

IUT DE COLMAR

SAE3 - ROM

ANNÉE 2022-23

Déployer un service de téléphonie

MARTIN BAUMGAERTNER

4 février 2023

Table des matières

1 Objectif 1	3
1.1 Configuration Serveur IPBX	3
1.1.1 Créer une machine virtuelle Debian11	3
1.1.2 Installez l'IPBX asterisk via le système de paquets Debian	3
1.1.3 Pour le service IPBX, déclarez les différents postes SIP	4
1.1.4 Pour le service IPBX, créer le plan de numérotation	7
1.2 Configuration Téléphone Clients	7
1.2.1 Téléphone logiciel Linphone	7
1.2.2 Téléphone matériel Nortel LIP6812	8
1.2.3 Téléphone Matériel Cisco 7941G	10
1.3 Validation des appels	11
2 Objectif 2 : Configuration inter-sites	12
2.1 Appels inter-sites	12
2.1.1 Sur l'IPBX, inscrire le faisceau SIP	12
2.1.2 Dans le plan de numérotation, ajoutez le préfix *	12
2.1.3 Validez les appels entrants et sortants entre cabinets médicaux	13
2.2 Appels externes	13
2.2.1 Ajoutez les numéros à 10 chiffres pour les appels extérieurs via le faisceau	13
2.2.2 Validez les appels externes vers (0112345678)	13
3 Objectif 3 : Fonctions téléphoniques	14
3.1 Transfert d'appels	14
3.1.1 Activez le transfert d'appels côté IPBX	14
3.1.2 Validez le fonctionnement dans l'ensemble des cas possibles .	14
3.1.3 Pour chaque téléphone, définissez un numéro de transfert .	15
3.2 Mise en attente	16
3.2.1 Trouvez une mise en attente personnalisée	16
3.2.2 Encodez la musique dans le format approprié	17
3.3 Interception d'appels	18
3.3.1 Définir un groupe d'appel Assistant et Secrétaire	18
3.3.2 Activez l'interception d'appel pour le groupe précédent . . .	18
3.3.3 Validez l'interception d'appels	19
3.4 Enregistrement de conversation	19
3.4.1 Activez l'enregistrement de conversation côté serveur	19
3.5 Prédécroché	20
3.5.1 Enregistrez un message d'accueil personnalisé	20

4	Objectif 4 : Boîtes vocales	21
4.1	Configuration serveur	21
4.1.1	Créer une boîte vocal pour chaque utilisateurs	21
5	Conclusion générale	22

Contexte

L'objectif de cette SAE et de créer un service de téléphonies "multi-sites". Je m'explique. Nous allons devoir créer, un serveur Asterisk qui est un serveur de téléphonie IP. Ce serveur est installé sur une machine virtuelle Debian11. Nous avons aussi à disposition, 2 téléphones IP matériel qui sont des téléphones SIP. Ces téléphones sont connectés dans le même réseau que notre serveur IPBX bien entendu. Puis, nous avons un troisième téléphone IP mais cette fois-ci sous forme de softphone qui est un téléphone SIP qui est installé sur un ordinateur. Nous utiliserons Linphone. Ce dernier est un logiciel libre et gratuit. Il est disponible sur Windows, Linux et Mac. Ce softphone est connecté sur le même réseau que notre serveur IPBX.

Chaque membre de la SAE doit choisir un contexte de travail différent. J'ai pour ma part choisi d'être la table 1, le médecin généraliste. Nous avons donc ces 3 postes IP pour illustrer un vrai cabinet. C'est-à-dire que nous avons un poste physique pour le praticien, un poste physique pour l'assistant et le softphone pour le secrétariat.

1 Objectif 1

1.1 Configuration Serveur IPBX

1.1.1 Créer une machine virtuelle Debian11

Pour cette SAE, j'ai décidé de choisir d'utiliser VMWare Workstation car c'était le logiciel que nous utilisions en cours surtout pendant les TP réseaux en première année. J'ai donc créé une VM Debian11.

1.1.2 Installez l'IPBX asterisk via le système de paquets Debian

Pour pouvoir installer asterisk, il faut d'abord installer le paquet suivant :

```
sudo apt install asterisk
```

Une fois Asterisk installé, pour l'utiliser il faut arrêter le service pour pouvoir démarrer Asterisk car sinon il y a un conflit. Pour arrêter le service, il faut écrire la commande suivante : `service asterisk stop`. Ensuite, nous pouvons démarrer Asterisk avec la commande suivante : `asterisk -vvvvc`.

1.1.3 Pour le service IPBX, déclarez les différents postes SIP

Pour déclarer les différents postes SIP, j'ai donc écrits les commandes suivantes dans les fichiers de configuration de Asterisk.

Premièrement, **pjsip.conf** :

```
[transport-udp]
type=transport
protocol=udp      ;udp,tcp,tls,ws,wss
bind=0.0.0.0
```

FIGURE 1 – Configuration de pjsip.conf

Ce sont des entrées de configuration pour la définition d'un transport PJSIP dans le fichier de configuration d'Asterisk.

- **[transport-udp]** définit un nom pour le transport, "transport-udp" en l'occurrence.
- **type = transport** indique le type de l'objet de configuration. Ici, il s'agit d'un transport PJSIP.
- **protocol = udp** définit le protocole de transport utilisé, en l'occurrence l'UDP (User Datagram Protocol).
- **bind = 0.0.0.0** définit l'interface réseau sur laquelle le transport écoutera les connexions. La valeur "0.0.0.0" signifie que le transport écoutera sur toutes les interfaces disponibles.

Puis, pjsip wizard.conf :

```
[TM1]
type = wizard
accepts_auth = yes
accepts_registrations = yes
transport = transport-udp
inbound_auth/type = userpass
inbound_auth/username = TM1
inbound_auth/password = toto
endpoint/allow = !all,alaw
aor/max_contacts = 1
aor/remove_existing = 1

[TM1BIS]
type = wizard
accepts_auth = yes
accepts_registrations = yes
transport = transport-udp
inbound_auth/type = userpass
inbound_auth/username = TM1BIS
inbound_auth/password = toto
endpoint/allow = !all,alaw
aor/max_contacts = 1
aor/remove_existing = 1

[TL1]
type = wizard
accepts_auth = yes
accepts_registrations = yes
transport = transport-udp
inbound_auth/type = userpass
inbound_auth/username = TL1
inbound_auth/password = toto
endpoint/allow = !all,alaw
aor/max_contacts = 1
aor/remove_existing = 1
```

FIGURE 2 – Configuration de pjsip wizard.conf

Nous avons ici donc la configuration de tous les téléphones. A chaque fois, j'ai donc déclaré un téléphone avec son numéro de téléphone, son mot de passe et son nom d'utilisateur. Certaines lignes permettent de faire différentes choses essentielles au bon fonctionnement du serveur Asterisk, je vais détailler toutes les explications à la page suivante.

Explication des lignes de configuration

Je vais prendre pour exemple l'utilisateur **[TL1]** car c'est après assez similaire pour les autres déclarations d'utilisateurs.

- **[TL1]** : définit un nom pour l'endpoint, "TL1" en l'occurrence.
- **accepts_auth = yes** et **accepts_registrations = yes** autorisent l'authentification et l'enregistrement des utilisateurs pour cet endpoint.
- **transport = transport-udp** définit le transport PJSIP à utiliser pour acheminer les appels à travers le réseau pour cet endpoint.
- **inbound_auth/type = userpass** définit le type d'authentification pour les appels entrants à cet endpoint, en l'occurrence l'authentification par nom d'utilisateur et mot de passe.
- **inbound_auth/username = TL1** et **inbound_auth/password = toto** définissent le nom d'utilisateur et le mot de passe pour l'authentification.
- **endpoint/allow = !all,alaw** définit les codecs audio autorisés pour les appels entrant et sortant à partir de cet endpoint. **alaw** est un codec audio couramment utilisé et **!all** signifie que tous les autres codecs ne sont pas autorisés.
- **aor/max_contacts = 1** et **aor/remove_existing = 1** définissent les paramètres de gestion des connexions pour cet endpoint. **Aor/max_contacts = 1** signifie qu'un seul contact peut être enregistré pour cet endpoint à tout moment, tandis que **aor/remove_existing = 1** signifie que les connexions existantes seront supprimées lorsqu'une nouvelle connexion est établie.
- **endpoint/call_group = 1** et **endpoint/pickup_group = 1** définissent les groupes d'appels et de prise en charge pour cet endpoint. Les numéros d'identification des groupes peuvent varier selon votre configuration.

Un **endpoint** est peut être un téléphone IP, un softphone, une interface de téléphonie sur un ordinateur, ou tout autre appareil capable de communiquer avec le système de téléphonie Asterisk.

1.1.4 Pour le service IPBX, créer le plan de numérotation

J'ai donc crée le plan de numérotation de la même manière que nous l'avons vu en cours et en TP comme peut en témoigner la capture d'écran suivante en rajoutant bien entendu ces lignes dans le contexte [default] :

```
exten => 555,1,Dial(PJSIP/TM1)
exten => 556,1,Dial(PJSIP/TM1BIS)
exten => 557,1,Dial(PJSIP/TL1)
```

FIGURE 3 – Configuration de extensions.conf

J'ai choisi de réutiliser dans un premier temps les numéros que l'on utilisait en cours pour me faciliter la compréhension. Ce sont des numéros, que j'ai à terme modifié en 1, 2 et 3.

Exten => 555,1 définit le numéro de l'extension à 555. Lorsqu'un appel est effectué à cet extension, les instructions suivantes seront exécutées. **Dial(PJSIP/TM1)** est l'instruction pour composer un appel. **Dial** est une commande intégrée d'Asterisk qui permet de composer un appel vers une destination donnée. **PJSIP/TM1** est la destination de l'appel.

PJSIP désigne le protocole de téléphonie IP utilisé (dans ce cas, PJSIP) et **TM1** est le nom de l'endpoint ou de l'utilisateur auquel l'appel doit être dirigé.

1.2 Configuration Téléphone Clients

1.2.1 Téléphone logiciel Linphone

Tout d'abord j'ai voulu configurer linphone sur la machine windows de l'IUT sur laquelle j'ai fait ma VM Debian. Cependant, j'ai eu de nombreux problèmes Linphone ne fonctionnait pas, même après avoir essayé plusieurs versions. J'ai donc essayé d'utiliser mon ordinateur personnel pour faire cette manipulation. J'ai donc installé Linphone sur mon ordinateur personnel et ça a fonctionné. Pour faire marcher Linphone j'ai utilisé la configuration suivante : *Où 10.129.10.164 est l'IP de mon serveur Asterisk*

UTILISER UN COMPTE SIP

Nom d'utilisateur	Nom d'affichage (optionnel)
<input type="text" value="TL1"/>	<input type="text" value="TL1"/>
Domaine SIP	
<input type="text" value="10.129.10.164"/>	
Mot de passe	<input type="password" value="....."/>
Transport	<input type="text" value="UDP"/> ▼
<input type="button" value="RETOUR"/> <input type="button" value="UTILISER"/>	

FIGURE 4 – Configuration de Linphone

1.2.2 Téléphone matériel Nortel LIP6812

Inscrivez le téléphone auprès du registrar de l'IPBX

Pour enregistrer le téléphone IP LG Nortel auprès du serveur IPBX, j'ai du le configurer directement depuis son interface pour rentrer tous les paramètres pour qu'il puisse se connecter.

Voici les différents paramètres que j'ai dû configurer pour que le téléphone remonte : J'ai du rentrer dans les paramètres du téléphone en appuyant sur la touche **settings**, puis choisir le sous menu **2. SIP configuration**



FIGURE 5 – Choix menu SIP configuration

Ensuite, je viens sélectionner la ligne 1. **Line 1 Settings**



FIGURE 6 – Choix menu Line 1 settings

Pour finir, je viens configurer l'ensemble des paramètres qui sont inscrits comme par exemple, l'adresse IP du serveur IPBX, le nom d'utilisateur, le mot de passe ainsi de suite...

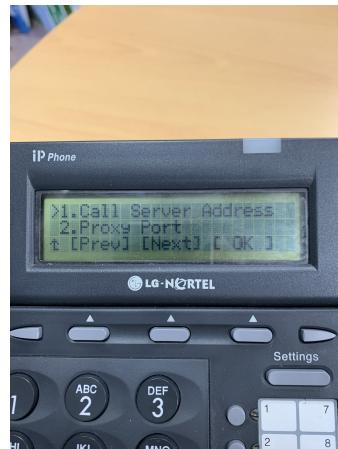


FIGURE 7 – Configuration du Nortel

Une fois configuré correctement le téléphone remonte automatiquement auprès du serveur IPBX et nous pouvons l'utiliser pour appeler en interne.

1.2.3 Téléphone Matériel Cisco 7941G

Etudiez la procédure d'installation d'un nouveau firmware

Pour installer un nouveau firmware sur le Cisco 7941G. Il a d'abord fallu monter un serveur TFTP afin de pouvoir envoyer le firmware sur le téléphone. Nous avons donc monter un serveur TFTP à l'aide de mes camarades, sur windows comme le démontre la photo qui suit :

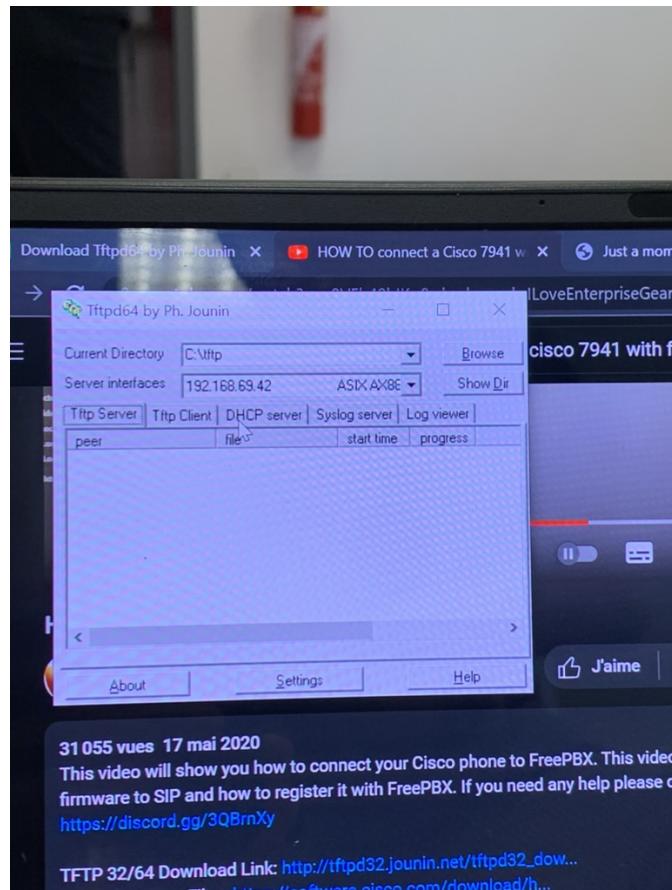


FIGURE 8 – Configuration du serveur TFTP

Puis, nous avons envoyé la configuration dans les téléphones. L'envoie du nouveau firmware a donc bel et bien fonctionné. Cependant, nous avons eu des problèmes pour configurer le téléphone pour pouvoir l'enregistrer auprès du serveur Astetisk. Etant bloqués par la configuration du téléphone. Nous n'avons pas réussi à l'enregistrer. Même après avoir essayé de différentes manières, comme en essayant plusieurs firmwares.

Comme nous pouvons le constater sur la photo qui suit, nous avons bien réussi à flasher les téléphones. Pour pouvoir les tenir prêt au flash il a fallu maintenir le bouton "#" pendant le démarrage du téléphone. Une fois que les boutons de volumes commençaient à clignoter il a fallu faire succéder les touches suivantes : **1, 2, 3, 4, 5, 6, 7, 8, 9, 0** Et, nous rentrions dans le mode de flashage comme le montre la photo qui suit :

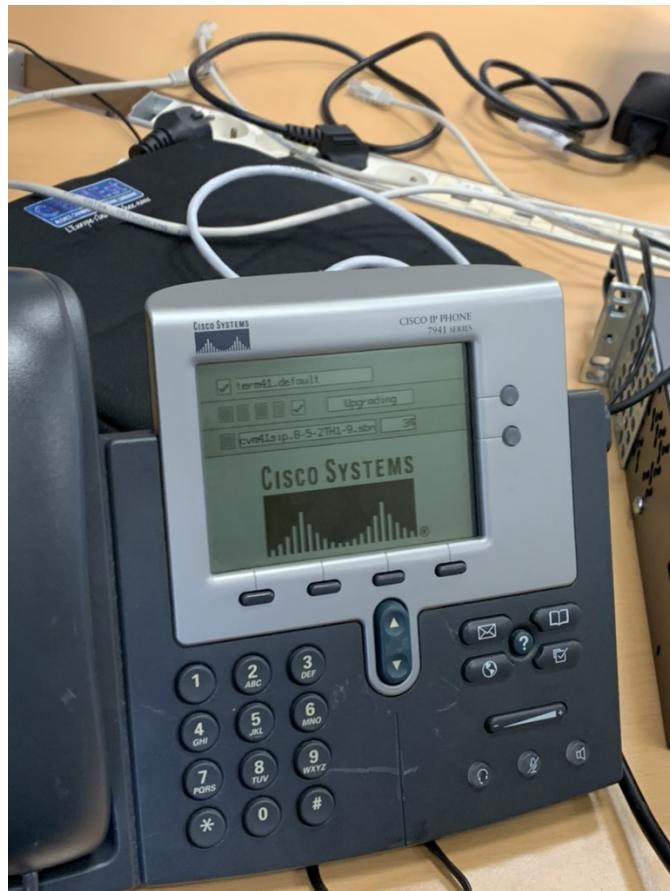


FIGURE 9 – Entrée du téléphone en mode flash

1.3 Validation des appels

Nous avons donc testé les différents appels internes possibles, et tout est fonctionnel, j'arrive bien à joindre la secrétaire, depuis le softphone et inversément.

2 Objectif 2 : Configuration inter-sites

2.1 Appels inter-sites

2.1.1 Sur l'IPBX, inscrire le faisceau SIP

Pour inscrire mon faisceau SIP auprès de l'opérateur Voix, j'ai configuré un nouveau contexte **[operateurvoix]** et que j'ai configuré comme ci-après :

```
[operateurvoix]
type = wizard
sends_auth = yes
sends_registrations = yes
remote_hosts = 10.129.10.20
outbound_auth/username=table1
outbound_auth/password = toto
endpoint/from_user = table1
endpoint/from_domain = 10.129.10.20
endpoint/allow=alaw
endpoint/call_group = 1
endpoint/pickup_group = 1
```

FIGURE 10 – Configuration de l'opérateur Voix

Je définis à chaque fois les bons paramètres que je vais expliquer ci-après :

- **remote_hosts** : Définis l'adresse IP du serveur Opérateur Voix
- **outbound_auth** : Définis le nom d'utilisateur et le mot de passe pour communiquer à travers le serveur voix

2.1.2 Dans le plan de numérotation, ajoutez le préfix *

Dans mon fichier **extensions.conf**, j'ai donc rajouté la ligne suivante dans le contexte **[default]** :

```
exten=>_*.1,Dial(PJSIP/${EXTEN:1}@operateurvoix)
```

FIGURE 11 – Configuration de l'opérateur voix dans extensions.conf

- **exten => _*.1** signifie que cette extension est associée à tout numéro composé, quelle que soit la longueur et les chiffres qui le composent. Le "" signifie n'importe quel nombre et le "." signifie n'importe quelle longueur.
- **Dial(PJSIP/\${EXTEN:1}@operateurvoix)** signifie que l'appel sera achevé via le protocole PJSIP en utilisant l'opérateur de voix défini. La variable "\${EXTEN:1}" est utilisée pour extraire le numéro composé par l'utilisateur, en excluant le premier caractère qui est un "_".

2.1.3 Validez les appels entrants et sortants entre cabinets médicaux

Suite à la configuration de l'opérateur voix, j'ai pu tester les appels entrants et sortants entre les différents cabinets de mes camarades. Tout est fonctionnel. Si par exemple je veux joindre Victor Uetwiller à la table7. Je compose **72 et j'arrive bien à l'appeler.

2.2 Appels externes

2.2.1 Ajoutez les numéros à 10 chiffres pour les appels extérieurs via le faisceau

Pour pouvoir appeler à l'extérieur de mon faisceau les appels à 10 chiffres, j'ai ajouté la ligne suivante dans le fichier **extensions.conf** :

```
exten=>_0.,1,Dial(PJSIP/${EXTEN}@operateurvoix)
```

FIGURE 12 – Configuration des appels à 10 chiffres extensions.conf

- **exten=>_0.,1** définit le numéro de l'extension avec un masque de correspondance général, où **_0.** signifie que cette extension sera activée pour tout numéro de téléphone commençant par **0**.
- **Dial(PJSIP/\${EXTEN}@operateurvoix)** est l'instruction pour composer un appel. **Dial** est une commande intégrée d'Asterisk qui permet de composer un appel vers une destination donnée. **PJSIP/\${EXTEN}@operateurvoix** est la destination de l'appel. **PJSIP** désigne le protocole de téléphonie IP utilisé (dans ce cas, PJSIP) et **\$EXTEN** est une variable qui contient le numéro d'appel entrant (sans le "0" initial). **Operateurvoix** est le nom d'un serveur ou d'un fournisseur de services de téléphonie IP auquel l'appel doit être dirigé.

2.2.2 Validez les appels externes vers (0112345678)

J'ai donc testé d'appeler les appels vers le 0112345678, le numéro hébergé par Victor sur son PC. Et l'appel fonctionne parfaitement.

3 Objectif 3 : Fonctions téléphoniques

3.1 Transfert d'appels

3.1.1 Activez le transfert d'appels côté IPBX

Pour activer le transfert d'appels côté IPBX, j'ai donc modifié les lignes suivante dans le fichier **extensions.conf** :

```
exten => 1,1,Dial(PJSIP/TM1,30,tT)
exten => 2,1,Dial(PJSIP/TM1BIS,30,tT)
exten => 3,1,Dial(PJSIP/TL1,30,tT)
```

FIGURE 13 – Configuration du transfert extensions.conf

Il ne faut aussi pas oublier de décommenter la ligne **blindxfer => #1** dans le fichier **features.conf** et dans le context **[featuremap]**.

J'ai modifié le plan de numérotation des téléphones en ajoutant l'option **tT** pour le transfert d'appels. Désormais quand je suis en appel et que je compose la touche **#1**, suivis du numéro de téléphone vers lequel je veux transférer l'appel et que je valide, l'appel est transféré.

En utilisant les options **tT** dans la ligne de configuration, j'autorisez les deux méthodes de transfert d'appel, soit en utilisant le code de transfert ou en utilisant la touche de transfert rapide.

3.1.2 Validez le fonctionnement dans l'ensemble des cas possibles

Pour tester les transferts d'appel, j'ai tout d'abord testé en interne de transférer mon appel vers un autre téléphone logiciel. Ce qui a fonctionné. Puis, pour compléxifier un peu le sujet, j'ai demandé à un camarade de m'appeler. J'ai receptionné l'appel et j'ai transféré l'appel vers une autre table. Ce qui a également fonctionné.

3.1.3 Pour chaque téléphone, définissez un numéro de transfert

Pour définir une touche rapide de transfert sur le LG Nortel. Je suis rentrer dans les paramètres, puis, j'ai cliqué sur le bouton numéro 3 qui correspond à **Phone settings** :

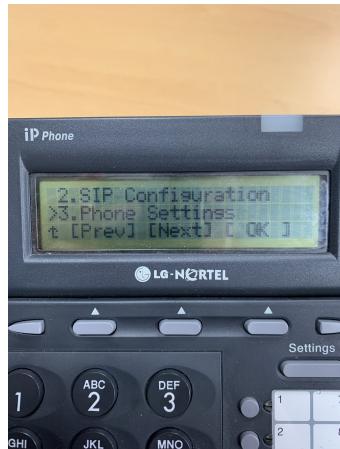


FIGURE 14 – Phone settings

Puis, je me suis rendu dans le sous-menu 6, qui correspond à **Flexible Key Settings** :



FIGURE 15 – Flexible Key Settings

J'ai ensuite choisi sur la touche que je voulais attribuer à ce raccourci en cliquant sur la touche que je souhaitais :



FIGURE 16 – Bouton à associer

Une fois fait je choisi le numéro que je veux associer à ce bouton et je valide. La procédure est exactement la même pour ajouter en raccourci un contact, le bouton de mise en attente ou encore la bouton pour accéder à la messagerie. C'est pour cette raison que je n'expliquerai pas comment faire après quand il faudra ajouter le bouton de mise en attente.

3.2 Mise en attente

3.2.1 Trouvez une mise en attente personnalisée

Pour la musique d'attente, j'ai décidé de la créer moi-même, en m'enregistrant avec mon téléphone. Puis que un célèbre logiciel de montage **Adobe Premiere Pro** j'ai rajouté une musique de fond pour donner un aspect plus professionnel.

3.2.2 Encodez la musique dans le format approprié

Il faut savoir que Asterisk n'accepte pas le mp3. J'ai donc converti mon fichier en g722. Qui est un codec que Asterisk accepte. Pour ce faire, je me suis rendu sur le site web <https://g711.org>. Sur ce dernier j'ai donc converti mon audio en g722.

Définissez la musique par défaut

Pour définir la musique d'attente par défaut. J'ai tout simplement modifié premièrement dans le fichier **musiconhold.conf** le contexte **[default]** :

```
[default]
mode=files
directory=/var/lib/asterisk/mohwav
random=yes
```

FIGURE 17 – Configuration du fichier musiconhold.conf

Voici quelques explications pour mieux comprendre ce que ces lignes de codes veulent dire :

- **mode=files** : définit le mode de lecture de la musique d'attente. Dans ce cas, le mode est "files", ce qui signifie que la musique sera lue à partir de fichiers audio.
- **directory=/var/lib/asterisk/mohwav** : définit le chemin du répertoire où le fichier audio de la musique d'attente est stocké. (Que j'ai au préalable créé)
- **random=yes** : définit si la musique d'attente sera jouée de manière aléatoire ou en ordre séquentiel. En utilisant "yes" pour la valeur "random", la musique d'attente sera jouée de manière aléatoire.

3.3 Interception d'appels

3.3.1 Définir un groupe d'appel Assistant et Secrétaire

Pour définir un groupe d'appel comprenant l'assistante et la secrétaire, j'ai d'abord créé le groupe :

```
[grp]
music = default
strategy = ringall
timeout = 15
member = PJSIP/TL1
member = PJSIP/TM1BIS
```

FIGURE 18 – Déclaration du groupe

Je viens définir une musique de prédécroché pour ce groupe. J'ai choisi d'utiliser la musique d'attente que j'ai créée précédemment pour faciliter l'explication. J'ai ensuite déclaré le mode de sonnerie qui est **ringall** ce qui signifie que tous les membres du groupe sonneront en même temps. Puis, je viens définir le temps de sonnerie et enfin, je viens définir les membres du groupe.

3.3.2 Activez l'interception d'appel pour le groupe précédent

Pour ce faire, j'ai juste rajouté ces lignes dans le contexte des déclaration de postes concernés dans le fichier **pjsip_wizard.conf** :

```
endpoint/call_group = 1
endpoint/pickup_group = 1
```

FIGURE 19 – Activation de l'interception

Ces lignes permettent au postes qui ont ces lignes de pouvoir intercepter les appels.

3.3.3 Validez l'interception d'appels

Pour pouvoir valider l'interception d'appels. J'ai généré comme vu en TP un appel "démo" vers mon poste TL1 à l'aide de la commande suivante : `originate PJSIP/TL1 application Playback demo-congrats`. Sur mon poste TM1BIS, lorsque l'appel sonne chez TL1, je peux tout simplement appuyer sur *8 pour récupérer l'appel sur TM1BIS.

3.4 Enregistrement de conversation

3.4.1 Activez l'enregistrement de conversation côté serveur

Je n'ai malheureusement pas réussi à faire fonctionner le déclenchement de l'enregistrement d'appel à partir du moment où on appuie sur une touche. Cependant, j'ai réussi à faire en sorte que les appels soient tous enregistrés par défauts. Pour ce faire, j'ai ajouté les lignes suivantes dans le fichier `extensions.conf` :

```
exten => 3,1,MixMonitor(${UNIQUEID}.wav)
exten => 1,1,MixMonitor(${UNIQUEID}.wav)
exten => 2,1,MixMonitor(${UNIQUEID}.wav)
```

FIGURE 20 – Activation de l'enregistrement d'appels

En rajoutant ces lignes aux 3 numéros je défini donc que les appels passés vers les 3 numéros seront tous automatiquement enregistrés.

La commande **MixMonitor(\$UNIQUEID.wav)** : permet d'enregistrer une conversation téléphonique en cours. La commande utilise la variable d'environnement **\$UNIQUEID** pour générer un nom de fichier unique pour chaque enregistrement. Le fichier enregistré aura donc un nom de type **\$UNIQUEID.wav**. La commande MixMonitor permet d'enregistrer à la fois les entrées et les sorties audio de la conversation téléphonique. Les fichiers d'enregistrements seront retrouvable dans le répertoire **/var/spool/asterisk/monitor**.

3.5 Prédécroché

3.5.1 Enregistrez un message d'accueil personnalisé

Je me suis donc crée un nouveau message d'accueil différent de celui de mise en attente de la même manière que j'ai crée le message d'attente. Pour le message de prédécroché j'ai crée une nouvelle classe dans le fichier **musiconhold.conf** :

```
[newclass]
mode=files
directory=/var/lib/asterisk/mohpre
random=yes
```

FIGURE 21 – Déclaration d'un nouveau message d'accueil

C'est exactement le même principe que pour le message de mise en attente, sauf le répertoire du fichier vu qu'il est différent est **/var/lib/asterisk/mohpre/**. Ensuite, dans le fichier **extensions.conf**, j'ai modifié le plan de numérotation pour rajouter la nouvelle classe du message d'accueil personnalisé :

```
exten => 1,2,Dial(PJSIP/TM1,30,tTwWm(newclass))
exten => 2,2,Dial(PJSIP/TM1BIS,30,tTwWm(newclass))
exten => 3,2,Dial(PJSIP/TL1,30,tTwWm(newclass))
```

FIGURE 22 – Ajout d'un nouveau message d'accueil

J'ai donc repris le plan de numérotation pour y rajouter à la fin **m(newclass)** ce qui signifie tout simplement que pendant la sonnerie le message d'accueil associé au contexte **newclass** dans le fichier **musiconhold.conf** sera joué chez l'appelant.

4 Objectif 4 : Boîtes vocales

4.1 Configuration serveur

4.1.1 Créer une boîte vocal pour chaque utilisateurs

Pour créer une boîte vocal à chaque utilisateur je viens définir dans le contexte [default] du fichier **extensions.conf** ces lignes de code :

```
exten => 1,3,Voicemail(10)
exten => 2,3,Voicemail(20)
exten => 3,3,Voicemail(30)
```

FIGURE 23 – Ajout d'une boîte vocal

Ces lignes signifient que pour accéder par exemple à la boîte vocale de l'utilisateur **TM1** ayant comme numéro **1**, il suffit de composer donc le numéro **8500** qui est le numéro de boîte vocale par défaut et de composer le numéro **10** qui correspond à sa boîte vocale. Puis, écrire le code de la boîte vocale qui est le **1-2-3-4** que je viens définir dans le fichier **voicemail.conf** :

```
1234 => 4242,Example Mailbox, root@localhost
10 => 1234,Praticien,praticien.saetel@gmail.com
20 => 1234,Assistante,assistante.saetel@gmail.com
30 => 1234,Secrétaire,secretaire.saetel@gmail.com
```

FIGURE 24 – Ajout de comptes et mot de passe

5 Conclusion générale

Ce projet fût pour nous tous une expérience très enrichissante. Nous avons fait face à différents types de problèmes que nous avons réussi à résoudre, ou non. Par exemple au commencement de la SAE, nous sommes tombés face à notre plus gros problème qui était le téléphone Cisco. Nous avons pris énormément de temps à comprendre comment il était possible de les flasher. Une fois tout monté, à savoir le serveur TFTP, et après avoir flashé les 12 téléphones, nous nous sommes rendu compte qu'il n'était pas possible de les paramétrier pour les connecter à nos serveurs IPBX. Nous avons donc essayé de réfléchir à d'autres solutions, sans en trouver...

L'autre gros problème que j'ai eu personnellement était le message de prédécroché en même temps que le transfert d'appels. Quand je mettais le bon paramètre pour avoir le transfert d'appels ça fonctionnait, et dès que je rajoutais le paramètre pour définir le musique de prédécroché, le transfert ne fonctionnait plus. J'ai réussi à résoudre ce problème en modifiant juste une virgule... C'est-à-dire qu'initialement, j'avais dans mon plan de numérotation ceci :

[...]PJSIP/TM1,30,tT,m(newclass). J'ai trouvé qu'en écrivant plutôt :
[...]PJSIP/TM1,30,tTm(newclass) ça fonctionnait. C'est une très petite et subtile modification, mais qui a su faire la différence.

En définitive, j'ai apprécié ce projet car il était en parfaite corrélation avec mon alternance. En tant qu'apprenti chez Orange, j'installe une offre téléphonique qui se prénomme **Connect pro**. Cette offre est destinée aux petites entreprises. Elle se base sur un serveur Asterisk, et donc, tout ce que je configure chez Orange est identique à ce que j'ai fait dans ce projet, jusqu'à même la possibilité d'envoyer par mails les messages vocaux. La seule différence, est que chez Orange tout a fait pour que l'installation soit la plus simple possible et rapide. Nous, les techniciens, passons tous par une interface graphique, tandis que pendant le projet tout était en ligne de commande. Je suis donc très content d'avoir pu voir et comprendre comment tout ce que je fais chez Orange fonctionne réellement en arrière plan, côté serveur.