise en place de la communication** entre 2 abonnés, via le serveur VoIP,
- la **phase de communication** à proprement parler : échanges de trames de voix de la conversation en cours,
- la **phase de terminaison**, mettant fin à la communication entre les 2 postes IP.

Les protocoles utilisés lors d'un appel téléphonique IP sont :

- **SIP = Session Initiation Protocol** : c'est un protocole standardisé de signalement (RFC 3261) au niveau de la couche applicative du modèle OSI, fonctionnant généralement sur la couche transport en UDP (ou quelquefois TCP...), typiquement sur le **port 5060**.
Son rôle est de créer, modifier ou terminer des sessions téléphoniques.

SIP est très similaire à HTTP dans son comportement parce que des clients SIP envoient des requêtes au serveur qui va répondre avec des réponses (status). La différence avec HTTP est que des clients SIP peuvent aussi répondre à des requêtes venant d'un serveur.

D'autres protocoles de signalement existent : H323, SCCP de Cisco... Mais SIP a progressivement supplanté ces deux protocoles.

- **SDP = Session Description Protocol** : protocole standardisé (RFC 4566) fournissant des infos sur les paramètres d'initialisation multimédia comme des appels VoIP.
- **RTP = Real-time Transport Protocol** : protocole de transport standardisé (RFC 3550) travaillant sur UDP au niveau de la couche transport du modèle OSI ; il est considéré par certains comme un protocole couche 4 OSI (transport), mais par d'autres comme protocole de couche supérieure (5) ; concrètement, l'en-tête UDP est enrichie de champs spécifiques (numérotation des paquets pour gérer les pertes et les dé-séquencements), horloge pour gestion de la gigue,...)
- **RTCP = Real-time Transport Control Protocol** : protocole étroitement lié à RTP (aussi défini dans la RFC 3550). Il ne transporte aucune donnée mais donne des informations sur la qualité de service fourni par RTP

Pour capturer les trames circulant pour une communication en VoIP, nous allons utiliser le logiciel de capture de trames **Wireshark**, déjà installé **sur votre PC physique** :

- Lancer Wireshark en mode admin, et choisir l'**interface physique Ethernet** à analyser.

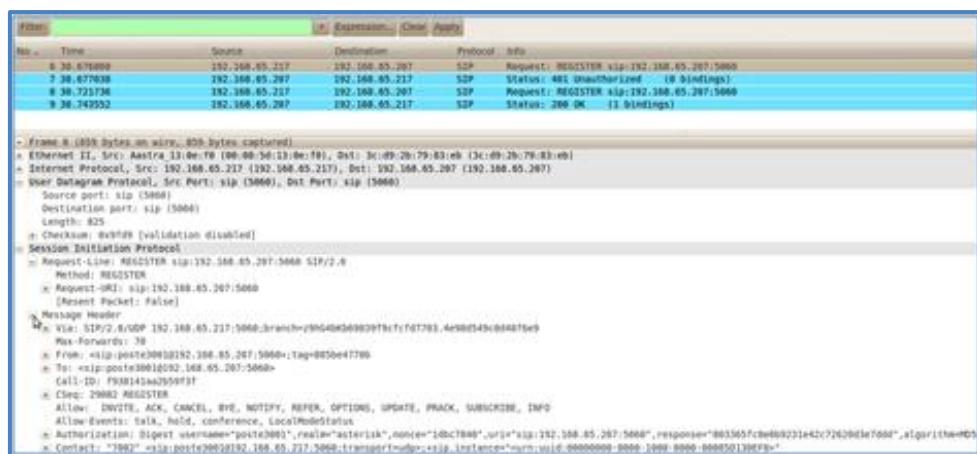
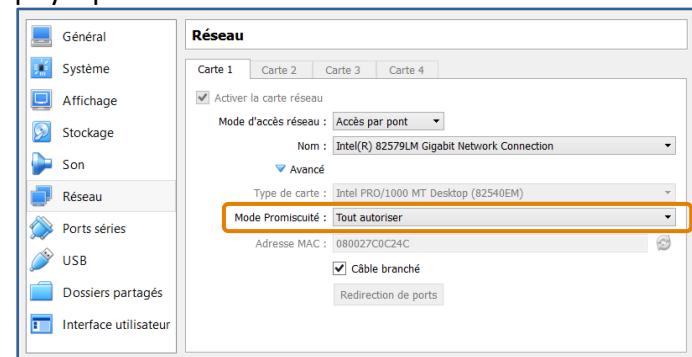
VI.1. Enregistrement d'un poste IP sur le serveur

Nous allons rebooter Asterisk et un téléphone afin de capturer les trames d'enregistrement d'un téléphone IP sur le serveur :

- Dans le terminal en interface CLI Asterisk, sortir de l'interface CLI (Ctrl-C).
- Quelle commande devez-vous exécuter pour visualiser dans ce terminal les processus contenant le mot clé Asterisk ? Si besoin, supprimer le(s) processus exécutant Asterisk afin d'être sûr qu'Asterisk ne soit plus actif.

Nous allons effectuer la manipulation suivante à partir du **hardphone 6757i** :

- Eteindre le téléphone ; redémarrer ensuite le serveur Asterisk, et enfin lancer avec Wireshark une capture sur l'interface physique Ethernet.
- Pour être certain de récupérer tout le trafic, vérifier que vos VM ont leur **mode promiscuité** sur « **Tout autoriser** » :
- Effectuer un filtrage sur l'**@ IP** du téléphone et sur le protocole **SIP** ; vous donnerez la syntaxe du filtre utilisé.
- Redémarrer le téléphone et observer son enregistrement. Quand l'interface CLI d'Asterisk confirme que l'enregistrement est effectué, arrêter la capture.



- Vous devez normalement avoir obtenu 4 trames SIP. Commenter les échanges et protocoles observés sur votre capture (que vous joindrez à votre compte-rendu), et représentez sous forme d'un graphe les échanges ayant eu lieu pour l'enregistrement du poste SIP sur le serveur.
- Détaillez le contenu des trames au niveau du protocole SIP, et en particulier le champ « **Authorization** » ; vous expliquerez pourquoi la 1^{ère} demande d'enregistrement a été refusée par le serveur alors que la 2^{ème} est acceptée.

VI.2. Initialisation et fermeture d'une communication

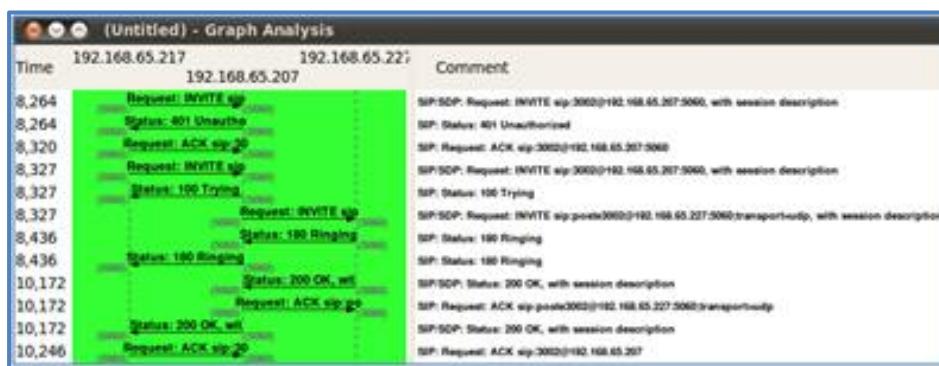
Observons les trames échangées lors d'une mise en communication de 2 téléphones :

- Supprimer le filtre entré dans Wireshark puis recommencer une capture ; lancer un appel du 3i03 vers le 3i01, discuter pendant 5 s environ, puis terminer la communication ; arrêter la capture de trames.
- Quels types de trames ont été capturés ? Commenter leur rôle, en dissociant bien les 3 phases :
 - initialisation de la communication
 - communication téléphonique en elle-même
 - fermeture de la communication
- Filtrer tout d'abord votre capture afin de ne garder que les trames **SIP**.

No.	Time	Source	Destination	Protocol	Info
6	18:284113	192.168.65.217	192.168.65.207	SIP/SDP	Request: INVITE sip:3i03@192.168.65.207:5060; with session description
7	18:284384	192.168.65.207	192.168.65.217	SIP	Status: 401 Unauthorized
8	18:284414	192.168.65.217	192.168.65.207	SIP	Request: ACK sip:3i03@192.168.65.207:5060
10	18:226795	192.168.65.217	192.168.65.207	SIP/SDP	Request: INVITE sip:3i01@192.168.65.207:5060; with session description
13	18:326929	192.168.65.207	192.168.65.217	SIP	Status: 503 Try-again
13	18:327411	192.168.65.207	192.168.65.217	SIP/SDP	Request: INVITE sip:3i01@192.168.65.207:5060;transport=tcp, with session description
13	18:459886	192.168.65.227	192.168.65.207	SIP	Status: 503 Ringing
14	18:436355	192.168.65.207	192.168.65.217	SIP	Status: 503 Ringing
17	18:171561	192.168.65.227	192.168.65.207	SIP/SDP	Status: 200 OK, with session description
18	18:171793	192.168.65.207	192.168.65.227	SIP	Request: ACK sip:3i01@192.168.65.207:5060;transport=tcp
19	18:171998	192.168.65.207	192.168.65.217	SIP/SDP	Status: 200 OK, with session description
20	18:245718	192.168.65.217	192.168.65.207	SIP	Request: ACK sip:3i01@192.168.65.207

- **Initialisation de la communication** : expliciter les **protocoles de plus haut niveau observés** sur votre capture (que vous joindrez à votre compte-rendu).
- Quel protocole de la couche « transport » est utilisé pour les échanges SIP ?
- Quel est le n° de port utilisé par le protocole SIP ?
- Représentez sous forme d'un graphe les échanges ayant eu lieu entre les 2 postes SIP et le serveur pour l'initialisation de la communication.

NB : vous pouvez utiliser la fonction « Graph Analysis » de Wireshark (dans le menu « Statistics > Flow Graph »).



- **Fermeture de la communication** : effectuer la même étude complète pour la fermeture de la communication : relevé des protocoles observés, graphe des échanges, explications...

VI.3. Transport de la voix pendant 1 communication

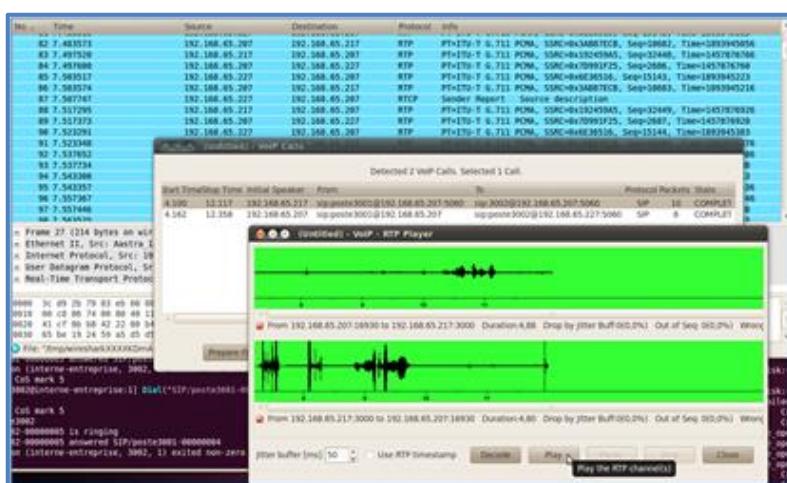
Une fois qu'une communication est établie, le protocole SIP n'est plus utilisé, jusqu'à la demande de fin de communication exprimée par un des postes. Pendant la communication proprement dite, d'autres protocoles sont utilisés pour le transport de la voix.

- Quels sont les protocoles de plus haut niveau utilisés pour le **transport de la voix** ?
- Donner le filtre à appliquer à votre capture précédente pour n'afficher que ces trames ; expliciter les protocoles observés sur votre capture (que vous joindrez à votre compte-rendu).
- Quel protocole de la couche « transport » est utilisé pour le transport de la voix ? Ce protocole permet-il un re-séquençage des données reçues dans le désordre ?
- Quels sont les n° de ports utilisés sur le serveur et les clients ToIP pour la transmission de la voix ?
- Existe-t-il un processus de remise en ordre des données reçues dans RTP ? Si oui, identifiez ce champ.
- Représenter un graphe montrant (pas en totalité, un extrait seulement !) les échanges entre les 2 postes ToIP et le serveur pour le transport de la voix.

VI.4. Ecoute de conversations

Nous allons utiliser des fonctionnalités de Wireshark afin de visualiser les conversations présentes dans une capture de trames, et tenter d'écouter une conversation :

- Aller dans le menu « Telephony > VoIP Calls » : quelles conversations y sont listées ?
- Sélectionner un appel VoIP (poste1 ⇔ serveur ou serveur ⇔ poste2), cliquer ensuite sur « Player », « Decode », sélectionner les 2 flux (entrant et sortant), et cliquer sur « Play » : vous devez entendre la conversation.



- Que peut-on en conclure en matière de sécurité de l'information ?
- Quel(s) protocole(s) de transport de la voix en temps réel permettant de chiffrer les données transportées (= la voix) connaissez-vous ?

VI.5. Sécurisation des flux

Il est possible de sécuriser les flux de transport de voix téléphonique en remplaçant le protocole de transport de la voix actuel (**RTP**) par le protocole **SRTP** (**Secure RTP**).

- Quel type de chiffrement est utilisé dans le protocole SRTP ?

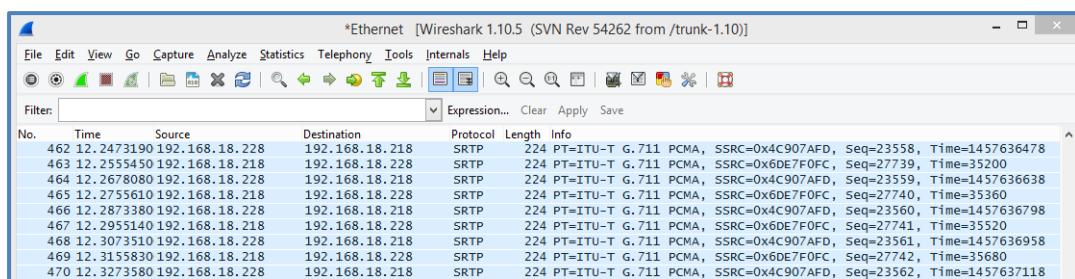
Les hardphones et softphones IP ne supportent pas tous le protocole STRP, mais le 6757i sait le gérer, et est compatible directement avec Asterisk. Nous allons donc faire uniquement un essai avec ce poste, et tenter d'observer une communication entre le 6757i et le serveur Asterisk.

Afin d'informer le poste 6757i et le serveur Asterisk que nous souhaitons utiliser le protocole STRP, effectuer les modifications suivantes :

- Sur le 6757i, aller dans le menu « Line 1 » de son portail web, et modifier le champ « RTP Encryption » en « SRTP only ».
- Sur le serveur Asterisk, rajouter dans leur profil SIP (fichier « `sip.conf` ») le paramètre : `encryption = yes`

Tout est maintenant prêt pour tester si un appel avec le 6757i génère maintenant un flux audio chiffré :

- Démarrer une nouvelle capture de trames sur Wireshark, composer le 3999 sur le 6757i (Vous pouvez parler pendant que le serveur diffuse son message), puis raccrocher.
- Arrêter la capture. Avez-vous récupéré des trames transportant la voix de manière sécurisée ?



- Utiliser les outils de Wireshark découverts précédemment afin de tenter d'écouter la conversation : peut-on toujours écouter la conversation ?
- Maintenant que nous avons vu l'effet de la sécurisation des flux, nous allons revenir à un mode non sécurisé pour le reste du TP : **supprimer les modifications faites sur le serveur et le 6757i afin de supprimer cette sécurisation des flux**.

VII. Déploiement automatique des postes

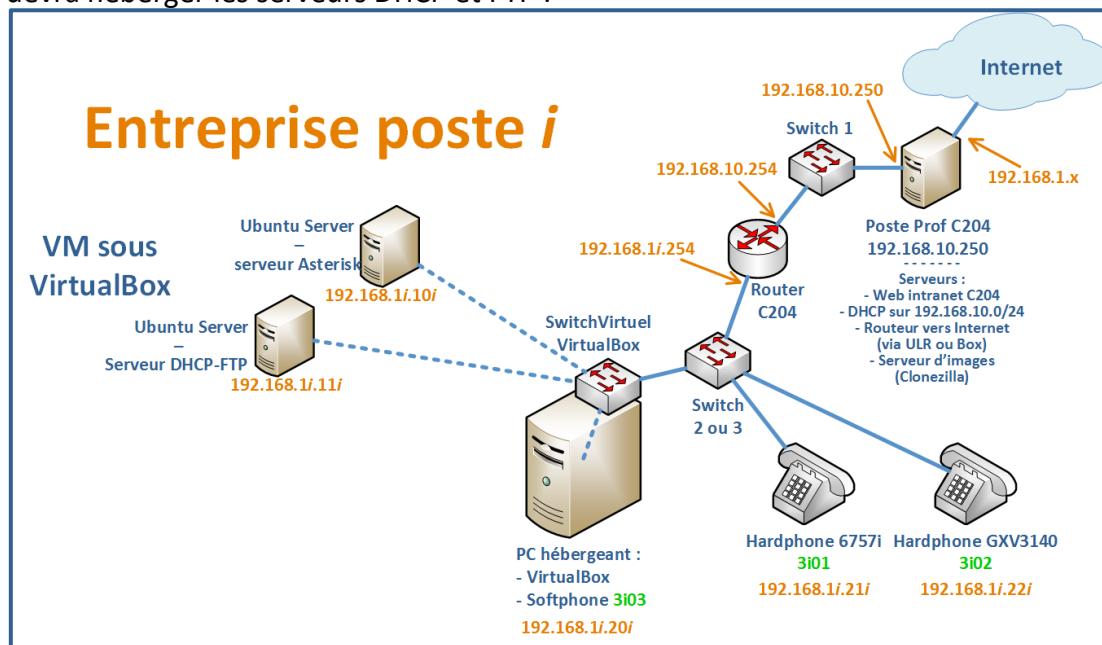
Nous allons voir comment améliorer le déploiement des hardphones IP car la configuration manuelle poste par poste deviendrait ingérable dès que le nombre de postes devient important. Pour gagner du temps, nous allons nous intéresser uniquement au déploiement des postes 6757i ; le déploiement d'un autre type de postes (les GXV3140 par exemple) se ferait selon un processus identique, il n'y a donc aucun intérêt à le faire deux fois.

Le **déploiement automatique (auto-provisioning)** de postes se base sur l'utilisation de :

- un **serveur DHCP** pour fournir automatiquement les **paramètres IP** aux téléphones et l'adresse du serveur FTP ou TFTP qu'ils devront utiliser
- un **serveur FTP ou TFTP** pour que les postes IP viennent chercher leur firmware (éventuellement, en cas de **mise à jour du firmware** des postes déployés), et leur **fichier de configuration** contenant les paramètres propres à chaque téléphone. Ici en l'occurrence, nous aurons besoin d'un serveur FTP pour les postes 6757i.

VII.1. VM hébergeant les services DHCP et FTP

Nous allons nous intéresser maintenant à une 2^{ème} VM, la VM « M4205-DHCP-FTP-Server » qui devra héberger les serveurs DHCP et FTP :



- Démarrer la VM DHCP-FTP-Server, s'identifier avec « user/rtrt », puis modifier le nom de la machine contenu dans le fichier « `/etc/hostname` » en « `M4205-DHCP-FTP-Server.poste1.c204.rtrt` ». Redémarrer ensuite votre VM en utilisant la commande `shutdown` et les options adéquates (obtenez plus d'aide sur cette commande avec « `shutdown --help` »).
- Comme pour la VM Asterisk-Server, configurer la carte réseau eth0 de la VM de façon permanente, en respectant les paramètres adéquats (cf schéma).

VII.2. Consultation des docs techniques Aastra

Pour la réalisation de ce déploiement automatique, la consultation des documents constructeurs des postes IP est fondamentale. En l'occurrence ici, le document « [AMT_PTD_TR_0014_7_1_FR.pdf](#) » disponible sur l'extranet d'Aastra (et dans les annexes du serveur Intranet de la salle) donne des indications sur le déploiement des postes 6757i :

Les postes Aastra A67xxi sont raccordés soit dans un VLAN ToIP dédié aux postes, soit cohabitent à la fois dans un VLAN Data et un VLAN ToIP en 802.1Q si un PC est chaîné au poste .

Le serveur FTP permettant le téléchargement des postes Aastra A67xxi (firmware et fichiers de configuration) peut être hébergé :

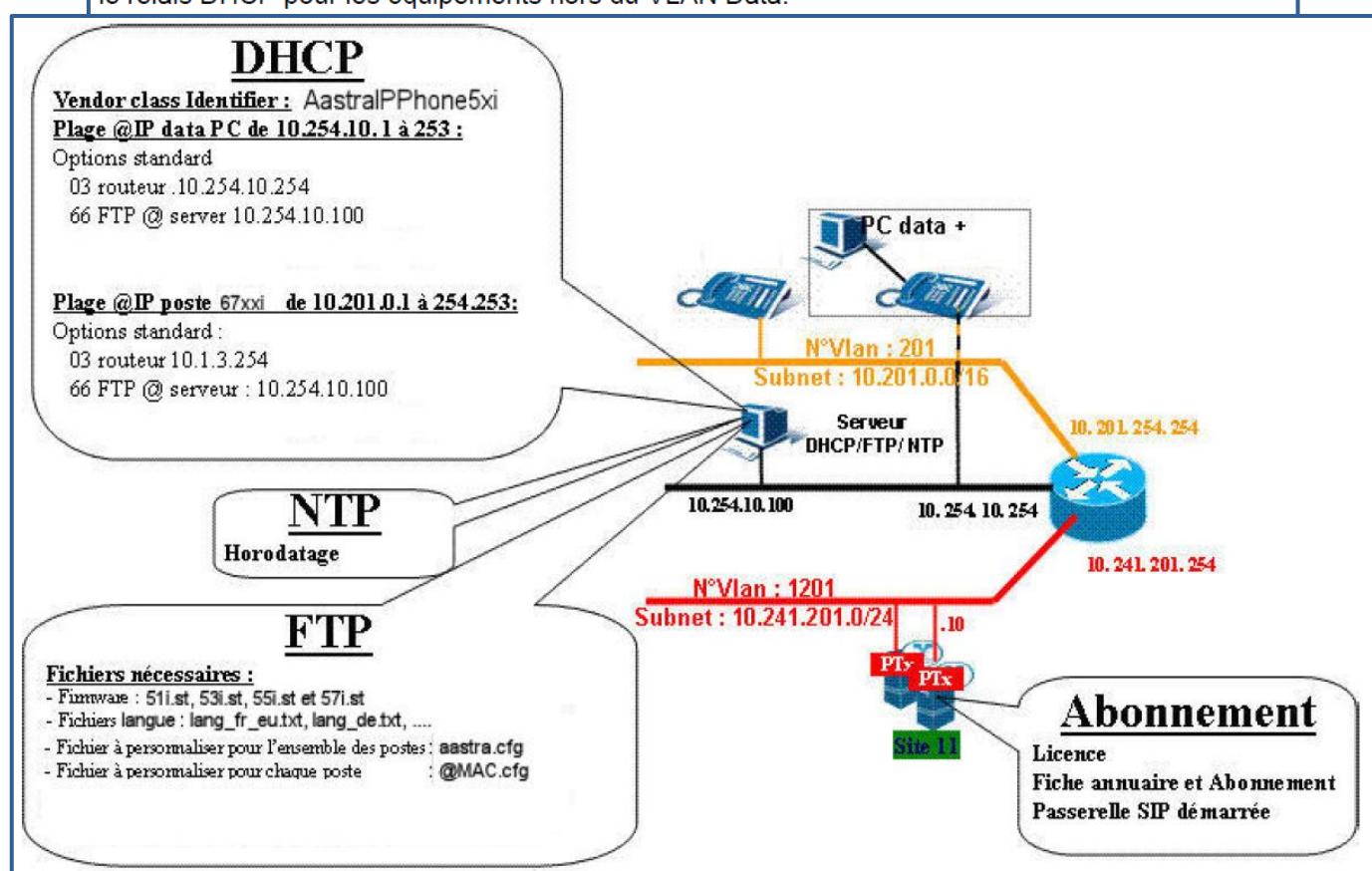
- > **sur une plate-forme Windows 2000/2003 Serveur** (sous la responsabilité de l'administrateur réseau)
- > **sur un système A5000 Server** (sous la responsabilité de l'administrateur réseau).
- > **sur un système Aastra X series**. Le serveur FTP, dans ce cas, est embarqué sur la carte UCV et le service correspondant sera géré complètement par le système.

Pour chaque poste, l'adresse du serveur FTP utilisé doit être déclarée :

- Soit par configuration manuelle (directement sur le poste ou via l'interface Web)
- Soit par configuration du serveur DHCP.

Un serveur DHCP pour la fourniture d'adresses IP doit être accessible depuis le VLAN ToIP.

Le commutateur Ethernet de niveau 3 (routeur en général) réalise le routage inter-VLAN ainsi que le relais DHCP pour les équipements hors du VLAN Data.



Que ce soit une configuration simplifiée ou complexe, Aastra préconise la configuration des postes via un serveur DHCP embarqué ou externe pour récupérer automatiquement ses paramètres réseaux standard et gérés ceux associés à la classe fournisseur des postes Aastra A67xxi) si nécessaire. La configuration est complétée par le téléchargement du logiciel et des fichiers de configuration associés aux postes A67xxi via un serveur FTP embarqué ou externe en déposant manuellement ces fichiers dans le répertoire de dépôt approprié.

Raccordement au Switch DATA

Postes A67xxi avec ou sans PC chainé:

Dans cette configuration, le poste est connecté sur un port supportant les VLAN 'Postes' et 'Data': il doit marquer ses trames dans le VLAN 'Postes'. Le trafic sur le port du Switch est marqué pour le poste et non marqué pour le PC.

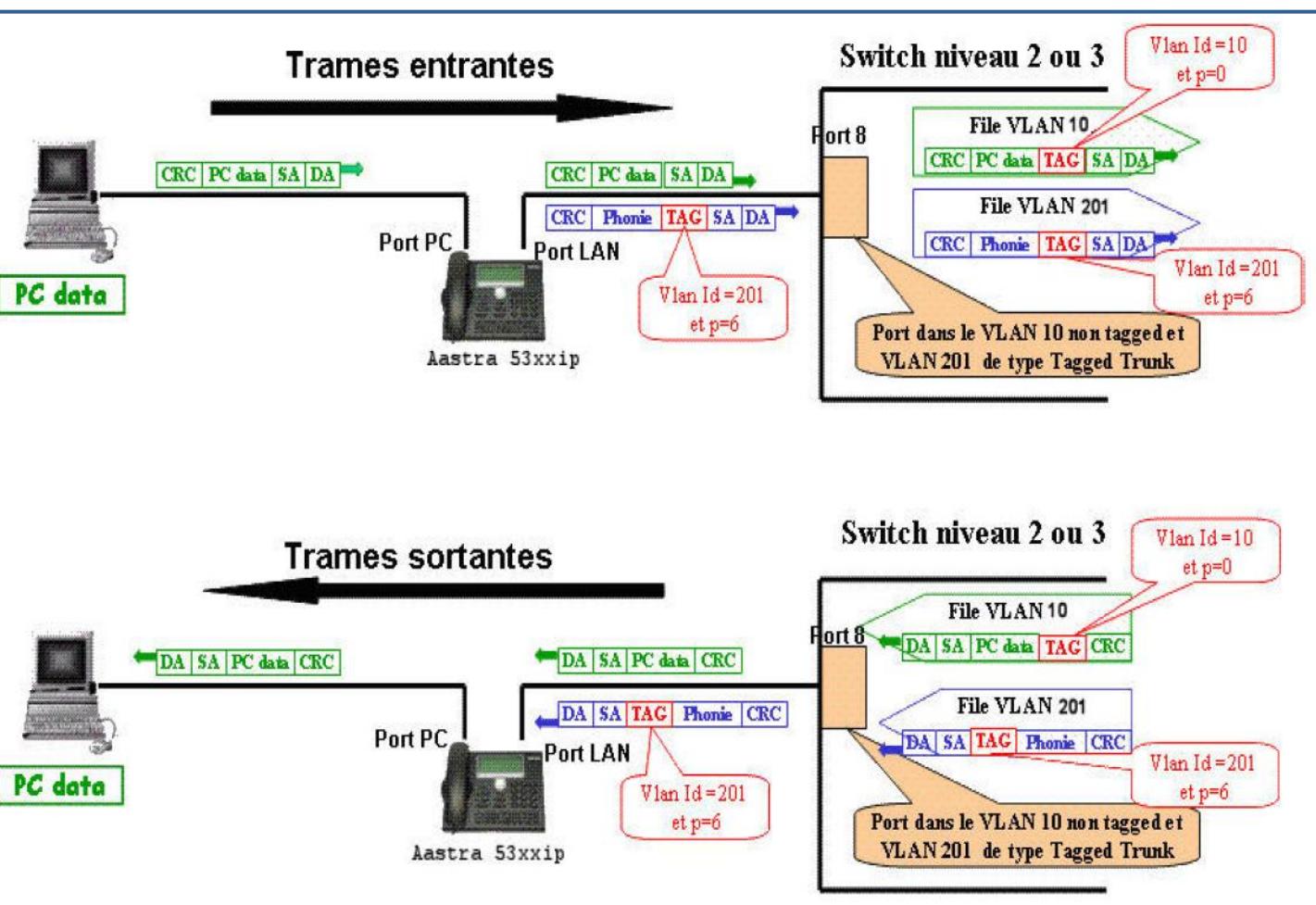
Aastra préconise de configurer tous les ports du Switch sur lesquels sont raccordés des postes A67xxi de la même manière que le poste soit seul ou chaîné avec un PC ce qui évite à l'administrateur réseau de reconfigurer les ports du Switch en fonction de la présence ou non d'un PC.

Principe de connexion du poste:

Connexion du poste sur le port du switch (VLAN 'Postes' + 'Data'): le poste va émettre sa requête DHCP dans le VLAN 'Data' et obtient sa configuration, notamment son VLAN ID, via le serveur FTP (firmware, packs langue et fichiers de configuration) en utilisant une adresse IP temporaire dans le VLAN 'Data'. Le poste va ensuite redémarrer sur le VLAN 'Postes'. Le poste va émettre une deuxième requête DHCP dans le VLAN 'Postes' et acquérir sa configuration complète.

Exemple de marquage de trames ethernet sur le port du Switch:

Il faut s'assurer que la prise réseau de raccordement au Switch DATA appartient bien au VLAN ToIP des postes Aastra A67xxi et au VLAN data des PC



VII.3. Installation et configuration du serveur DHCP

Nous allons installer sur la VM « DHCP-FTP-Server » le serveur DHCP « isc-dhcp-server » :

- Mettre à jour la liste des paquets puis installer le paquet « isc-dhcp-server ».

Commençons la configuration du serveur DHCP en sélectionnant l'interface sur laquelle le serveur devra écouter et distribuer des adresses IP :

- Indiquer dans « /etc/default/isc-dhcp-server » l'**interface d'écoute** :
INTERFACES="eth0"

La configuration du service DHCP se fait dans le fichier dans « /etc/dhcp/dhcpd.conf », en utilisant notamment des options (paramètres) définis tels que dans le tableau présenté ici : <http://www.ipamworldwide.com/ipam/isc-dhcpv4-options.html> .

La configuration DHCP peut être très différente selon les besoins. Différents comportements peuvent être souhaités :

- dans certaines installations, on ne souhaite **adresser aucune @ IP à des inconnus**, et **seules les cartes réseau d'@ MAC connues et déclarées se verront attribuer une @ IP** qui peut être **réservée (fixe)** ; la config ci-dessous en est un exemple :

```
subnet 192.168.18.0 netmask 255.255.255.0 {
    default-lease-time 21600;
    max-lease-time 21600;
    # option 3 :
    option routers              192.168.18.254;
    # option 1 :
    option subnet-mask          255.255.255.0;
    # option 28 :
    option broadcast-address     192.168.18.255;
    # option 6 :
    option domain-name-servers   192.168.1.1, 10.1.30.24;
    # option 42 :
    option ntp-servers          192.168.18.108;
    # option 66 :
    option tftp-server-name      "192.168.18.118";

    host 6757i-bureauB101 {
        hardware ethernet A1:B1:C1:D1:E1:F1;
        fixed-address 192.168.18.50;
    }
    host 6757i- bureauB102 {
        hardware ethernet A2:B2:C2:D2:E2:F2;
        fixed-address 192.168.18.51;
    }
    ...
}
```

- dans d'autres installations, on préférera **ne pas figer les @ IP sur les @ MAC** des postes si on estime que la tâche de suivi des @ MAC est trop « lourde » ; on peut alors se baser sur l'utilisation d'un paramètre émis par les équipements lors d'une demande d'attribution d'adresse : le « **vendor-class-identifier** » ; dans ce cas, **un pool d'adresses IP est réservé pour tous les équipements de même « vendor-class-identifier »**, et dans ce cas, pas besoin de gérer le suivi des @ MAC des téléphones. La config ci-dessous en est un exemple :

```

class "hardphonesIPAAstra" {
    match if substring (option vendor-class-identifier, 0, 8) = "AastralP";
}

subnet 192.168.18.0 netmask 255.255.255.0 {
    default-lease-time 21600;
    max-lease-time 21600;
    # option 3 :
    option routers                 192.168.18.254;
    # option 1 :
    option subnet-mask             255.255.255.0;
    # option 28 :
    option broadcast-address       192.168.18.255;
    # option 6 :
    option domain-name-servers    192.168.1.1, 10.1.30.24;
    # option 42 :
    option ntp-servers            192.168.18.108;
    # option 66 :
    option tftp-server-name        "192.168.18.118";

    pool {
        allow members of "hardphonesIPAAstra";
        range 192.168.18.50 192.168.18.99;
    }
}

```

- En vous inspirant de l'exemple précédent, ajouter un bloc pour votre sous-réseau dans « `/etc/dhcp/dhcpd.conf` » respectant les consignes suivantes :

- **délivrance d'une @ IP uniquement aux équipements 6757i**, membres d'une classe « 6757i », qui effectue une **discrimination sur le « vendor-class-identifier » complet** défini pour les 6757i (voir les extraits des docs constructeurs données pages précédentes, où le 'x' sera à remplacer par un '7')
- pool d'adresses disponibles : `192.168.1i.21i` à `192.168.1i.219`

- Vérifier la syntaxe de votre fichier de configuration avec la commande suivante, et le cas échéant, corriger vos erreurs :

`dhcpd -t`

```

root@dhcp-server:~# dhcpcd -t
Internet Systems Consortium DHCP Server 4.2.4
Copyright 2004-2012 Internet Systems Consortium.
All rights reserved.
For info, please visit https://www.isc.org/software/dhcp/
root@dhcp-server:~#

```

Mais dans notre cas, comme indiqué dans les docs Aastra, les postes Aastra 6757i ne sont pas conçus pour utiliser un serveur TFTP, mais un serveur **FTP** ; l'option 66 classiquement utilisée n'est donc pas suffisante, et par contre, il faut rajouter une option non standardisée, pour renseigner un champ propre au concepteur du téléphone. Pour cela, nous allons utiliser l'option « **43** » :

- Pourquoi Aastra a-t-il préféré utiliser FTP et non pas TFTP ?
- En vous aidant d'Internet, expliquer quels sont le nom et l'utilité de l'option « 43 ».

Toujours en consultant la documentation constructeur « [AMT_PTD_TR_0014_7_1_FR.pdf](#) », dans le chapitre « 10 - Configuration d'un serveur DHCP externe », on peut connaître la liste des paramètres que le poste 6757i va demander au serveur DHCP :

<i>Paramètre ou option DHCP</i>	<i>Remarque</i>
Adresse IP et masque de sous réseau (option 1)	paramètre demandé dans l'option 55
Adresse IP de la passerelle (option 3)	paramètre demandé dans l'option 55
Adresse IP serveur DNS (option 6)	paramètre demandé dans l'option 55
Adresse IP serveur NTP (option 42)	paramètre demandé dans l'option 55
Option spécifique Vendeur (option 43)	paramètre demandé dans l'option 55 Voir "Paramètres négociables via l'option 43" à la page 282
Information spécifique Vendeur Class (option 60)	*
Adresse IP du serveur FTP (option 66)	

Il faut en fait rajouter une **option spécifique (43)**, identifiée avec le numéro spécifique constructeur « **2** », pour **préciser l'@ du serveur FTP ainsi que les identifiants de connexion à utiliser**, comme dans l'exemple ci-dessous :

Tableau 2: Paramètres négociables via l'option 43

Paramètre	Code	Valeur Hex.	Longueur/ type	Remarque
cfg-server	02	02	String	Adresse IP du serveur ftp. Exemple: ftp://connexio:connexio@192.168.0.100

Un exemple de config de serveur DHCP montre qu'il est possible de rajouter deux morceaux de code similaires à ceux-ci-dessous, le premier avant la déclaration du subnet, le 2^{ème} à l'intérieur de la définition du pool, afin de **faire construire l'option 43 souhaitée, de code 2** :

```
# Declaration de la structure du terminal de la gamme 67xxi(57i), modele 57i
option space Connexio-57i;
option Connexio-57i.cfg-server-address code 2 = string;
```

```
if substring(option vendor-class-identifier,0,16) = "AastralPPhone57i" {
    # Parametres dhcp propres au sous-reseau et au terminal 67xxi(57i) 57i
    option server.vendor-option-space Connexio-57i;
    option Connexio-57i.cfg-server-address "ftp://connexio:connexio@50.1.1.1";
}
# fin de condition pool
```

NB : Pour mieux comprendre le rôle de ces commandes, vous pouvez consulter <http://manpages.ubuntu.com/manpages/precise/man5/dhcp-options.5.html> :

VENDOR ENCAPSULATED OPTIONS

The DHCP protocol defines the **vendor-encapsulated-options** option, which allows vendors to define their own options that will be sent encapsulated in a standard DHCP option. It also defines the **Vendor Identified Vendor Sub Options** option ("VIVSO"), and the DHCPv6 protocol defines the **Vendor-specific Information Option** ("VSIO"). The format of all of these options is usually internally a string of options, similarly to other normal DHCP options. The VIVSO and VSIO options differ in that they contain options that correspond to vendor Enterprise-ID numbers (assigned by IANA), which then contain options according to each Vendor's specifications. You will need to refer to your vendor's documentation in order to form options to their specification.

The value of these options can be set in one of two ways. The first way is to simply specify the data directly, using a text string or a colon-separated list of hexadecimal values. For help in forming these strings, please refer to **RFC2132** for the DHCPv4 **Vendor Specific Information Option**, **RFC3925** for the DHCPv4 **Vendor Identified Vendor Sub Options**, or **RFC3315** for the DHCPv6 **Vendor-specific Information Option**. For example:

```
option vendor-encapsulated-options
 2:4:
  AC:11:41:1:
  3:12:
  73:75:6e:64:68:63:70:2d:73:65:72:76:65:72:31:37:2d:31:
  4:12:
  2f:65:78:70:6f:72:74:2f:72:6f:6f:74:2f:69:38:36:70:63;
option vivso
  00:00:09:bf:0E:
  01:0c:
  48:65:6c:6c:6f:20:77:6f:72:6c:64:21;
option dhcp6.vendor-opts
  00:00:09:bf:
  00:01:00:0c:
  48:65:6c:6c:6f:20:77:6f:72:6c:64:21;
```

The second way of setting the value of these options is to have the DHCP server generate a vendor-specific option buffer. To do this, you must do four things: define an option space, define some options in that option space, provide values for them, and specify that that option space should be used to generate the relevant option.

To define a new option space in which vendor options can be stored, use the option space statement:

```
option space name [ [ code width number ] [ length width number ] [ hash size number ] ] ;
```

Where the numbers following **code width**, **length width**, and **hash size** respectively identify the number of bytes used to describe option codes, option lengths, and the size in buckets of the hash tables to hold options in this space (most DHCPv4 option spaces use 1 byte codes and lengths, which is the default, whereas most DHCPv6 option spaces use 2 byte codes and lengths).

The code and length widths are used in DHCP protocol - you must configure these numbers to match the applicable option space you are configuring. They each default to 1. Valid values for code widths are 1, 2 or 4. Valid values for length widths are 0, 1 or 2. Most DHCPv4 option spaces use 1 byte codes and lengths, which is the default, whereas most DHCPv6 option spaces use 2 byte codes and lengths. A zero-byte length produces options similar to the DHCPv6 Vendor-specific Information Option - but not their contents!

The hash size defaults depend upon the **code width** selected, and may be 254 or 1009. Valid values range between 1 and 65535. Note that the higher you configure this value, the more memory will be used. It is considered good practice to configure a value that is slightly larger than the estimated number of options you plan to configure within the space. Previous versions of ISC DHCP (up to and including DHCP 3.0.*), this value was fixed at 9973.

The name can then be used in option definitions, as described earlier in this document. For example:

```
option space SUNW code width 1 length width 1 hash size 3;
```

```
option SUNW.server-address code 2 = ip-address;
```

```
option SUNW.server-name code 3 = text;
```

```
option SUNW.root-path code 4 = text;
```

```
option space ISC code width 1 length width 1 hash size 3;
```

```
option ISC.sample code 1 = text;
```

```
option vendor.ISC code 2495 = encapsulate vivso-sample;
```

```
option vendor-class.ISC code 2495 = text;
```

```
option ISC.sample "configuration text here";
```

```
option vendor-class.ISC "vendor class here";
```

```
option space docsis code width 2 length width 2 hash size 17;
```

```
option docsis.tftp-servers code 32 = array of ip6-address;
```

```
option docsis.cablelabs-configuration-file code 33 = text;
```

```
option docsis.cablelabs-syslog-servers code 34 = array of ip6-address;
```

```
option docsis.device-id code 36 = string;
```

```
option docsis.time-servers code 37 = array of ip6-address;
```

```
option docsis.time-offset code 38 = signed integer 32;
```

```
option vsio.docsis code 4491 = encapsulate docsis;
```

Once you have defined an option space and the format of some options, you can set up scopes that define values for those options, and you can say when to use them. For example, suppose you want to handle two different classes of clients. Using the option space definition shown in the previous example, you can send different option values to different clients based on the vendor-class-identifier option that the clients send, as follows:

```
class "vendor-classes" {
  match option vendor-class-identifier;
}
```

```
subclass "vendor-classes" "SUNW.Ultra-5_10" {
  vendor-option-space SUNW;
  option SUNW.root-path "/export/root/sparc";
}
```

```
subclass "vendor-classes" "SUNW.i86pc" {
  vendor-option-space SUNW;
  option SUNW.root-path "/export/root/i86pc";
}
```

```
option SUNW.server-address 172.17.65.1;
option SUNW.server-name "sundhcp-server17-1";
```

```
option vivso-sample.sample "Hello world!";
```

```
option docsis.tftp-servers ::1;
```

As you can see in the preceding example, regular scoping rules apply, so you can define values that are global in the global scope, and only define values that are specific to a particular class in the local scope. The **vendor-option-space** declaration tells the DHCP server to use options in the SUNW option space to construct the DHCPv4 **vendor-encapsulated-options** option. This is a limitation of that option - the DHCPv4 VIVSO and the DHCPv6 VSIO options can have multiple vendor definitions all at once (even transmitted to the same client), so it is not necessary to configure this.

- Rajouter dans votre configuration les lignes manquantes, en prenant bien-sûr le soin de les **personnaliser** ; vous choisirez notamment comme identifiants de connexion login / password : « **posteip** » / « **AastrA6757i** »
- Redémarrer votre serveur DHCP (et le cas échéant, corriger les erreurs de syntaxe...) :
`service isc-dhcp-server restart`

Tout semble maintenant opérationnel côté serveur DHCP. Les postes 6757i vont recevoir les informations nécessaires pour savoir comment se connecter à leur serveur FTP.

VII.4. Configuration du serveur FTP

Nous allons utiliser le **serveur FTP Pure-ftp** :

- Installer le paquet « **pure-ftpd** » sur la VM DHCP-FTP-Server.
- Créer l'utilisateur/groupe système avec lequel le serveur FTP sera lancé :
`groupadd ftpgroup
useradd -g ftpgroup -d /dev/null -s /usr/sbin/nologin ftpuser`

L'ensemble des fichiers de configuration de Pure-ftpd se trouve dans le répertoire « `/etc/pure-ftpd/` ».

- Activer l'authentification liée à Pure-FTP (création d'un lien symbolique pour activer l'authentification des utilisateurs virtuels) :
`ln -s /etc/pure-ftpd/conf/PureDB /etc/pure-ftpd/auth/`
- Créer un dossier pour l'utilisateur virtuel « `posteip` », l'affecter à l'utilisateur/groupe du serveur FTP (pour ne pas le laisser en root/root), puis lui créer un compte avec le même mot de passe que celui déclaré sur le serveur DHCP :
`mkdir -p /home/ftp/posteip
chown -R ftpuser:ftpgroup /home/ftp/posteip
pure-pw useradd posteip -u ftpuser -g ftppgroup -d /home/ftp/posteip`

La dernière commande renseigne le fichier « `/etc/pure-ftpd/pureftpd.passwd` » avec le nouvel utilisateur de Pure-ftp ; vous pouvez d'ailleurs consulter son contenu avec la commande « `cat` ». Enfin il faut transformer ce fichier dans un format sécurisé et lisible par le serveur FTP (« `/etc/pure-ftpd/pureftpd.pdb` ») :

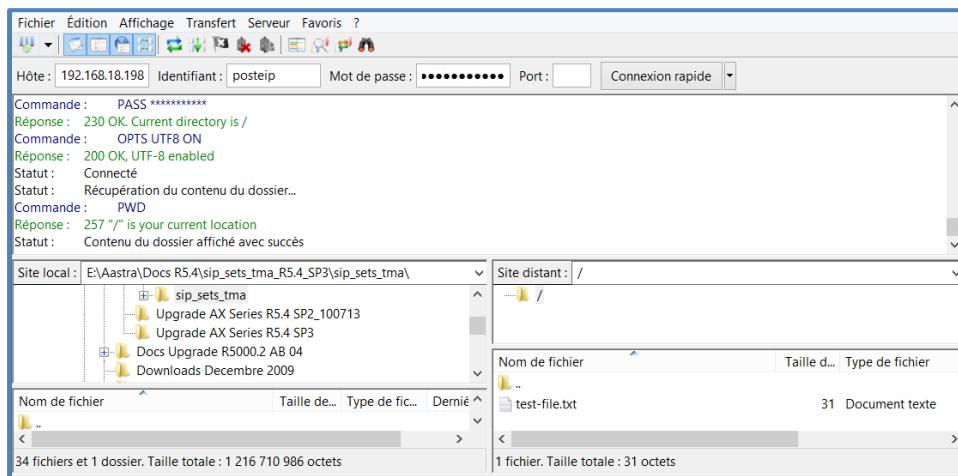
- Régénérer le fichier des utilisateurs :
`pure-pw mkdb`
NB : cette commande est à exécuter après chaque modification ou rajout d'utilisateur afin de régénérer le fichier des utilisateurs.
- Redémarrer votre serveur FTP pour que la configuration soit prise en compte :
`/etc/init.d/pure-ftpd restart`
- Vérifier la liste des utilisateurs qui ont un compte sur ce serveur FTP :
`pure-pw list`

Avant de tester si tout ce que nous venons de faire va fonctionner avec les téléphones, testons une connexion sur le serveur depuis le client Filezilla installé sur le PC physique :

- Créer un fichier texte et le placer dans le dossier FTP de l'utilisateur « posteip ».

```
root@dhcp-server:~# echo 'fichier de test du serveur ftp' > /home/ftp/posteip/test-file.txt
root@dhcp-server:~#
```

- Tenter une connexion avec Filezilla depuis le PC physique avec les identifiants « posteip » / « Aastra6757i » : la connexion fonctionne-t-elle ? Voyez-vous votre fichier et pouvez-vous le télécharger sur votre PC physique ?



- Tester également l'envoi d'un fichier depuis votre PC vers le serveur FTP pour vérifier que le transfert est possible (et donc que le droit en écriture est correctement configuré, même si cela n'est pas nécessaire pour notre configuration).

Maintenant que nos 2 serveurs fonctionnent, il ne reste plus qu'à nous intéresser aux fichiers nécessaires pour le **provisioning** (**déploiement**) des postes 6757i.

```
Commande : STOR lang_fr.txt
Réponse : 150 Accepted data connection
Réponse : 226-File successfully transferred
Réponse : 226 0.006 seconds (measured here), 9.13 Mbytes per second
Statut : Transfert de fichier réussi, 58 333 octets transférés en 1 seconde
```

VII.5. Préparation des fichiers de provisioning

Les postes 6757i vont venir consulter le serveur FTP pour rechercher notamment :

- la présence d'un firmware, pour le cas échéant se mettre à jour automatiquement
- la présence de fichiers sons, en langue française par exemple, personnalisés ou non
- la présence d'un fichier de configuration contenant des paramètres globaux ou personnalisés par adresse MAC.

Les formats des fichiers utilisés par les 6757i pour le provisioning sont les suivants :

- firmware : « 57i.st »
- fichier de configuration global des postes : « aastral.cfg »
- fichier de configuration spécifique à un poste : « @MAC.cfg » (ex: 00085D3A2451.cfg)
- fichiers pack langue: lang_<ISO 639>_<ISO 3166>.txt ou lang_<ISO 639>.txt (ex: lang_fr_ca.txt ou lang_de.txt)

- Récupérer ces fichiers sur l'intranet de la C204 (dans les documents Annexes) ; consulter leur contenu.
- Placer déjà les 2 fichiers que nous n'avons pas besoin de modifier « 57i.st » et « lang_fr.txt » sur le serveur FTP.
- **Ouvrir le fichier « aastr้า.cfg » et effectuer les ajouts ou modifications suivants** afin que les paramètres du serveur ToIP soient automatiquement renseignés lors du provisioning pour n'importe quel poste 6757i :

```
time server1: @ IP de votre serveur Asterisk-NTP
sip line1 proxy ip: @ IP de votre serveur Asterisk
sip line1 proxy port : 5060
sip line1 registrar ip: @ IP de votre serveur Asterisk
sip line1 registrar port : 5060
```

Le fichier précédent est un fichier standard, qui sera exploité par les hardphones 6757i qui n'auront aucun fichier de configuration personnalisé. Mais nous allons aussi créer maintenant un **fichier personnalisé** pour notre poste 6757i afin qu'il soit téléchargé lors du démarrage, et qui renseignera les **paramètres personnels** que le poste doit prendre. Nous allons ainsi directement préciser à chaque poste, en se basant sur leur **@ MAC**, leur paramètres globaux ainsi que leur identité propre (nom d'abonné, n° d'abonné, nom à afficher, password...) :

- **Ouvrir le fichier « MAC_57i.cfg » et le sauvegarder** sous le nom de l'**@ MAC** de votre 6757i (lettres en majuscules !), avec l'**extension « .cfg »** (ex : « A1B2C3D4E5F6.cfg ») ; y effectuer ensuite les ajouts ou modifications suivants :

```
time server1 : @ IP de votre serveur Asterisk - NTP
sip line1 proxy ip : @ IP de votre serveur Asterisk
sip line1 proxy port : 5060
sip line1 registrar ip : @ IP de votre serveur Asterisk
sip line1 registrar port : 5060
sip screen name: "poste 3i01"
sip user name: poste3i01
sip line1 user name: poste3i01
sip line1 auth name: poste3i01
sip line1 password: rtrt
sip line1 screen name 2: " "
#sip line2 user name:
#sip line2 screen name: " "
#
topsoftkey5 value: http://192.168.1.10i/annuaire/i5xi.php
```

Ainsi, les hardphones IP téléchargeront préférentiellement :

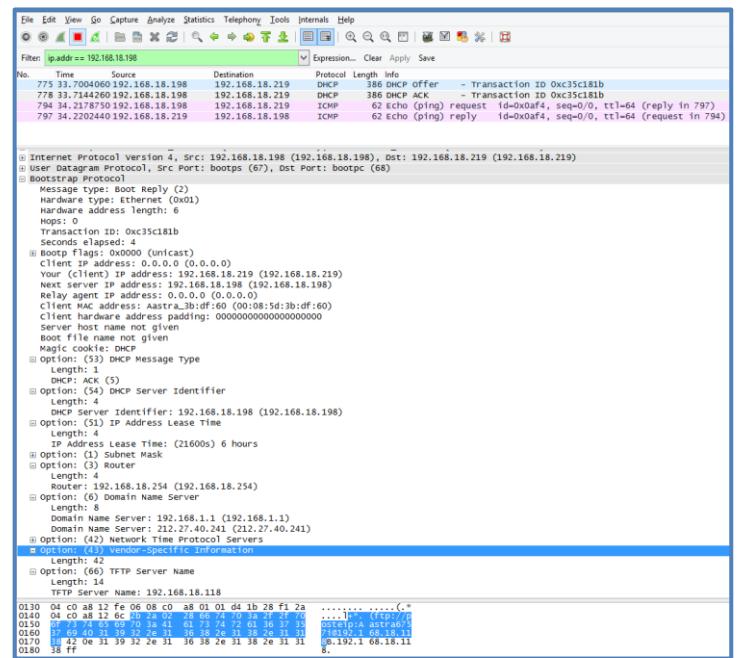
- soit leur fichier spécifique portant le nom de leur **@ MAC**
 - soit le fichier généraliste « aastr้า.cfg », dans le cas où aucun fichier ne portant leur **@ MAC** n'est présent sur le serveur FTP.
- Placer sur le serveur FTP vos 2 fichiers modifiés « aastr้า.cfg » et « **votre@MAC.cfg** ».

Tout est maintenant prêt pour le grand test...

VII.6. Test du provisioning du 6757i

Afin de voir plus en détails ce qu'il se passe, nous allons à nouveau utiliser **Wireshark** afin d'observer :

- tout d'abord les **échanges DHCP** entre le serveur et le poste 6757i à sa mise sous tension
 - ensuite, les **fichiers échangés** entre le serveur **FTP** et le poste IP.
- Lancer Wireshark en mode admin, choisir l'**interface physique Ethernet** à analyser et lancer une capture.
- Effectuer un reset usine du 6757i. Observer les messages indiqués sur le poste ; quand son redémarrage est terminé et son enregistrement auprès du serveur Asterisk effectué, arrêter la capture.
- Analyser les paramètres fournis par le serveur DHCP, et vérifier qu'ils correspondent à ce que nous avons configuré dans le serveur DHCP.
- Lister également les fichiers échangés entre le serveur FTP et le poste IP (retrouver les trames concernées).
- Consulter l'@ IP affectée au téléphone et se connecter sur l'interface web du 6757i afin de vérifier les paramètres affectés automatiquement par le provisioning.

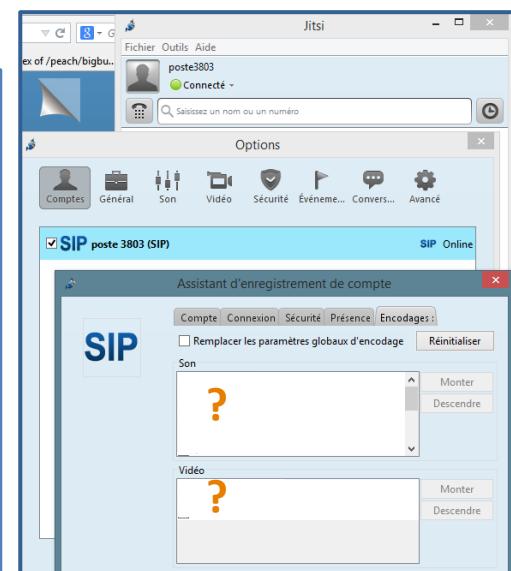


Vous savez maintenant comment fonctionne et comment faire un provisioning (auto-configuration) de postes IP. C'était exactement le même principe qui était utilisé lorsque nous avons mis en service les postes 6757i sur l'IPBX Aastra AXS12, mis à part le fait qu'ici nous allons plus loin, chaque poste recevant sa config via un fichier personnel pour éviter une configuration fastidieuse à la main de chaque poste IP.

VIII. ~~Mise en place d'appels visio (pas le temps)~~

Les postes IP GXV3140 et Jitsi offrent des fonctionnalités d'appel visio. Découvrons comment le configurer avec Asterisk.

- Quels sont les principaux codecs vidéo rencontrés aujourd'hui ? Parmi eux, quel codec est à utiliser préférentiellement aujourd'hui ?
- Dans le menu de config du compte 3803 sur Jitsi (> Outils > Options > poste 3803 > Encodage), quel codec vidéo est activé par défaut ?
- Même question pour le GXV3140 ?



Il n'y a visiblement pas beaucoup de choses à faire sur les téléphones IP pour que la visio soit opérationnelle. Intéressons-nous à Asterisk pour activer les facultés vidéos dans les profils des 2 postes SIP :

- Dans le fichier « sip.conf », ajouter aux contextes 3i02 et 3i03 paramètres suivants :


```
videosupport = yes
allow = h264
allow = h263p
allow = h263
```
- Brancher une webcam sur votre PC physique, et configurez-la dans Jitsi.
- Tenter un appel visio du GXV3140 vers le softphone Jitsi : les appels se font-ils en visio ?
- Commentez la qualité des 2 flux vidéo. Quelles améliorations devrions-nous absolument faire si l'installation de ToIP devait réellement être mise en service ? Quelles sont les bases pour une bonne QoS dans un réseau convergent ?

IX. Fonctionnalités et services sous Asterisk

Afin que notre système puisse être suffisamment convivial et appréciable afin d'être réellement déployé dans une entreprise, nous allons maintenant mettre en service quelques fonctionnalités et services supplémentaires de traitement des appels, tels que par exemple :

- renvoi, transfert, interception d'appel
- groupement d'abonnés,
- enregistrement de conversation,
- mise en attente (parcage), musique d'attente,
- conférences, serveur vocal interactif (SVI),
- queue de distribution automatique des appels,
- messagerie vocale, messagerie unifiée,
- listing des appels, taxation,
- connexions externes (trunk SIP),
- intégration dans une base de données

IX.1. Renvoi d'appels sur non-réponse

Dans le fichier « `/etc/asterisk/extensions.conf` », pour définir le comportement du plan de numérotation, on utilise des lignes de commande avec la structure suivante :

	n° d'abonné	n° de l'instruction (ordre)	Instruction à effectuer	Optionnel : temps avant de passer à l'instruction suivante
exten =>	3i01,	1	Dial(SIP/poste3i01	,10)

Ce qui donne pour notre exemple :

```
exten => 3i01, 1, Dial(SIP/poste3i01,10)
```

Nous allons essayer de mettre en place un renvoi d'appel en cas de non-réponse :

- **Rajouter une ligne d'instruction (ordre = 2) pour l'extension 3i01 afin que les appels soient automatiquement renvoyés vers l'abonné 3i02 au bout de 8s en cas de non-réponse du 3i01.**
- **Composer le 3i01 à partir de la console CLI (« console dial 3i01») et vérifier que le renvoi vers le 3i02 fonctionne correctement lorsque le 3i01 ne répond pas.**

Si l'appel renvoyé du 3i01 vers le 3i02 n'aboutit toujours pas (3i02 également absent), il peut être souhaitable de faire reboucler l'appel de nouveau vers le 3i01, puis 3i02, ... jusqu'à ce qu'un des 2 abonnés décroche (ou que l'appelant raccroche...) ; pour cela, même si ce type de commande n'est pas plébiscité par les informaticiens purs, il est possible d'utiliser l'instruction « `Goto(nom_du_contexte,n°_extension,n°_instruction)` ».

- **Rajouter une ligne d'instruction (ordre = 3) pour l'extension 3i01 afin que les appels soient basculés du 3i01 au 3i02 et réciproquement jusqu'à ce que l'un des 2 réponde.**
- **Vérifier le bon fonctionnement.**
Une fois que vos renvois fonctionnent, supprimer les instructions ajoutées dans cette partie (afin de ne pas être perturbé par la suite par ces renvois...).

Rq1 : afin de ne pas avoir à donner un numéro à toutes les instructions pour une extension, il est possible de **remplacer les n° d'instructions** 2, 3, 4,... par **n** ; Asterisk exécutera alors ces instructions dans l'ordre d'écriture dans extensions.conf ; dans l'exemple suivant, on peut remplacer « 2 » par « n » :

```
exten => 3999, 1, Set(CCHANNEL(language)=fr)
exten => 3999, 2, BackGround(demo-congrats)
```



Rq2 : afin de ne pas d'écrire à chaque fois « exten => xxxx,n,... », il est possible d'écrire « same => n,... », comme dans cet exemple :

```
exten => 3999, 1, Set(CCHANNEL(language)=fr)
same => n, BackGround(demo-congrats)
```

IX.2. Groupement d'abonnés

Comme pour les AXS12 d'Aastra, il est possible de créer un groupement de plusieurs abonnés, comme par exemple un service « hotline » regroupant plusieurs postes de techniciens. Cela est relativement facile à réaliser avec Asterisk grâce à l'instruction Dial : au lieu de faire classiquement « Dial(SIP/abonne1) », il est possible d'écrire « Dial(SIP/abonne1&SIP/abonne2) »

- Créez l'extension 3i10 permettant d'appeler le groupement formé des abonnés 3i01 et 3i02, et tester le bon fonctionnement de ce groupement d'abonnés 3i10.

IX.3. Services accessibles depuis des touches programmées sur les postes IP (pas le temps)

Pour réaliser un transfert d'appel, une conférence à 3, un enregistrement de conversation, une mise en attente d'appel (parcage), ... pendant une communication, les concepteurs des hardphones IP programment généralement des touches en leur affectant un service particulier. L'appui sur une des touches en question génère directement des trames SIP avec le serveur Asterisk pour effectuer le traitement d'appel souhaité.

Nous allons tester quelques-uns de ces services, avec le(s) hardphone(s) IP dont vous disposez :

- Tester un transfert d'appel à partir du 6757i ou du GXV3140 (« Xfer » ou ).
- Tester une conférence à 3 à partir du 6757i ou du GXV3140 (touche « Conf » pour le 6757i ou pour le GXV3140 les touches  pour la sélection des lignes et  pour le lancement de la conférence).

IX.4. Services accessibles par composition de suffixes

Mais les services accessibles par touches préconfigurées sur un poste SIP n'existent pas toujours ou ne sont parfois pas disponibles ; c'est le cas par exemple de certaines versions gratuites de softphones, dont la version gratuite du softphone X-Lite 4.

Il est néanmoins toujours possible de mettre en place ces services, et de faire appel à eux en composant des suffixes ; c'est ce que nous allons voir maintenant. Pour avoir accès à des facilités, il faudra alors composer des **suffixes** à l'aide de combinaisons de touches (*, #, ..), qui informeront le serveur Asterisk de notre souhait d'utiliser le service désiré.

Pour activer ces services sur notre serveur ToIP Asterisk, il suffira de modifier le contenu des instructions Dial dans le plan de numérotation en ajoutant un ou des paramètres tels que par exemple :

- le paramètre « t » autorise la fonction transfert pour l'appelé
- le paramètre « T » autorise la fonction transfert pour l'appelant.
- le paramètre « x » autorise la fonction enregistrement pour l'appelé
- le paramètre « X » autorise la fonction enregistrement pour l'appelant.

Exemple pour le poste 3/01 :

```
exten => 3/01,1,Dial(SIP/poste3/01,10,txX)
```

Nous allons aujourd'hui mettre en service les fonctionnalités suivantes :

- transfert d'appel
- interception d'appel
- enregistrement de conversation
- mise en attente (parcage)
- musique d'attente

IX.4.1. Transfert d'appel

- Rajouter aux extensions des postes 3/01, 3/02 et 3/03 les paramètres autorisant la facilité de transfert pour l'appelant et l'appelé.
- Testons si les transferts fonctionnent : par exemple, depuis le GXV3140, appeler le 6757i ; une fois la communication établie, composer le suffixe # suivi du n° d'abonné du softphone : le transfert a-t-il fonctionné ?

NB : il faut être relativement rapide pour composer “ # ” puis les chiffres du numéro du poste de transfert ; mais il est possible de modifier ce paramètre :

- Consulter le fichier de configuration « /etc/asterisk/features.conf » et consulter le paramètre « transferdigittimeout » pour connaître le temps dont on dispose entre chaque nombre composé.
- Modifier ce timer à 4s.

IX.4.2. ~~Interception d'appel~~ (pas le temps)

Comme pour la gamme A5000 d'Aastra, il est possible de créer des « groupements d'intercom », dénomination Aastra pour désigner un **groupement d'abonnés** ayant autorité d'interception d'appel entre eux. Ainsi, pour éviter de devoir se lever de son bureau pour aller répondre au téléphone d'un collègue absent, il est possible de composer un préfixe afin d'intercepter l'appel depuis son propre poste téléphonique.

Pour activer le service d'interception d'appel, nous allons consulter certains paramètres du fichier de configuration « features.conf » :

- Ouvrir le fichier « /etc/asterisk/features.conf » et rechercher le paramètre « pickupexten » ; quelle est le préfixe à composer pour intercepter un appel ?
Décommenter cette ligne (même si cela fonctionnera quand-même sans cela...).

Pour déclarer les groupements de postes ayant autorité d'interception entre eux, il faut :

- déclarer les abonnés qui feront partie d'un groupement d'appel, dont les appels seront susceptibles d'être interceptés
- définir quels abonnés auront autorité pour intercepter les appels du groupe.

- Dans « sip.conf », ajouter à la fin de la description du contexte [poste3i01] la ligne suivante :

callgroup=1

- Ajouter maintenant à la fin de la description du contexte [poste3i02] la ligne suivante :
pickupgroup=1

- Appeler le 3i01 depuis le softphone, ne pas décrocher, et tenter l'interception de l'appel depuis le GXV3140, en composant le préfixe adéquat. Le test est-il concluant ?

- Essayer maintenant d'intercepter sur le 3i01 un appel du softphone destiné au 3i02. Le test est-il également concluant ? Effectuer les modifications nécessaires pour que ces 2 postes puissent s'intercepter mutuellement.

Rq : il est possible également de faire appartenir un abonné à plusieurs groupes d'appel de son réseau téléphonique :

- Parmi les exemples en commentaire dans le fichier « sip.conf », rechercher la syntaxe qu'il faudrait utiliser pour le contexte [poste3i01] afin d'inclure l'abonné 3i01 dans les groupes 1, 2, 3 et 5. (ne pas faire la modif dans « sip.conf », c'est juste une question théorique...)

IX.4.3. **Enregistrement de conversation** (pas le temps)

Il est possible avec Asterisk d'enregistrer une conversation en cours, avec génération de fichiers audio mémorisant la conversation. Encore un service que l'on peut développer gratuitement avec Asterisk mais souvent soumis à péage (option payante) chez des constructeurs d'IPBX.

- Afin d'activer la facilité d'enregistrement, modifier le contexte [featuremap] du fichier « /etc/asterisk/features.conf » en décommentant la ligne suivante :
automixmon => *3
- En s'aidant du commentaire associé à l'instruction précédente, quelles options faut-il rajouter aux extensions des abonnés afin d'autoriser la facilité d'enregistrement de conversations ? Et quel est le suffixe à composer pour activer l'enregistrement ?

Rq : il faudra être relativement rapide pour composer le suffixe ; mais pour que ce soit plus pratique, nous allons modifier ce paramètre :

- Consulter le fichier de configuration « /etc/asterisk/features.conf » et modifier le paramètre « featuredigittimeout » pour l'augmenter d'1 s.
- Rajouter aux extensions des postes 3i01, 3i02 et 3i03 ces options afin d'autoriser l'enregistrement de conversations.
- Engager une conversation entre le 3i01 et le 3i02, puis composer sur un des 2 postes le suffixe permettant l'enregistrement. En consultant les informations de verbose affichées dans l'interface CLI Asterisk, pouvez-vous dire si l'enregistrement se passe correctement ou non ? Donner le nom du fichier créé contenant les flux audio enregistrés. De quel type est-il ? (wav, mp3, gsm, ... ?)

NB : si l'enregistrement ne fonctionne pas à cause d'un problème de droit d'ouverture de fichiers (voir verbose CLI Asterisk), vérifier que vous avez bien lancé Asterisk en tant qu'administrateur (compte root) et non avec votre profil utilisateur.

Intéressons-nous au fichier audio généré par l'enregistrement de la conversation :

- Vérifier la présence de ce nouveau fichier dans « /var/spool/asterisk/monitor ».

Comme nous ne pouvons brancher un casque sur notre VM pour écouter le contenu du fichier, nous allons l'importer sur notre machine physique en liaison FTP sécurisée (SFTP) :

- Si tel n'est pas le cas déjà, installer sur votre PC physique le client Filezilla, téléchargeable sur <https://filezilla-project.org/> ; lancer Filezilla, et se connecter à la VM sur le port 22 (SFTP et SSH)
- Télécharger le fichier audio de l'enregistrement, puis brancher un casque sur les connecteurs audio de votre PC et vérifier que vous entendez bien la conversation enregistrée.

Rq : la législation impose de demander au préalable une autorisation d'enregistrement à la CNIL et d'informer les usagers d'un éventuel enregistrement de conversation en cours. Donc pour être totalement légal, notre service ne peut pas être utilisé de la sorte, il faudrait rajouter un message d'information diffusé à chaque début de procédure d'enregistrement ou par défaut avant chaque mise en communication, informant qu'il est possible que la conversation soit enregistrée...

Nous allons maintenant essayer de créer une fonction d'enregistrement automatique des appels pour un numéro. Prenons par exemple le cas du groupement « hotline » 3i10 créé au §VIII-2, qui regroupe les abonnés 3i01 et 3i02 : on peut très bien imaginer que leur responsable souhaite pouvoir écouter les conversations des appels reçus afin d'améliorer la qualité de la hotline. On souhaite donc avoir le fonctionnement suivant :

- dès réception d'un appel pour le 3i10, diffusion d'un message informant que la conversation est susceptible d'être enregistrée
 - lancement de l'enregistrement de l'appel dans un fichier de forme « 3i10_date_heure.gsm » stocké dans « /var/spool/asterisk/monitor/ »
 - appel des abonnés membres du groupement.
- Pour réaliser ceci, modifier dans « extensions.conf » les lignes du contexte [plan-num-prive] concernant l'extension 3i10 :
- ```
exten => 3i10,1,Set(CHANNEL(language)=fr)
exten => 3i10,n,Playback(avertissement_recording)
exten => 3i10,n,Set(FILENAME=3i10-${STRFTIME(${EPOCH},,%C%y%m%d%H%M%S)})
exten => 3i10,n,MixMonitor(${FILENAME}.wav,b)
exten => 3i10,n,Dial(SIP/poste3i01&SIP/poste3i02,10)
exten => 3i10,n,Hangup()
```
- Commenter la fonction de ces lignes.
- Pour créer le fichier audio contenant le message « avertissement\_recording.wav », il peut être astucieux d'utiliser la fonction enregistrement activée précédemment : effectuer un appel entre 2 téléphones, lancer l'enregistrement de la conversation, et prononcer le message que vous souhaitez créer ; raccrocher.
- Récupérer le fichier.wav, aux caractéristiques directement exploitables par Asterisk, dans le dossier « /var/spool/asterisk/monitor/ » et le copier dans « /var/lib/asterisk/sounds/fr/ » sous le nom « avertissement\_recording.wav ».
- Composer le 3i10 sur le softphone et vérifier que le comportement est correct :
  - avant que les 2 postes sonnent, diffusion intégrale de l'annonce informant d'un possible enregistrement de la conversation
  - lancement de l'enregistrement de la conversation avant que les postes 3i01 et 3i02 ne sonnent.

Nous allons maintenant essayer d'améliorer la convivialité pour la consultation des messages vocaux, en permettant la consultation à partir d'un navigateur web :

- Installer sur la VM le serveur web « apache2 ».
- Nous allons modifier certains fichiers de configuration afin de rediriger le chemin par défaut vers le dossier « monitor » d'Asterisk, et autoriser son accès à tous :
  - dans « /etc/apache2/sites-available/000-default.conf », mettre en commentaire « DocumentRoot /var/www/html » et rajouter à la place « DocumentRoot /var/spool/asterisk/monitor »
  - dans « /etc/apache2/apache2.conf », rajouter dans les modèles de sécurité par défaut une autorisation d'accès pour le dossier /var/spool/asterisk/monitor :

```
<Directory /var/spool/asterisk/monitor/>
 Options Indexes FollowSymlinks
 AllowOverride None
 Require all granted
</Directory>
```

- Redémarrer le serveur Apache.
- Ouvrir le navigateur Firefox de votre PC physique, et composer l'URL « 192.168.1*i*.10*i* » : visualisez-vous la liste des fichiers audio enregistrés par Asterisk ? Cliquer sur le lien pour écouter la (les) conversation(s) enregistrée(s) précédemment.

Rq : Bien-sûr, il reste à mettre en place toute la structure de contrôle afin que chaque abonné n'ait accès qu'à ses enregistrements. Travailler avec une structure de base de données gérant des profils sera alors bien plus intéressant.

#### IX.4.4. ~~Mise en attente d'appels (parcage), musique d'attente (pas le temps)~~

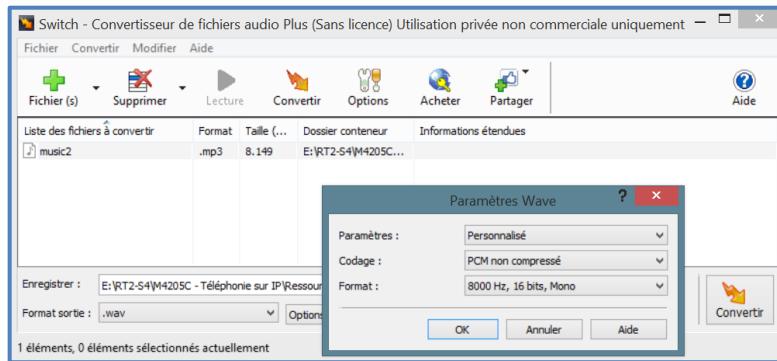
Nous allons mettre en place maintenant des fonctionnalités de de parcage (mise en attente) d'appels :

- Dans le fichier « extensions.conf », rajouter juste après le début du contexte [plan-num-prive] la ligne suivante :  
include => parkedcalls
- Consulter « features.conf » afin de savoir dans quel fichier se configure le parcage d'appels (« ; Asterisk 12 Note – All parking lot configuration is now done in ... ») ; ouvrir ce fichier.
- Quel suffixe faut-il composer pour parquer un appel ?
- Lancer une communication entre le 3*i*01 et le 3*i*02, et composer sur le poste 3*i*01 le suffixe adéquat pour mettre en attente le 3*i*02. Quel numéro de parking l'opératrice renvoie-t-elle ? A quoi peut servir ce numéro ?
- Récupérer l'appel mis en attente.
- Que se passe-t-il lorsqu'on ne récupère pas assez rapidement l'appel (laisser en parcage pendant près d'1 min) ?  
Quel est le nom du paramètre dans le fichier de configuration qui permet de personnaliser ce timer ?
- Expliquer également comment personnaliser le pool des numéros de parking distribués par le serveur.
- Quels renseignements renvoie la commande CLI Asterisk « parking show default » ?

Il est possible de personnaliser la musique d'attente diffusée à l'interlocuteur mis en attente. La musique d'attente diffusée lors de l'attente est issue par défaut de fichiers au format .wav stockés dans le répertoire « /var/lib/asterisk/moh » (« moh » pour « Music On Hold »).

- Visualiser les fichiers présents dans le dossier « /var/lib/asterisk/moh ».

Nous allons placer des fichiers audio personnels de format .wav dans ce répertoire ; pour info, les fichiers, à l'origine en .mp3, ont été convertis par un outil tel que Switch Audio Converter en format .wav avec comme options de codec : PCM – mono - 16 bits/Sa - 8kSa/s :



- Récupérer les fichiers music1.wav, music2.wav et music3.wav sur le serveur web intranet de la salle (dossier « [http://192.168.10.250/annexes/ressources\\_annexes/](http://192.168.10.250/annexes/ressources_annexes/) ») et les placer dans le dossier « /var/lib/asterisk/moh/ ».
- Changer l'extension des fichiers déjà présents dans le dossier moh (les renommer en .wav2 par exemple) afin qu'Asterisk ne les détecte plus. Redémarrer le serveur Asterisk, et réessayer de mettre en attente l'abonné 3i02 ; quel musique ce dernier entend-il alors pour patienter ?  
NB : si vous n'entendez rien, pensez à vérifier les droits alloués aux fichiers audio (744) (*chmod*), et faire en sorte qu'ils appartiennent au même propriétaire (root) (*chown*) et même groupe (root) (*chgrp*).
- Consulter et modifier le contenu du fichier « /etc/asterisk/musiconhold.conf » afin d'entendre les musiques d'attente en ordre aléatoire. Tester à nouveau des mises en attente successives d'abonnés : entendez-vous bien des musiques différentes à chaque fois ?
- Pour « s'amuser », modifier le contenu de votre fichier « musiconhold.conf » afin de permettre aux personnes mises à attente d'appuyer sur « # » pour changer de musique (passage à la suivante) si ils n'aiment pas.

NB : encore un point de législation : si vous diffusez de la musique, il faut être à jour des droits d'auteurs à reverser à la SACEM. Sinon, utilisez uniquement des fichiers musicaux libres de droit.

Remarque : il est possible de diffuser également un flux audio disponible sur Internet, comme une radio par exemple, afin de s'en servir comme musique d'attente. C'est ce que nous allons essayer de faire maintenant :

- Modifier dans « /etc/asterisk/musiconhold.conf » le contexte [default] comme ceci :
  - ajouter les lignes suivantes :
 

```
mode=custom
application=/usr/bin/mpg123 -q -s -mono -r 8000 -f 8192 -@
http://www.alouette.fr/alouette.m3u
```
  - commenter toutes les autres lignes

NB : vous pouvez choisir le flux streaming radio .m3u de votre choix sur [http://www.linuxpedia.fr/doku.php/flux\\_radio](http://www.linuxpedia.fr/doku.php/flux_radio) par exemple.

- Commentez le rôle des différentes options insérées à la commande mpg123.

Testons tout d'abord si le serveur est capable de diffuser le flux radio, et pour tester facilement cela, nous allons créer une extension spécifique :

- Créez dans « extensions.conf » l'extension suivante :

```
exten => 3i11,1,Answer()
 same => n,Musiconhold(100)
 same => n,Hangup()
```
- Tester la réception du flux en composant cette extension (si besoin, recharger complètement Asterisk), puis tester la diffusion de la musique d'attente lors d'une mise en attente : recevez-vous le flux audio live ? Cool, non ??

Rq : ce n'est pas l'objectif de ce module, mais nous pourrions dans le même esprit aller plus loin, en diffusant une vidéo sur les visiophones lors d'une mise en attente, voire diffuser un flux streaming vidéo tv.

## IX.5. ~~Mise en place de conférences (pas le temps)~~

Mettre en place une conférence téléphonique est un service demandé très souvent à un administrateur réseau. Asterisk permet de mettre en place des conférences facilement, chaque utilisateur composant un n° d'appel pour entrer dans l'espace de conférence.

La mise en place de conférence passe par le paramétrage du fichier « /etc/asterisk/conference.conf » et la modification du plan de num « extensions.conf » dans lequel nous utiliserons la fonction ConfBridge() aux particularités suivantes :

- audio haute définition qui peut être mélangé à des fréquences d'échantillonnage allant de 8 kHz à 96kHz
- capacités vidéo : un seul flux vidéo transmis aux conférenciers, mais avec une commutation dynamique des flux vidéo sur la base du participant parlant le plus fort
- système de menu à contrôle dynamique pour les administrateurs de la conférence qui peuvent agir sur les participants, les rendre muets ou non, les expulser...
- personnalisation des conférences en instaurant un mot de passe d'accès, un système d'annonce d'utilisateurs, un compteur d'utilisateurs, mode discours, mode écoute...
- Vérifier dans « conference.conf » que les sections default\_user et default\_bridge sont conformes à ceci :

```
[general]
...
[default_user]
type=user
...
[default_bridge]
type=bridge
```

- Dans « extensions.conf », créer juste avant notre section principale [plan-num-prive] une section [salons-conferences], à inclure dans notre section principale, déclarant une extension 3i50 pour accéder au pont de conférence :

```
...
[salons-conferences]
exten => 3/50,1,NoOp()
same => n,ConfBridge(${EXTEN},default_bridge,default_user)

[plan-num-prive]
include => salons-conferences
...
```

- Tester l'entrée en conférence des 3 postes dans le salon 3/50.

Lors de ce test, nous n'avions qu'un seul type de profil d'utilisateur. Mais il est préférable de créer 2 types de participants, par exemple un profil administrateur, et un profil participant : l'administrateur pourrait avoir la possibilité de rendre muet un participant, d'empêcher la diffusion du son tant que le membre leader de la conférence n'est pas connecté, ...

Il est également possible de définir différents profils de ponts (bridges) de conférences, utilisables pour définir des options telles que le nombre max de participants, la possibilité d'enregistrer la conférence et la localisation de l'enregistrement, ...

Nous allons commencer l'amélioration de nos salons de conférences par la mise en place d'un code PIN pour accéder à la conférence :

- Modifier l'extension 3/50 afin de demander un mot de passe (PIN : Personal Identification Number) pour rejoindre la conférence :

```
...
[salons-conferences]
exten => 3/50,1,NoOp()
same => n,Set(CONFBRIDGE(user,pin)=1234)
same => n,ConfBridge(${EXTEN},default_bridge,default_user)
...
```

- Retenter l'entrée en conférence des 3 postes dans le salon 3/50, en vérifiant bien que le mot de passe est demandé (puis # pour valider...).

Dans certains cas, il est utile de créer une conférence où tous les participants peuvent se joindre à la conférence avant l'heure de départ et attendre dans une salle d'attente jusqu'à ce que le leader de la conférence les ait rejoints. Nous pouvons activer cette fonctionnalité en rajoutant des options et en identifiant un (des) utilisateur(s) comme leader(s) de la conférence ; quand le leader se joint à la conférence, tous les participants en attente vont être basculés dans la salle de conférence et pourront tous se parler normalement.

- Modifier la section [salons-conferences] comme ci-dessous et expliquer le rôle des instructions :

```
...
[salons-conferences]
; participants standards :
exten => 3/50,1,NoOp()
same => n,Set(CONFBRIDGE(user,pin)=1234)
same => n,Set(CONFBRIDGE(user,admin)=no)
same => n,Set(CONFBRIDGE(user,wait_marked)=yes)
```

```

same => n,Set(CONFBRIDGE(user,end_marked)=yes)
same => n,Goto(conference,1)

; participants leaders (admin)
exten => 3i51,1,NoOp()
same => n,Set(CONFBRIDGE(user,pin)=9876)
same => n,Set(CONFBRIDGE(user,admin)=yes)
same => n,Set(CONFBRIDGE(user,marked)=yes)
same => n,Goto(conference,1)

; même pont de conférence pour tous :
exten => conference,1,NoOp()
same => n,ConfBridge(primary,default_bridge,default_user)

```

- Retenter une conférence, en supposant que le 6757i et le softphone sont des participants standards et le GXV3140 est maintenant le leader de la conférence. Quand la conférence commence-t-elle ? Que se passe-t-il pour les participants lorsque le leader de la conférence se retire ?

Une dernière manipulation afin d'améliorer encore nos salons de conférences : nous allons utiliser des fonctionnalités de ConfBridge() pour que les participants puissent utiliser des séquences DTMF de leur téléphone pour ajuster leurs paramètres audio (niveau micro, écouteur...) :

- Créer dans « confbridge.conf » une section [menu\_ctrl\_vol] pour permettre à chaque participant de contrôler le volume son d'écoute et de parole :
- ```

[menu_ctrl_vol]
type=menu
*5=toggle_mute
1=increase_listening_volume
4=decrease_listening_volume
7=reset_listening_volume
3=increase_talking_volume
6=decrease_talking_volume
9=reset_talking_volume

```
- Modifier l'instruction ConfBridge de « extensions.conf » pour prendre en compte cette section de contrôle de volume :
- ```

; même pont de conférence pour tous :
exten => conference,1,NoOp()
same => n,ConfBridge(primary,default_bridge,default_user,menu_ctrl_vol)

```
- Relancer une conférence ; après avoir logé les postes, composer sur l'un d'eux \*5 : entendez-vous un message ? Que dit-il ? Composer à nouveau \*5 : que se passe-t-il ?

Il est également possible de faire une vidéoconférence. Les restrictions d'usage de la vidéo pour les conférences avec Asterisk sont les suivantes :

- tous les participants doivent utiliser le même codec vidéo, car Asterisk ne réalise aucun transcodage.
- Il n'y a pas de fonctionnalité de multiplexage vidéo dans Asterisk ; un seul flux vidéo peut être diffusé à la fois à un participant.

Asterisk possède 4 modes de pont de conférence vidéo :

- « follow\_talker » : le flux vidéo diffusé sur tous les téléphones est celui du participant en train de parler (ayant le plus haut niveau sonore de parole)
- « last\_marked » : flux du dernier arrivé dans la conférence
- « first\_marked » : flux du premier arrivé dans la conférence
- « none » : pas de vidéo

- Dans « confbridge.conf », préciser dans la section de pont par défaut le mode de vidéo souhaité :

```
[default_bridge]
type=bridge
video_mode=follow_talker ; par exemple...
```

- Même si ici il sera difficile avec seulement 2 postes vidéo de tester une vidéoconférence (et le basculement du flux vidéo lors du changement d'interlocuteur !), tester quand-même une conférence vidéo avec le GXV3140 et le softphone.

## IX.6. Distribution automatique des appels (pas le temps)

La « distribution automatique d'appels » (**ACD : Automatic Call Distribution**), ou la « mise en attente d'appels », fournit un moyen pour un serveur de téléphonie de placer les appels entrants dans une file d'attente (queue). **Chaque appel arrivant destiné à un service particulier est placé dans une queue, avec affectation d'un rang.** Il y aura autant de queues à gérer que de services différents existant dans l'entreprise.

La fonction de distribution automatique des appels détermine l'ordre dans lequel chaque appel doit être livré à un agent disponible (typiquement : **premier entré, premier sorti**). Lorsqu'un agent est disponible, l'appelant de plus haut rang dans la file d'attente est livré à cet agent. Si vous avez déjà appelé une organisation et entendu "tous nos représentants sont malheureusement occupés", « votre temps d'attente est de... » ou « merci de renouveler votre appel... », c'est que vous avez eu affaire à un ACD.

Avantages d'avoir un ACD pour une organisation :

- les appelants n'ont pas à recomposer en permanence le n° jusqu'à avoir la chance de joindre quelqu'un de l'organisation ou du service souhaité
- meilleur service auprès des clients, meilleure gestion des situations où il y a davantage d'appelants que d'agents disponibles.

Commençons par créer une queue ACD simple, qui accepte les appels et essaie de les transférer à un des membres du service souhaité :

- Créer deux queues dans le fichier « /etc/asterisk/queues.conf », appelées « ventes » et « support » :

[general]

```

persistentmembers=yes ; store each dynamic member in the Asterisk database
autofill=yes ; distribute all waiting callers to available members
monitor-type=MixMonitor ; specify the application to use to record queue member
 ; conversations
shared_lastcall=yes ; respect the wrapup time for members logged into more
 ; than one queue

```

[StandardQueue](!) ; template to provide common features

```

musicclass=default ; play [default] music
strategy=rrmemory ; use the Round Robin Memory strategy
joinempty=no ; do not join the queue when no members available
leavewhenempty=yes ; leave the queue when no members available
ringinuse=no ; don't ring members when already InUse (prevents
 ; multiple calls to an agent)

```

[ventes](StandardQueue) ; create the sales queue using the parameters in the
 ; StandardQueue template

[support](StandardQueue) ; create the support queue using the parameters in the
 ; StandardQueue template

- Dans l'interface CLI d'Asterisk, utiliser la commande « **queue show** » pour vérifier que les 2 queues « ventes » et « support » existent bien.

Nous allons maintenant permettre l'ajout d'agents dans les différentes queues pour répondre aux appels distribués par l'ACD. Nous n'allons pas configurer de façon fixe la liste des agents membres d'une queue, car alors des appels pourraient être renvoyés vers eux même lorsqu'ils ne sont pas à leur poste de travail. Il faut donc **permettre aux agents d'être capables de se loger et déloger seuls**. Pour cela, nous disposons des instructions :

- AddQueueMember()
- RemoveQueueMember()

- Ajouter dans le plan de numérotation une section « gestion-queues-agents » pour permettre aux agents de gérer leur appartenance aux queues :

...

[gestion-queues-agents]

```
exten => *21,1,Verbose(2,Loggin agent dans la queue "Ventes")
same => n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
same => n,AddQueueMember(ventes,${MemberChannel})
same => n,Verbose(1,${AQMSTATUS}) ; ADDED, MEMBERALREADY, NOSUCHQUEUE
same => n,Playback(agent-loginok)
same => n,Hangup()
```

```
exten => *23,1,Verbose(2,Loggout agent de la queue "Ventes")
same => n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
same => n,RemoveQueueMember(ventes,${MemberChannel})
same => n,Verbose(1,${RQMSTATUS}); REMOVED, NOTINQUEUE, NOSUCHQUEUE
same => n,Playback(agent-loggedoff)
same => n,Hangup()
```

```
exten => *54,1,Verbose(2,Loggin agent dans la queue "Support")
same => n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
same => n,AddQueueMember(support,${MemberChannel})
same => n,Verbose(1,${AQMSTATUS}) ; ADDED, MEMBERALREADY, NOSUCHQUEUE
same => n,Playback(agent-loginok)
same => n,Hangup()
```

```
exten => *56,1,Verbose(2,Loggout agent de la queue "Support")
same => n,Set(MemberChannel=${CHANNEL(channeltype)}/${CHANNEL(peername)})
same => n,RemoveQueueMember(support,${MemberChannel})
same => n,Verbose(1,${RQMSTATUS}); REMOVED, NOTINQUEUE, NOSUCHQUEUE
same => n,Playback(agent-loggedoff)
same => n,Hangup()
```

[plan-num-prive]

include => gestion-queues-agents

...

Il ne nous reste plus qu'à créer des extensions pour que les clients puissent joindre les services « ventes » et « support », et diriger ces appels vers les queues respectives :

- Ajouter dans le plan de numérotation une section « acces-queues-clients » configurant les extensions 3i21 et 3i22 pour accéder aux queues des 2 services :

```
...
[acces-queues-clients]
exten => 3i21,1,Verbose(2,${CALLERID(all)}) entre dans la queue « Ventes »)
same => n,Queue(ventes)
same => n,HangUp()
```

```
exten => 3i22,1,Verbose(2,${CALLERID(all)}) entre dans la queue « Support »)
same => n,Queue(support)
same => n,HangUp()
```

```
[plan-num-prive]
include => acces-queues-clients
...
```

- Incrire le 6757i comme agent du service « Ventes » ; appeler ce service depuis le softphone et engager la communication ; appeler également ce service depuis le GXV3140 : l'appel est-il mis en attente ? Une fois la communication avec le softphone terminée, l'appel du GXV3140 est-il correctement présenté au 6757i ?

Nous n'irons pas plus loin aujourd'hui dans la gestion de cet ACD, mais sachez qu'il est possible d'insérer un agent dans plusieurs queues s'il est multi-compétent, de gérer les priorités de queues, d'insérer des messages diffusés lors de l'attente, ...

## IX.7. Messagerie vocale, messagerie unifiée

Même si aujourd’hui les usages d’e-mail (courriel) et la messagerie instantanée sont devenus prépondérants, la messagerie vocale reste une composante populaire essentielle de n’importe quel serveur de téléphonie, pour laisser un message à un interlocuteur distant.

Asterisk dispose d’un système de messagerie vocale avec les caractéristiques suivantes :

- boîtes de messagerie vocale protégées par mot de passe
- nombre illimité de boîtes de messagerie vocale
- chaque boîte vocale est constituée de dossiers pour une meilleure organisation de la messagerie vocale
- messages d’accueil personnalisables et différenciables pour les états « occupé » et « non disponible »
- **notification par e-mail de la présence d’un message vocal** (« **messagerie unifiée** »), avec éventuellement le fichier audio joint
- indication de message en attente (lumière clignotante ou tonalité saccadée) sur les téléphones

### IX.7.1. Messagerie vocale classique

La configuration de la messagerie vocale se fait principalement dans le fichier « /etc/asterisk/voicemail.conf ».

- Créer dans la section « default » les boîtes vocales des 3 postes, sans préciser pour l’instant d’@ mail :

```
[default]
; extension_boite_vocale => mot_de_passe[,prénom nom[,adresse_mail
 [,2nde_adresse_mail[,options[|options]]]]]
3i01 => 4625,Katniss Everdeen
3i02 => 3159,Peeta Mellark
3i03 => 8790,Gale Hawthorne
```

Pour intégrer maintenant l’utilisation de la messagerie vocale dans le plan de numérotation, nous allons utiliser l’instruction VoiceMail(boîte\_messagerie,options) :

- Dans le plan de numérotation, modifier le comportement du serveur lors d’un appel vers l’abonné 3i01 comme ceci :

```
exten => 3i01,1,NoOp()
same => n,Dial(SIP/poste3i01,10,tTxX)
same => n,GoToIf($"${DIALSTATUS}" = "BUSY"?busy:unavail)
same => n(unavail),VoiceMail(3i01@default,u)
same => n,Hangup()
same => n(busy),VoiceMail(3i01@default,b)
same => n,Hangup()
```

- Faire de même pour les abonnés 3i02 et 3i03.
- Que signifie la différence entre « busy » et « unavailable » ?

- Tester la messagerie en occupant le 6757i avec le softphone, et en appelant le 6757i depuis le GXV3140 : basculez-vous vers la messagerie ?

Tester les différences de comportement de la messagerie lorsque le 6757i refuse l'appel (appui sur Drop ou Ignore) ou ne répond simplement pas.

Il nous reste maintenant à créer une extension afin que chaque abonné puisse consulter ses messages déposés sur sa boîte vocale :

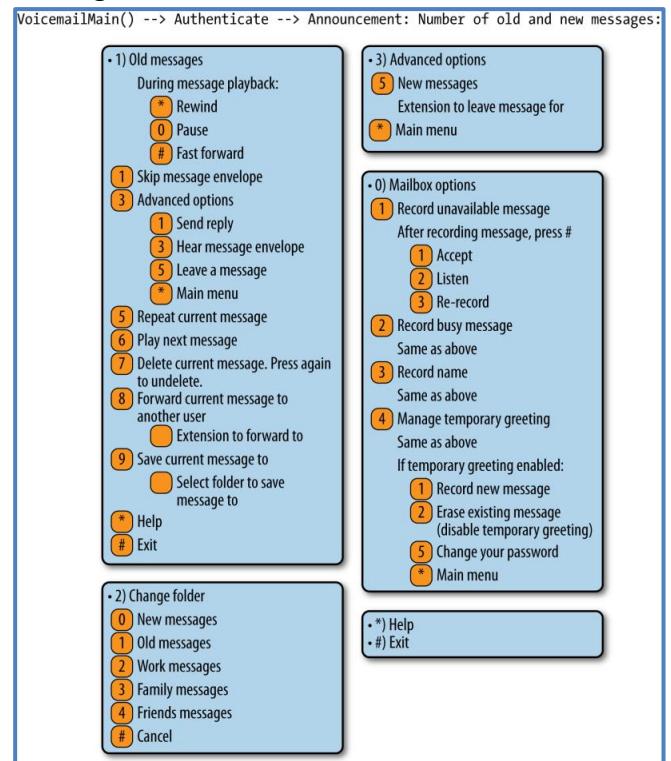
- Ajouter au plan de numérotation une extension \*98 qui permettra de consulter sa messagerie :

exten => \*98,1,NoOp()

same => n,VoiceMailMain()

- Tester la consultation de messages pour l'abonné 3/01. Vous devez entendre les messages précédemment laissés.

*Exemple d'arborescence de messagerie :*



Afin que l'heure indiquée dans les messages soit correcte, il nous faut synchroniser la machine Mail-Server avec le serveur de temps ntp installé sur VM Asterisk-Server ; pour cela, nous allons installer le client ntpdate :

- Installer le paquet « ntpdate » et modifier le contenu du fichier « /etc/default/ntpdate » afin que la machine se synchronise sur le serveur NTP d'Asterisk-Server.

NB : ntpdate est lancé lors de l'activation d'une interface réseau ; donc si vous souhaitez resynchroniser l'horloge de la VM, désactiver puis réactiver l'interface eth0 (ou redémarrer la VM).

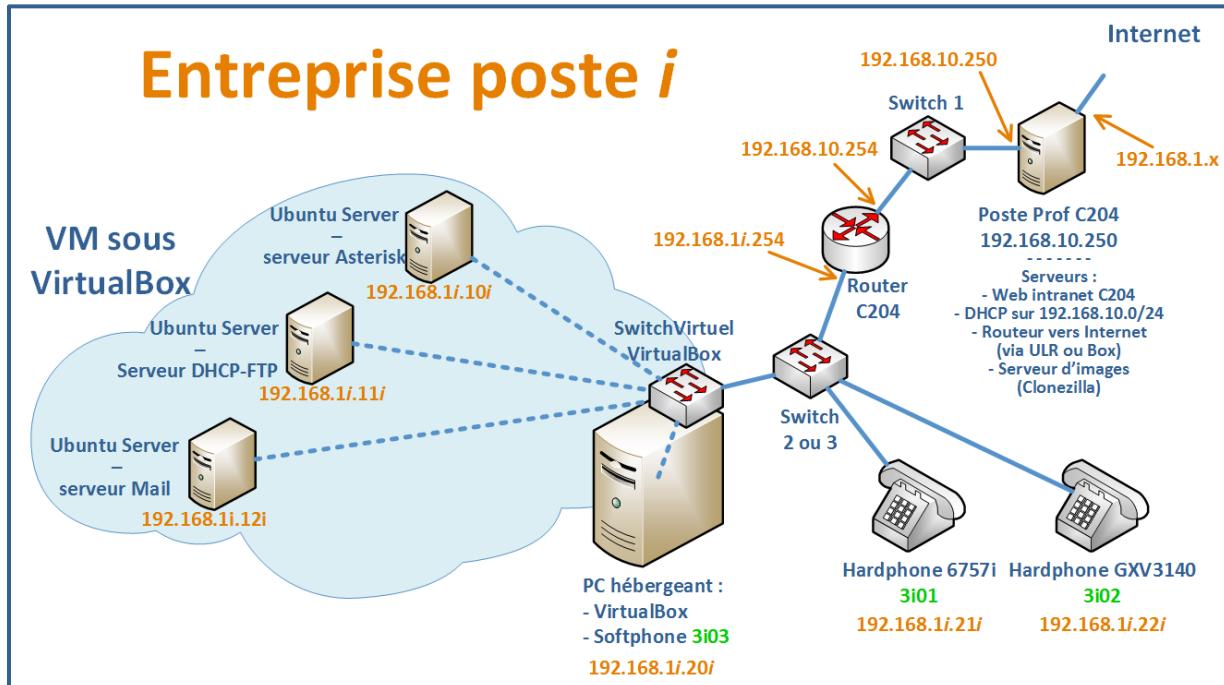
Notre messagerie vocale traditionnelle est maintenant opérationnelle. Néanmoins il est préférable aujourd'hui de mettre également en place un service de **messagerie unifiée**, en complément de la messagerie vocale traditionnelle.

- Rappeler ce qu'est la « messagerie unifiée ».

Nous allons donc maintenant commencer par installer un serveur de messagerie afin de créer des comptes de messagerie pour les abonnés téléphoniques, puis nous configurerons Asterisk afin de paramétrier la messagerie unifiée.

### IX.7.2. Configuration de la VM dédiée à la messagerie

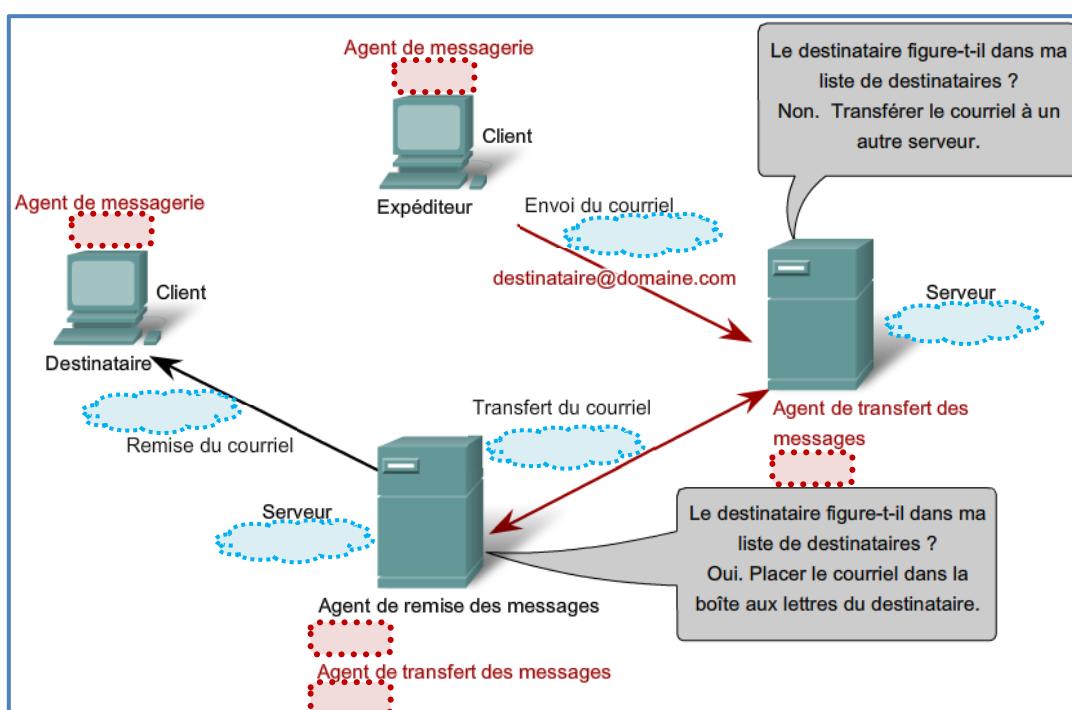
Pour mettre en place des solutions de messagerie unifiée (partie suivante §IX.7.3), nous devons au préalable configurer la VM « **M4205-Mail-Server** » devant accueillir un serveur mail, qui hébergera les services mails ainsi que les boîtes mail des abonnés de l'entreprise.



- Comme pour le serveur Asterisk, configurer la carte réseau eth0 de la VM Mail-Server de façon permanente, avec les paramètres adéquats (cf schéma).

Quelques rappels tout d'abord sur le transport de messages mail :

- Que signifient les acronymes **MUA**, **MTA**, **MDA** ?



- Quel est le nom du protocole classique utilisé pour l'envoi de mails ? Quel est le n° de port réservé à ce service ?
- De même, quels protocoles utilisés pour la livraison des messages mails connaissez-vous ? Quels sont leurs n° de ports réservés, en version classique et sécurisée ?
- Dans le schéma de la page précédente, indiquer dans les encadrés s'il s'agit d'un MUA, MDA ou MTA ; indiquer également dans les bulles le nom d'un protocole adéquat :

Nous allons installer sur la VM Mail-Server les services suivants :

- **Postfix** pour envoyer des mails
- **Dovecot** pour délivrer, remettre des mails à un abonné
- **Roundcube** pour permettre un accès Web aux serveurs de messagerie (WebMail)

### IX.7.3. Installation basique de Postfix

Nous allons installer le serveur **SMTP Postfix** avec, pour éviter que des spameurs utilisent le serveur pour envoyer du spam, une **authentification SMTP-AUTH** ainsi qu'un **chiffrement SSL/TLS**.

Commençons tout d'abord par installer Postfix sans le sécuriser :

- Quel protocole est utilisé par Postfix pour envoyer les mails ?
- Modifier le nom de la machine en « mail-server.postei.c204.rtlr » (dans « /etc/hostname »)
- Ajouter l'entrée suivante dans « /etc/hosts » :  
192.168.1.i.12i mail-server.postei.c204.rtlr  
(postei.c204.rtlr sera le nom de domaine fictif de votre groupe de travail)
- Installer le paquet « ssh ».
- Installer ensuite le paquet « postfix » avec les paramètres suivants :
  - type de serveur de messagerie : « site internet » (*pour avoir config de base*)
  - nom de courrier : « postei.c204.rtlr »
- Créer un utilisateur « admin » avec la commande « adduser » (mot de passe « rtrt ») ; cet utilisateur servira à faire rediriger le trafic destiné au compte « root » ou « postmaster » du serveur mail.
- Exécuter « dpkg-reconfigure postfix » pour continuer la configuration :
  - destinataire des courriels de « root » et « postmaster » : « admin »
  - autres destinations pour lesquelles le courrier sera accepté :  
postei.c204.rtlr, posteit, localhost.localdomain, localhost
  - forcer des mises à jour synchronisées de la file d'attente des courriels : non
  - réseaux internes: 127.0.0.0/8 [::ffff:127.0.0.0]/104 [::1]/128 192.168.1.i.0/24
  - taille max des boîtes aux lettres : 0 (valeur par défaut 51.200.000)
  - caractère d'extension des adresses locales : +
  - protocoles internet à utiliser : tous

Postfix prend en charge les **2 principaux formats de boîtes à lettres** existant : **mbox** ou **Maildir**. Entre les 2 types de format de messagerie supportés, nous allons choisir maildir, qui propose davantage de fonctionnalités pour la gestion des fichiers de messagerie (1 dossier par utilisateur, et 1 fichier par mail...) ; nous allons le préciser dans Postfix :

- Dans « /etc/postfix/main.cf », ajouter la ligne suivante pour créer un dossier Maildir dans le dossier home de chaque utilisateur :

```
home_mailbox = Maildir/
```

Cela placera les nouveaux courriels dans « /home/nom\_utilisateur/Maildir ».

#### IX.7.4. Sécurisation du serveur Postfix

Sécurisons maintenant notre serveur Postfix avec **authentification SMTP-AUTH** (pour éviter que des hackers utilisent le serveur pour envoyer des spams), ainsi qu'un **chiffrement SSL/TLS**. SMTP-AUTH permet à un client de s'identifier par le mécanisme d'authentification **SASL** (**Simple Authentication and Security Layer**) : couche d'authentification et de sécurité simple). **Transport Layer Security (TLS)** devrait être utilisé afin de chiffrer le processus d'authentification. Une fois authentifié, le serveur SMTP autorisera le client à relayer les mails.

Modifions donc le contenu de « /etc/postfix/main.cf » afin de configurer Postfix pour SMTP-AUTH en utilisant SASL (Dovecot SASL) :

- Ajouter (ou modifier les lignes correspondantes) dans « /etc/postfix/main.cf » :  

```
smtpd_sasl_type = dovecot
smtpd_sasl_path = private/auth (c'est un chemin relatif au
 répertoire de la file d'attente (queue) de Postfix, qui sera utilisé dans un fichier
 installé ultérieurement (10-master.conf)...)
smtpd_sasl_local_domain =
smtpd_sasl_auth_enable = yes
smtpd_sasl_security_options = noanonymous
broken_sasl_auth_clients = yes
smtpd_recipient_restrictions =
 permit_sasl_authenticated,permit_mynetworks,reject_unauth_destination
```

Nous allons générer un certificat numérique pour TLS. Les utilisateurs qui voudront se connecter au serveur de messagerie via **TLS** devront reconnaître le certificat utilisé pour TLS.

Un certificat est une méthode utilisée pour distribuer une clé publique et d'autres informations à propos d'un serveur et de l'organisation responsable de ce serveur. Les certificats peuvent être signés numériquement par une **autorité de certification CA** (**Certification Authority**), qui est un tiers de confiance attestant de la véracité des informations contenues dans le certificat. Dans la plupart des cas, pour configurer un serveur sécurisé utilisant le chiffrement par clé publique, on envoie notre demande de certificat (avec notre clé publique) à la CA, qui vérifie notre identité et la demande de certificat, puis nous renvoie un certificat pour notre serveur sécurisé. Une alternative consiste à créer son propre **certificat auto-signé** (NB : les certificats auto-signés ne devraient pas être utilisés dans la plupart des environnements de production). Mais ici, c'est ce que nous allons quand-même faire : générer notre propre **certificat auto-signé pour TLS**, que les utilisateurs pourront accepter et installer.

### Génération d'une demande de signature de certificat (Certificate Signing Request ou CSR) :

Que nous ayons obtenu notre certificat d'une autorité de certification, ou qu'il s'agisse d'un certificat auto-signé, la première étape consiste à **générer une clé**. Lorsqu'un certificat est utilisé par des démons tels que Apache, Postfix, Dovecot,..., avoir une clé sans phrase de passe est souvent utile et pratique : ne pas avoir de phrase de passe permet le démarrage automatique de ces démons (pas besoin de saisir cette phrase à chaque fois). Mais il faut néanmoins être conscient que cela peut constituer une faille de sécurité pouvant mettre en danger tout le serveur.

- Pour **générer une clé**, exécutez la commande suivante dans un terminal :

```
openssl genrsa -des3 -rand /etc/hosts -out ma_cle_smtpd.key 2048
```

Entrer « rtrt » comme phrase de passe (4 caractères minimum, avec normalement minuscules, majuscules, chiffres ou signes de ponctuation...)

Re-saisir la phrase de passe pour vérification. La clé du serveur est alors générée et stockée dans le fichier `ma_cle_smtpd.key`.

- A partir de la clé générée précédemment, créer maintenant la clé non sécurisée (sans phrase de passe) (c'est cette clé que nous utiliserons), et échanger le nom des clés :

```
openssl rsa -in ma_cle_smtpd.key -out ma_cle_smtpd.key.insecure
```

```
mv ma_cle_smtpd.key ma_cle_smtpd.key.secure
```

```
mv ma_cle_smtpd.key.insecure ma_cle_smtpd.key
```

Il vous sera demandé la phrase de passe définie dans `ma_cle_smtpd.key` (pour pouvoir l'« extraire »...).

La clé non sécurisée se nomme maintenant « `ma_cle_smtpd.key` » et nous pourrons utiliser ce fichier pour générer le CSR sans saisir de mot de passe.

- Pour créer la CSR (demande de signature de certificat), lancer la commande suivante dans un terminal :

```
openssl req -new -key ma_cle_smtpd.key -out ma_cle_smtpd.csr
```

Il ne nous sera demandé aucune phrase de passe, car il n'y en a plus dans cette clé.

Entrer le nom de l'entreprise, le nom du site, courriel, ... ; notre CSR sera créé avec toutes ces infos renseignées et conservé dans le fichier « `ma_cle_smtpd.csr` » :

Country name : FR

State ou Province : Charente

Locality : La Rochelle

Organization Name : IUT

Organization Unit Name : RT

Common name (FQDN) : postei.c204.rtlr

Email : admin @postei.c204.rtlr

### Création d'un certificat auto-signé (Self-Signed Certificate ou SSC) :

Une fois cette demande de signature de certificat générée, il serait possible de soumettre ce fichier CSR à une autorité de certification (CA) pour traitement. La CA vérifierait les informations contenues dans ce fichier CSR et émettrait une validation, une signature de ce certificat. Mais ici, comme nous ne travaillons pas dans un environnement de production, nous allons ici créer notre propre certificat auto-signé en utilisant ce CSR :

- Pour créer le certificat auto-signé (.crt), entrer la commande suivante :

```
openssl x509 -req -days 365 -in ma_cle_smtpd.csr -signkey
ma_cle_smtpd.key -out ma_cle_smtpd.crt
```

Ici, encore une fois aucune phrase de passe ne vous sera demandée car nous travaillons avec la clé générée sans phrase de passe.

Notre certificat auto-signé sera créé et stocké dans le fichier « ma\_cle\_smtpd.crt ».

#### Installation du certificat :

- Placer le fichier de clé « ma\_cle\_smtpd.key » et le fichier de certificat « ma\_cle\_smtpd.crt » (à défaut du fichier de certificat émis par l'autorité de certification si nous nous étions adressés à une CA) dans le bon dossier :

```
cp ma_cle_smtpd.crt /etc/ssl/certs
cp ma_cle_smtpd.key /etc/ssl/private
```

Il nous suffira maintenant de simplement **configurer les applications** ayant la capacité d'utiliser le chiffrement par clé publique (ex : Apache en HTTPS, Dovecot en IMAPS et POP3S,...) pour qu'elles utilisent les **fichiers de certificat et de clé** que nous venons de créer.

#### Configuration des fichiers de certificat et clé TLS dans Postfix :

Nous allons fournir à Postfix les fichiers de certificat et de clé à utiliser pour l'encryptage TLS des e-mails entrants et sortants :

- Ajouter (ou modifier les lignes correspondantes) dans « /etc/postfix/main.cf » :

```
smtp_tls_security_level = may
smtpd_tls_security_level = may
smtp_tls_note_starttls_offer = yes
smtpd_tls_key_file = /etc/ssl/private/ma_cle_smtpd.key
smtpd_tls_cert_file = /etc/ssl/certs/ma_cle_smtpd.crt
smtpd_tls_loglevel = 1
smtpd_tls_received_header = yes
myhostname = mail-server.postei.c204.rtlr
```

Postfix supporte maintenant SMTP-AUTH, basé sur SASL. Toutefois, il est encore nécessaire de mettre en place l'authentification SASL avant de pouvoir utiliser le protocole SMTP-AUTH. Plusieurs implémentations de SASL existent ; nous allons aujourd'hui utiliser une d'elles appelée Dovecot SASL, implémentées dans des paquets de Dovecot, qui est un serveur MDA, dont nous aurons justement besoin d'installer après. Alors autant nous servir de ses fonctionnalités SASL dès maintenant.

## IX.7.5. Installation de Dovecot SASL

Postfix prend en charge deux implémentations SASL : Cyrus SASL et Dovecot SASL, et c'est cette dernière que nous allons choisir. Nous commencerons par installer les paquets de base de Dovecot, et configurerons SASL :

- Installer le paquet de base « dovecot-core » ; choisir de créer un certificat auto-signé, et préciser comme nom d'hôte « mail-server.poste1.c204.rtlr ».
- Vérifier que Dovecot a correctement démarré avec la commande :

```
ps aux | grep dovecot
```
- Afin d'activer l'implémentation Dovecot SASL, dans « /etc/dovecot/conf.d/10-master.conf », modifier la partie concernant le « service auth » comme ceci :

```
service auth {
 # auth_socket_path points to this userdb socket by default. It's typically
 # used by dovecot-lda, dovecadm, possibly imap process, etc. Its default
 # permissions make it readable only by root, but you may need to relax these
 # permissions. Users that have access to this socket are able to get a list
 # of all usernames and get results of everyone's userdb lookups.
 unix_listener auth-userdb {
 mode = 0666
 user =
 group =
 }

 # Postfix smtp-auth
 unix_listener /var/spool/postfix/private/auth {
 mode = 0666
 user = postfix
 group = postfix
 }
}
```
- Afin de permettre aux clients Outlook d'utiliser le protocole SMTP-AUTH, dans la section authentication mechanisms de /etc/dovecot/conf.d/10-auth.conf, changez la ligne « auth\_mechanisms = plain » en ceci :

```
auth_mechanisms = plain login
```
- Redémarrer Dovecot avec la commande :

```
service dovecot restart
```

(ou « service dovecot start » si il n'était pas en activité...)

Testons le fonctionnement du service d'authentification :

- Vérifier que le port SMTP est ouvert :  

```
lsof -i -P
```

(pour voir l'ouverture des ports, notamment SMTP)
- Afin de vérifier que SMTP-AUTH et TLS fonctionnent correctement, exécuter :  

```
telnet localhost 25
```

Après avoir établi la connexion avec postfix, entrer :  

```
ehlo localhost
```

L'authentification est fonctionnelle si les lignes suivantes (entre autres) s'affichent :

```
250-STARTTLS
250-AUTH PLAIN LOGIN
250-AUTH=PLAIN LOGIN
250-8BITMIME
```

Entrer « quit » pour se déconnecter.

- Faire le même test en remplaçant cette fois-ci « localhost » par « mail-server.poste8.c204.rtlr ».
- Autre amélioration : si on souhaite activer le port 587 pour se connecter de manière sécurisé par SMTP avec chiffrement du trafic avec n'importe quel client mail, il nous faut décommenter les lignes suivantes dans le fichier « /etc/postfix/master.cf » :

```
submission inet n - - - - smtpd
-o smtpd_tls_security_level=encrypt
-o smtpd_sasl_auth_enable=yes
-o smtpd_client_restrictions=permit_sasl_authenticated,reject
```
- Redémarrer postfix ; revérifier que les ports SMTP classique et sécurisé sont ouverts :  
`lsof -i -P`
- Quelles modifications devrait-on faire si on ne voulait exploiter **que** le port 587 (et pas le 25) ? (NB : c'est juste une question « théorique », nous n'allons pas le faire ici.)

## IX.7.6. Installation du serveur IMAP de Dovecot

Le paquet « dovecot-imapd » contient les éléments nécessaires pour installer le service **MDA IMAP** de Dovecot :

- Installer le paquet « dovecot-imapd ».
- Ajoutez le protocole IMAP à la liste des protocoles activés dans le fichier « /etc/dovecot/dovecot.conf » grâce à l'ajout de la ligne suivante :  
`protocols = imap`

Comme pour le serveur MTA Postfix, il faut préciser au serveur MDA le type de boîte de messagerie (mbox ou Maildir), et bien-sûr prendre le même type de boîte mail :

- Modifier dans « /etc/dovecot/conf.d/10-mail.conf » la ligne suivante :  
`mail_location = mbox:~/mail:INBOX=/var/spool/mail/%u`  
en :  
`mail_location = maildir:~/Maildir`

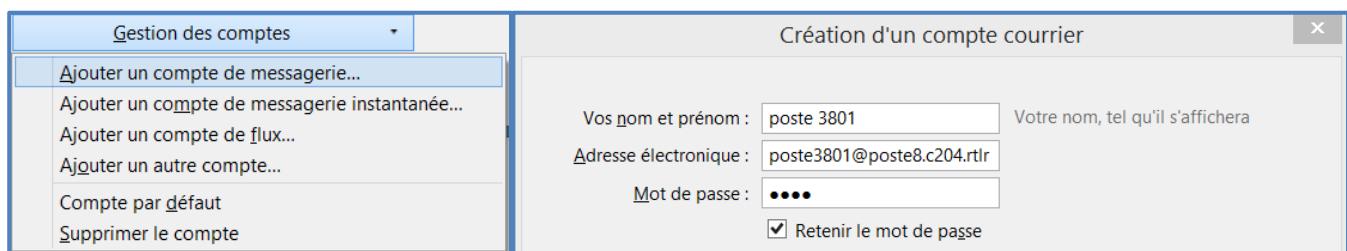
Il n'y a rien de plus à configurer sur ce serveur, les fonctionnalités de sécurité sont déjà préconfigurées depuis l'installation et la configuration de « dovecot-core ».

- Redémarrer dovecot puis revérifier que les ports IMAP classique et sécurisé sont ouverts :  
`lsof -i -P`

### IX.7.7. Crédation des comptes de messagerie et tests

Nous allons tester la messagerie sur les 3 postes de base : 3i01, 3i02 et 3i03.

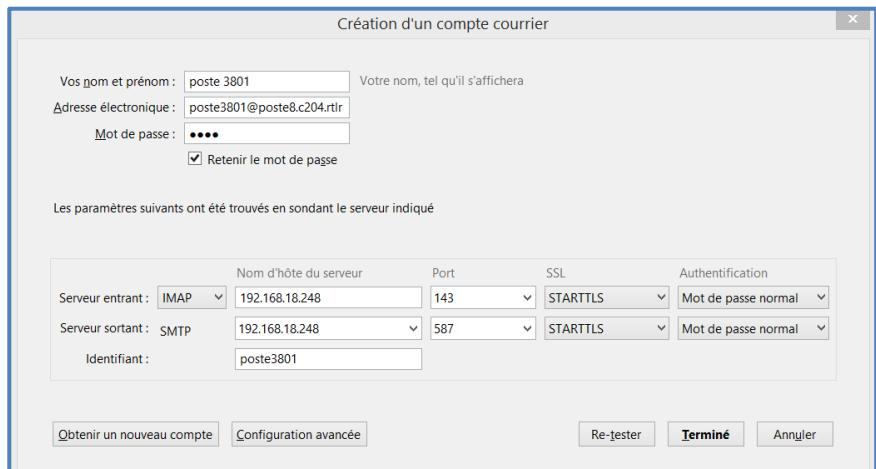
- Créer les utilisateurs « poste3i01 », « poste3i02 » et « poste3i03 » avec la commande « adduser » (mot de passe « rtrt »). Une boîte mail sera alors disponible pour eux sur ce serveur.
- Installer sur le PC physique le **client de messagerie (MUA) Thunderbird** (<https://www.mozilla.org/fr/thunderbird/>)
- Une fois installé, créer le premier compte de messagerie pour le poste 3i01 à partir du menu « Outils > Paramètres des comptes > Gestion des comptes > Ajouter un compte de messagerie... » avec les paramètres suivants :
  - Nom & prénom : poste 3i01
  - Adresse mail : [poste3i01@postei.c204.rtlr](mailto:poste3i01@postei.c204.rtlr)
  - Mot de passe : rtrt



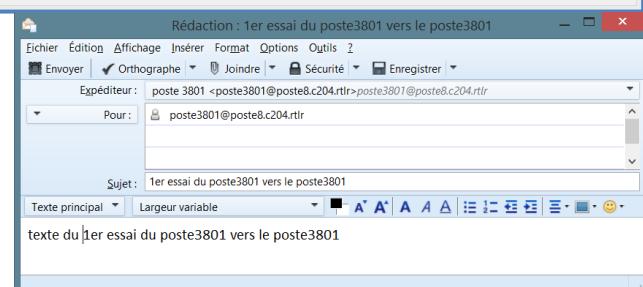
- Cliquer sur « Continuer », mais normalement il ne va pas trouver les paramètres de config, vu que notre domaine poste3i.c204.rtlr n'est pas connu, et référencé dans aucun serveur DNS. Préciser alors l'@ IP de la VM Mail-Server pour les serveurs entrant et sortant (ou le nom « mail-server.c204.rtlr » si vous le renseignez au préalable dans le fichier « hosts »), puis re-tester la détection du reste des paramètres (même si nous les connaissons...)

Thunderbird doit normalement trouver des paramètres similaires à ceux figurant dans l'exemple ci-contre :

- Cliquer sur « Terminé ».



- Tester l'envoi d'un message du poste 3i01 vers lui-même ; normalement tout doit bien se passer.



- Sur le serveur, vérifier avec « cat /var/log/mail.log » si l'envoi du mail s'est bien déroulé, un peu comme dans l'exemple de capture ci-dessous :

```
Oct 20 11:47:23 mail-server postfix/smtpd[8546]: connect from unknown[192.168.18.208]
Oct 20 11:47:23 mail-server postfix/smtpd[8546]: Anonymous TLS connection established from unknown[192.168.18.208]: TLSv1 with cipher ECDHE-RSA-AES128-SHA (128/128 bits)
Oct 20 11:47:23 mail-server postfix/smtpd[8546]: 5BDE980A6C: client=unknown[192.168.18.208], sasl_method=PLAIN, sasl_username=poste3801
Oct 20 11:47:23 mail-server postfix/cleanup[8551]: 5BDE980A6C: message-id=<5444D26.703020@poste8.c204.rtir>
Oct 20 11:47:23 mail-server postfix/qmgr[8455]: 5BDE980A6C: from=<poste3801@poste8.c204.rtir>, size=707, nrcpt=1 (queue active)
Oct 20 11:47:23 mail-server postfix/smtpd[8546]: disconnect from unknown[192.168.18.208]
Oct 20 11:47:23 mail-server postfix/local[8552]: 5BDE980A6C: to=<poste3801@poste8.c204.rtir>, relay=local, delay=0.06, delays=0.02/0.01/0.02, dsn=2.0.0, status=sent (delivered to maildir)
Oct 20 11:47:23 mail-server postfix/qmgr[8455]: 5BDE980A6C: removed
root@mail-server:"#
```

Vous devez également constater la création de dossiers supplémentaires dans Thunderbird pour le compte poste3/01 suite à l'envoi de ce mail :



Depuis l'envoi (et la réception, ce mail étant adressé à lui-même) de ce 1<sup>er</sup> mail, un dossier « Maildir/ » a été créé dans le home de l'utilisateur : vérifions si tout s'est bien passé :

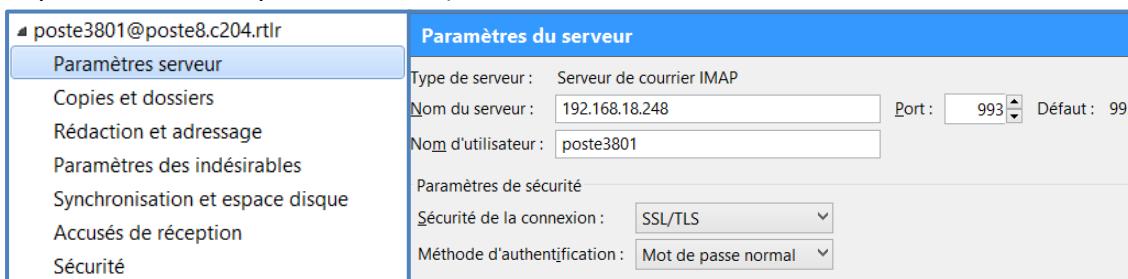
- Consulter le contenu du dossier « /home/poste3801 » et de ses sous-dossiers « new », « cur » et « tmp » : quels fichiers y sont présents ?
- Consulter le contenu d'un des fichiers présents (normalement il n'y en a qu'un vu que nous n'avons envoyé qu'un seul mail), comme par exemple ci-contre :

```
root@mail-server:~# ls -l /home/poste3801/Maildir/cur/
total 4
-rw-r--r-- 1 poste3801 poste3801 889 oct. 20 12:23 1413800599.U8011c1ie98MS25520.mail-server:2,S
root@mail-server:# cat /home/poste3801/Maildir/cur/1413800599.U8011c1ie98MS25520.mail-server:2,S
Return-Path: <poste3801@poste8.c204.rtir>
X-Original-To: poste3801@poste8.c204.rtir
Delivered-To: poste3801@poste8.c204.rtir
Received: from [192.168.18.208] (unknown [192.168.18.208])
 using TLSv1 with cipher ECDHE-RSA-AES128-SHA (128/128 bits)
 (No client certificate requested)
 by mail-server.poste8.c204.rtir (Postfix) with ESMTPSA id 7A7A7803BA
 for <poste3801@poste8.c204.rtir>; Mon, 20 Oct 2014 12:23:19 +0200 (CEST)
Message-ID: <5444E292.20100@poste8.c204.rtir>
Date: Mon, 20 Oct 2014 12:23:14 +0200
From: poste3801 <poste3801@poste8.c204.rtir>
User-Agent: Mozilla/5.0 (Windows NT 6.3; WOW64; rv:24.0) Gecko/20100101 Thunderbird/24.6.0
MIME-Version: 1.0
To: poste3801@poste8.c204.rtir
Subject: 1er essai du poste3801 vers le poste3801
Content-Type: text/plain; charset=ISO-8859-1; format=flowed
Content-Transfer-Encoding: 7bit
texte du 1er essai du poste3801 vers le poste3801
root@mail-server:~#
```

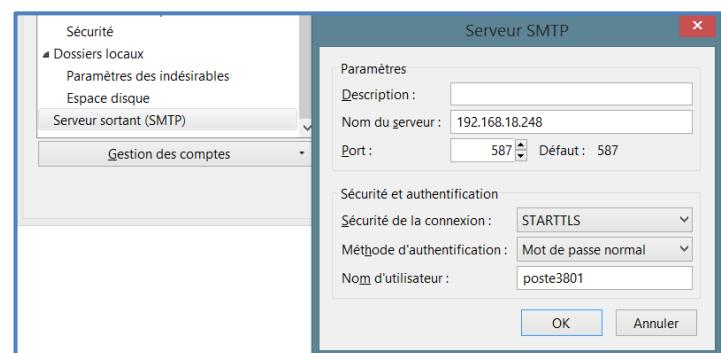
- Que se passe-t-il sur le serveur quand on supprime le mail reçu depuis Thunderbird ?

Enfin, essayons juste une dernière modification pour renforcer le processus d'accès IMAP :

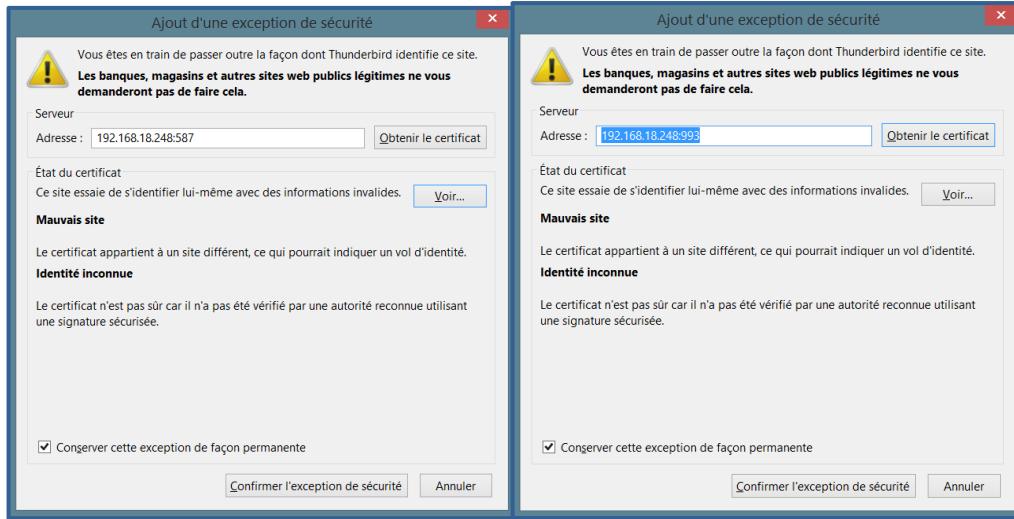
- Aller dans « Outils > Paramètres des comptes > Paramètres serveur » et modifier le type de sécurité de la connexion : passer en mode **SSL/TLS** ; le n° de port doit alors passer automatiquement à **993**).



- Toujours dans le menu « Paramètre des comptes », dans « Serveur sortant (SMTP) », vérifier que l'accès au serveur sortant SMTP est bien dirigé vers le port **587** :



NB : au final, lors de ces configurations, il vous aura été demandé 2 acceptations de certificats, un pour le port 587, et un pour le 993. Et si vous cliquez sur « Voir », vous retrouverez les infos indiquées lors de la construction de notre certificat auto-signé :



- De la même façon, créer sous Thunderbird les 2 autres comptes « poste3i02 » et « poste3i03 » pour les postes 3i02 et 3i03.

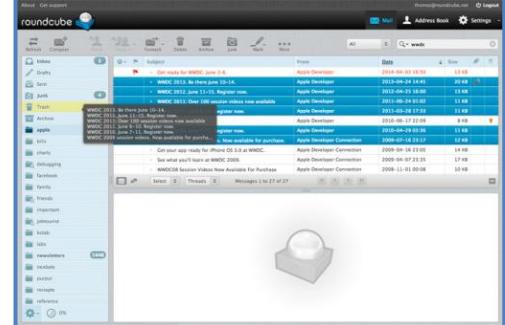
### IX.7.8. Installation de la plate-forme Roundcube

Le projet **Roundcube** est une solution **webmail** libre et open source avec une interface utilisateur facile à installer et configurer, et qui s'exécute sur une machine ayant un serveur **base de données (Mysql)** par exemple) et un **serveur web (Apache2** par exemple). Nous devons donc au préalable installer ces 2 services :

- Installer « mysql-server », avec « rtrt » comme mot de passe.
- Installer « apache2 », puis rajouter le nom du serveur web dans « /etc/apache2/apache2.conf » :
 

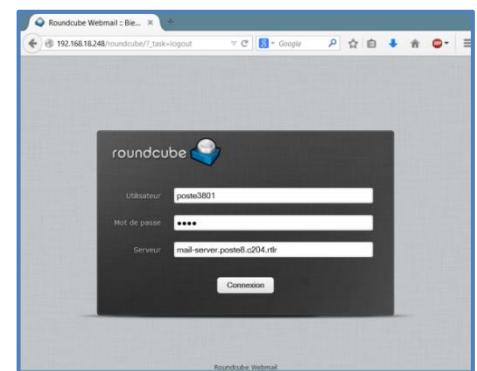
```
ServerName mail-server.postei.c204.rtrtlr
```
- Redémarrer apache2.
- Installer le paquet « roundcube-mysql », puis le paquet « roundcube », en renseignant :
  - configuration de la BD de roundcube avec dbconfig-common : yes
  - type de BD : mysql
  - mot de passe : rtrt
- Décommenter les deux lignes suivantes dans /etc/roundcube/apache.conf :
 

```
Alias /roundcube/program/js/tiny_mce/ /usr/share/tinymce/www/
Alias /roundcube /var/lib/roundcube
```
- Activer le module « mcrypt » dans PHP avec :



```
php5enmod mcrypt
```

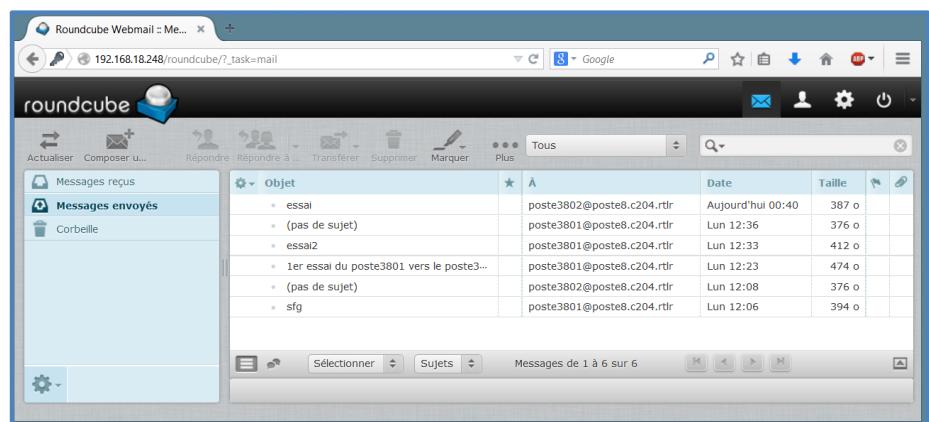
- Redémarrer apache2.
- Le webmail RoundCube est désormais accessible sur « <http://192.168.1.12i/roundcube> ».
- Se connecter sur le serveur avec les identifiants du poste 3i01 par exemple :
  - Utilisateur : poste3i01
  - Mot de passe : rtrt
  - Serveur : mail-server.postei.c204.rtlr



Vous devez retrouver les mails précédemment émis et reçus pour cet utilisateur.

- Tester l'envoi et la réception de mails depuis Roundcube sur les 3 utilisateurs poste3i01, poste3i02 et poste3i03.

Notre serveur de mail et sa plate-forme WebMail sont maintenant opérationnels.



### IX.7.9. Unification de la messagerie sur Asterisk

Nous allons retourner dans le fichier de configuration de la messagerie « `/etc/asterisk/voicemail.conf` » et renseigner les paramètres de messagerie du serveur Asterisk :

- Dans le contexte [general], modifier (ou vérifier) les lignes suivantes :
 

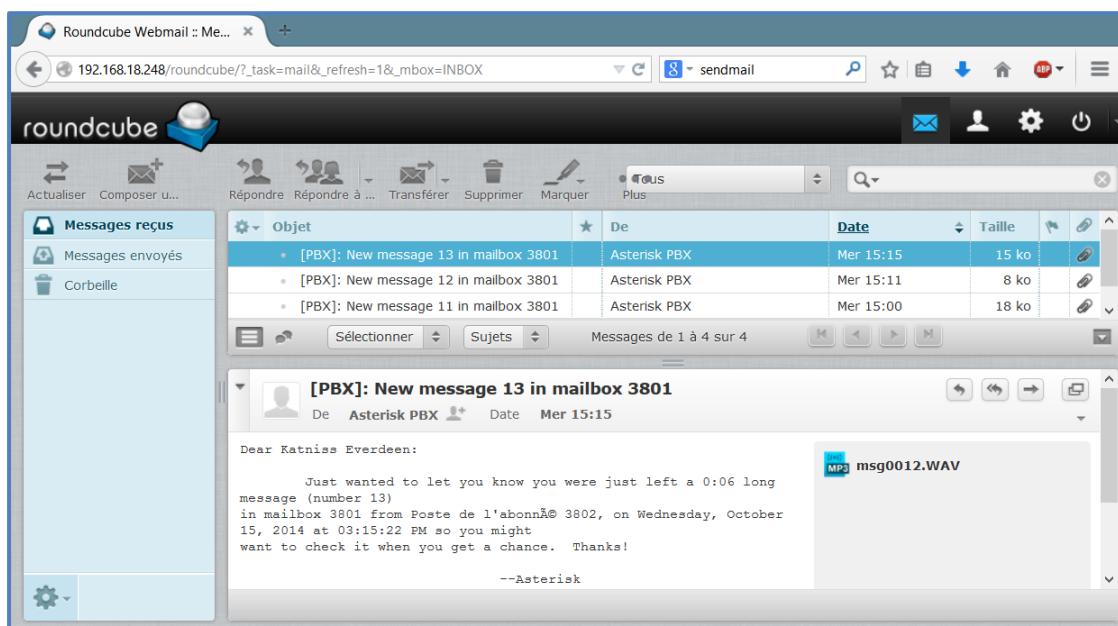
```
serveremail = asterisk-server.poste8.c204.rtlr ; ce compte n'existe pas,
; mais cela évitera les retours inutiles de mails vers le serveur Asterisk
sendvoicemail = yes ; envoi d'un mail d'alerte lors du dépôt d'un
; message sur la boîte vocale
attach = yes ; un fichier audio contenant le message est joint
```
- Dans le contexte [default], ajouter aux profils de nos 3 abonnés principaux leur @ mail, afin que le serveur Asterisk utilise cette adresse comme destinataire du mail :
 

```
[default]
3i01 => 4625,Katniss Everdeen, poste3i01@postei.c204.rtlr
3i02 => 3159,Peeta Mellark, poste3i02@postei.c204.rtlr
3i03 => 8790,Gale Hawthorne, poste3i03@postei.c204.rtlr
```

Pour envoyer les mails, le serveur Asterisk aura recours à un serveur mail **MTA** très simple appelé « **sendmail** » (NB : postfix est aujourd’hui bien plus répandu que sendmail). Il nous faut installer ce serveur (mais la configuration sera très légère, car seule la machine elle-même enverra des mails avec).

- Installer le paquet « sendmail » sur Asterisk-Server.
- Ajouter (ou modifier) dans le fichier « hosts » du serveur Asterisk les liens suivants :

|                                 |                                  |
|---------------------------------|----------------------------------|
| 127.0.0.1                       | asterisk-server.postei.c204.rtlr |
| 127.0.1.1                       | asterisk-server                  |
| 192.168.1 <i>i</i> .12 <i>i</i> | mail-server.postei.c204.rtlr     |
| 192.168.1 <i>i</i> .12 <i>i</i> | postei.c204.rtlr                 |
- Relancer Asterisk, laisser un message sur un des postes, et vérifier qu'après quelques temps le message est disponible dans Thunderbird et sur le webmail Roundcube avec comme pièce jointe le fichier audio ; vérifier qu'il soit bien audible.



## IX.8. Standard auto – ~~Serveur vocal interactif~~ (pas le temps)

Il est souvent nécessaire de mettre en place sur les serveurs de téléphonie un système de réponse automatiquement aux appels entrants, permettant aux appelants de se diriger vers diverses extensions d'abonnés ou services grâce à des choix de menu. Ce système est appelé un **Standard Automatique** (AA : **Automated Attendant**), ou encore **Serveur Vocal Interactif** (IVR : **Interactive Voice Response**) lorsqu'il s'agit d'un standard auto complexe (avec consultation de base de données, ...).

Un standard automatique fournit normalement les services suivants :

- transfert à une extension
- transfert vers une file d'attente
- transfert vers la messagerie vocale
- écouter un message (ex : « nos horaires d'ouverture sont ...»)
- se connecter à un sous-menu
- joindre la réception
- répétition des choix si absence de réponse ou mauvais choix

La conception de l'arborescence doit être bien pensée, afin que le standard auto soit fonctionnel, réponde bien aux besoins de l'entreprise, et avant tout reste simple pour les utilisateurs : les clients de l'entreprise ne doivent pas se perdre ni se lasser devant ce qui pourrait représenter une « barrière d'accès » pour eux. Il est notamment préconisé de toujours proposer en dernier recours la mise en communication avec une standardiste « humaine ».

Nous allons maintenant réaliser sur notre serveur Asterisk un standard automatique dont le cahier des charges est le suivant :

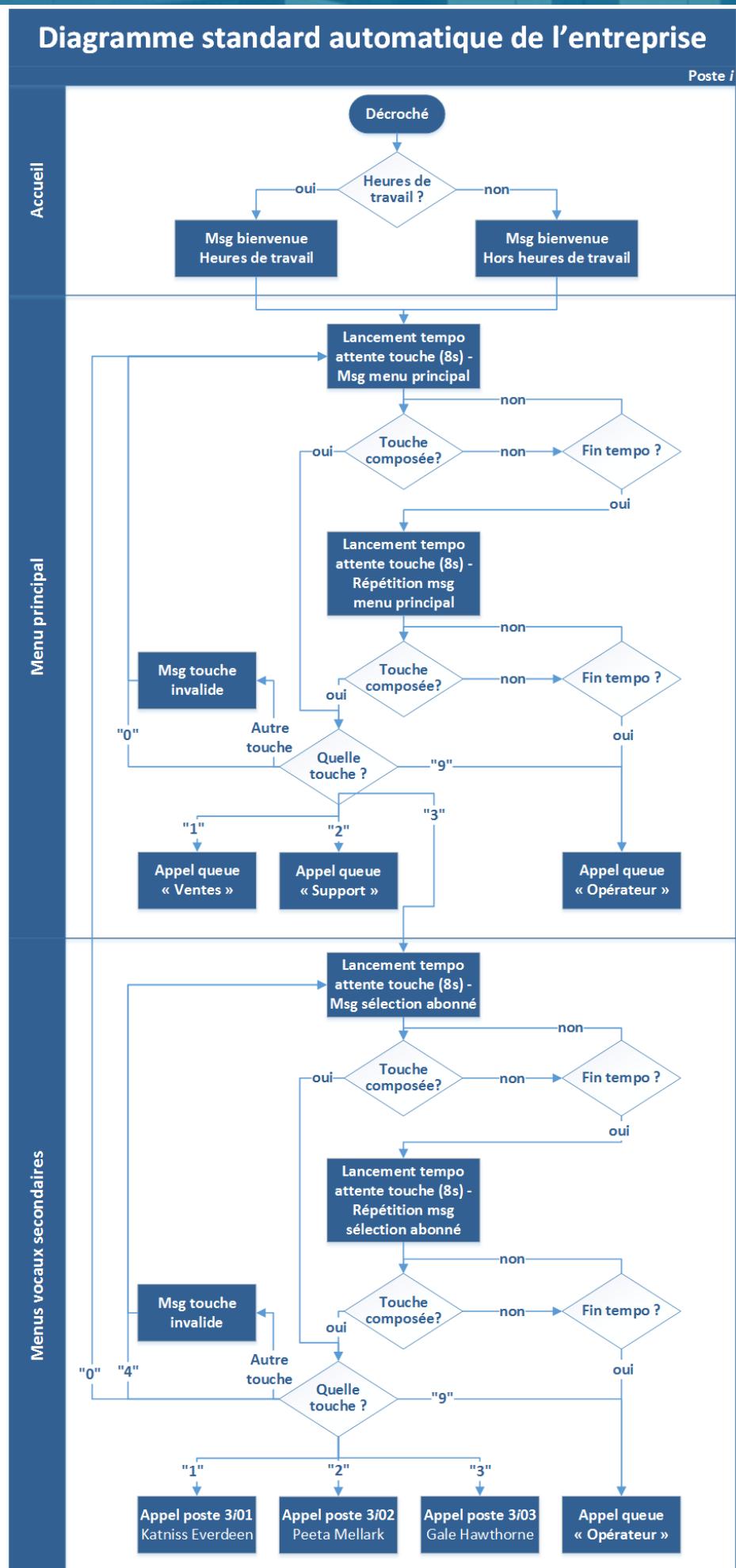
- l'architecture du standard automatique doit respecter le diagramme de la page suivante
- le standard automatique aura comme extension 3i00, qui sera ultérieurement associé un n° SDA d'entête de l'entreprise.

Au préalable, nous allons réaffecter les membres aux queues existantes, à savoir « ventes » et « support », et créer une 3<sup>ème</sup> queue « opérateur » :

- Créer une 3<sup>ème</sup> queue « opérateur » avec pour extension le n° 3i23.
- Incrire les abonnés comme ceci :
  - le 6757i comme agent du service « opérateur » (ce sera le seul membre de cette queue, vu que nous n'avons pas assez de postes à disposition)
  - le GXV3140 comme agent du service « ventes » (seul également)
  - le softphone Jitsi comme agent du service « support » (").

Il nous faut créer les fichiers audio qui seront diffusés pendant la navigation dans le serveur vocal interactif. Pour enregistrer le message vocal directement au bon format, il peut être astucieux d'utiliser la messagerie vocale que nous venons de mettre en place : en effet, il suffit de laisser un message sur la boîte vocale d'un utilisateur, et d'aller récupérer ce fichier audio.

- Créer les messages audio nécessaires, les copier dans le dossier « /usr/share/asterisk/sounds/f » et les renommer (pensez aux droits des fichiers).



Ecriture de l'arborescence :

- Dans quel fichier doit-on écrire le comportement à avoir pour ce standard automatique ?
- S'inspirer de l'exemple ci-dessous pour écrire votre standard automatique :

; signification :

; "s" : comportement par défaut, « standard »  
; "i" : « incorrect », quand on tape une mauvaise touche  
; "t" : « timeout », quand la tempo est terminée

```
exten => 3800,1,Goto(my_auto_attendant,s,1)
```

```
[my_auto_attendant]
```

```
exten => s,1,Verbose(1, Caller ${CALLERID(all)} has entered the auto attendant)
 same => n,Answer()
```

```
; this sets the inter-digit timer
 same => n,Set(TIMEOUT(digit)=2)
```

```
; wait one second to establish audio
 same => n,Wait(1)
```

```
; If Mon-Fri 9-5 goto label daygreeting
 same => n,GotolftTime(9:00-17:00,mon-fri,*,*? daygreeting:afterhoursgreeting)
```

```
same => n(daygreeting),Playback(msg_daygreeting) ; DAY GREETING
 same => n,Goto(menuprompt)
```

```
same => n(afterhoursgreeting),Playback(msg_nightgreeting) ; AFTER HOURS GREETING
 same => n,Goto(menuprompt)
```

```
same => n(menuprompt),Background(msg_mainmenu) ; MAIN MENU PROMPT
 same => n,WaitExten(6)
 same => n,Background(msg_repeat&msg_mainmenu)
 same => n,WaitExten(6)
 same => n,Goto(0,1) ; Treat as if caller has pressed '0' if no extension dialed
```

```
exten => 1,1,Verbose(1, Caller ${CALLERID(all)} has entered the sales queue)
 same => n,Goto(Queues,5001,1) ; Sales Queue
```

```
exten => 2,1,Verbose(1, Caller ${CALLERID(all)} has entered the service queue)
 same => n,Goto(Queues,5002,1) ; Service Queue
```

```
exten => 3,1,Verbose(1, Caller ${CALLERID(all)} has requested address and fax info)
 same => n,Background(faxandaddress) ; Address and fax info
 same => n,Goto(s,menuprompt) ; Take caller back to main menu prompt
```

```
exten => 0,1,Verbose(1, Caller ${CALLERID(all)} is calling the secretary)
 same => n,Dial(SIP/secretary) ; Secretary extension

exten => i,1,Verbose(1, Caller ${CALLERID(all)} has entered an invalid selection)
 same => n,Playback(invalid)
 same => n,Goto(s,menuprompt)

exten => t,1,Verbose(1, Caller ${CALLERID(all)} has timed out)
 same => n,Goto(0,1)
```

- Tester l'intégralité de votre SVI ; dès que le test est concluant, appeler votre enseignant pour qu'il valide votre travail.

## X. Connexion externe – Trunk SIP

SIP est de loin le plus populaire des protocoles VoIP. C'est un protocole peer-to-peer, et il n'y a pas vraiment de spécificités formelles pour réaliser un trunk en SIP ; si on connecte un seul téléphone à son serveur ou si on réalise une connexion entre deux serveurs, les connexions SIP seront similaires.

Les trunks SIP sont réalisés dans 2 cas principalement :

- besoin d'interconnecter les serveurs ToIP de 2 sites distants d'une même entreprise (maison mère et succursale)
- connecter son réseau privé au serveur de son fournisseur d'accès au réseau téléphonique public (SIP provider).

Nous allons réaliser les 2 possibilités.

### X.1. Interconnexion de 2 serveurs SIP privés

Nous allons réaliser un trunk entre votre serveur Asterisk et celui d'un autre binôme, afin de réaliser un pseudo-réseau unique, où il sera possible d'appeler un poste du site distant en composant simplement les 4 chiffres internes du n° de poste visé.

- Dans « /etc/asterisk/sip.conf », modifier pour les 3 postes le champ context en « default » au lieu de « plan-num-prive ».
- Selon l'état d'avancement de chacun des binômes, former des associations de 2 binômes.

La mise en place du trunk SIP va reposer ici sur une **configuration symétrique** de la liaison sur les 2 serveurs Asterisk. Dans les instructions suivantes, les paramètres *j* et *k* feront référence aux numéros de binômes qui réaliseront le trunk ; vous devrez les remplacer par les n° de binômes (1 à 6) correspondants.

- Sur le poste *j*, dans « /etc/asterisk/sip.conf », créer le profil SIP suivant :

```
[serverk]
; Specify the SIP account type as 'peer'. This means that incoming
; calls will be matched on IP address and port number. So, when Asterisk
; receives a call from 192.168.1k.10k and the standard SIP port of 5060,
; it will match this entry in sip.conf. It will then request authentication
; and expect the password to match the 'secret' specified here.
type = peer

; This is the IP address for the remote box (serverk). This option can also
; be provided a hostname.
host = 192.168.1k.10k

; When we send calls to this SIP peer and must provide authentication,
; we use 'serverj' as our default username.
defaultuser = serverj
```

```

; This is the shared secret with serverk. It will be used as the password
; when either receiving a call from serverk, or sending a call to serverk.
secret = rtrt

; When receiving a call from serverk, match it against extensions
; in the 'plan-num-prive' context of extensions.conf.
context = plan-num-prive

; Start by clearing out the list of allowed codecs.
disallow = all

; Allow the alaw and gsm codecs
allow = gsm
allow = alaw

```

- Sur le poste *k*, effectuer la même configuration symétrique.
- Vérifier avec la commande « `sip show peers` » que le lien SIP avec le serveur distant est bien déclaré, comme sur l'exemple ci-dessous :

| Name/username       | Host           | Dyn | Forcerport | Comedia | ACL | Port  | Status      |
|---------------------|----------------|-----|------------|---------|-----|-------|-------------|
| poste3801/poste3801 | 192.168.18.228 | D   | Auto (No)  | No      |     | 5060  | Unmonitored |
| poste3802/poste3802 | 192.168.18.238 | D   | Auto (No)  | No      |     | 12482 | Unmonitored |
| poste3803/poste3803 | 192.168.18.208 | D   | Auto (No)  | No      |     | 5060  | Unmonitored |
| serv9/server8       | 192.168.19.219 |     | Auto (No)  | No      |     | 5060  | Unmonitored |

4 sip peers [Monitored: 0 online, 0 offline Unmonitored: 4 online, 0 offline]

Il nous reste à configurer le plan de numérotation pour tenir compte du fait qu'il faille rediriger les appels pour les n° d'abonnements distants vers le trunk SIP :

- En s'inspirant de l'exemple ci-dessous, rajouter une section [appels-vers-site-distant] de votre plan numérotation pour rediriger les appels vers le serveur du site distant :

```

...
[default]
include => appels-vers-site-distant
...
[appels-vers-site-distant]
exten => _39XX,1,Dial(SIP/${EXTEN}@server9) ; tous les appels vers les n° de
 ; 4 chiffres commençant par 39 seront transférés
 ; sur le lien SIP au serveur SIP distant
; signification :
; _ correspond à un template (modèle) pour décrire un type (liste) de n°
; X correspond à un chiffre entre 0 et 9
; Z correspond à un chiffre entre 1 et 9
; N correspond à un chiffre entre 2 et 9
; [1247-9] correspond à un chiffre dans la liste suivante : 1,2,4,7,8,9
; . correspond à un chiffre/lettre et plus
; ! correspond à un chiffre/lettre et plus, ou à rien
...

```

- Vérifier que les appels entre les 2 sites fonctionnent correctement.

## X.2. Connexion à un fournisseur SIP

Lorsqu'on s'inscrit à un **fournisseur SIP (SIP provider)**, on dispose alors d'un service d'envoi et de réception d'appels téléphoniques publics.

La configuration peut légèrement différer en fonction du fournisseur SIP qui, logiquement, fournit les paramètres nécessaires et parfois des exemples de configuration d'Asterisk pour nous aider à se connecter plus facilement.

Afin de pouvoir recevoir des appels, le provider aura probablement besoin que notre serveur s'enregistre auprès de l'un de ses serveurs.

Nous allons aujourd'hui avoir recours à un abonnement chez le prestataire SIP « **RTL TELECOM** » très économique ☺, aux caractéristiques suivantes :

- @ IP serveur : 192.168.10.251
- login / mot de passe : *postei* / *passwordi*
- service « plutôt restreint », limité aux n° suivants :

| N° poste de travail | n° internes | n° publics (entête + SDA) |
|---------------------|-------------|---------------------------|
| 1                   | 31xx        | 05 46 01 31 xx            |
| 2                   | 32xx        | 05 46 02 32 xx            |
| 3                   | 33xx        | 05 46 03 33 xx            |
| 4                   | 34xx        | 05 46 04 34 xx            |
| 5                   | 35xx        | 05 46 05 35 xx            |
| 6                   | 36xx        | 05 46 06 36 xx            |
| 7                   | 37xx        | 05 46 07 37 xx            |

Pour que notre serveur puisse se connecter au provider SIP, il nous faut ajouter une ligne d'enregistrement :

- Rajouter dans la section [general] de « */etc/asterisk/sip.conf* » la ligne suivante :
 

```
...
[general]
register => postei:passwordi@192.168.10.251
...
```
- Quelles différences y-a-t-il entre **contrôle centralisé** et **distribuée des appels** ? Qu'est-ce qu'une ré-invitation ?
- Rajouter à vos profils [poste3i01], [poste3i02], ... le paramètre suivant :
 

```
insecure = invite ; permet d'autoriser les ré-invitations entre terminaux
; finaux (NB : le paramètre « canreinvite » est normalement déjà à « yes »)
```
- Créer également dans « *sip.conf* » un profil de type « *peer* » pour votre provider :
 

```
[mon-provider-sip]
type = peer
host = 192.168.10.251
defaultuser = postei ; ou « from_user = postei », qui modifie les
; champs From: et Contact: de l'instruction
; INVITE lors de l'envoi d'un appel au provider
```

```

secret = password
; Most providers won't authenticate when they send calls to you,
; so you need this line to just accept their calls.
insecure = invite
context = appels-entrants
dtmfmode = rfc2833
disallow = all
allow = alaw
allow = gsm
deny = 0.0.0.0/0
permit = 192.168.10.251/32

```

Rq : voici un exemple de configuration pour une connexion sur l'opérateur OVH, où on retrouve la commande pour l'enregistrement :

```

[general]
language=fr
bindport=5060
bindaddr=0.0.0.0
srvlookup=yes
canreinvite=no
defaultexpiry=3600
registertimeout=30
registerattempts=0
disallow=all
allow=ulaw
allowguest=yes
nat=yes

;Connexion au compte SIP ovh.com
;register => numéro-compte-sip:mot-depasse-compte-sip@fournisseur.sip.com
register => 0033XXXXXXXX:XXXX@sip.ovh.fr

;Création du compte Asterisk pour OVH
[vers-ovh]
disallow=all
type=friend
secret=XXXX
host=sip.ovh.fr
fromdomain=sip.ovh.fr
fromuser=0033XXXXXXXX
username=0033XXXXXXXX
nat=yes
context=depuis-ovh
insecure=invite,port
qualify=yes
dtmfmode=inband
allow=ulaw

```

- Exécuter la commande « `sip show registry` » pour consulter si l'enregistrement auprès du provider SIP s'est bien déroulé. Vous devez obtenir une réponse ressemblant à l'exemple ci-dessous, pour un enregistrement auprès d'OVH :

| Asterisk*CLI> sip show registry | Host                 | dnsmgr | Username     | Refresh State   | Reg. Time                 |
|---------------------------------|----------------------|--------|--------------|-----------------|---------------------------|
|                                 | sip.ovh.fr:5060      | N      | 0033XXXXXXXX | 3585 Registered | Thu, 20 Sep 2012 11:53:43 |
|                                 | 1 SIP registrations. |        |              |                 |                           |

- Maintenant que le compte a été défini, il reste à configurer dans « `extensions.conf` » le plan de numérotation pour nous permettre d'envoyer des appels via le provider SIP, ici sans avoir besoin de composer le 0 pour indiquer qu'on souhaite sortir du réseau privé :

```

...
[default]
include => plan-num-prive
include => appels-sortants
...
```

```
[appels-sortants]
exten => _0ZXXXXXXX,1,Dial(SIP/${EXTEN}@mon-provider-sip)
```

```
[appels-entrants]
exten => _3iXX,1,Ringing(1) ; Attendre une seconde en faisant retentir la
 ; sonnerie du telephone de l'appelant
 same => n,Answer() ; Répond à l'appel
 same => n,Goto(plan-num-prive,${EXTEN},1)
...

```

- Quelle instruction aurait-on dû utiliser dans la section [appels-sortants] si on souhaitait que les usagers du réseau privé aient besoin de composer le 0 pour sortir sur le réseau public via le provider SIP ?
- Qu'aurait-on dû mettre dans la section [appels-entrants] si on avait voulu que tous les appels aillent directement sur le standard automatique ?
- Tester les appels publics en composant les 10 chiffres d'un abonné. Appeler votre enseignant pour qu'il vérifie que tout fonctionne correctement.

Rq : Pour information, la configuration réalisée sur le serveur ToIP du provider SIP « RTL Telecom » est la suivante :

Dans « sip.conf » :

```
...
[postei]
type = peer
secret = passwordi
host = dynamic ; ne pas mettre l'@ IP car alors l'enregistrement serait
 ; impossible (car inutile en fait...)
context = routage-appels
disallow = all
allow = alaw
allow = gsm
dtmfmode = rfc2833
language=fr
deny = 0.0.0.0/0
permit = 192.168.1i.10i/32
...

```

Et lorsqu'un poste s'enregistre, le message suivant apparait sur la console du serveur de « RTL Telecom » ; et quand on fait sip show peers sur le serveur opérateur, on voit (avant de configurer le plan de num dans extensions.conf, ce qui peut expliquer le « poste8/*s* ») :

```
Asterisk Ready.
== Parsing '/etc/asterisk/cli.conf': Found
 > Saved useragent "Asterisk PBX 12.5.0" for peer poste8

*CLI> sip show registry
Host dnsmgr Username Refresh State Reg. Time
0 SIP registrations.

*CLI> sip show peers
Name/username Host Dyn Forcerport Comedia ACL Port Status Description
poste8/s 192.168.18.218 D Auto (No) No A 5060 Unmonitored

1 sip peers [Monitored: 0 online, 0 offline Unmonitored: 1 online, 0 offline]
*CLI>
```

Dans « extensions.conf » :

```
...
[default]
include => routage-appels
...
[routage-appels]
exten => _05460131XX,1,Dial(SIP/${EXTEN:-4:4}@poste1) ; on ne transmet
 same => n,Hangup() ; que les 4 derniers chiffres composés
exten => _05460232XX,1,Dial(SIP/${EXTEN:-4:4}@poste2)
 same => n,Hangup()
...
```

Rq : La variable  **\${EXTEN}**  a la syntaxe  **\${EXTEN:x:y}** , où x désigne la position du début et y le nombre de chiffres retenus. Par exemple, pour le n° composé « 94169671111 », cela donne :

- **\${EXTEN:1:3}**  commence 1 chiffre après le début et garde 3 chiffres, soit 416
- **\${EXTEN:4:7}**  commence 4 chiffre après le début et garde 7 chiffres, soit 9671111
- **\${EXTEN:-4:4}**  commence 4 chiffres avant la fin et garde 4 chiffres, soit 1111
- **\${EXTEN:2:-4}**  commence 2 chiffres après le début et exclut les 4 derniers, soit 16967
- **\${EXTEN:-6:-4}**  commence 6 chiffres avant la fin et exclut les 4 derniers, soit 67
- **\${EXTEN:1}**  commence 2 chiffres après le début et garde tout le reste : 4169671111 (si le nombre de chiffres à retourner est vide, la fonction retourne l'ensemble des chiffres restants)

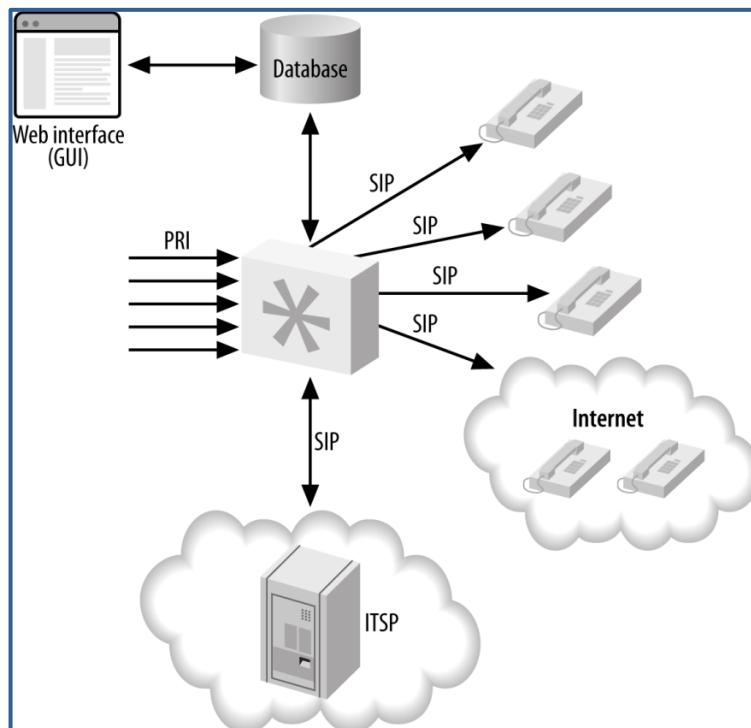
## XI. Intégration dans une base de données

La limitation actuelle de notre système est qu'au moindre changement d'un profil d'abonné SIP par exemple, il faut redémarrer certains modules d'Asterisk afin que les modifications effectuées dans les fichiers de configuration soient prises en compte. Une solution afin que l'on puisse **modifier certaines données en « temps réel »** est de **stocker les données nécessaires au fonctionnement d'Asterisk dans une base de données**.

**Paramétriser Asterisk afin qu'il exploite les informations stockées dans une base de données** est l'un des aspects fondamentaux de la construction d'un grand système fiable et performant, déployable en cluster ou distribué.

La puissance d'une base de données permet de :

- utiliser et modifier dynamiquement les données des abonnés et des plans de numérotation, sans dépendre uniquement de quelques fichiers clés (sip.conf, extensions.conf, ...).
- intégrer facilement des outils de gestion des données par interfaces web
- partager les informations entre le serveur ToIP Asterisk et d'autres outils ou services réseaux.



Nous allons donc chercher dans cette partie à **stocker la configuration des postes SIP et le plan de numérotation** (informations actuellement rentrées dans les fichiers « sip.conf » et « extensions.conf ») **dans une base de données**. Il existe différents types de bases de données, nous allons utiliser ici une **base de données MySQL**.

### Remarque sur le fonctionnement d'Asterisk en temps réel :

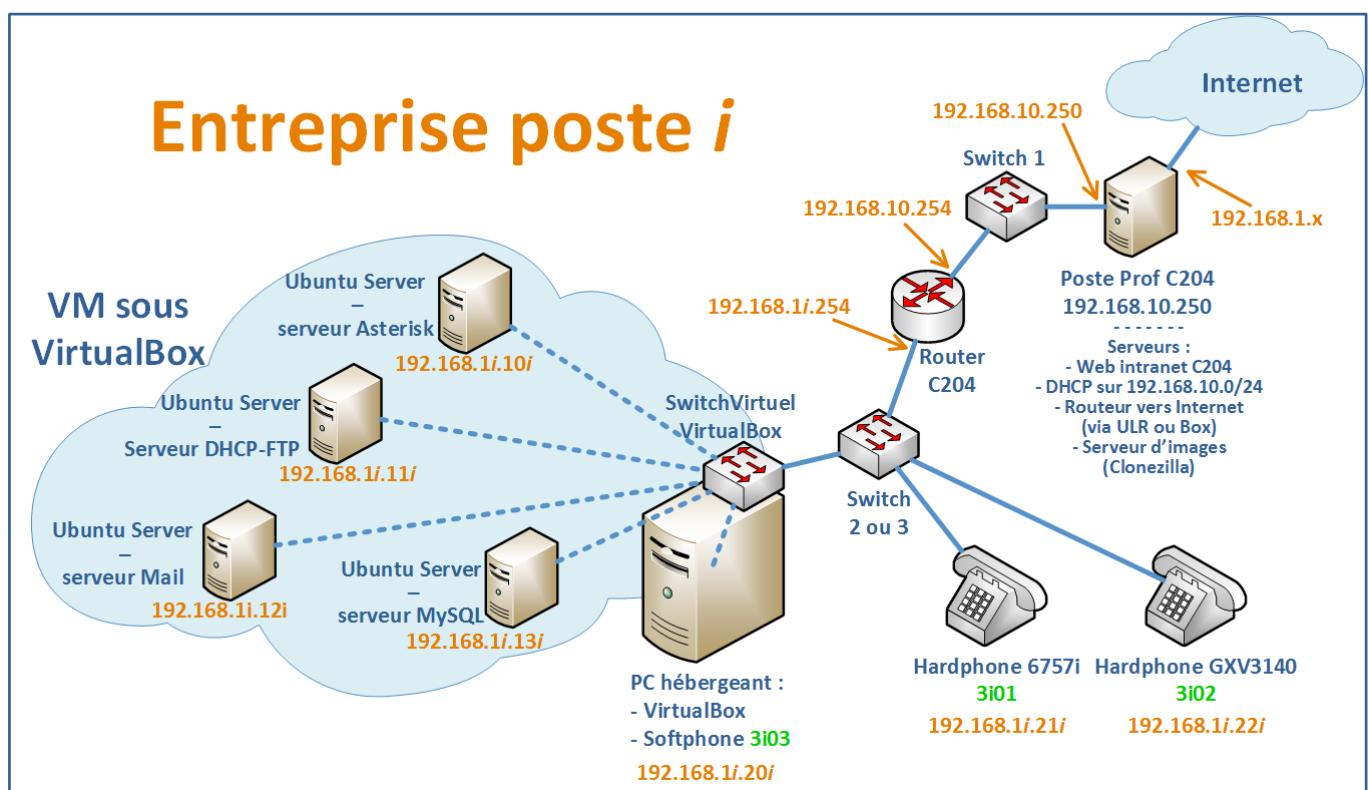
L'architecture temps réel d'Asterisk (Asterisk Realtime Architecture) permet de stocker tous les paramètres, normalement stockés dans les fichiers de configuration Asterisk, dans une base de données. Il existe deux types de temps réel: statique et dynamique :

- méthode statique : elle est similaire à la méthode traditionnelle de lecture d'un fichier de configuration (l'information est chargée uniquement en cas de déclenchement d'un « reload » depuis la CLI), excepté le fait que les données sont lues sur la base de données et non plus dans les fichiers de configuration ; avec ce mode statique, apporter des changements aux données nécessite un rechargement, tout comme si on avait changé un fichier de configuration
- méthode dynamique : Asterisk charge et met à jour l'information telle qu'elle est utilisée par le système en direct ; ceci est couramment utilisé pour par exemple les profils SIP et les boîtes vocales car il n'y a pas besoin de recharger Asterisk lorsque des modifications ont été apportées à ces données.

Le type de temps réel est configuré dans le fichier « /etc/asterisk/extconfig.conf ». Ce fichier indique à Asterisk les informations qu'il faut charger depuis la base de données et où le trouver. Cela permet par exemple de charger une partie des paramètres et données depuis la base de données où ils sont stockés, et les autres paramètres ou données à partir des fichiers de configuration standard.

## XI.1. Crédation d'une VM pour la base de données

Nous aurions pu installer une base de données sur la même VM à côté du serveur ToIP Asterisk, mais ici, afin de **bien dissocier les rôles de chaque machine**, nous allons exploiter une 3<sup>ème</sup> VM, « **M4205-MySQL-Server** », qui hébergera la base de données MySQL :



- Comme pour le server Asterisk, configurer la carte réseau eth0 de la VM MySQL-Server de façon permanente, avec les paramètres IP adéquats (cf schéma).

## XI.2. Installation de la base de données MySQL

- Installer le paquet « mysql-server », et choisir « rtrt » comme mot de passe pour le compte administrateur « root » de MySQL.

Avec la base de données MySQL active maintenant, nous allons commencer par sécuriser notre installation. Nous allons exécuter un script permettant d'entrer un nouveau mot de passe pour l'utilisateur root, avec quelques options supplémentaires :

- Exécuter le script suivant :

```
/usr/bin/mysql_secure_installation
```

Dans l'exécution de ce script assez simple, choisir de supprimer l'utilisateur « anonymous », maintenir l'autorisation d'accès distant (pas exclusivement depuis « localhost ») ; pour les autres choix, sélectionner les valeurs par défaut.

- Se connecter à l'ILC de la base MySQL avec la commande suivante :

```
mysql -u root -p
```

Le prompt de la console MySQL doit s'afficher : **mysql >**

- Créer un utilisateur « asterisk » (mot de passe « rtrtsql ») avec la commande suivante, où le % indique que l'utilisateur « asterisk » peut se connecter depuis n'importe quel hôte :

```
mysql> create user 'asterisk'@'%' identified by 'rtrtsql';
```

- Créer une base de données « asteriskdb », que nous exploiterons avec Asterisk :

```
mysql> create database asteriskdb ;
```

- Visualiser si la base de données a bien été créée :

```
mysql> show databases ;
```

- Maintenant qu'ont été créés d'une part un utilisateur « asterisk » et d'autre part une base de données « asteriskdb », autoriser cet utilisateur à accéder à la base de données :

```
mysql> grant all privileges on asteriskdb.* to 'asterisk'@'%' ;
```

- Enfin, sortir de la console MySQL et vérifier que vous pouvez correctement vous loguer sur la base de données « asteriskdb » en tant qu'utilisateur « asterisk » :

```
mysql> exit
```

```
root@asterisk-server:# mysql -u asterisk -p asteriskdb
```

- Ressortir de la console MySQL.

Il nous faut maintenant vérifier que la VM Asterisk-Server pourra accéder à la base de données « asteriskdb ». Pour cela, effectuons les vérifications suivantes :

- Sur la VM MySQL-Server, afficher les ports ouverts avec « netstat ». Qu'en pensez-vous ?
- Depuis la VM Asterisk-Server, tester avec « nmap » l'ouverture du port 3306 sur la VM MySQL-Server. Est-ce que cela confirme le test précédent ?

Dernier test : depuis la VM Asterisk-Server, tentons de nous connecter sur la base de données distante :

- installer sur Asterisk-Server le client « mysql-client-core-5.6 », puis tenter la connexion sur la base « asteriskdb » :

```
mysql -u asterisk -p asteriskdb -h 192.168.1.13
```

La connexion est-elle possible ?

Remédions à cela :

- Sur MySQL-Server, modifier la ligne suivante dans « /etc/mysql/my.cnf » :  
bind-address = ~~127.0.0.1~~ 0.0.0.0
- Redémarrer le serveur MySQL et observer les différences sur les ports d'écoute.
- Retenter la connexion depuis Asterisk-Server : la connexion à distance fonctionne-t-elle maintenant ?
- Fermer la connexion.

C'est tout pour le moment concernant la config du serveur MySQL.

### XI.3. Installation d'ODBC sur Asterisk-Server

Le **connecteur logiciel ODBC** est une couche d'abstraction qui permet à Asterisk de communiquer avec un large éventail de bases de données sans nécessiter de développer un connecteur séparé pour chacun des types de bases de données que nous voudrions utiliser avec Asterisk. Cela permet d'économiser beaucoup d'efforts de développement et la maintenance du code. Il y a une légère baisse des performances, car nous ajoutons une autre couche d'application entre Asterisk et la base de données, mais cela peut être atténué avec une bonne conception et en vaut la peine quand vous avez besoin de puissantes fonctionnalités, et de base de données flexibles dans votre système Asterisk.

- Avant d'installer le connecteur dans Asterisk, vous devez installer ODBC dans Linux lui-même. Pour installer les pilotes ODBC, utiliser la commande suivante :  

```
apt-get install unixodbc unixodbc-dev
```
- Installer le connecteur ODBC pour MySQL :  

```
apt-get install libmyodbc
```

Configurons maintenant ce driver ODBC pour MySQL :

- Dans le fichier « /etc/odbcinst.ini » (vierge initialement), ajouter les lignes suivantes :  
[MySQL]  
Description = ODBC for MySQL  
Driver = /usr/lib/x86\_64-linux-gnu/odbc/libmyodbc.so  
Setup = /usr/lib/x86\_64-linux-gnu/odbc/libodbcmyS.so  
FileUsage = 1
- Vérifier que le système est capable de voir le pilote en exécutant la commande suivante, qui doit retourner le nom de l'étiquette [MySQL] si tout va bien :  

```
odbcinst -q -d
```

- Ensuite, configurer le fichier « /etc/odbc.ini », qui est utilisé pour créer un identifiant qu'Asterisk va utiliser pour référencer cette configuration :

```
[asteriskdb-connector]
Description = MySQL connection to 'asteriskdb' database
Driver = MySQL
Database = asteriskdb
Server = 192.168.1.i.13i
Port = 3306
Socket = /var/run/mysqld/mysqld.sock
```

NB : le socket mentionné se situe sur la VM distante (192.168.1*i*.13*i*) hébergeant le serveur MySQL.

Nous allons maintenant vérifier que nous pouvons nous connecter à la base de données MySQL afin de **valider le fonctionnement du connecteur ODBC « asteriskdb-connector »** :

- Utiliser l'application « isql » pour se connecter à la base en utilisant « asteriskdb-connector » et afficher la 1<sup>ère</sup> colonne de la base de données :

```
echo "select 1" | isql -v asteriskdb-connector asterisk rtrtsql
```

Vous devriez obtenir le résultat suivant (ou au moins quelque chose de similaire) :

```
root@asterisk-server:~# echo "select 1" | isql -v asteriskdb-connector asterisk rtrtsql
+-----+
| Connected! |
+-----+
| sql-statement |
| help [tablename] |
| quit |
+-----+
SQL> select 1
+-----+
| 1 |
+-----+
| 1 |
+-----+
SQLRowCount returns 1
1 rows fetched
SQL> root@asterisk-server:~#
```

Maintenant que le connecteur logiciel ODBC est installé, configuré et fonctionne, il nous faut recompiler Asterisk pour créer et installer les modules spécifiques ODBC :

- Retourner dans le dossier où a été décompressé l'archive Asterisk (/home/user/asterisk-12.5.0 par exemple...), et relancer :

```
./configure
make menuselect
make install
```

Vérifier dans « menuselect » que les modules suivants relatifs à ODBC soient activés : cdr\_odbc, cdr\_adaptive\_odbc, func\_odbc, func\_realtime, pbx\_realtime, res\_config\_odbc et res\_odbc ; sélectionner également ODBC\_STORAGE dans le menu Voicemail Build Options afin que les messages vocaux ne soient plus enregistrés sous forme de fichiers locaux (dans /var/...) mais dans la base de données.

NB : vous pourrez vérifier après compilation que ces modules sont présents dans le dossier « /usr/lib/asterisk/modules ».

Il ne nous reste maintenant plus qu'à indiquer à Asterisk les paramètres du connecteur logiciel ODBC à utiliser pour se connecter et exploiter la base de données MySQL :

- Dans le fichier « /etc/asterisk/res\_odbc.conf », modifier les lignes suivantes :

```
[asteriskdb]
enabled => yes
dsn => asteriskdb-connector
username => asterisk
password => rtrtsql
pooling => no
limit => 1
pre-connect => yes
```

- Redémarrer Asterisk, et vérifier que la connexion avec la base de données fonctionne :

\*CLI> odbc show

Vous devez obtenir une réponse similaire à celle-ci-contre :

```
*CLI> odbc show
ODBC DSN Settings

Name: asterisk
DSN: asteriskdb-connector
Last connection attempt: 2014-11-08 04:07:24
Pooled: No
Connected: Yes
*CLI>
```

La connexion entre Asterisk et la base MySQL est maintenant fonctionnelle. Nous pouvons donc commencer à l'exploiter.

## XI.4. Enregistrement des postes SIP dans la base

Tous les postes SIP déclarés jusque-là dans le fichier « sip.conf » le seront désormais dans une table (que nous appellerons « sip ») de la base de données MySQL « asteriskdb »).

- Afin d'informer le serveur ToIP Asterisk du fait qu'il doit désormais chercher les **profils SIP aussi dans la base de données** (et plus uniquement dans le fichier « sip.conf »), ajouter à la fin du fichier « /etc/asterisk/extconfig.conf » :

sippeers => odbc,asteriskdb,sip

Pour **créer une table « sip » dans la base de données « asteriskdb »**, deux méthodes sont envisageables :

- 1<sup>ère</sup> méthode fastidieuse (archaïque) : utilisation de la commande mysql « create table », où il faut écrire tous les paramètres en une seule instruction, sans se tromper... ; ... nous ne ferons pas cette méthode, mais voici à quoi cela ressemble :

```
mysql> use asteriskdb
CREATE TABLE `sip` (
`id` int(11) NOT NULL auto_increment,
`name` varchar(80) NOT NULL default '',
`accountcode` varchar(20) default NULL,
`amaflags` varchar(13) default NULL,
`callgroup` varchar(10) default NULL,
```

```

`callerid` varchar(80) default NULL,
`canreinvite` char(3) default 'yes',
`context` varchar(80) default NULL,
`defaulttip` varchar(15) default NULL,
`dtmfmode` varchar(7) default NULL,
`fromuser` varchar(80) default NULL,
`fromdomain` varchar(80) default NULL,
`fullcontact` varchar(80) default NULL,
`host` varchar(31) NOT NULL default '',
`insecure` varchar(4) default NULL,
`language` char(2) default NULL,
`mailbox` varchar(50) default NULL,
`md5secret` varchar(80) default NULL,
`nat` varchar(5) NOT NULL default 'no',
`deny` varchar(95) default NULL,
`permit` varchar(95) default NULL,
`mask` varchar(95) default NULL,
`pickupgroup` varchar(10) default NULL,
`port` varchar(5) NOT NULL default '',
`qualify` char(3) default NULL,
`restrictcid` char(1) default NULL,
`rtptimeout` char(3) default NULL,
`rtpholdtimeout` char(3) default NULL,
`secret` varchar(80) default NULL,
`type` varchar(6) NOT NULL default 'friend',
`username` varchar(80) NOT NULL default '',
`disallow` varchar(100) default 'all',
`allow` varchar(100) default 'g729;ilbc:gsm;ulaw;alaw',
`musiconhold` varchar(100) default NULL,
`regseconds` int(11) NOT NULL default '0',
`ipaddr` varchar(15) NOT NULL default '',
`regexten` varchar(80) NOT NULL default '',
`cancallforward` char(3) default 'yes',
`setvar` varchar(100) NOT NULL default '',
PRIMARY KEY (`id`),
UNIQUE KEY `name` (`name`),
KEY `name_2` (`name`),
) TYPE=MyISAM ROW_FORMAT=DYNAMIC ;

```

Et dans ce cas, l'enregistrement de profils SIP se ferait comme ceci :

```
insert into sip (name,type,host,username,secret) values ('postxxx','friend',
'dynamic','nomxxx','pwdxxx') ;
```

La vérification de l'enregistrement dans la table « sip » se fait avec la commande :

```
select name,type,host,username,secret from sip ;
```

- une 2<sup>ème</sup> **méthode plus intelligente** : utilisation d'une **interface web pour créer et remplir les tables de données**, « sip » notamment ; pour cela, nous allons utiliser **phpmyadmin**, exploitable avec l'aide d'un serveur web « **apache2** » (que nous avons d'ailleurs déjà installé et utilisé dans ce projet). C'est cette méthode que nous allons utiliser maintenant :

- Installer « **phpmyadmin** » sur **MySQL-Server** (Attention : ne pas laisser dbconfig-common modifier la config de la base !)
- Dans « `/etc/apache2/apache2.conf` », rajouter la ligne suivante pour inclure dans le serveur apache2 la configuration de phpmyadmin :
 

```
Include /etc/phpmyadmin/apache.conf
```
- Dans « `/etc/apache2/apache2.conf` », vérifier l'autorisation d'accès à tous au dossier `/var/www/` et ses sous-dossiers :
 

```
<Directory /var/www/>
 Options Indexes FollowSymlinks
 AllowOverride None
 Require all granted
</Directory>
```
- Redémarrer le serveur Apache.  
NB : si un message d'erreur concernant le paramètre « `ServerName` » non affecté apparaît, il suffit de rajouter dans « `/etc/apache2/apache2.conf` » la ligne « `ServerName localhost` » par exemple.
- Ouvrir le navigateur Firefox de votre PC physique, composer l'URL « `192.168.1.i.13i/phpmyadmin/` » et se loguer avec les identifiants créés précédemment.

- Aller dans la base « `asteriskdb` » et créer la table « `sip` » contenant les éléments du tableau page suivante.

Nous allons maintenant déclarer nos 3 postes `3i01` à `3i03` dans la base de données :

- Toujours depuis le portail web de phpmyadmin, dans la table « `sip` », commencer par ajouter (onglet « `Insérer` ») le profil « `poste3i01` », aux paramètres conformes à ce que nous avions préalablement écrit dans le fichier « `sip.conf` ».
- En utilisant la fonction « `Copier` », créer également les profils `poste3i02` et `poste3i03`.

| Nom            | Type    | Taille/Valeurs     | Défaut        | A_I (Auto Incrémenté) |
|----------------|---------|--------------------|---------------|-----------------------|
| id             | INT     | 11                 |               | X                     |
| name           | VARCHAR | 50                 |               |                       |
| type           | ENUM    | 'friend','peer'    | friend        |                       |
| secret         | VARCHAR | 80                 |               |                       |
| host           | VARCHAR | 80                 | dynamic       |                       |
| context        | VARCHAR | 80                 | default       |                       |
| disallow       | VARCHAR | 80                 | all           |                       |
| allow          | VARCHAR | 80                 | alaw,gsm,ulaw |                       |
| dtmfmode       | ENUM    | 'rfc2833','inband' | rfc2833       |                       |
| language       | CHAR    | 2                  | fr            |                       |
| canreinvite    | ENUM    | 'yes','no'         | yes           |                       |
| deny           | VARCHAR | 80                 | NULL          |                       |
| permit         | VARCHAR | 80                 | NULL          |                       |
| callerid       | VARCHAR | 80                 | NULL          |                       |
| defaultuser    | VARCHAR | 80                 | NULL          |                       |
| fromuser       | VARCHAR | 80                 | NULL          |                       |
| fromdomain     | VARCHAR | 80                 | NULL          |                       |
| insecure       | VARCHAR | 80                 | invite        |                       |
| callgroup      | VARCHAR | 80                 | NULL          |                       |
| pickupgroup    | VARCHAR | 80                 | NULL          |                       |
| mailbox        | VARCHAR | 80                 | NULL          |                       |
| nat            | ENUM    | 'yes','no'         | no            |                       |
| ipaddr         | VARCHAR | 80                 | NULL          |                       |
| mask           | VARCHAR | 80                 | NULL          |                       |
| port           | VARCHAR | 80                 | NULL          |                       |
| cancallforward | ENUM    | 'yes','no'         | yes           |                       |
| accountcode    | VARCHAR | 80                 | NULL          |                       |
| fullcontact    | VARCHAR | 80                 | NULL          |                       |
| lastms         | VARCHAR | 80                 | NULL          |                       |
| useragent      | VARCHAR | 80                 | NULL          |                       |
| regseconds     | VARCHAR | 80                 | NULL          |                       |
| regserver      | VARCHAR | 80                 | NULL          |                       |

Vérifions le fonctionnement des postes sur le serveur :

- Si tout fonctionne normalement, nous n'avons plus besoin des descriptions dans le fichier « sip.conf » ; pour s'en assurer, mettre en commentaire les lignes concernant la définition des postes.
- Redémarrer Asterisk, ainsi que le poste 6757i (plutôt capricieux).
- Tester les appels entre les 3 postes SIP : l'essai est-il concluant ?

Rq : si vous utilisez la commande « `sip show peers` », ne vous inquiétez pas si vous ne voyez plus les 3 profils poste3i0x, car cette commande ne s'intéresse qu'aux profils SIP déclarés dans les fichiers de configuration locaux.

Il nous reste à nous occuper de nos 2 derniers profils SIP, utilisés pour les **Liens trunk SIP** : « `serverk` » et « `mon_provider_sip` » :

- Ajouter dans la table « `sip` » 2 autres entrées, pour les profils trunk SIP déclarés au préalable dans « `sip.conf` » pour **[serverk]** et **[mon-provider-sip]** ; penser également à mettre en commentaire ces sections dans « `sip.conf` ».
- Après avoir redémarré Asterisk, vérifier que les enregistrements vers les 2 serveurs distants (ou au moins « `mon-provider-sip` ») a été correctement effectuée, à l'aide de la commande suivante :

```
sip show registry
```

## XI.5. Enregistrement du plan de num dans la base

Le comportement du serveur de téléphonie, défini jusque-là dans le fichier « `extensions.conf` », sera désormais décrit et accessible dans la table « `extensions` » que nous allons créer dans la base « `asteriskdb` ».

- Afin d'informer le serveur ToIP Asterisk du fait qu'il doit désormais **aussi** chercher les actions liées aux extensions **dans la base de données** (et plus exclusivement dans le fichier « `extensions.conf` »), ajouter à la fin du fichier « `/etc/asterisk/extconfig.conf` » :  
`extensions => odbc,asteriskdb,extensions`
- Dans la base « `asteriskdb` », créer une 2<sup>ème</sup> table que vous appellerez « `extensions` », contenant les éléments ci-dessous :

| Nom      | Type    | Taille/Valeurs | Défaut  | A_I (Auto Incrémenté) |
|----------|---------|----------------|---------|-----------------------|
| id       | INT     | 11             |         | X                     |
| context  | VARCHAR | 50             | default |                       |
| exten    | VARCHAR | 50             | NULL    |                       |
| priority | INT     | 4              | 0       |                       |
| app      | VARCHAR | 50             | NULL    |                       |
| appdata  | VARCHAR | 255            | NULL    |                       |

Nous allons modifier le contenu du fichier « `extensions.conf` » afin de n'y garder que les configurations d'extensions particulières, mais nous allons passer tout le traitement des extensions classiques des postes dans la base de données :

- Dans la section `[plan-num-prive]`, commentez toutes les lignes concernant les extensions 3i01, 3i02 et 3i03 des postes internes. Ajouter la ligne suivante dans cette section :

```
include => postes-internes
```

- Rajouter avant la section [plan-num-prive] la section suivante :
 

```
...
[postes-internes]
switch => Realtime/postes-internes@
```
- En vous inspirant de ce que nous avions écrit au préalable dans « extensions.conf », créer dans la table « extensions » de la base 3 entrées pour décrire l'action à effectuer (« Dial(...) ») pour les extensions 3i01 à 3i03 (contexte « postes-internes »).
- Tester les appels entre les 3 postes.
- Rajouter des lignes supplémentaires pour chaque extension afin de gérer le basculement sur la messagerie lorsque l'abonné n'est pas joignable.

## XI.6. Exportation des boîtes vocales dans la base

Nous allons maintenant exporter les données concernant les boîtes vocales dans la base de données afin que, comme pour les profils SIP et le plan de numérotation des postes internes, toutes les modifications effectuées dessus soient prises en considération en live, sans avoir besoin de redémarrer le service adéquat d'Asterisk (ou Asterisk en entier).

- Afin d'informer le serveur ToIP Asterisk du fait que les profils des **boîtes vocales** sont désormais aussi hébergées **dans la base de données** (et non plus exclusivement dans le fichier « voicemail.conf »), ajouter à la fin de « /etc/asterisk/extconfig.conf » :
 

```
voicemail => mysql,asteriskdb,voicemail_users
```
- Dans la base « asteriskdb », créer la table « voicemail\_users » pour stocker les déclarations des boîtes vocales, contenant les éléments ci-dessous :

| Nom      | Type    | Taille/Valeurs | Défaut  | A_I (Auto Incrémenté) |
|----------|---------|----------------|---------|-----------------------|
| id       | INT     | 11             |         | X                     |
| name     | VARCHAR | 80             |         |                       |
| context  | VARCHAR | 80             | default |                       |
| mailbox  | INT     | 5              |         |                       |
| password | VARCHAR | 80             |         |                       |
| fullname | VARCHAR | 80             | NULL    |                       |
| email    | VARCHAR | 80             | NULL    |                       |

Nous allons modifier le contenu du fichier « /etc/asterisk/voicemail.conf » pour que les configurations précédentes des boîtes vocales ne soient plus prises en compte :

- Dans la section [default], commentez toutes les lignes concernant les extensions 3i01, 3i02 et 3i03 des postes internes.
- En vous inspirant de ce que nous avions écrit au préalable dans « voicemail.conf », créer 3 entrées dans la table « voicemail\_users » pour les 3 boîtes vocales des postes 3i01 à 3i03.
- Tester l'accès aux boîtes vocales des 3 postes et le dépôt de messages vocaux entre eux.

La définition des boîtes vocales et l'authentification aux services de messagerie se fait désormais via la base de données. Seul le stockage des messages se fait encore sur le serveur Asterisk, mais quoiqu'il arrive, les messages vocaux étant envoyés par mail aux abonnés, les messages sont en sécurité sur le serveur IMAP.

Nous allons créer une dernière table pour la messagerie vocale, pour inventorier les messages vocaux des abonnés ; les messages vocaux seront toujours stockés comme actuellement dans « /var/spool/asterisk/voicemail/ », la base de données ne contiendra qu'un listing des messages, avec leurs propriétés et également la possibilité de les écouter :

- Dans « /etc/asterisk/voicemail.conf », modifier les paramètres suivants :
 

```
odbstorage = asteriskdb
odbtable = voicemail_msg
```
- Dans la base « asteriskdb », créer la table « voicemail\_msg » pour stocker les messages vocaux, contenant les éléments ci-dessous :

| Nom            | Type     | Taille/Valeurs | Défaut | A_I (Auto<br>Incrémenté) |
|----------------|----------|----------------|--------|--------------------------|
| id             | INT      | 11             |        | X                        |
| mailboxuser    | VARCHAR  | 80             | NULL   |                          |
| dir            | VARCHAR  | 80             | NULL   |                          |
| msg_id         | VARCHAR  | 40             | NULL   |                          |
| recording      | LONGBLOB |                | NULL   |                          |
| mailboxcontext | VARCHAR  | 80             | NULL   |                          |
| callerid       | VARCHAR  | 40             | NULL   |                          |
| origtime       | VARCHAR  | 40             | NULL   |                          |
| duration       | VARCHAR  | 20             | NULL   |                          |
| context        | VARCHAR  | 80             | NULL   |                          |
| macrocontext   | VARCHAR  | 80             | NULL   |                          |
| read           | BOOLEAN  |                | false  |                          |
| msgnum         | INT      | 11             | NULL   |                          |
| flag           | VARCHAR  | 10             | NULL   |                          |

- Laisser un message sur la boîte vocale d'un des abonnés. Retrouvez-vous une nouvelle entrée dans la table « voicemail\_msg » ?

Maintenant qu'il repose sur une base de données, notre serveur de téléphonie privée est maintenant beaucoup plus fonctionnel et opérationnel. Un dernier service très demandé et prisé en environnement professionnel reste néanmoins à mettre en place : un service de journalisation et taxation des appels.

## XII. Listing des appels - Taxation

Dans un réseau téléphonique, il est important d'avoir une traçabilité du trafic téléphonique, afin notamment de gérer le coût de fonctionnement global de l'entreprise, mais aussi de maîtriser les consommations par service. Un **outil de journalisation ou de taxation des appels** est donc indispensable pour avoir une bonne maîtrise de son réseau téléphonique privé. Un **service de taxation des appels**, éditant des « **tickets de communication** », sera d'ailleurs **indispensable pour un hôtel ou un hôpital** par exemple. Nous allons tenter de déployer ce type de service pour notre entreprise.

Le système **CDR (Call Detail Records)** d'Asterisk est utilisé pour mémoriser (log) l'historique des appels dans le système. Dans certains déploiements, ces documents sont utilisés à des fins de **facturation**. Dans d'autres, les enregistrements d'appels sont utilisés pour **analyser les volumes d'appels** dans le temps. Ils peuvent également être utilisés comme un outil de débogage par les administrateurs.

Les enregistrements des détails d'appels (CDR) contiennent des informations sur les appels passés par le serveur Asterisk. Le stockage CDR à l'aide d'une base de données est un usage populaire dans Asterisk, car cela rend la gestion beaucoup plus facile. Placer les enregistrements dans une base de données ouvre de nombreuses possibilités comme :

- la construction de sa propre interface Web pour le suivi des statistiques telles que l'utilisation d'appels et la localisation des destinataires,
- la facturation, ou la vérification de la facture de l'opérateur téléphonique routant ses appels.

Le stockage des enregistrements détaillés des appels se fait via le module « `cdr_adaptive_odbc` », qui permet de choisir et personnaliser les colonnes de données intégrées dans sa table. La table peut même être alimentée grâce à la fonction de plan de numérotation `CDR()` : par exemple, si on voulait implémenter le coût par minute du routage d'un appel externe, on pourrait rajouter une colonne « `cout_par_min` » dans la table « `cdr` », et dans le plan de num, rajouter une instruction (avant le `Dial(...)`) du genre « `Set(CDR(cout_par_min)=0.005)` ».

Configurons l'utilisation du Call Detail Record sur notre serveur Asterisk :

- Dans « `/etc/asterisk/cdr_adaptive_odbc.conf` », ajouter les lignes suivantes :

```
[mysql_cdr_connection]
connection=asteriskdb
table=cdr
```
- Dans la base « `asteriskdb` », créer la table « `cdr` » pour stocker l'historique des appels, contenant les éléments listés dans le tableau page suivante.
- Vérifier que la connexion CDR avec la base est opérationnelle :

```
cdr show status
```
- Passer plusieurs appels, en interne et également via l'opérateur public, puis vérifier que la liste détaillée des appels est contenue dans la table « `cdr` ».

| Nom         | Type     | Taille/Valeurs | Défaut              | A_I (Auto<br>Incrémenté) |
|-------------|----------|----------------|---------------------|--------------------------|
| id          | INT      | 11             |                     | X                        |
| calldate    | DATETIME | 80             | 0000-00-00 00:00:00 |                          |
| clid        | VARCHAR  | 80             |                     |                          |
| src         | VARCHAR  | 80             |                     |                          |
| dst         | VARCHAR  | 80             |                     |                          |
| dcontext    | VARCHAR  | 80             |                     |                          |
| channel     | VARCHAR  | 80             |                     |                          |
| dstchannel  | VARCHAR  | 80             |                     |                          |
| lastapp     | VARCHAR  | 80             |                     |                          |
| lastdata    | VARCHAR  | 80             |                     |                          |
| duration    | INT      | 11             | 0                   |                          |
| billsec     | INT      | 11             | 0                   |                          |
| disposition | VARCHAR  | 45             |                     |                          |
| amaflags    | INT      | 11             |                     |                          |
| accountcode | VARCHAR  | 20             |                     |                          |
| uniqueid    | VARCHAR  | 32             |                     |                          |
| userfield   | VARCHAR  | 255            |                     |                          |
| peeraccount | VARCHAR  | 20             |                     |                          |
| linkedid    | VARCHAR  | 32             |                     |                          |
| sequence    | INT      | 11             | 0                   |                          |

That's all folks !!! Nous n'irons pas plus loin, mais c'est déjà pas mal... ;-)

Pour information, un produit libre comme [a2billing](http://www.asterisk2billing.org) ([www.asterisk2billing.org](http://www.asterisk2billing.org)) offre une solution évoluée intéressante pour la **gestion du trafic téléphonique et la taxation des appels**. Il est basé sur le CDR d'Asterisk. Quelques screenshots de ce produit :

The image displays four screenshots of the a2Billing software interface, showcasing its reporting and analytical capabilities:

- Screenshot 1 (Top Left): Agent List**  
Shows a list of agents with access to the system. It includes columns for Agent ID, Name, and Last Activity.
- Screenshot 2 (Top Right): Call History Report**  
A detailed report table showing call records. Columns include Date, CallID, DID, Phone Number, Duration, and various financial metrics like Total, Average, Std Dev, Margin, and Weight.
- Screenshot 3 (Bottom Left): Accounts Info**  
Provides an overview of account statistics. It shows the total number of accounts (372), new accounts (18), and accounts suspended (25). It also includes sections for Refills Info, Calls Info Today, and Payments Info.
- Screenshot 4 (Bottom Right): Call Volume by Hour**  
A bar chart titled "Mastodon - Load by Hours" showing call volume over a 24-hour period. The x-axis represents hours from 0 to 23, and the y-axis represents the count of calls.

## XIII. Conclusion

Comme pour chaque compte-rendu, conclure sur ce que vous avez appris et retenu d'important dans ce TP-projet.

Et n'oubliez-pas vos remarques et impressions sur ce projet, toutes les critiques (objectives !) étant constructives...

*Rappel : votre compte-rendu est à déposer au plus tard 7 jours après votre dernière séance.*