

# TP Voix sur IP

## SIP et Asterisk

Ce TP propose une installation et un test d'un serveur et de clients afin de mettre en oeuvre le protocole SIP. L'objectif est d'analyser les comportements des logiciels, et de comprendre le protocole SIP en lui même à travers plusieurs exemples. Le sujet de ce TP est un guide pour la mise en oeuvre minimale du service SIP, mais il est demandé aux élèves de se poser des questions sur le fonctionnement de SIP, et de profiter de cette mise en oeuvre pour tester et observer des comportements du protocole (en plus de ce qui est demandé dans ce document). Par ailleurs, la configuration des logiciels clients n'est pas expliquée, il est laissé à l'étudiant le soin de rechercher comment les configurer pour s'enregistrer avec le serveur et émettre des appels.

Ce TP se fait sous le système d'exploitation LINUX (natif ou par une image virtuelle).

Le schéma du TP est le suivant : nous allons installer un serveur VoIP (nommé *Asterisk*), créer des comptes utilisateurs pour des clients, et définir un plan de numérotation qui permettent d'appeler ces clients. D'un autre côté, nous allons installer des clients VoIP qui utiliseront ce serveur pour émettre des appels et être appelés. Les tâches à réaliser sont les suivantes :

- Créer les comptes client sur le serveur
- Configurer le logiciel client
- Enregistrer le client
- Réaliser un appel sur un numéro de test
- Créer et enregistrer un deuxième client
- Réaliser des appels entre les deux clients, en changeant les options des différents comptes, et en changeant la configuration du réseau

Ce TP est à faire en binôme. Un rapport au format PDF sera à rendre (nicolas.montavont@telecom-bretagne.eu) par email à pour répondre aux questions posées dans le sujet

# Prise en main du TP

## 1. Suite logiciels à utiliser

Afin de mettre en œuvre un service de VoIP, et d'analyser les comportements serveur et client les logiciels suivants seront utilisés :

- jitsi (client SIP)
- linphone (client SIP)
- Ekiga (client SIP, H323)
- xmeeting (client SIP, H323)
- Asterisk (serveur VoIP – serveur SIP) : Asterisk est un serveur VoIP / IP PBX très riche, qui permet de mettre en œuvre de nombreuses fonctionnalités. Entre autres, Asterisk permet d'installer les quelques services suivants :
  - Serveur SIP (registrar, proxy)
  - Serveur H323 (gatekeeper)
  - Passerelle IP – RTC
  - Messages de répondeur par email
  - Music On Hold
  - Transcodage

Dans le cadre de ce TP, nous allons mettre en place certaines de ces fonctionnalités

- Wireshark : Le logiciel Wireshark permet de visualiser les trames échangées sur le réseau, et il est capable d'analyser et d'interpréter ces dernières. Lancez le logiciel avec l'utilisateur root, et lancez la capture sur l'interface réseau utilisée. En fin de capture, il est possible de demander à Wireshark d'analyser les transactions SIP, en allant dans le menu analyse / SIP transactions.

## 2. Tutorial Asterisk

### 2.1. Manipulation

Asterisk est déjà installé sur l'ordinateur. L'ensemble des fichiers de configuration se trouve dans /etc/asterisk. Il y a notamment les fichiers extensions.conf et sip.conf qui sont les fichiers principaux pour configurer un service SIP. Le fichier sip.conf regroupe la configuration du protocole SIP sur Asterisk, à savoir les proxy / provider / clients. Le fichier extensions.conf est le fichier principal du serveur Asterisk, celui qui indique le plan d'appel du serveur (voir plus loin pour plus d'informations sur la configuration de ces fichiers). Prendre soin de faire une copie des fichiers initiaux (les renommer <nom\_fichier>.old par exemple), et copier ensuite les fichiers sip.conf et extensions.conf du bureau vers le répertoire /etc/asterisk pour pouvoir éditer ces fichiers. Le logiciel lit exclusivement les fichiers dans le répertoire /etc/asterisk.

## 2.2. Manipulation d'Asterisk

Pour lancer Asterisk, il faut taper la commande suivante dans un terminal avec les droit administrateur :

```
asterisk -vvvc
```

Il est également possible de se connecter à la console Asterisk afin d'interroger le serveur, et de voir tous les messages de fonctionnement (si jamais vous quittez le terminal). Pour se connecter à une instance d'Asterisk en cours, dans un terminal, il faut taper la commande :

```
asterisk -r
```

Pour sortir de la console, il suffit de faire un ctrl-C (attention, cette commande ne tue pas le processus asterisk, mais quitte uniquement la console). Pour tuer une instance d'asterisk, le plus simple est de tuer le processus (kill -9 <numéro\_de\_processus>).

## 2.3. La console Asterisk

La console Asterisk permet d'interroger le démon Asterisk, de regarder la liste des utilisateurs, les modules en place, de les redémarrer et d'avoir d'éventuels messages d'erreurs. Une fois entré dans la console (asterisk -r), voici quelques commandes utiles :

- help : liste l'ensemble des commandes disponibles
- dialplan reload : recharge le fichier extensions.conf – nécessaire après une modification dans le fichier
- sip reload : recharge le fichier sip.conf – nécessaire après une modification dans le fichier
- sip show peers : liste l'ensemble des clients SIP configurés sur Asterisk avec certain paramètre, comme l'adresse IP, et indique combien de clients sont connectés
- sip show users : liste l'ensemble des clients SIP avec leur mot de passe
- sip show user <username> : décrit les paramètres de l'utilisateur SIP <username>
- sip set debug : permet de passer le protocole SIP en mode debug, où vous verrez tous les messages et états du protocoles SIP dans la console
- dialplan debug : permet de passer le plan de numérotation en mode debug, pour voir les opérations effectués par Asterisk en temps réel

Note : Lorsque vous jouerez avec les différentes configurations possibles, il n'est pas forcément nécessaire de redémarrer le serveur Asterisk. Seul un module peut être redémarré (en tapant le nom du module et reload)

## 2.4. Sip.conf

Le fichier /etc/asterisk/sip.conf regroupe l'ensemble des options configurables pour installer des services SIP. Cette section donne une aide pour la configuration de ce fichier.

Le fichier sip.conf commence avec une section [general] qui donne la configuration par défaut pour tout utilisateur et provider. Tout élément de cette section sera écrasée par des définitions spécifiques définies plus loin dans le fichier. Ces extensions (appelés canaux SIP) sont définies entre crochet.

Voici quelques paramètres possibles pour la section [general] :

- Domain (example.com) : configure le nom de domaine sur lequel travaille Asterisk.
- Bindaddr / bindport (adresse IP x.x.x.x ou 0.0.0.0 pour toutes les adresses du serveur / numero de port par défaut : 5060) : les adresses IP et numéros de port sur lesquels le serveur Asterisk doit tourner
- Srvlookup (yes, no) : indique à Asterisk de faire des requêtes DNS, notamment pour utiliser des adresses logiques contenant des noms de domaines
- Support NAT : plusieurs paramètres peuvent être configurés si le serveur Asterisk est derrière un NAT :
  - Externip (=x.x.x.x) : adresse publique du boîtier NAT
  - Localnet (=x.x.x.x/n.n.n.n => adresse IP/netmask) : réseau local sur lequel se trouve généralement les clients (par exemple : 192.168.0.0/255.255.255.0)
  - Nat (= yes | no) : influence la déclaration des utilisateurs SIP (peers, users) pour indiquer que ces utilisateurs sont eux-mêmes derrière un boîtier NAT par rapport à l'Asterisk

Les canaux SIP suivants sont possibles :

**User** : authentifie les appels entrants (peut seulement appeler)

**Peer** : authentifie les appels sortants (peut seulement être appelé)

**Friend** : authentifie les appels entrants et sortants (peut appeler et être appelé)

Voici quelques paramètres disponibles pour configurer les canaux SIP (user, peer, friend) :

- Type (= <type du client SIP>) : indique le type du client, c'est-à-dire friend, user ou peer
- Context (<nom\_contexte>) : indique le contexte dans le plan de numérotation (extensions.conf) dans lequel les instructions pour cette extension seront exécutées. Plusieurs extensions peuvent avoir le même contexte
- Callerid (<nom <numéro>> - exemple : =John Smith <555>) : Nom complet de l'utilisateur, qui sera utilisé en dernier recours si rien d'autre n'est disponible pour afficher un nom d'utilisateur lors d'un appel
- Fromuser (<from\_ID>) : Définit l'utilisateur à mettre dans le champ from à la place du caller ID (comportement étrange observé sur l'Asterisk – ne pas utiliser)
- Username (<username[@realm]>) : Nom d'utilisateur utilisé pour l'authentification. Le username doit être le même que le nom de l'extension entre []
- secret : mot de passe utilisé pour l'authentification
- Host (= dynamic | x.x.x.x) : indique l'adresse IP d'une extension. Si la valeur de host vaut *dynamic*, le client est autorisé à se connecter par n'importe quelle adresse.
- Mailbox (= <number>@<context>) : Numéro de mailbox avec le contexte associé
- directmedia (yes, no, nonat, update) : indique si Asterisk doit être au cœur de tous les échanges entre deux clients, c'est-à-dire si les flux RTP doivent obligatoirement passer par Asterisk, ou s'ils peuvent directement être échangés entre les clients.
- Allow / disallow (= all | gsm | alaw | ulaw | g729 | g726 | g723.1 | ilbc | speex) : Permet ou interdit l'utilisation de codecs. Les règles définies dans un canal SIP s'ajoutent aux règles

définies dans la section [general]. Afin de ré-initialiser ces règles pour un certain canal, il faudra commencer par *disallow=all*

- Dtmfmode (= inband | rfc2833 | INFO) : défini comment les tonalités DTMF sont transmises (dans la bande, i.e., dans le flux audio, ou hors bande comme de la signalisation)
- Nat (= yes | no | never): indique si un NAT se trouve entre le client et Asterisk. Dans le cas où nat=yes, Asterisk ignorera le champs Contact Information et utilisera l'adresse du paquet

Exemple :

```
[general]
context=default
port=5060
bindaddr=0.0.0.0
srvlookup=yes
```

```
[peter]
type=friend
nat=no ; Telephone without NAT
host=dynamic ; the devices can be registered with different IPs each time
```

## **2.5. Extensions.conf**

Comme nous l'avons déjà dit, Asterisk propose une multitude de protocoles. Il lui faut alors un moyen de connaître les actions à mener lors d'un appel, pour rediriger les appels vers le bon protocole et / ou le bon destinataire. Le fichier /etc/asterisk/extensions.conf indique à Asterisk comment router / commuter les appels, c'est le plan de numérotation. Extensions.conf indique les actions à entreprendre lorsque des INVITE vers un numéro ou un utilisateur arrivent sur Asterisk.

Ce fichier est composé en trois zones : la zone générale où se trouve les paramètres généraux pour toute l'architecture, la zone globale où on insère des variables globales qui seront utilisées dans la troisième zone qui est constituée de plusieurs sous-zones appelées « contextes ».

### **Section générale**

Ici sont définis les paramètres généraux du serveur :

autofallthrough= yes ; raccrochage automatique après l'appel (par défaut, le champ est à yes)

static= yes ; en binôme avec l'option suivante

writeprotect = no ; modification possible depuis la CLI

### **Section globale**

Dans cette section, on déclare des variables pour, par exemple, des numéros de téléphone ou des variables faisant appel à des fichiers son pour le parcage ou autre.

<Nom\_variable> => <valeur ou chemin du fichier>

### ***Section contexte***

Chaque contexte définit une portion du plan d'appel, les contextes permettent de diviser de manière précise la gestion des appels. On peut créer par exemple un contexte pour filtrer les appels entrants et un autre pour filtrer les appels sortants. Les filtres qui seront mis en place seront fonction des règles que vous voulez mettre en place : autoriser certains appels seulement, utiliser une certaine ligne téléphonique pour un certain type de numéro (portable – fixe – international) selon les tarifs des opérateurs.

Le contexte en lui-même est défini par une série de règles par numéro (ou ensemble de numéro). Pour un même numéro, on pourra définir une série d'instructions séquentielles, identifiées par un numéro de séquence.

Exemple :

[default]

Exten => 118,1,Dial(Sip/toto)

Exten => 119,1,Dial(Sip/obelisk)

Syntaxe :

- [] : définition des contextes (ex : general)
- exten => <numéro>, #, Action
  - <numéro> : donne le numéro de téléphone ou le nom de client qui appelle, et donc qui doit être redirigé vers le bon destinataire
  - # : numéro de l'action, utilisé pour le séquençage d'action. On commence par 1, et si d'autres actions doivent suivre, on incrémente de ce numéro
  - Action : représente la tâche à réaliser par Asterisk.

Les actions suivantes sont possibles :

- Answer()
- Wait()
- Voicemail()
- Hangup()
- Dial(SIP/<canal>)
- Goto(<context>, <numéro>, #)

*Notez que les fichiers d'origine de la configuration du serveur (qui sont nommés sip.conf.old et extensions.conf.old sont dans /etc/asterisk et donnent beaucoup d'exemples de configuration.*

## Travail à faire

### 3. Configuration du réseau

Nous utiliserons deux topologies réseau pour réaliser ce TP. La première consiste en une architecture simple, où tous les noeuds appartiennent au même sous-réseau, alors que nous utiliserons une architecture avec des NAT dans un deuxième temps.

Pour la première topologie, nous pourrions utiliser le réseau sans fil nommé "TPVOIP". Il n'y a pas de serveur DHCP sur ce réseau, il vous appartient donc de configurer vos machines (soit en configurant manuellement les adresses IP sur les postes, soit en installant un serveur DHCP). Dans cette première topologie, nous allons configurer un serveur VoIP, et deux clients au moins pour faire des tests de fonctionnalités.

1. Proposer un plan d'adressage pour un ensemble d'un serveur et deux clients. Représenter la topologie du réseau sur un schéma, en indiquant les adresses IP de chaque équipement, et leur rôle. Préciser quelle méthode de configuration (IP) a été retenue.

### 4. Registrar SIP

La première étape du TP est de configurer un compte pour un client SIP et de vérifier que le client est à même de s'enregistrer et d'appeler le serveur. Dans cette première étape, il s'agit de configurer un client logiciel (choisissez jitsi ou linphone pour commencer), et le fichier sip.conf sur Asterisk.

Commencez par créer les informations de compte client sur le serveur. Pour cela, il faut éditer le fichier sip.conf de l'asterisk (situé dans /etc/asterisk) qui regroupe l'ensemble de la définition du service SIP. Faire une sauvegarde du fichier actuel, et repartir d'un fichier vierge.

Les champs minimums à définir dans sip.conf par client sont les paramètres suivants :

- Type
- Codecs (*allow=all* pour commencer)
- host (adresse IP du client)

Une fois que les informations client ont été sauvegardées, démarrer le serveur asterisk (*asterisk -c*) ou alors recharger la configuration SIP du serveur (*sip reload* dans la console Asterisk) si le serveur est déjà lancé. A présent ouvrir jitsi ou linphone, et configurer un compte client qui correspond au même nom d'utilisateur que celui créé sur le serveur. Indiquer l'adresse IP de votre serveur comme serveur / domaine.

Une fois les configurations nécessaires faites sur le client et le serveur, le client doit pouvoir s'enregistrer auprès de l'Asterisk. Vous pourrez le vérifier avec la commande sip show peers ou encore sur le logiciel client qui doit indiquer *connecté* ou *online*. Vous pouvez également vérifier l'enregistrement du client par le logiciel de captures de trames *wireshark*.

2. Comment se déroule l'enregistrement ? Expliquer cet enregistrement et donner une illustration des messages échangés.

Installer un client sur une machine différente du serveur, et un client sur la machine où tourne le serveur.

3. Quels sont les numéros de port utilisés par ces clients SIP lors de leur enregistrement ? Expliquer.

Ensuite, pour tester si le client arrive à émettre des appels, vous pouvez configurer un numéro de téléphone pour appeler le service echo. La fonction echo se configure de la façon suivante :

```
exten => 600,1,Playback(demo-echotest)
exten => 600,n,Echo
exten => 600,n,Playback(demo-echodone)
exten => 600,n,Goto(s,6)
```

Ce service fonctionne en mono-utilisateur, et renvoie simplement ce que vous dites. Ceci permet de tester la latence entre le serveur et le client. Après avoir ajouté ces lignes dans le fichier extensions.conf, ne pas oublier de le recharger pour Asterisk (taper *dialplan reload*)

A présent, ajouter un mot de passe pour empêcher qu'un autre utilisateur puisse se connecter au serveur à votre place. Pour cela, ajouter le champs *secret* dans la configuration SIP du serveur concernant ce client et recharger la configuration SIP. Observer dans Wireshark les messages envoyés pour l'enregistrement.

4. Comment se déroule l'enregistrement ? Expliquer cet enregistrement et donner une illustration des messages échangés lorsque le mot de passe entré par l'utilisateur est correct, et lorsqu'il ne l'est pas.

A présent indiquer au logiciel client de se mettre *offline* (*déconnecté*). Observez l'échange de messages dans Wireshark.

5. Comment se déroule la déconnexion ?

## 5. Communication entre deux clients

A présent, nous allons effectuer des appels entre les deux clients, à travers le serveur Asterisk. Pour cela, ajouter le paramètre suivant pour chacun des client :

```
directmedia=No
```

Une fois que les deux clients sont configurés et enregistrés, il faut indiquer à Asterisk comment diriger les appels vers ces clients. Il faut pour cela ajouter une ligne pour chaque utilisateur dans le fichier extensions.conf, qui indique pour le numéro appelé, de rediriger l'appel vers le protocole SIP/<nom\_utilisateur>. (voir tutorial en début de document).

Tester la communication entre les deux clients.

### **5.1. Analyse des comportements des clients et serveurs**

Maintenant que la configuration client et serveur est en place, analysez les messages dans Wireshark.

6. Quels sont les messages utilisés pour établir la session (émettre l'appel) ?

7. Quels sont les messages utilisés pour transporter la voix ?

8. Quels sont les messages utilisés pour clore la session (terminer l'appel) ?

9. Identifiez les sources et destinations de chaque message (faire un diagramme d'échange de messages). Que peut-on en conclure ?

10. Quelles sont les numéros de port utilisés par chaque message ? Où sont négociés les numéros de ports pour le média ?

Changez à présent dans le fichier sip.conf le paramètre directmedia des deux clients (mettre la valeur <Yes>). (Il peut être préférable de relancer les clients une fois cette configuration chargée dans Asterisk).

11. Quelle est la différence dans une communication ? Qu'en déduisez-vous sur le comportement du serveur ?

Vous pouvez également mettre un client offline (quitter le logiciel client par exemple), et avec l'autre client l'appeler.

12. Que se passe-t-il ?

## 5.2. Négociation des codecs et qualité audio

Dans les logiciels clients, désactiver tous les codecs, sauf un, le même sur chaque client, qui sera successivement :

- g711a
- gsm
- g723a
- speex

Configurer un taux d'erreur et de délai sur l'interface de votre serveur par la commande *tc* :

- Ajoute un délai fixe de 100ms : *tc qdisc add dev eth0 root netem delay 100ms*
- Ajoute un délai variable +/-20ms : *tc qdisc add dev eth0 root netem delay 100ms 20ms*
- Ajoute une perte de 3%, avec un corrélation de 25% avec la perte précédente :  
*tc qdisc change dev eth0 root netem loss 0.3% 25%*

Etablir les appels entre les deux clients.

13. Observez vous la même qualité audio selon les différents codecs utilisés ? Si non, quelle observation pouvez vous faire ?

Remettre tous les codecs disponibles du côté des logiciels client. Mettre à présent *disallow=all* pour les deux comptes clients sur le serveur, et uniquement autoriser le codec GSM pour un des clients et Speex pour l'autre.

14. Est ce que l'appel est bien réalisé (décrire ce que l'on observe)

Remettre *directmedia=no*.

15. Est ce que l'appel est bien réalisé (décrire ce que l'on observe)

## 6. VoIP en environnement de NAT

A présent, nous allons considérer des réseaux différents entre le serveur et le client, dont l'un de ces réseaux sera derrière un NAT. Le réseau sans fil TPVOIP sera derrière un NAT, alors que le réseau filaire restera en adressage "public".

### 6.1. Les clients sont derrière le NAT

Mettre dans un premier temps le serveur dans l'espace public, et les clients dans l'espace d'adressage naté.

Dans un premier temps, enlever toute configuration *directmedia* des clients dans le fichier sip.conf.

16. Représenter l'architecture réseau, et l'adressage de chacune de vos machines.

17. Le client peut-il s'enregistrer ? Expliquer.

Changer à présent la configuration des clients dans le fichier sip.conf en ajoutant le champs *nat=yes*.

18. Le client peut-il s'enregistrer ? Expliquer.

Taper la commande sip show peers dans la console asterisk.

19. A quoi correspond l'adresse IP / numéro de port affiché pour le client ?

Réaliser le test d'appel echo (600) avec l'un de vos clients.

20. Comment se déroule l'appel ? (Détailler l'établissement de session, et le chemin des flux RTP pour chaque étude)

Réaliser un appel entre les deux clients.

21. Comment se déroule l'appel ?

Déplacer un client dans l'espace d'adressage public, et essayer à nouveau l'appel entre les deux clients. Tester la symétrie de l'appel : réaliser un appel à partir de chaque client

22. Comment se déroule les appels ? Dans quel sens la communication est-elle possible ? Pourquoi ?

Mettre le champs *directmedia* à *no* pour le client situé derrière le NAT et réessayer l'appel entre les deux clients. (appel à partir de chaque client)

23. Comment se déroule les appels ? Dans quel sens l'appel est-il possible ?

## 6.2. Le serveur est derrière le NAT

A présent, inverser la configuration, et mettre le serveur derrière le NAT, et les clients dans l'espace d'adressage public. Sur le NAT, rediriger le trafic à destination du port 5060 pour qu'il arrive sur le serveur Asterisk.

24. Par quelle adresse le serveur Asterisk sera-t-il joignable à partir d'Internet?

Ajouter dans la configuration générale du fichier sip.conf le paramètre suivant :

```
[general]
```

```
externalip=80.123.123.123
```

```
localnet=192.168.1.0/255.255.255.0
```

Modifier la configuration des téléphones. Vérifier que les deux téléphones s'enregistrent auprès du serveur. Réaliser un appel entre les deux téléphones.

25. La session est-elle établie entre les deux téléphones ? Est ce que le son passe entre les deux téléphones ?

Editer le fichier /etc/asterisk/rtp.conf pour identifier la plage de port à rediriger vers le serveur, et configurer le NAT.

26. La session est-elle établie entre les deux téléphones ? Est ce que le son passe entre les deux téléphones ?

## 7. Service de messagerie vocale

Nous allons à présent configurer une messagerie vocale pour les utilisateurs. Pour cela, il faut configurer la messagerie dans voicemail.conf, puis indiquer le numéro de messagerie dans la configuration de l'utilisateur SIP, et enfin ajuster le plan de numérotation.

### *Modification dans le fichier voicemail*

Commencer par indiquer le numéro de la messagerie, associé à la mailbox de la manière suivante :

```
<numéro de voicemail> => <mot de passe>, <nom d'utilisateur>, <email>, <option(s)>
```

```
Exemple : 1000 => 0000, Claude Chabrou, claudio@email.com
```

### *Modification dans le fichier sip.conf*

Ajouter une ligne dans les clients indiquer leur numéro de voicemail à l'aide de la syntaxe suivante :

mailbox=<numéro de mailbox>

### ***Modification dans le fichier extensions.conf***

Il va falloir changer le plan de numérotation de telle sorte que si l'utilisateur pour lequel vous créez le voicemail ne répond pas à un appel, sa messagerie se met en route. Il faut donc retrouver le plan de numérotation pour cet utilisateur, et le modifier en appelant la commande voicemail, qui est défini de la façon suivante :

*Voicemail* – Enregistre un canal en sauvant l'audio dans un fichier pour un numéro de boîte vocale donné.

VoiceMail(boxnumber[@context][&boxnumber2[@context]][&boxnumber3],[flags])

Les flags peuvent être les suivants :

- s: Evite le message "Please leave your message after the tone. When done, hang up, or press the pound key."
- u: Joue le message d'indisponibilité. Par défaut, le message est : "The person at extension ... 1234 ... is unavailable,", mais ce message est modifiable en interrogeant sa boîte vocale.
- b: Joue le message Occupé. Par défaut, ce message dit : "The person at extension ... 1234 ... is busy."

L'appel de notre extension doit donc sonner pendant une durée, et si personne ne décroche, appeler la messagerie correspondante :

```
exten => <phone_number>,1,Dial(SIP/<phone_number>,20,tr) ;le tel sonne pendant 20s
exten => <phone_number>,2,VoiceMail(<numéro_messagerie>@<context>,u)
exten => <phone_number>,102,VoiceMail,b<numero_messagerie>
```

La dernière ligne est utilisée si l'utilisateur <phone\_number> est en communication alors qu'un appel arrive pour lui. La priorité 102 (101+1) est du au fait qu'Asterisk fait un saut de 101 dans la priorité lorsque l'utilisateur demandé est occupé.

### ***Ecouter sa messagerie***

A présent, il n'y a plus qu'à ajouter un plan de numérotation pour consulter sa messagerie. Notez que le numéro de répondeur sera le même pour tout le monde, et qu'un serveur vocal vous demandera votre numéro de messagerie ainsi que le mot de passe. Cela peut être ajouté par l'entrée suivante :

```
exten => <numéro_répondeur>,1, Ringing
exten => <numéro_répondeur>,2, Wait(2)
exten => <numéro_répondeur>,3, VoiceMailMain()
```

Tester le service de messagerie.