

Administrateur de réseaux

Groupe alternance

Didier OURABAH, Maxime GUILLET, Léonard LECOUEY, Vincent BATOUFFLET, William ZIVIC

RAPPORT DE PROJET

<< ASTERISK >>

TABLE DES MATIERES

1.	Introduction	2
2.	VoIP	3
a.	Fonctionnement	3
	Numérisation	3
	Compression	3
	Transport	3
	Etablissement de la connexion	4
b.	Intérêts	4
3.	Protocole SIP	5
a.	Fonctionnement	5
	Mode Point à point	5
	Mode diffusif.....	6
b.	Différence H.323/SIP	7
4.	Asterisk	8
	Historique :	8
	Fonctionnalités :	8
	De la téléphonie à moindre coût :	9
	Une particularité modulaire :	9
	Principales fonctions :	9
	Les APIs :	10
	Fonctionnement évolué :	10
5.	Mise en place du serveur	11
a.	Installation du système	11
b.	Configuration de base	11
	Définition des sources d'installation :	11
	Installation de paquets supplémentaires :	11
	Configuration de l'interface réseau :	12
c.	Mise en place du service Asterisk	12
	Installation des paquets :	12
	Configuration de base d'Asterisk	12
d.	IVR (Intelligent Voice Response)	13
	Descriptif des fonctions	13
	Conversion du son.....	15
	Mise en place du menu	15
e.	Messagerie Vocale	16
f.	Les files d'attente et les agents	17
6.	Clients	19
a.	Clients	19
	X-Lite	19
	Windows Messenger	20
	SJPhone	21
	Liphone.....	21
b.	Comparatif	22
7.	Repartition des tâches	23
8.	Gestion du projet	24
9.	Conclusion.....	25
10.	Lexique.....	26

1. INTRODUCTION

Dans le cadre de nos études en licence ISDRN, nous avons dû répondre à un sujet combinant de nombreux domaines et permettant de mettre en pratique nos connaissances théoriques et pratique acquises durant nos derniers mois de licence.

L'objectif est d'installer un standard téléphonique sous VoIP (Asterisk) dans une entreprise nommée "Gaulois Télécom". Ceci permet de mettre en place un système téléphonique avancé pour un coût moindre.

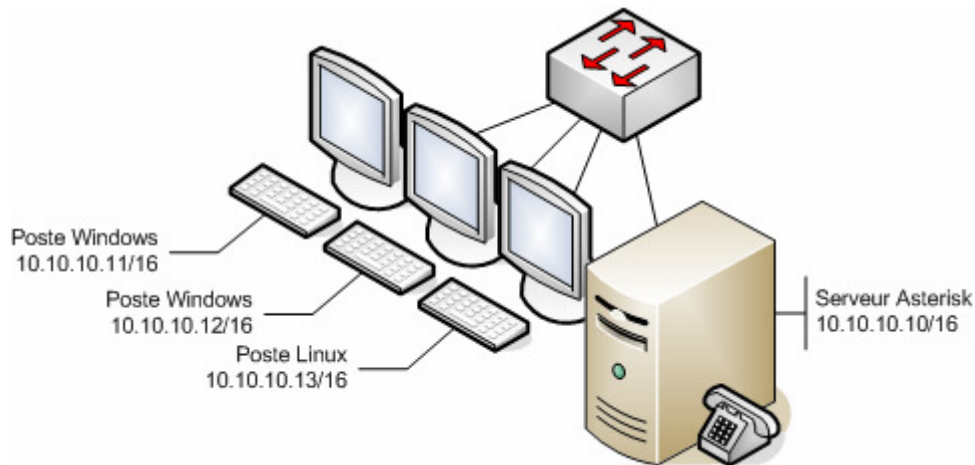


Figure 1 : schéma réseau simplifié

Une grande partie de notre travail s'est orienté sur la mise en forme d'un rapport qui, en plus de présenter nos démarches et nos résultats, met à disposition de tous, une documentation synthétique des différents outils et logiciels nécessaires à la conception de ce projet.

2. VOIP

"VoIP" est un abrégé de l'anglais Voice Over IP parfois appelé téléphonie IP ou téléphonie sur Internet. Cette technologie permet de communiquer par voix via le réseau Internet ou autres réseaux supportant le protocole **TCP/IP**^[1].

a. Fonctionnement

Lorsqu'un utilisateur veut entrer en communication avec un autre, une connexion est alors établie entre les deux terminaux. L'utilisateur peut alors émettre un son par le biais d'un micro (signal analogique) qui est ensuite numérisé et compressé par la machine (signal par synthèse).

Une fois les données encapsulées dans un paquet, il est envoyé au destinataire qui procèdera aux opérations inverses assurant ainsi la mise en forme d'un message audible.

Schématiquement le transport de la voix se fait ainsi :

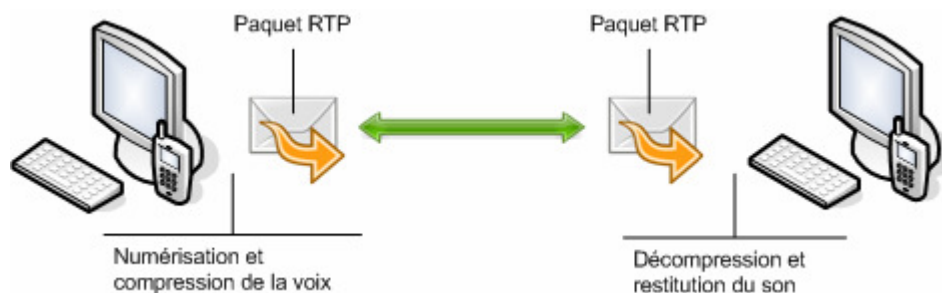


Figure 2: transport de la voix

Numérisation

Les signaux de la voix (analogiques) doivent d'abord être convertis sous forme numérique suivant le format PCM (Pulse Code Modulation) à 64kbits/s. La modulation d'impulsion codée est une technique d'échantillonnage quantifiée sur une série de symbole dans un code numérique (*binaire*). L'ordinateur ne comprenant que le code binaire, la numérisation est donc primordial.

Compression

Lors de la numérisation, le codage PCM se contente de mesurer des échantillons indépendamment des uns des autres. Un échantillon du signal n'est pas isolé, mais corrélé avec d'autres (précédent ou suivant).

En tenant compte des informations, il est possible de prévoir la valeur du nouvel échantillon et donc de transmettre qu'une partie de l'information. C'est ce qu'on appelle la prédiction. Cela permet de réduire la taille du paquet pour optimiser la bande passante.

Il existe deux grands types de compressions : le codage différentiel et le codage par synthèse. Pour notre projet, nous allons uniquement nous intéresser au codec GSM (*Global System for Mobile Communication*) qui utilise une fréquence d'échantillonnage de 8kHz/s (codage synthèse).

Le GSM utilise le format de codage appelé RPE-LTP (*Regular Pulse Excitation-Long Term Prediction*) avec un débit binaire de 13kbits/s (*160 échantillons du signal seront codés sur 260 bits*).

Transport

L'information voyage dans des datagrammes **UDP**^[2] ne garantissant pas la livraison car il n'effectue aucune vérification concernant la perte de paquet et ne transmet aucune information sur les configurations utilisés.

Il a donc fallu définir un nouveau protocole fournissant plusieurs fonctionnalités :

- Le numéro de séquence pour la remise en ordre des paquets ;
- Un champ horodatage (*timestamp*) pour la restauration de la base de temps ;
- Détecte la perte de paquets pour informer la source dans des délais compatibles avec le service ;
- Identifier le contenu des données et permettre leur transmission ;
- Intègre des solutions pour traverser des passerelles de certains réseaux locaux

Ce protocole est appelé RTP (*Real-Time Transport Protocol*) qui se complète par un protocole de contrôle qui transmet des rapports de réception RTCP (*Real-Time Transport Control Protocol*). Par exemple lors d'une conférence regroupant plusieurs participant, RTCP permet d'identifier différentes sources d'émissions contribuant à la session, mais il n'est cependant pas obligatoire.

Etablissement de la connexion

Avant de pouvoir communiquer directement, les membres de la discussion doivent établir un protocole pour la démarrer.

Les principaux protocoles utilisés pour l'établissement de la communication sont :

- H323 ;
- SIP ;
- IAX (SIP amélioré, issu du projet de **PABX**^[3] Asterisk) ;
- MGCP ;
- SCCP ;
- Jingle (*basé sur Jabber*).

SIP est utilisé pour notre projet et est développé dans le chapitre suivant.

b. Intérêts

La VoIP doit non seulement simplifier la mise en place (*installation et configuration*) mais aussi faire économiser de l'argent.

En effet les entreprises dépensent énormément en communication téléphonique, hors le prix des communications de la VoIP est dérisoire en comparaison. Il suffit simplement de louer des hébergeurs pour transmettre les communications sans avoir à payer tous les services. Cela offre donc une grande indépendance. L'entreprise peut installer facilement son propre **PABX**^[3] au lieu d'utiliser des offres onéreuses en pure téléphonie (**PABX**^[3] Ericsson, Alcatel,...).

Ajouté à cela le fait qu'il faille un personnel compétent pour administrer ces **PABX**^[3] ordinaire, il revient donc très vite essentiel de faire de la VoIP.

Le transport se faisant aussi par le biais du réseau informatique, il n'est donc pas nécessaire de devoir mettre en place un réseau téléphonique à part. Le tout est alors centralisé sur une même entité.

De plus, en positionnant la voix comme une application supplémentaire sur le réseaux IP, l'entreprise ne va pas uniquement substituer un transport opérateur RTC à un transport IP, mais simplifié la gestion de la voix, des données et vidéo par ce seul transport.

3. PROTOCOLE SIP

SIP (Session Initiation Protocol, RFC 3261) est un protocole de signalisation de niveau Application pour la gestion de session audio indépendant du protocole de transport IP (**UDP**^[2] / **TCP**^[4]). Ce protocole avancé permet aux utilisateurs de maintenir, transférer, renvoyer, mettre en attente un appel ou encore de réaliser des conférences.

a. Fonctionnement

SIP permet donc de mettre en place une communication. Pour cela avant que la connexion soit établie, il se charge d'envoyer plusieurs paquets entre les postes afin de définir le début et la fin de la conversation, son type, et sa composante (type d'encodage utilisé pour l'audio). Ces requêtes sont répertoriées sous divers codes :

- 1xx : Information - La requête a été reçue par le destinataire et continue à être traitée (ex : 180 = 'en train de sonner')
- 2xx : Succès (ex : 200 = 'OK', 202 = 'acceptée')
- 3xx : Redirection - Une autre action doit avoir lieu afin de valider la requête
- 4xx : Erreur du client - La requête contient une syntaxe fautive ou bien elle ne peut pas être traitée par ce serveur (ex : 404 = 'Not found')
- 5xx : Erreur du serveur - Le serveur n'a pas réussi à traiter une requête qui semble être correcte
- 6xx : Echec général, la requête ne peut être traitée par aucun serveur

On distingue également 2 modes précis d'ouverture de sessions avec SIP que l'on va détailler :

- Mode Point à point : communication entre deux postes ;
- Mode diffusif : Plusieurs postes membre d'un serveur.

Mode Point à point

Le mode point à point est donc une communication simple entre deux sans passer par une passerelle.



Figure 3 : mode point à point

Pour ouvrir une session, un utilisateur émet une invitation transportant un descripteur de session permettant aux utilisateurs souhaitant communiquer de s'accorder sur la comptabilité de leur média.

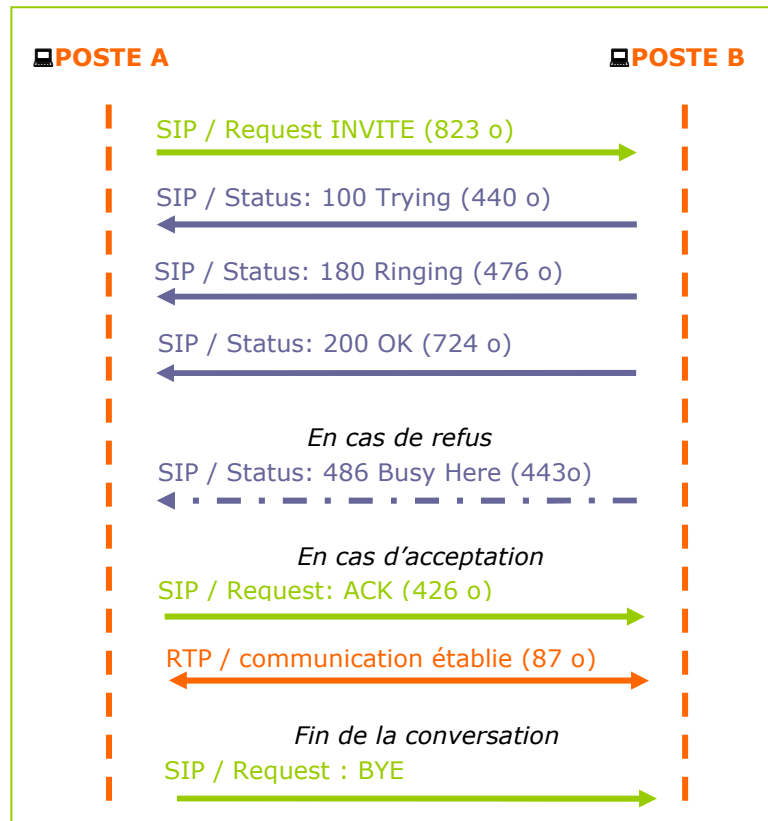
L'appelant et l'appelé doivent être identifiés via son URL SIP qui est du même type qu'une URL mailto (utilisateur@machine). Pour le mode point à point on utilise donc l'adresse IP du poste à joindre dans le logiciel de communication : *sip:nom@adresseip*.

Pour ouvrir une session, l'appelant envoie une requête contenant l'URL SIP du destinataire.

Lors de la mise en place de cette communication, plusieurs paquets sont échangés entre les deux postes :

- Invite : Permet d'informer le destinataire qu'une communication veut être établie avec lui et l'appelant.
- Trying : Essai d'établir la connexion,
- Ringing : Emet une sonnerie en attendant le décrochage du combiné distant.
- OK : Permet d'acquiescer une fois le combiné distant décroché.

- ACK : Cette requête permet de confirmer que le terminal appelant a bien reçu une réponse définitive à une requête Invite.
- RTP : Retrouver les explications de ce protocole dans la partie VoIP.
- BYE : Cette requête est utilisée par le terminal de l'appelé à fin de signaler qu'il souhaite mettre un terme à la session.



Mode diffusif

Le mode diffusif, contrairement au mode point à point, utilise une passerelle pour réaliser une communication entre deux éléments. Les clients sont enregistrés sur un serveur appelé **M.C.U.**^[5] qui va les identifier par rapport à un numéro. Lorsqu'un client veut appeler quelqu'un, il ne va donc plus utiliser l'adresse IP mais sont identifiant.

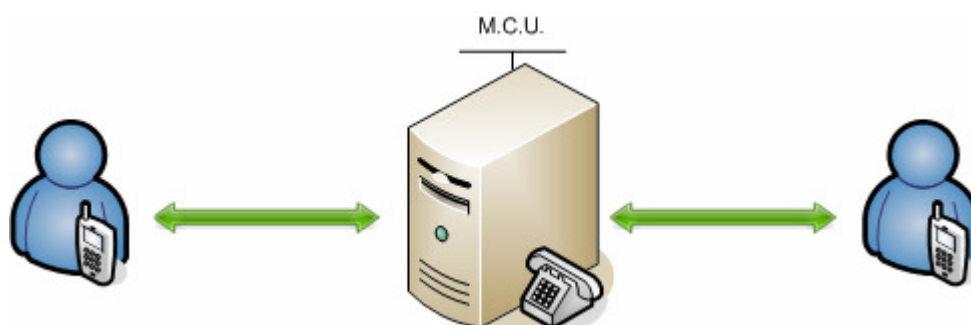
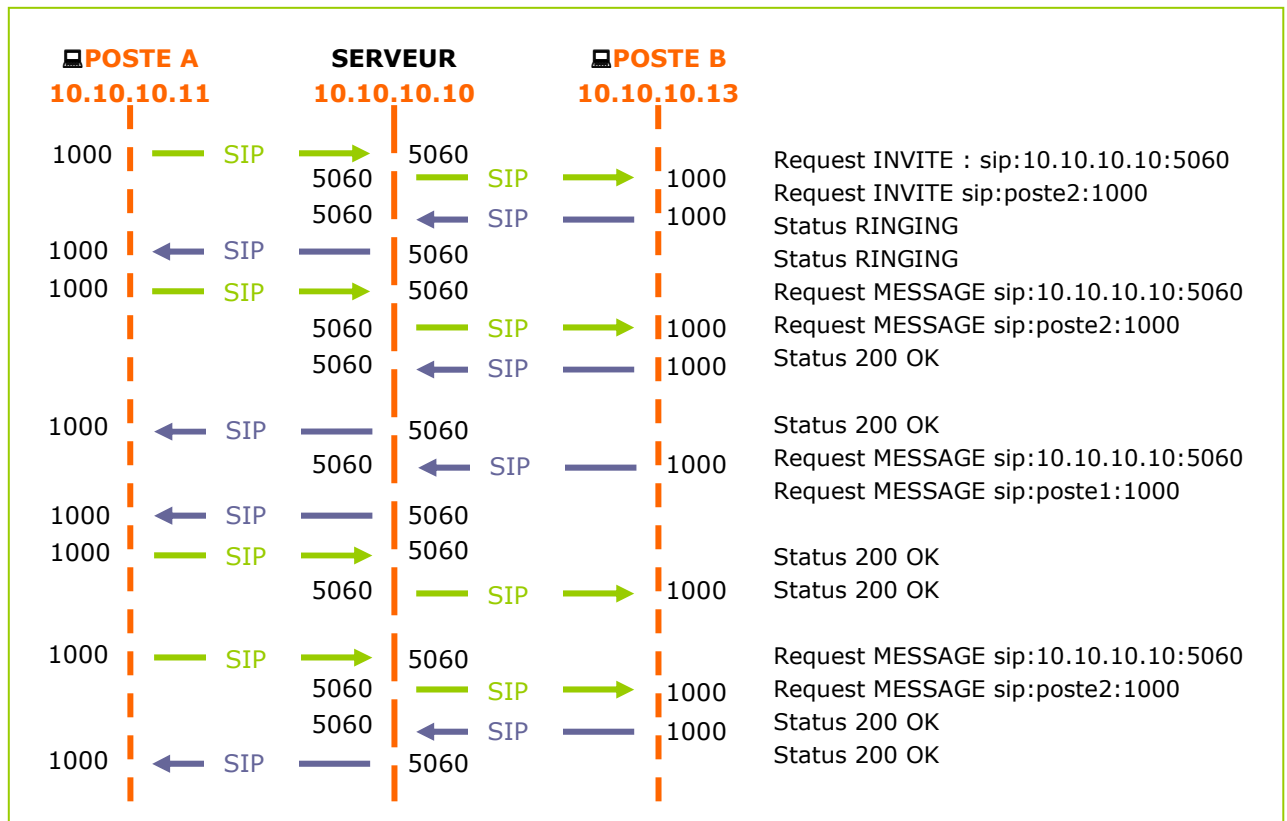


Figure 4 : mode diffusif



b. Différence H.323/SIP

SIP est un protocole jeune, contrairement à H.323 qui existe depuis un certain temps, il propose plus d'option tout en restant plus léger et plus facile d'utilisation.

	SIP	H.323
Nombre échanges pour établir la connexion	1,5 aller-retour	6 à 7 aller-retour
Maintenance du code protocolaire	Simple	Complexe et nécessitant un compilateur
Evolution du protocole	Protocole ouvert	Ajout d'extensions propriétaires
Fonction de conférence	Distribuée	Centralisée par le M.C.U. ^[5]
Fonction de téléservices	Oui	H.323 v2 + H.450
Détection d'un appel en boucle	Oui	Non
Signalisation multicast	Oui	Non

4. ASTERISK

Asterisk est un **PABX**^[3] logiciel libre, multi plateforme, publié sous licence **GPL**^[6]. Asterisk permet, entre autres, la messagerie vocale, la conférence, les serveurs vocaux, la distribution des appels. Asterisk implémente les protocoles H.323 et SIP, ainsi qu'un protocole spécifique nommé IAX. Il permet la communication entre client et serveur Asterisk ainsi qu'entre deux serveurs.

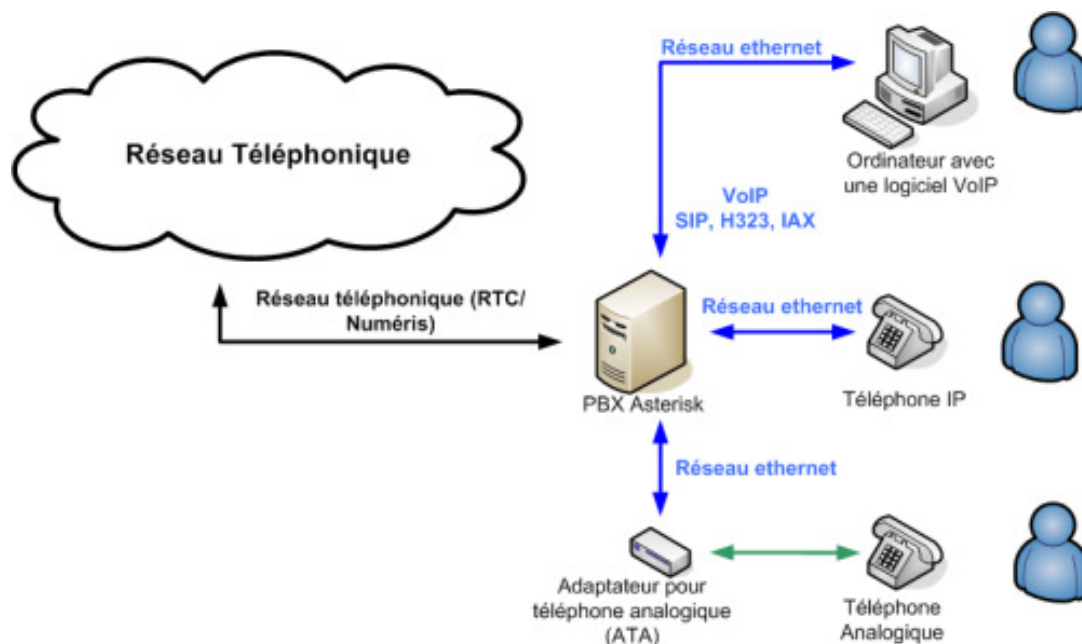
Historique :

Le PBX open source Asterisk a vu le jour quand Mark Spencer, a voulu acquérir un PBX traditionnel pour sa société. Le créateur d'Asterisk, trouvant que le prix d'acquisition d'un PBX traditionnel était démesuré, initia un projet open source. Il a donc commencé à développer Asterisk.

Remarque : Il faut savoir que le prix d'un petit **PABX**^[3] (100 postes) est de 3000 euros.

Son équipe s'est rapidement rapprochée de celle de Jim Dixon (Zapata Telephony Project), dans le but de construire des cartes d'interface "bon marché" avec le réseau téléphonique classique. Leur but était de construire des cartes compatibles avec des plateformes basées sur Intel et ainsi de permettre à n'importe quel PC, avec le système d'exploitation Linux, une carte d'interface et le logiciel Asterisk d'avoir un PBX entièrement fonctionnel.

Pour info : Le nom Asterisk fait référence au symbole "*" qui signifie "wildcard" en ligne de commande Unix et DOS. Ce choix a été fait car Asterisk est conçu dans le but d'offrir une très grande souplesse dans des réseaux de voix.



Fonctionnalités :

Asterisk offre les fonctionnalités suivantes :

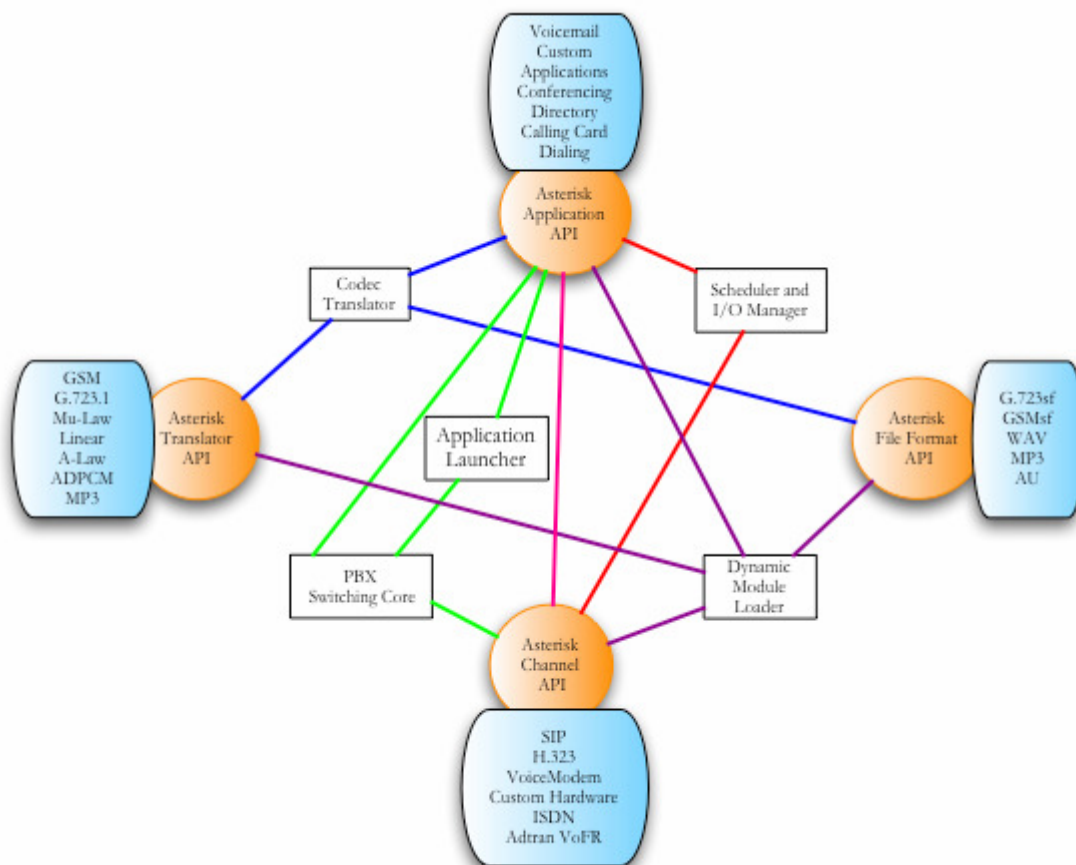
- Messagerie vocale
- Conférence téléphonique
- Répondeur vocal interactif
- Mise en attente d'appels
- Services d'identification de l'appelant
- VoIP

De la téléphonie à moindre coût :

Dans l'interconnexion avec les équipements de téléphonie numériques et analogiques, Asterisk reconnaît une large gamme de dispositifs matériels, et notamment ceux fabriqués par ses sponsors, tels que digium ou encore Quicknet.

Une particularité modulaire :

Asterisk est composé d'un noyau central de commutation, de quatre API (Interface de programmation d'applications) de chargement modulaire des applications téléphoniques, des interfaces matérielles, de traitement des formats de fichier, et des codecs. Il assure la commutation transparente entre toutes les interfaces supportées, permettant à cette commutation de relier entre eux une diversité de systèmes téléphoniques en un unique réseau commuté.



Principales fonctions :

PBX Switching Core :

Système de commutation de central téléphonique privé, reliant ensemble les appels entre divers utilisateurs et des tâches automatisées. Le noyau de commutation relie d'une manière transparente des appels arrivant sur divers interfaces de matériel et de logiciel.

Application Launcher :

Lance les applications qui assurent des services pour des usagers, tels que la messagerie vocale, la lecture de messages et le listage de répertoires (annuaires).

Codec Translator :

Utilise des modules de codec pour le codage et le décodage de divers formats de compression audio utilisés dans l'industrie de la téléphonie. Un certain nombre de codecs sont disponibles pour palier aux divers besoins et pour arriver au meilleur équilibre entre la qualité audio et l'utilisation de la bande passante.

Scheduler & I/O Manager :

Ils traitent la planification des tâches de bas niveau et la gestion du système pour une performance optimale dans toutes les conditions de charge.

Les APIs :**Asterisk Application API :**

Elle autorise différents modules de tâches à être lancé pour exécuter diverses fonctions. Communication, audioconférence, pagination, liste d'annuaire, messagerie vocale, transmission de données intégrée, et n'importe quelle autre tâche qu'un système PBX standard exécute actuellement ou exécuterait dans l'avenir, sont mises en oeuvre par ces modules distincts.

Asterisk Translator API :

Charge les modules de codec pour supporter divers formats de codage et de décodage audio tels que le GSM, la Mu-Law, l'A-Law, et même le MP3.

Asterisk Channel API :

Cette API gère le type de raccordement sur lequel arrive un appelant, que ce soit une connexion VoIP, un RNIS, un PRI, une signalisation de bit dérobé, ou une autre technologie. Des modules dynamiques sont chargés pour gérer les détails de la couche basse de ces connexions.

Asterisk File Format API :

Elle permet la lecture et l'écriture de divers formats de fichiers pour le stockage de données dans le file system.

Sa particularité modulaire permet à Asterisk d'intégrer de façon continue le matériel de commutation téléphonique actuellement mise en oeuvre, et les technologies de Voix par paquet en constante augmentation, émergeant aujourd'hui.

La capacité de charger des modules de codec permet à Asterisk d'être compatible avec le codec extrêmement compact nécessaire à la Voix sur IP sur des connexions lentes comme un modem téléphonique tout en maintenant une haute qualité audio sur des types de connexion moins "étroites".

Fonctionnement évolué :

Asterisk ne permet pas seulement l'utilisation d'équipements traditionnels de téléphonie, il augmente aussi en nombre leurs capacités.

En utilisant le protocole de voix sur IP Inter-Asterix eXchange (IAX ou inter central Asterisk), Asterisk mêle progressivement la voix et le trafic de données à travers des réseaux disparates. Tant que l'on transporte la voix par paquets, il est possible d'envoyer des données telles que des documents URL et des images, en conformité avec le trafic Voix, permettant ainsi une intégration plus grande des informations.

5. MISE EN PLACE DU SERVEUR

a. Installation du système

Le serveur qui héberge la plate-forme Asterisk est un serveur **GNU/Linux**^[7] fonctionnant avec un système d'exploitation Debian Sarge 3.1r2. L'installation se fait à partir du disque d'installation par le réseau disponible sur le site Web : www.debian.org.

Lors du boot du CD, initialiser l'installation par cette ligne :

```
linux26 DEBCONF_PRIORITY=medium vga=773
```

Cette ligne permet d'installer avec les options suivantes :

- **linux26** : Noyau 2.6.8.
- **DEBCONF_PRIORITY=medium** : Mode intermédiaire pour lequel plus de questions seront posées lors de l'installation.
- **vga=773** : Affichage 1024x768 avec un mode de 256 couleurs pour le terminal.

b. Configuration de base

Définition des sources d'installation :

Les sources d'installation sont utiles au système afin qu'il sache où il doit récupérer les paquets dont il pourrait avoir besoin. Pour définir les différentes sources d'installation, on peut utiliser la commande `apt-setup` ou bien modifier directement le fichier `/etc/apt/sources.list`.

On ajoute le serveur FTP de l'Université de Versailles Saint-Quentin-en-Yvelines en tant que miroir :

```
deb ftp://ftp.uvsq.fr/pub/debian/debian sarge main contrib non-free
deb-src ftp://ftp.uvsq.fr/pub/debian/debian sarge main contrib non-free
```

On peut également ajouter le serveur Debian regroupant les différentes mises à jour de sécurité:

```
deb http://security.debian.org/ sarge/updates main contrib non-free
```

Installation de paquets supplémentaires :

Afin d'ajouter quelques fonctionnalités utiles, on ajoute les paquets suivants :

- **gpm** : Service de support de la souris en mode console
- **less** : Utilitaire d'affichage relativement similaire à `more`
- **bzip2** : Compression Bzip2
- **host** : Utilitaire de résolution de nom
- **lftp** : Client FTP en mode console
- **manpages-fr** : Versions françaises des manuels
- **mutt** : Client de messagerie en mode console
- **ntpdate** : Configuration de l'horloge
- **ssh** : Serveur et client SSH (Secure Shell)

Configuration de l'interface réseau :

Il faut tout d'abord fixer une adresse IP au serveur. La configuration s'effectue dans le fichier `/etc/network/interfaces`, on donnera au serveur l'adresse IP 10.10.10.10 avec un masque de 255.255.0.0.

```
auto eth0
iface eth0 inet static
    address 10.10.10.10
    netmask 255.255.0.0
    gateway 10.10.0.1
    broadcast 10.10.255.255
```

Ensuite on relance le service réseau à l'aide de la commande :

```
/etc/init.d/networking restart
```

c. Mise en place du service Asterisk

Installation des paquets :

Liste des paquets nécessaires pour la mise en fonction du serveur :

- **asterisk** : Paquet principal ;
- **asterisk-config** : Fichiers de configuration d'Asterisk ;
- **asterisk-doc** : Documentation sur Asterisk ;
- **asterisk-sounds-main** : Fichiers sons pour Asterisk ;
- **asterisk-prompt-fr** : Fichiers de langue française pour Asterisk ;
- **mpg123** : Lecteur audio mpeg1 et 2 ;
- **sox** : Utilitaire d'édition et de conversion audio (wav, gsm, mp3,...).

Configuration de base d'Asterisk

Ajout de clients

La configuration de nouveaux clients SIP se fait dans le fichier `/etc/sip.conf` dans lequel on ajoute une entrée pour chaque client.

Plusieurs options permettent de définir et de paramétrer un client :

- **type** : Type de client (*peer, user ou friend*)
- **username** : Identifiant de l'utilisateur
- **secret** : Mot de passe de l'utilisateur
- **host** : Méthode pour trouver le client (*dynamique, nom d'hôte ou adresse IP*)
- **callerid** : Identité de l'utilisateur
- **language** : Langue par défaut pour l'utilisateur

▪ Description des paramètres :

Pour chacun des paramètres précédents, plusieurs valeurs sont disponibles selon la configuration désirée.

type :

- **peer** : Client SIP auquel Asterisk pourra envoyer des appels
- **user** : Client SIP qui pourra passer des appels via Asterisk
- **friend** : Client qui sera à la fois en mode 'peer' et 'user'

host :

- **dynamic** : Le client s'enregistre auprès du serveur
- **nom d'hôte** : Nom d'hôte du client
- **adresse IP** : Adresse IP du client

▪ *language* :

- **us** : Langue par défaut
- **fr** : Langue française

Exemple : Création de l'utilisateur Maxime

```
[1001]
type=friend
username=1001
secret=1001
host=dynamic
callerid="Maxime"
language=fr
```

Attribution de numéros d'appel

Pour attribuer un numéro à un client on édite le fichier `/etc/asterisk/extensions.conf` :

```
exten => 1001,1,Dial(SIP/1001,20,tr)
exten => 1001,2,VoiceMail(1001)
```

exten permet de définir une nouvelle extension :

- **1001** : numéro d'appel (ou d'extension) du serveur vocal ;
- **[1,2,3]** : ordre de priorité pour l'exécution ;
- **Dial, VoiceMail...** : Commande à exécuter.

Les fonctions `Dial()` et `VoiceMail()` seront détaillées plus bas avec les autres fonctions Asterisk. Ici l'appel vers la destination 1001 sera redirigé vers le serveur de messagerie vocale si jamais l'appel n'a pas été prit au bout du temps imparti.

d. IVR (Intelligent Voice Response)

Description des paramètres généraux du fichier `/etc/asterisk/extensions.conf` nécessaire au fonctionnement pour l'IVR :

Sauvegarder le plan de numérotation depuis la console :

```
static=yes
writeprotect=no
```

Le contexte global permet de définir des variables globales qui seront utilisées par la suite

```
[global]
```

Descriptif des fonctions

Pour effectuer la configuration des différentes extensions d'Asterisk nous n'avons utilisé que certaines fonctions du serveur définies dans `/etc/asterisk/extensions.conf`. Celles-ci vont être définies ci-après avec les options que nous avons retenues pour une utilisation standard de la téléphonie sur IP.

AgentLogin([numero_agent][options])

Si le numéro d'agent n'est pas spécifié, l'agent sera invité à saisir ses identifiants. La seule option disponible ici est l'option **s**. Si elle est ajoutée, l'agent n'entendra pas de message lors du succès de son authentification.

Answer()

Répondre à un appel.

Background(nom_fichier)

Lire un fichier son en tâche de fond. C'est généralement la fonction qui est utilisée pour les menus IVR.

Dial(type/identifiant[timeout][options][url]) :

Engager une connexion en fonction du type (*ici SIP*) vers un identifiant particulier. On peut également passer un timeout (*en secondes*) en paramètre, qui sera le temps maximal d'attente pour la réponse.

▪ *Options :*

- **A(fichier_audio)** : faire entendre le fichier audio_file à l'appelé ;
- **C** : réinitialiser les données CBR (*Call Detail Record*), ces informations sont utilisées pour la facturation ;
- **D(code_dtmf)** : permet l'envoi de codes **DTMF**^[8] à l'appelé lorsqu'il acceptera l'appel ;
- **f** : force le *CallerID* à être défini comme l'extension ;
- **g** : continuer dans l'exécution du contexte lorsque l'appelé raccroche ;
- **h** : autoriser l'appelé à raccrocher en pressant la touche * ;
- **H** : autoriser l'appelant à raccrocher en pressant la touche * ;
- **L(limite[:alerte][:répétition])** : mettre en place une limite de durée (*en millisecondes*) pour l'appel. Il est possible de définir un temps d'alerte au bout duquel une alerte sera donnée, ainsi que spécifier une répétition ;
- **m** : jouer la musique d'attente à la place de la sonnerie standard ;
- **M(macro)** : exécuter une macro lorsque la connexion s'établit ;
- **P()** : demander le numéro de l'appelant si aucun *CallerID* n'est envoyé ;
- **r** : génère une tonalité pour l'appelant (*par exemple : ligne occupée*) ;
- **S(timer)** : définir une durée maximale de communication, le timer démarre lorsque l'appelé prend l'appel ;
- **t** : autoriser l'appelé à transférer l'appel ;
- **T** : autoriser l'appelant à transférer l'appel ;
- **w** : autoriser l'appelé à démarrer l'enregistrement en pressant *1 ;
- **W** : autoriser l'appelant à démarrer l'enregistrement en pressant *1 ;

Goto([contexte][extension]priorité)

Se rendre à une priorité d'extension précise d'un contexte. Les paramètres contexte et extension sont facultatifs. Dans le cas où ils ne sont pas spécifiés les contexte et extension actuels sont utilisés.

GotoIfTime(horaires|jours_semaine|jours_mois|mois?[contexte][extension]priorité)

Similaire à la fonction *Goto()* précédente, sauf qu'il est possible de spécifier des plages de temps.

Hangup()

Terminaison d'appel.

MusicOnHold()

Jouer une musique d'attente indéfiniment.

Playback(nom_fichier,options)

Lire un fichier son (*il ne faut pas spécifier l'extension*) entièrement avant de rendre le contrôle, contrairement à la fonction *Background()* précédente.

Queue(nom_queue)

Placer l'appel dans la file d'attente spécifiée en paramètre.

Ringin()

Obtenir une tonalité sur le canal.

SetGlobalVar(variable=valeur)

Définir une variable à utiliser pour la configuration. On passe en paramètres les nom et valeur de la nouvelle variable.

VoiceMail(numero_boite@contexte)

Laisser un message à un ou plusieurs correspondants. L'appelant est alors invité à laisser son message.

VoiceMailMain([s]numero_boite@contexte)

Accéder à la messagerie d'un correspondant. Si le paramètre **s** est ajouté en début de chaîne, aucun mot de passe ne sera demandé.

Wait(durée)

Définir une durée d'attente avant de passer à la commande suivante.

Conversion du son

Asterisk utilisant un codage audio GSM, il faut donc convertir les sons dans ce format afin qu'ils puissent être intégrés dans l'application. Pour cela, l'utilitaire *sox* sur Linux permet de convertir des formats MP3 et WAV en GSM.

Il suffit d'utiliser une commande pour lancer cette conversion :

```
sox son.wav -r 8000 -c 1 son.gsm
```

Cette commande encode un fichier format WAV avec un échantillonnage de 8000Hz en format GSM.

Mise en place du menu

```
Lecture du message d'accueil
# > Accès au menu
  1 > Appeler un correspondant
    * > Numéro du correspondant
    0 > Retour au menu précédent
  2 > Accéder au serveur de messagerie
  3 > Joindre le support technique
  0 > Réécouter le message
```


La configuration du serveur vocal s'effectue dans le fichier `/etc/asterisk/extensions.conf`. On ajoute les différents éléments du menu et créant plusieurs extensions.

```
; Serveur vocal
exten => 100,1,Ringing
exten => 100,2,Wait(4)
exten => 100,3,Goto(accueil,s,1)
```

Entrée principale du menu

L'entrée principale du menu est celle à laquelle on attribut le numéro d'appel du serveur vocal, ici on lui a attribué le numéro 100.

e. Messagerie Vocale

La configuration se passe dans le fichier `/etc/asterisk/voicemail.conf`, la messagerie vocal peut être invoquer grace a l'action `Voicemail`, et peut être consulté grâce a l'action `VoicemailMain`, elle permet de laisser un message, de notifier l'arrivé d'un nouveau message par mail, de consulter et de sauvegarder ses messages.

Différente options sont disponibles pour la configuration globale du `Voicemail` :

Il est possible de définir les différents formats dans lesquels seront sauvegardé les messages vocaux. (plusieurs formats séparées par un | permettent de sauvegarder dans plusieurs format) :

```
format=wav49|gsm|wav
```

Ne pas mettre en attachement du mail le fichier son :

```
attach=no
```

Durée maximal du message (180 sec) :

```
maxmessage=180
```

Durée minimum du message (2 sec) :

```
minmessage=2
```

Définir l'expéditeur du message :

```
serveremail=asterisk@localhost
```

Nom de l'expéditeur :

```
fromstring=Service de Messagerie
```

Le sujet du message mail :

```
emailsubject=Nouveau message numéro ${VM_MSGNUM} dans la boite ${VM_MAILBOX}
```

Les différentes variables suivantes peuvent être intégré au sujet et corps du mail.

- **VM_NAME** : Le nom d'utilisateur
- **VM_DUR** : La durée du message
- **VM_MSGNUM** : Le numéro du message

- **VM_MAILBOX** : Le numéro de la boîte vocal
- **VM_CALLERID** : L'identifiant de l'appelant
- **VM_DATE** : La date actuelle

Corps du message mail :

```
emailbody=Cher(e) ${VM_NAME},\n\n\tVous avez un nouveau message de la part de
${VM_CALLERID}.\n Ce message dure ${VM_DUR} et il a ete envoye ${VM_DATE},\nvous
pouvez consulter votre boîte ${VM_MAILBOX} pour ecouter ce message.\n\n\t\t\t\tLe
Service de Messagerie\n
```

Permet de laisser un message à un utilisateur depuis le VoiceMail

```
sendvoicemail=yes
```

On peut choisir le fuseau horaire :

```
tz=central
```

Il est possible de relire et modifier le message lorsqu'on laisse un message :

```
review=yes
```

La configuration des boîtes vocal des utilisateurs sera placée dans le contexte [default] du fichier de la façon suivante :

```
numéro de la boîte => mot de passe,nom d'utilisateur,adresse mail, ,|les options
```

Exemple :

```
1001 => 1001,Maxime,mguillet@localhost,,|attach=no|nextaftercmd=yes|review=yes
```

Les options globales peuvent être définies dans les options pour chaque utilisateur.

f. Les files d'attente et les agents

Il est possible de créer des files d'attente et de définir des agents qui seront chargés de répondre au appel de ces files d'attentes.

3 fichiers doivent être modifiés :

- `agents.conf` : La définition des agents ;
- `queues.conf` : La définition des files et les agents qui leur sont associés ;
- `extensions.conf` : Le numéro d'appel de la file d'attente.

▪ `agents.conf` :

Dans le contexte [`agents`], il est nécessaire de créer des agents de la façon suivante :

```
agent => numéro d'agent,mot de passe de l'agent,Nom de l'agent
```

exemple : `agent => 2001,2001,Maxime`

▪ `queues.conf` :

Les agents vont être associés au contexte :

```
member => Agent/numéro d'agent
```

exemple : member => Agent/2001

- *extensions.conf* :

L'option *AgentLogin()* permet a l'agent de s'authentifier et de recevoir des appels depuis la file d'attente.

L'option *Queue(nom de la file)* permet aux utilisateurs de contacter les agents de la file ou d'être mis en attente si aucun ne sont disponible.

6. CLIENTS

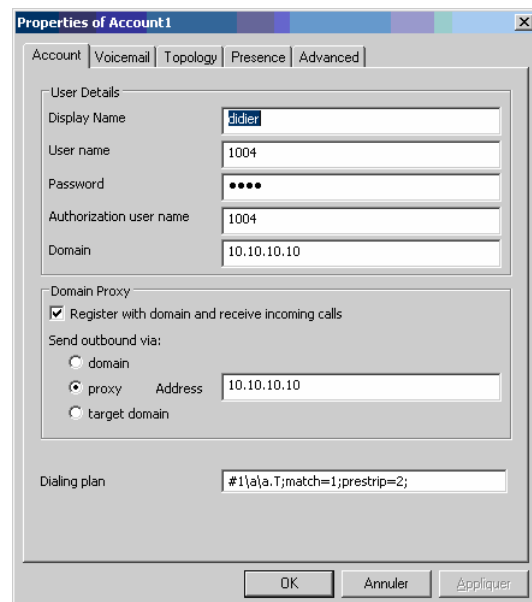
Le client SIP (Soft Phone) permet de jouer le rôle d'un téléphone IP de manière logiciel. Il requière un système de son (carte son, baffles, micro,...) pour l'utiliser. Ces clients permettent de remplacer des téléphones IP matériels onéreux (100, 200€).

Beaucoup de client existe plus au moins évoluée, cependant certains sont des client propriétaire et ne sont pas utilisables avec Asterisk.

a. Clients

X-Lite

Ce client est un des plus abouti en terme de fonctionnalités, de fiabilités et de simplicité. Il permet la gestion de contacts et de groupes et fait également la messagerie instantané. Il existe en version payante (EyeBeam) (Plusieurs comptes SIP, support de la vidéo,...). Il fonctionne également sous Windows.



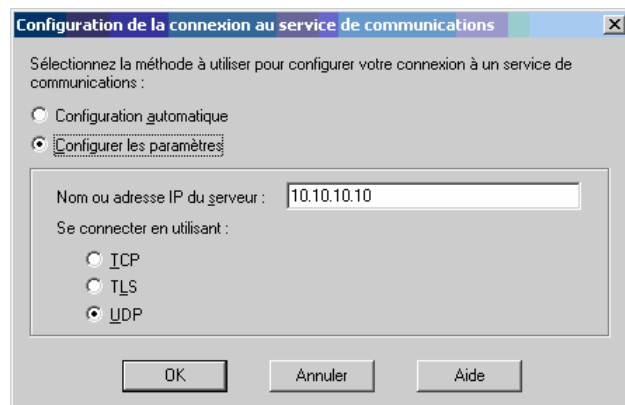
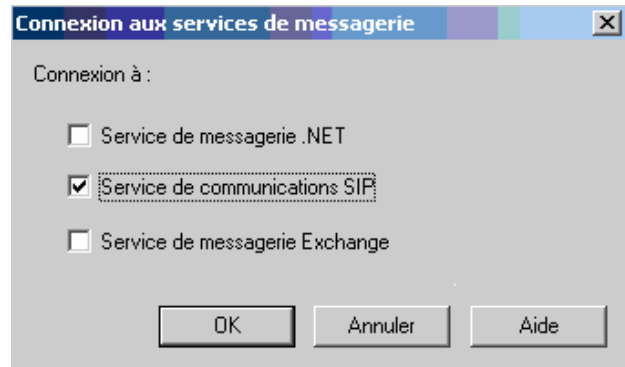
Configuration du client X-lite en protocole SIP :

Pour le mettre en place avec la passerelle Asterisk, cliquer droit sur l'interface *X-lite* ► *SIP Account Settings* puis *Proprieties*. On renseigne les champs *Display Name*, *User name* et *Password* qui permettent au client de s'identifier sur le serveur.

Les champs *Authorization user name* est défini sur le Serveur Asterisk et on rentre l'adresse IP du serveur dans le champ *Domain* (on notera que l'adresse IP du proxy est la même que celle du serveur).

Windows Messenger

C'est un logiciel de messagerie instantanée et de visioconférence développé par Microsoft et n'existe pas sous Linux. Il est compatible avec SIP et permet de communiquer avec un serveur type Asterisk.



Configuration du client en protocole SIP :

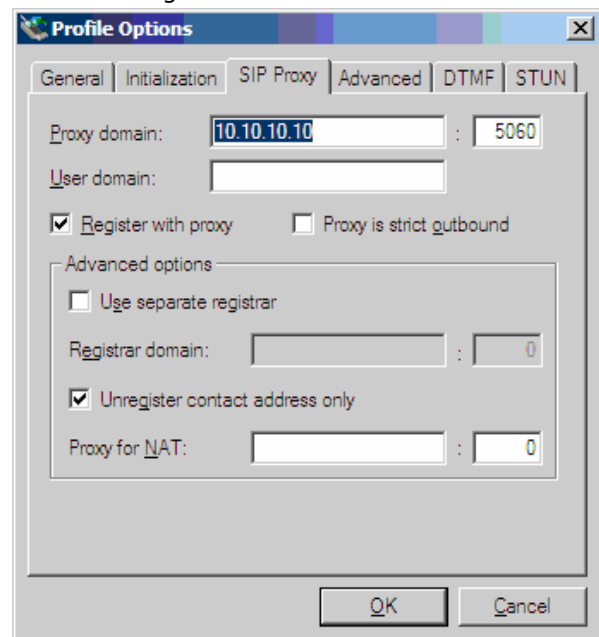
- Cocher *Service de communications SIP* au démarrage du logiciel
- Aller dans le menu *Outils* ► *options* ► *comptes* ► *Avancé* et renseigner le nom ou l'adresse IP du serveur en précisant le protocole UDP

Pour passer un appel

Aller dans *Actions* ► *démarrer une conversation vocale* et dans le champ *adresse de messagerie* on renseigne l'ID (ou le nom) de l'appelant et l'adresse IP ou le nom du serveur Asterisk.

SJPhone

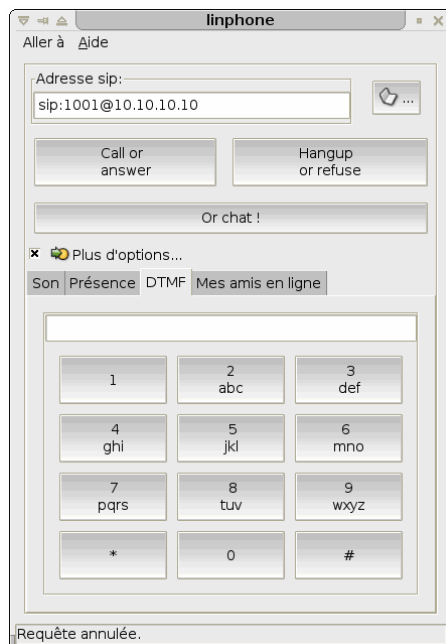
Ce logiciel gère les deux normes SIP et H.323 ce qui le rend compatible avec la plupart des vendeurs et fournisseur de services en téléphonie IP et existe également sous Linux.



Configuration du client SJphone en protocole SIP :

- Dans le menu *Options* ► *Profiles* on crée un nouveau profil "New"
- Dans le champ *Proxy domain* on rentre l'adresse IP du serveur avec le numéro de port : 5060 correspondant au protocole SIP.

Linphone



Linphone est un client exclusivement Linux. Etant assez basique, il fait uniquement les fonctions nécessaire pour une communication SIP.

Configuration du client en protocole SIP :

- Dans le Menu *Aller à* ► *Préférences*, un menu apparaît
- Dans l'onglet *SIP*, spécifier l'ID de l'utilisateur et l'adresse IP du serveur

b. Comparatif

	X-lite	SJphone	Linphone	Windows Messenger
Mode conférence	OUI	OUI		NON
Gestion de l'IVR	OUI	OUI	OUI	NON
Carnet de contacts	NON	NON	OUI	OUI
Gestion de la messagerie	OUI	OUI	OUI	NON
Messagerie	OUI	NON	NON	OUI
Qualité d'écoute	+	++	+	-
Simplicité d'utilisation	++	+	++	-
Compatibilité H323	NON	OUI	NON	NON
Codecs	DVI4, G.711, uLaw/aLaw, GSM, iLBC	GSM 6.10, iLBC codec, G.711 A-law, G.711 U-law	PCMU, GSM, PCMA, Speex, 1015	

7. REPARTITION DES TACHES

Tâches	Didier OURABAH	Maxime GUILLET	Vincent BATOUFFLET	Léonard LECOUEY	William ZIVIC
Installation DEBIAN et configuration					
Installation d'Asterisk					
Configuration de sip.conf, création des utilisateurs					
Configuration IVR dans extensions.conf					
Configuration de VoiceMail dans voicemail.conf -					
Création des files d'attentes					
Configuration des différents clients SIP					
Configuration d'un VPN distant					
Configuration Asterisk pour appel freephonie					

Rédaction	Didier OURABAH	Maxime GUILLET	Vincent BATOUFFLET	Léonard LECOUEY	William ZIVIC
La VoIP					
Protocole SIP					
Asterisk					
Les clients SIP					
Configuration Asterisk					
Mise en place du How-to					
Power-Point					

8. GESTION DU PROJET

Etant en groupe de cinq pour la conception et le développement du projet, il nous a été plus complexe de nous accorder et de travailler en commun.

Pour respecter les délais, nous nous sommes fixés une feuille de route à suivre, nous permettant de connaître le temps restant à chaque fois qu'une tâche était accomplie.

Ce projet a été pour nous l'occasion de travailler en groupe et de révéler nos capacités. Au vue de l'ensemble du travail demandé, il nous a paru judicieux de travailler le plus possible ensemble, et de manière complémentaire afin de collecter le plus possible les informations et les actions personnelles.

Nous nous sommes impliqués de manière à apprendre, ensemble, sur tous les aspects du projet.

Le schéma ci-dessous rend compte de la démarche adoptée au cours de l'évolution du projet :

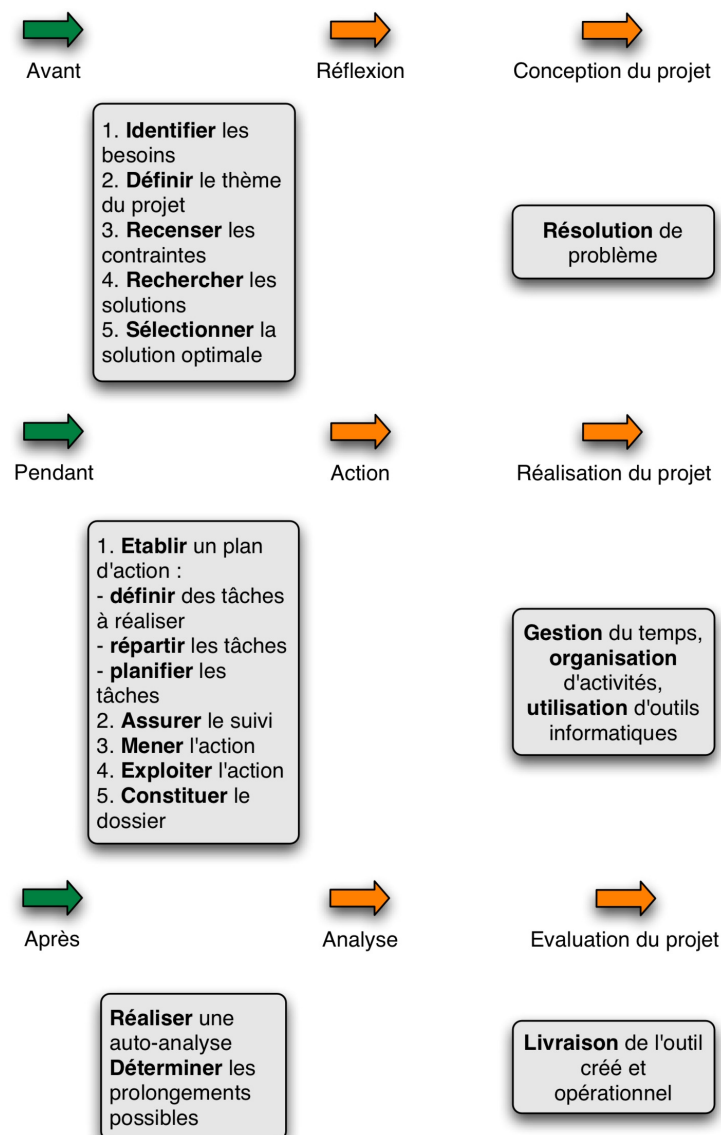


Figure 5 : Gestion de projet

9. CONCLUSION

Ce projet nous a permis de mettre en pratique nos connaissances théoriques acquises en cours de réseaux et télécommunications.

Nous nous sommes de ce fait, initié au travail collectif et avons acquis de multiples connaissances tant dans la conception d'un réseau VoIP, que dans les multiples fonctionnalités avancées d'un PABX logiciel.

Nous avons de ce fait constaté que le logiciel libre Asterisk pouvait faire de l'ombre aux gros centraux téléphoniques.

Cette expérience enrichissante représente pour notre futur professionnel un atout majeur dont nous ne manquerons pas de nous y référer.

10. LEXIQUE

1. **TCP/IP** : La suite des protocoles Internet est l'ensemble des protocoles qui constituent la pile de protocoles utilisée par Internet. Elle est souvent appelée TCP/IP, d'après le nom de deux de ses protocoles : TCP (*Transmission Control Protocol*) et IP (*Internet Protocol*).
2. **UDP** : (*User Datagram Protocol*) est contrairement au protocole TCP, il travaille en mode non connecté : il n'y a pas de moyen de vérifier si tous les paquets envoyés sont bien arrivés à destination et ni dans quel ordre.
3. **PABX** : Un PABX (*Private Automatic Branch eXchange*) est un commutateur téléphonique qui sert principalement à relier les postes téléphoniques d'un établissement (*lignes internes*) avec le réseau téléphonique public (*lignes externes*). Il permet en plus la mise en œuvre d'un certain nombre de fonctions.
4. **TCP** : Transmission Control Protocol est un protocole de transport fiable, en mode connecté.
5. **MCU** : Un multipoint control unit est un logiciel informatique ou une machine servant à établir simultanément plusieurs communications.
6. **GPL** : General Public Licence ou Licence publique générale GNU (*GNU GPL*), une licence pour les logiciels libres.
7. **GNU/Linux** : UNIX est le nom d'un système d'exploitation créé en 1969, à usage principalement professionnel, conceptuellement ouvert et fondé sur une approche par laquelle il offre de nombreux petits outils chacun dotés d'une mission spécifique, multitâche et multi-utilisateur. Il a donné naissance à une famille de systèmes, dont les plus populaires en 2005 sont GNU/Linux, BSD et Mac OS X. Les UNIX sont aujourd'hui tous proches de la norme POSIX.
8. **DTMF** : (*Dual-Tone Multi-Frequency*) Combinaisons de fréquences utilisées pour la téléphonie à impulsion moderne.