

IUT DE COLMAR

SAE3 - ROM

ANNÉE 2022-23

---

## Déployer un service de téléphonie

---

MARTIN BAUMGAERTNER

3 février 2023

---

# Table des matières

<b>1 Objectif 1</b>	<b>2</b>
1.1 Configuration Serveur IPBX . . . . .	2
1.1.1 Créer une machine virtuelle Debian11 . . . . .	2
1.1.2 Installez l'IPBX asterisk via le système de paquets Debian . . . . .	2
1.1.3 Pour le service IPBX, déclarez les différents postes SIP . . . . .	3
1.1.4 Pour le service IPBX, créer le plan de numérotation . . . . .	6
1.2 Configuration Téléphone Clients . . . . .	6
1.2.1 Téléphone logiciel Linphone . . . . .	6
1.2.2 Téléphone matériel Nortel LIP6812 . . . . .	7
1.2.3 Téléphone Matériel Cisco 7941G . . . . .	9
1.3 Validation des appels . . . . .	9
<b>2 Objectif 2 : Configuration inter-sites</b>	<b>10</b>
2.1 Appels inter-sites . . . . .	10
2.1.1 Sur l'IPBX, inscrire le faisceau SIP . . . . .	10
2.1.2 Dans le plan de numérotation, ajoutez le préfix * . . . . .	10
2.1.3 Validez les appels entrants et sortants entre cabinets médicaux	11
2.2 Appels externes . . . . .	11
2.2.1 Ajoutez les numéros à 10 chiffres pour les appels extérieurs via le faisceau . . . . .	11
2.2.2 Validez les appels externes vers (0112345678) . . . . .	11

---

# 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. De plus, 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 écoute 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...

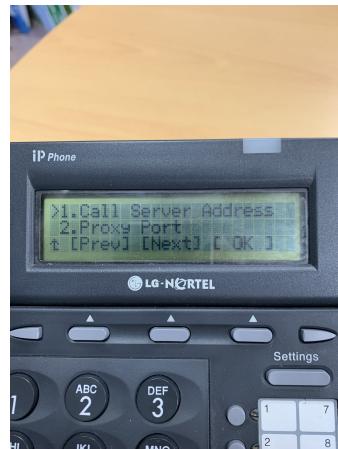


FIGURE 7 – Configuration du Nortel

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**

**Installez, avec toutes les précautions nécessaires, le dernier firmware SIP**

**Inscrivez le téléphone auprès du registrar de l'IPBX**

**Parametrez les touches d'appels rapides et collez des étiquettes**

### **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 inversenement.

---

## 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 8 – 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 9 – 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 10 – 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.