REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
– MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE SCIENTIFIQUE –
ECOLE NATIONALE POLYTECHNIQUE.

DEPARTEMENT D'ELECTRONIQUE

Mémoire de fin d'études

En vue de l'obtention du
Diplôme d'ingénieur d'état en Electronique

ETUDE ET MISE EN ŒUVRE D'UN IPBX SOUS LINUX

Proposé et dirigé par :

M[r] R.SADOUN

Réalisé par:

Mlle GADOUCHE Amina Lamia
Mlle MOULLA Fatima

Table des matières

Chapitre 2 – Le protocole SIP-

Chapitre 3 – Etude et mise en œuvre d'un IPBX « Asterisk »

Conclusion générale

Annexes

Annexe A : Les codes SIP les plus courants

Annexe B : Les champs d'en-têtes SIP

Annexe C : Commandes et applications Asterisk

Références bibliographiques

LES ACRONYMES

A

ACD	Automatic Call Distribution

C

CallID	Call Identifier
CID	Call Identifier
CR LF	Cariage Return, Line Feed
CR	Cariage return
CSRC	Contributing SouRCe(protocole RTP)
CTI	Computer Telephony Integration

D

DHCP	Dynamic Host Configuration Protocol
DNS	Domain Name System
DTMF	Dual Tone Multi-Frequency

F

FCFS	First Come First Saved

G

GSM	Global System of Mobile communication

I

IETF	Internet Engeneering Task Force
IGPM	Internet Group Management protocol
INTRSERV	INTegrated SERVices
IP	Internet Protocol
IPBX	Internet Protocol Private Branch eXchange (ou encore IP-PBX)
ISDN	RNIS
ISO	International Standardization Organization
IVR	Interactive Voice Response

L

LAN	Local Area Network
LDAP	Lighweight Directory Acess Protocol
LS	Algorithme de routage par état de lien

M

MCU	Multipoint Cotrol Unit
MMUSIC	Multiparty Miltimedia Session Control
MTU	Maximum Transmission Unit

P

PBX	Private Branch eXchange
PDU	Protocol Data Unit
POT	Post Old Telephony(téléphone analogique)
PPP	Point to Point Protocol
PSTN	Public Switched Telephone Network

Q

QoS	Quality of Service

R

RFC	Request For Comment
RNIS	Réseau Numérique à Intégration de
RTC	Réseau Téléphonique Commuté
RTCP	Real Time Control Protocol (voir la RFC 1889)
RTP	Real Time Protocol (voir la RFC 1889)
RTP/AVT	Real Time Protocol /Audio Video Transport
RTSP	Real Time Streaming Protocol

S

SAP	Session Announcement Protocol
SDP	Session Description Protocol
SIP	Session Initialisation Protocol
SLIP	Serial Line Internet Protocol
SMPTP	Simple Mail Transmission Protocol
SP	Single sPace
SVI	Serveur Vocal Interactif

T

TCP	Transport Control Protocol
TDM	Time Division Multiplex
ToIP	Telephony over Intenet Protocol
TOS	Type Of Service
TTC	Telecommunication Technologies Committee

	TTL	Time To Live
U		
	UAC	User Agent Client
	UAS	User Agent Server
	UDP	User Datagram Protocol
	UIT	Union Internationle des Télécommunications
	URI	Uniform Resource Identifier
	URL	Uniform Ressource Locator
V		
	VoIP	Voice over Intenet Protocol

Liste des figures

Liste des tableaux

Introduction générale

INTRODUCTION GENERALE

La voix sur IP (Voice over IP) est une technologie de communication vocale en pleine émergence. Elle fait partie d'un tournant dans le monde de la communication. En effet, la convergence du triple play (voix, données et vidéo) fait partie des enjeux principaux des acteurs de la télécommunication aujourd'hui. Plus récemment l'Internet s'est étendu partiellement dans l'Intranet de chaque organisation, voyant le trafic total basé sur un transport réseau de paquets IP surpasser le trafic traditionnel du réseau voix (réseau à commutation de circuits). Il devenait clair que dans le sillage de cette avancée technologique, les opérateurs, entreprises ou organisations et fournisseurs devaient, pour bénéficier de l'avantage du transport unique IP, introduire de nouveaux services voix et vidéo. Ce fût en 1996 la naissance de la première version voix sur IP appelée H323. Issu de l'organisation de standardisation européenne ITU-T , ce standard a maintenant donné suite à de nombreuses évolutions, quelques nouveaux standards prenant d'autres orientations technologiques.

Pour être plus précis, le signal numérique obtenu par numérisation de la voix est découpé en paquets qui sont transmis sur un réseau IP vers une application qui se chargera de la transformation inverse (des paquets vers la voix). Au lieu de disposer à la fois d'un réseau informatique et d'un réseau téléphonique commuté (RTC), l'entreprise peux donc, grâce à la VoIP, tout fusionner sur un même réseau. Ça par du fait que la téléphonie devient de la "data". Les nouvelles capacités des réseaux à haut débit devraient permettre de transférer de manière fiable des données en temps réel. Ainsi, les applications de vidéo ou audioconférence ou de téléphonie vont envahir le monde IP qui, jusqu'alors, ne pouvait raisonnablement pas supporter ce genre d'applications (temps de réponse important, Qos). Jusque vers le milieu des années 90, les organismes de normalisation ont tenté de transmettre les données de manière toujours plus efficace sur des réseaux conçus pour la téléphonie. A partir de cette date, il y a eu changement. C'est sur les réseaux de données, que l'on s'est évertué à convoyer la parole. Il a donc fallu développer des algorithmes de codage audio plus tolérants et introduire des mécanismes de contrôle de la qualité de service dans les réseaux de données. Faire basculer différents types de données sur un même réseau permet en plus, de simplifier son administration.

Comme toute innovation technologique qui se respecte, la VoIP doit non seulement simplifier le travail mais aussi faire économiser de l'argent. Les entreprises dépensent énormément en communications téléphoniques, or le prix des communications de la ToIP (Téléphonie sur IP) est dérisoire en comparaison. En particulier, plus les interlocuteurs sont éloignés, plus la différence de prix est intéressante. De plus, la téléphonie sur IP utilise jusqu'à dix fois moins de bande passante que la téléphonie traditionnelle. Ceci apportant un grand intérêt pour la voix sur réseau privée. Il semblerait que les entreprises après avoir émis un certain nombre de doutes sur la qualité de services soient désormais convaincues de la plus grande maturité technologique des solutions proposées sur le marché. Qu'il s'agisse d'entreprises mono-site ou multisites, les sondages montrent que le phénomène de migration vers les systèmes de téléphonie sur IP en entreprise est actuellement engagé.

Les premières technologies de VoIP imaginées étaient propriétaires et donc très différentes les unes des autres. Pourtant, un système qui est censé mettre des gens et des systèmes en relation exige une certaine dose de standardisation. C'est pourquoi sont apparus des protocoles standards, comme le H323 ou le SIP (Session Initiation Protocol).

Donc le premier bénéfice de la VoIP est l'allégement des factures téléphoniques intra entreprises, voire entreprises- fournisseurs-clients dans le cas de réseaux étendus. Mais l'entreprise peut aller plus loin. Elle peut faire le choix du tout-IP et miser sur la téléphonie sur IP. Elle pourra alors remplacer ses anciens combinés téléphoniques par des terminaux IP (téléphones IP ou PC équipés d'un logiciel de téléphonie) et cet exemple se place dans un contexte de WAN ou de réseau étendu. Reste qu'un changement d'architecture n'est pas toujours facile à justifier. Certains défendent néanmoins que la téléphonie sur IP est déjà rentable au niveau d'un LAN ou réseau local. Ici, le surcoût de la téléphonie classique est à mettre au compte du déplacement fréquent des téléphones (déménagements, aménagements de nouveaux bureaux...) et à la gestion du câblage. Avec la téléphonie sur IP et l'IPBX, ce souci disparaît car les terminaux, dotés chacun d'une adresse IP, peuvent être connectés instantanément à n'importe quel endroit du réseau en bénéficiant de toutes les fonctionnalités d'un PBX traditionnel. C'est là que se situe le but de ce projet, à travers lequel nous tenterons de mettre en œuvre un IPBX au niveau d'un LAN. Pour cela nous ferons une étude de la technologie utilisée qui est la VoIP et des réseaux téléphoniques dans un premier chapitre.

Puis nous nous intéresserons dans le deuxième chapitre au protocole de signalisation SIP que nous étudierons en détail.

Dans le troisième chapitre, nous exposerons notre application qui consiste en l'étude d'un IPBX et de la mise en œuvre de ce dernier au niveau d'un réseau local utilisant le protocole SIP.

Chapitre 1

La Téléphonie

1.1. Réseau téléphonique public

1.1.1. Introduction

Le réseau téléphonique public, PSTN ("Public Switched Telephone Network") ou RTPC (Réseau Téléphonique Public Commuté) constitue l'un des plus importants réseaux au monde ayant quelques centaines de millions d'abonnés.

Essentiellement analogique au départ, le réseau s'est progressivement numérisé : la transmission dans le réseau d'abord, suivie par la commutation ensuite. La dernière partie numérisée reste la partie locale, c'est-à-dire la connexion de l'abonné au réseau : c'est un des objectifs du RNIS.

1.1.2. Histoire de la téléphonie

Du premier télégraphe de Chappe en 1790 au RTC actuelle, l'histoire des communications a connu de grands moments et de grandes avancés dû à l'ingéniosité de certains et aux progrès technologique et électronique. Nous retiendrons quelques grandes dates tel que :

- 1837 Premier télégraphe électrique inventé par Samuel Morse
- 1889 Almon B. Strowger (USA) invente le premier « sélecteur » automatique et donne ainsi naissance à la commutation téléphonique automatique
- 1938 Alec Reeves (Français) dépose le brevet des futurs systèmes à modulation par impulsion et codage (MIC) : quantification et échantillonnage du signal à intervalles réguliers, puis codage sous forme binaire.
- 1962 Les premiers systèmes de transmission multiplex de type MIC apparaissent aux Etats-Unis ils permettent une liaison à 24 voies entre centraux téléphonique, à la même époque en France on installe des MIC à 32 voies.
- 1970 Un nouveau pas est franchi dans le domaine de la commutation électronique avec la mise en service en France, par le CNET, des premiers centraux téléphoniques publics en commutation électronique temporelle.
- 1979 Lancement du minitel en France

- 1987 Le RNIS est mis en service en France.
- 1990 De nouveaux concepts apparaissent tel que la commutation temporelle asynchrone (ATM) et la hiérarchie numérique synchrone.

1.1.3. Principe du RTC

Le réseau téléphonique public commuté (RTCP, ou simplement RTC) est un outil de communication composé de nœuds (commutateurs) a essentiellement pour objet la transmission de la voix. L'échange d'informations se fait au moyen de protocoles de communication appelés systèmes de signalisation. Ils sont basés la plupart du temps sur l'émission de fréquences. L'ensemble des données du réseau doit être géré localement au niveau de chaque commutateur.

1.1.4. Schéma global du RTC

- Le réseau téléphonique commuté (RTC) met en relation deux postes d'abonné
- Le protocole pour établir, maintenir et rompre la relation s'appelle la signalisation
- Les 3 étages du réseau :

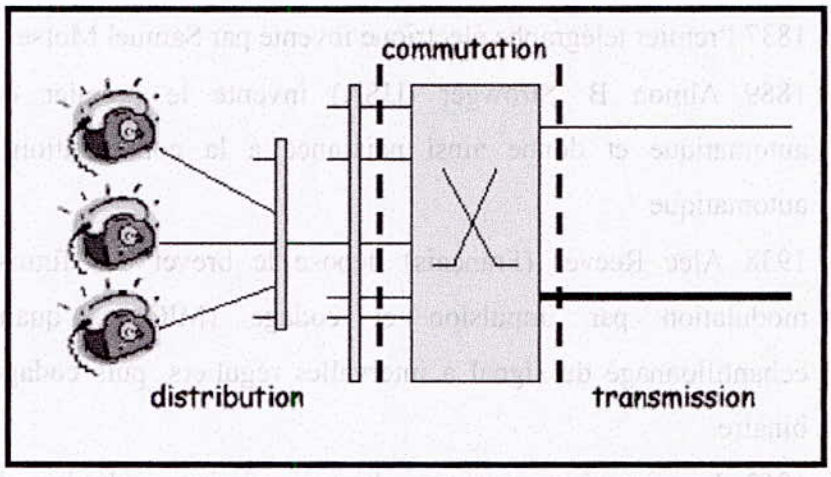


***Figure 1. 1** . Les 3 étages du réseau RTC*

- **La commutation** : partie centrale du réseau qui mets en relation les abonnés. Un commutateur peut être un concentrateur ou un aiguilleur ;
- **La transmission :** la liaison de l'ensemble des commutateurs (réseau de transmission ou réseau de transport) ;
- **La distribution** : le réseau reliant les abonnés au commutateur le plus proche (le commutateur de rattachement).

1.1.5. La hiérarchie du réseau

- **Zone à autonomie d'acheminement (ZAA)** : les commutateurs (CAA) accueillent les abonnés et établissent les communications locales. A noter aussi les concentrateurs de trafic dans les zones dispersées : quelques dizaines de milliers d'abonnés
- **Zone de transit secondaire (ZTS)** : contient les commutateurs "internes" (CTS). Assure le routage si nécessaire.
- **Zone de transit principale (ZTP)** : un CTS est relié à un CTP, lui-même éventuellement à un commutateur de transit international (CTI)

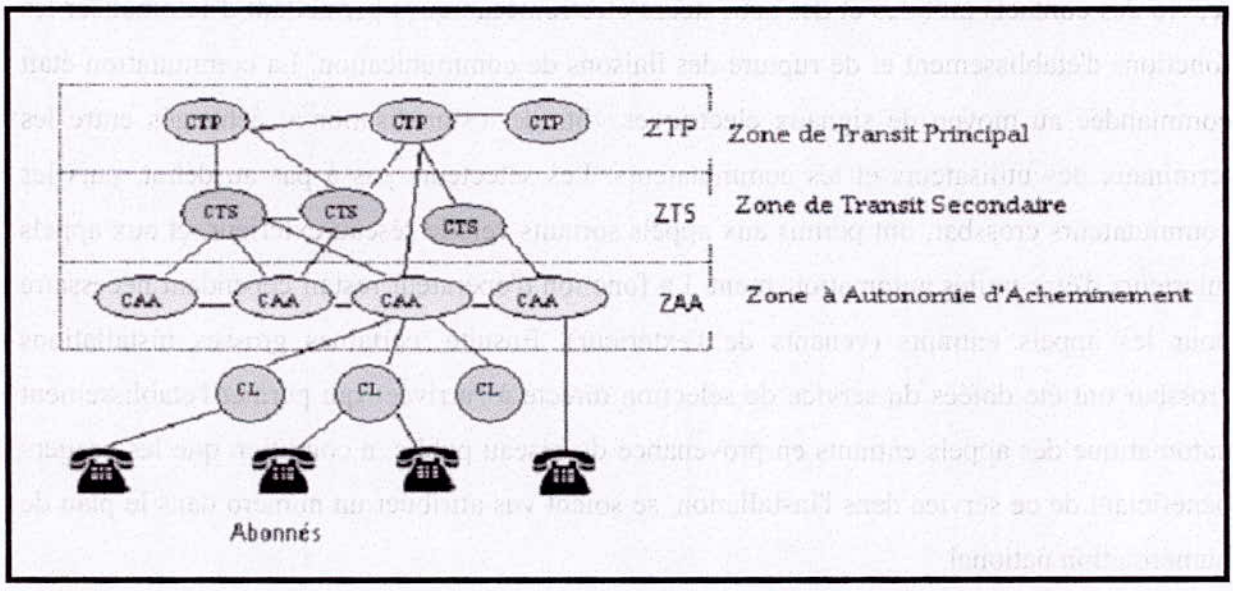


Figure 1. 2. *La hiérarchie du réseau RTC*

1.2. Les installations téléphoniques privées

1.2.1. Introduction

Les installations téléphoniques privées se sont développées en même temps que les grands réseaux téléphoniques publics, dont elles étendent et développent, au sein d'une même organisation, les fonctions de mise en relation de personnes.

À l'époque de la commutation manuelle, l'installation téléphonique privée constituait à la fois un outil de mise en communication intérieure et de démultiplication de l'utilisation des lignes extérieures, reliant le groupe au réseau public. Une fonction d'opérateur était chargée de relier les lignes intérieures entre elles ou aux lignes extérieures, suivant la demande des utilisateurs. Cela se faisait manuellement à l'aide d'un cordon de liaison appelé dicorde s'enfichant des deux côtés dans des prises dites jacks correspondant aux lignes à relier, établissant ainsi une liaison de communication entre les utilisateurs.

Des systèmes de commutation automatique ont ensuite été introduits. Ils mettaient en œuvre des contacts mobiles et des activateurs électromécaniques permettant d'automatiser les fonctions d'établissement et de rupture des liaisons de communication. La commutation était commandée au moyen de signaux électriques, dits de « signalisation », échangés entre les terminaux des utilisateurs et les commutateurs. Les sélecteurs pas à pas au début, puis les commutateurs crossbar, ont permis aux appels sortants vers le réseau extérieur, et aux appels intérieurs d'être traités automatiquement. La fonction d'opérateur restait cependant nécessaire pour les appels entrants (venants de l'extérieur). Ensuite, certaines grosses installations crossbar ont été dotées du service de sélection directe à l'arrivée qui permet l'établissement automatique des appels entrants en provenance du réseau public, à condition que les usagers bénéficiant de ce service dans l'installation, se soient vus attribuer un numéro dans le plan de numérotation national.

Les techniques de télécommunications ont été marquées par d'importantes mutations à la suite du développement de l'informatique et de l'électronique. Cela s'est traduit par les évolutions fondamentales suivantes :

- la généralisation de l'électronique en remplacement de l'électromécanique pour les dispositifs de commutation et de signalisation ;

- le passage à la commutation temporelle puis à la commutation par paquets en remplacement de la commutation spatiale ;
- la généralisation des dispositifs à programme enregistré pour la commande des installations, à la place des logiques dites câblées de l'époque.

Ces évolutions ont rapidement été appliquées au domaine des PABX (Private Automatic Branch Exchange), modifiant totalement la définition, la forme et même les fonctions des installations téléphoniques privées. Elles ont permis en particulier de faire du PABX un outil offrant de multiples services utiles aux utilisateurs, et débordant largement la simple fonction de commutation de la voix pour laquelle il avait été imaginé au début. [17]

Le PABX actuel, est une machine qui utilise les mêmes technologies de base que l'informatique, mais il met en œuvre des programmes d'applications spécifiques au domaine de la communication vocale, fournissant ce qu'il est habituel d'appeler des « services » de communication (en informatique « applications »). Ces services restent centrés sur la communication « de personne à personne » (dérivé du terme anglais person to person communication), mais ils débordent largement la simple liaison vocale, pour offrir toute une panoplie de services dits « complémentaires », justement par rapport à la pure liaison vocale. Le PABX est maintenant étroitement associés aux équipements et applications informatiques sur lesquels il s'appuie pour les fonctions orientées vers la gestion, que l'informatique est mieux placée pour traiter.

Si l'on veut décrire très simplement la forme que prend l'installation téléphonique privée, on pourrait dire qu'il s'agit d'un ensemble d'entités fonctionnelles interconnectées entre elles, dont on distinguera trois types :

- le terminal d'utilisateur : poste téléphonique, poste multifonction, micro-ordinateur, téléphone IP .... ;
- le nœud de communication, plate-forme PABX ;
- les serveurs, qui sont des ordinateurs portant des applications qui entrent dans le cadre des services offerts par les installations téléphoniques privées.

1.2.2 La commutation

L a commutation est la réalisation d'une liaison entre les équipements terminaux.

Le tableau suivant résume les différents types de commutation :

Commutation.	Principe
De circuit	Établissements à la demande d'une liaison temporaire à usage exclusif par la mise en connexion de circuit de données à chaque commutateur traversé. Exemple RTC
De message	Les commutateurs réceptionnent, stockent et transfèrent des messages complets. Fonction de mémorisation ; Taille quelconque ; risque de saturation.
De paquets	Les données à transmettre sont structurées en paquets acheminés par les commutateurs. Le message source va être segmenté en sous messages de taille fixe ; aux quels sont ajoutés des identifiants. Une voie est occupée uniquement durant le temps de transmission d'un paquet, elle est ensuite disponible pour le transfert d'autres paquets. L'acheminement des paquets s'effectue après l'établissement d'un chemin virtuel (logique) entre les deux systèmes d'extrémité.
En mode datagramme	Il n'y a pas d'établissement préalable de chemin logique, les paquets d'un même

	message sont acheminés indépendamment les uns des autres. Exemple : Internet.
Spatiale	Un chemin physique de bout en bout est réalisé pour chaque commutation.
Temporelle	Est basée sur le multiplexage temporel. Elle permet de partager dans le temps un même support physique de transmission entre des utilisateurs différents.
ATM	Permet de transporter par commutation de paquets (de taille petite et fixe, dénommé cellule) des données de nature quelconque sans mettre en oeuvre toute une panoplie de services de réseau comme cela nécessairement X. 25. Asynchrone précise que la commutation permet un fonctionnement asynchrone des horloges des systèmes émetteurs et récepteurs. Exemple : RNIS large bande.

Tableau 1. 1. *Les différents types de commutation*

1.2.3. Description fonctionnelle des PABX

1.2.3.1. Architecture générale

On considère les PABX comme offrant quatre types de fonctions :

- **les fonctions de raccordement**, qui consiste à adapter les signaux circulant entre les entités consécutives des installations téléphoniques privés, aux contraintes des lignes de transport qui les relie entre elles ;
- **les fonctions de commutation,** qui consiste à aiguiller en transparence les signaux, acheminés par le système (signaux vocaux, image, etc.) en fonction des demandes des utilisateurs : émetteurs et destinataires ;

- **les fonctions de signalisation et d'adressage**, qui consiste à élaborer et à échanger les informations nécessaires à l'invocation et à la fourniture des services, entre le système et les utilisateurs ;
- **les fonctions de commande**, qui incluent d'une part la commande des fonctions de commutation à partir du traitement des signalisations échangés et, d'autre part, les opérations de gestion, d'administration, de maintenance et d'exploitation.

1.2.3.2. Modes de réalisation des fonctions PABX

Ces fonctions ne sont en général pas localisées dans des sous-ensembles spécifiques à chacune d'elles comme cela était le cas dans les PABX à l'origine. La réalisation des PABX s'appuie en effet de plus en plus sur des architectures informatiques, définies par le concept client /serveur. Le client est l'usager interagissant avec le terminal et le serveur est porté par le central « plate forme PABX », pour indiquer qu'il est constitué d'une plate forme matériel et logiciel sur laquelle sont portés les applicatifs permettant d'offrir les services du PABX.
Le central peut être client vis-à-vis de serveurs informatiques fournissant des services auxiliaires de gestion ou d'administration. Le tableau 3.1 montre comment ces fonctions se répartissent entre les trois entités : terminal, central et le serveur.[17]

Support	Fonctions			
	Raccordement	**Signalisation**	**Commutation**	**Commande**
Terminal	Extrémité terminale	Client utilisateur	Repartie (mode paquet)	Fonctions locales du terminal
Plate forme PABX	Extrémité centrale	Serveur PABX Client CTI	Centralisée et répartie (circuit et paquet)	Fonctions PABX
Serveur informatique	Extrémité informatique	Serveur CTI Client CTI	Répartie (mode paquet)	Fonctions applicatives

Tableau 1. 2. *Répartition des fonctions du PABX*

Les logiciels applicatifs réalisant les fonctions de commande peuvent être portés par la plate forme PABX ou par des plates formes indépendantes de type informatique. La séparation entre ces deux catégories a été mouvantes et évolutive. Au fur et à mesure que les plates formes informatiques évoluaient et étaient en mesure de supporter des

applications de type téléphonique, ce qui a permis de rapprocher le mode de réalisation des PABX de celui des ordinateurs. Le tableau 3.2 présente une suite de cinq modes de réalisations échelonnés dans l'évolution historique des technologies des PABX.

Les paramètres considérés dans la description de cette évolution sont les suivants :

- les systèmes d'exploitation qui supportent les applications, qui vont des solutions spécifiques d'autrefois, aux solutions tout informatiques ;
- la technologie de commutation de la voix qui va de la commutation de circuit à la commutation de paquets ;
- les services de base offerts par les PABX qui vont de la communication vocale aux services supplémentaires (fonctionnalités PABX) ;
- les applications associées qui se développent sur des serveurs informatiques liés au PABX.

Mode de réalisation	**Système d'exploitation**	**Commutation de la voix**	**Services de bases**	**Applications associées**
1. plate-forme PABX traditionnelle	Spécifique temps réel	Circuit centralisé	Communication vocale et CP à P	Serveurs reliés en mode terminal (CM)
2. plate-forme PABX	Mixte base UNIX (chorus,OSE..) (1)	Circuit réparti	Enrichissement R,Cm ,A,Mob	Liens CTI avec serveurs (CM,A,Mob)
3. plate-forme PABX évoluée	Mixte informatique (windows NT …)	Circuit réparti	Enrichissement R,Cm ,A,Mob	Liens CTI avec serveurs (CM, A, Mob)
4.plate-forme PC avec cartes	Windows NT, linux …	Circuit sur bus	CPàP, R, CM, Mob Sur plate forme	CM et A sur serveurs associé
5.plate-forme PC dédiée	Windows NT, linux …	Paquet	CPàP, R, CM, Mob Sur plate forme	CM et A sur serveurs associé

(1) windows NT, chorus, OSE et linux sont des systèmes d'exploitations mixtes supportant à la fois des applications temps réel et informatique A : les services Axiliaires(liés à la gestion, l'exploitation, installations…) CM : services de communications médiatisées (fonction d'opérateur, centre d'appel, messagerie…) CP à P : communication de personne à personne Mob : les services de mobilité (terminaux portables) R : services réseau (interconnexions)

Tableau 1. 3. *Modes de Réalisation des PABX*

Le mode de réalisation n°1 n'est plus utilisé que pour les très petits PABX (moins de six à huit usagers). Il repose sur plate forme entièrement spécifique sans liens avec l'informatique.

Pour les structures 2 et 3 les plates-formes sont capables de recevoir simultanément des applications temps réel pour gérer la téléphonie et des applications de type informatique supportées par un système d'exploitation différent, soit orienté comme UNIX, soit purement informatique comme NT. La tendance de l'offre est de séparer les plates-formes en plusieurs serveurs reliés entre eux par un réseau local, chaque serveur étant optimisé pour certaines fonctions : temps réel ou informatique.

Le mode réalisation n° 4, n'a pas connu un grand succès, compte tenu de la mauvaise adaptation de la plate forme PC aux contraintes d'exploitation et de permanence de service des PABX.

Le mode de réalisation n°5 repose sur des éléments empruntés au monde des ordinateurs mais rassemblés en une plate-forme spécifique dédiée à la téléphonie. Ce mode utilise les modes d'interconnexion mis au point pour le PC : protocole de réseaux locaux et le protocole IP. Ce mode représente la génération qui donnera naissance aux PABX IP, dont l'architecture est tout IP basée sur des standards tels que le H.323 et SIP .Ce mode va permettre aux installations téléphoniques privées de passer à la ToIP.[17]

1.2.4. Fonctionnalités des PABX

1.2.4.1. Fonctionnalités standard

- Annuaire central

- Conférence (interne/externe) : permet d'établir une communication entre plusieurs utilisateurs.
- double appel : vous êtes en communication avec une ligne , et vous avez besoin d ' un renseignement sur une question de votre interlocuteur, vous mettez alors votre premier correspondant en attente, et reprenez une autre ligne , vous avez alors deux appels sur votre poste.
- groupe de postes :
    - groupement d'appel : permet de faire sonner plusieurs postes :
        - en mode parallèle : tous les postes sonnent en même temps
        - en mode cyclique : les postes sonnent les un après les autres dans un ordres préalablement définit, et identique à chaque appel
        - séquentiel : les postes sonnent les uns après les autres dans un ordre tenant compte de l'occupation des postes.
    - groupement d'interception: permet par la composition d'un code, ou une touche de prendre une communication arrivant sur un autre poste du groupe.
    - groupement de diffusion : permet par l'intermédiaire de l'équipement du postes numérique de diffuser une annonce dans un service de l'entreprise.

- Musique d'attente : déclenchement d'une musique au niveau de l'utilisateur mis en attente.
- parcage : permet de stocker un usager externe sur un dispositif de musique en vue de la libération de son correspondant intérieur et de la reprise éventuelle de cet usager par celui -ci
- Prise de ligne (automatique)
- Rappel automatique en cas d'occupation et de non-réponse
- Renvoi de nuit/renvoi de jour
- Renvoi de poste
- Renvoi temporisé sur non-réponse
- Retour d'appel
- Transfert de communication (interne/externe) : transfert : vous recevez une communication qui ne vous est pas destinée, vous pouvez la transférer, par une manoeuvre, sur un autre poste intérieur ou extérieur.

1.2.4.2. Fonctionnalités particulières

- Annuaire interne et externe (tous les numéros de postes sont répertoriés dans l'annuaire interne du système, avec le nom qui correspond.
- Entrée en tiers : les postes autorisés peuvent intervenir directement dans la communication en cours.
- Interception : un abonné peut intercepter sur son propre poste les appelants destinés à un groupe ou à certains collègues.
- Liste des appelants : sur les postes numériques à afficheur, les appels sont inscrits avec la date et l'heure dans une liste à partir de laquelle un rappel peut être réactivé. [7]

1.2.5. Les stratégies de migration vers la ToIP

L'élément central de la téléphonie traditionnelle en entreprise est le PABX, ou central téléphonique privé. Trois solutions sont envisageables pour permettre le passage à la ToIP.

Il est possible d'adapter un PABX par l'ajout d'une passerelle IP : le commutateur peut donc être connecté sur le réseau local (LAN) haut débit, ce qui ouvre le système informatique de l'entreprise aux applications de téléphonie. C'est le choix d'une évolution progressive.

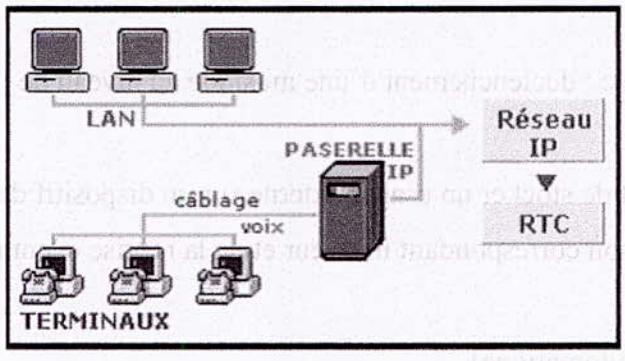


Figure 1.3. *PABX et passerelle IP*

Il est également possible d'externaliser les fonctions de téléphonie vers un IP Centrex, service fourni par un opérateur ou autre fournisseur de solution de VoIP, qui gère le service de bout en bout. L'externalisation du service est bien adaptée aux sites de petites tailles (PME et TPE) puisque, hormis l'installation de téléphones IP, aucun investissement n'est nécessaire.

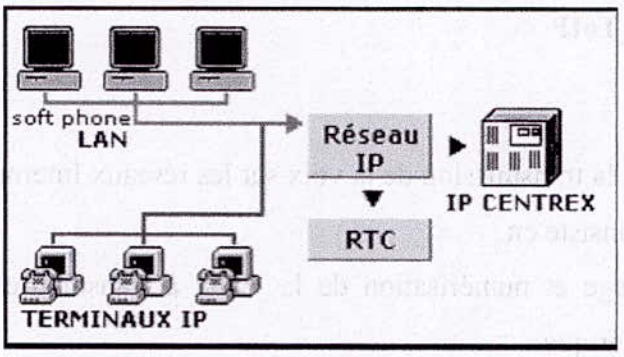


Figure 1. 4. *IP CENTREX*

Enfin est possible de remplacer le PABX classique par un IP PBX (serveur d'appel ToIP et serveur d'application) impliquant le remplacement des terminaux téléphoniques analogiques classiques par des téléphones IP. C'est le choix de la rupture, impliquant un renouvellement complet des infrastructures. (C'est le cas traité dans le chapitre 3)

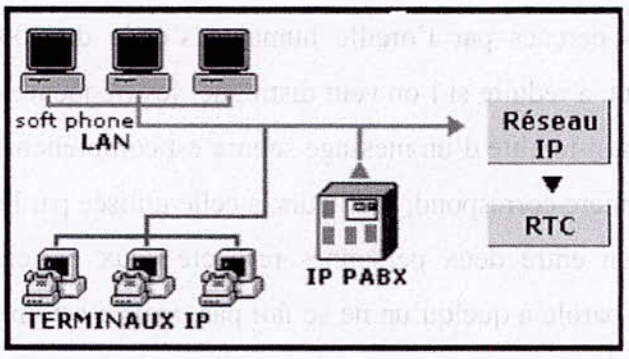


Figure 1. 5. *IP PABX (IPBX) et équipements ToIP*

Définition de l'IPBX

L'IPBX, ou PABX-IP, est un autocommutateur compatible avec la ToIP. Il permet, comme un commutateur téléphonique standard, d'établir une communication téléphonique entre deux abonnés distants, remplissant ainsi le rôle des anciennes opératrices. A l'intérieur d'une entreprise, l'IPBX définit le routage des paquets pour que la communication parvienne au bon poste de l'entreprise. Un PABX-IP peut être soit un autocommutateur auquel l'entreprise ajoute une carte d'extension IP, soit une machine nativement IP. Un autocommutateur IP sert de serveur de messagerie, capable de stocker l'historique des communications ou éventuellement des messages.

1.3. La téléphonie IP _ToIP

1.3.1. Principe

La téléphonie sur IP est la transmission de la voix sur les réseaux Internet. Le principe de base de cette transmission consiste en :

1. Echantillonnage et numérisation de la voix à transmettre par un convertisseur analogique numérique.
2. Compression et encodage du signal numérique obtenu grâce à des algorithmes de compression spécifiques.
3. Découpage du signal en paquets.
4. Transmission des paquets à travers le réseau Internet à la réception, les paquets sont rassemblés, le signal de données ainsi obtenu est, décomprimé puis convertis en signal analogique sonore. [15]

1.3.2. La voix

Le système vocal est complexe et basé sur des ondes sonores de fréquences différentes. Le spectre des fréquences perçues par l'oreille humaine s'étale de 100 Hz à 20kHz. Cette fourchette est, cependant, à réduire si l'on veut distinguer les fréquences utiles des fréquences audibles. En effet, la quasi-totalité d'un message sonore est compréhensible dans la fourchette 300-3400 Hz. Cette dernière correspond, d'ailleurs, à celle utilisée par le téléphone standard.

Une conversation entre deux personnes respecte deux principes : intelligibilité et interactivité. Couper la parole à quelqu'un ne se fait pas, mais c'est un gage d'interactivité et de dialogue. En terme de transmission numérique, cela se traduit par le terme duplex. Une conversation full duplex assure cette interactivité car chaque locuteur peut parler en même temps, ce qui arrive quand deux personnes parlent de leur propre expérience sans s'écouter... Un mode half duplex induit une conversation unidirectionnelle.

Cette interactivité implique des notions de délais dans le transport de la voix (avec le téléphone, par exemple). Les mesures effectuées montrent qu'un temps de transit inférieur à 150 ms garantit un dialogue actif. Jusqu'à 400 ms (limite supérieure) le dialogue reste tout de même assez réactif. Au-delà de cette limite le contradicteur aura l'impression de parler dans le vide.

1.3.3. Les difficultés et limites associées à la VOIP

- Délai : temps de transmission d'un paquet (doit rester inférieur à 400ms pour respecter les contraintes d'une conversation interactive).
- Perte : disparition de paquets au cours de la communication (fait partie de la transmission IP mais doit être soit réduite, soit inhibée).
- Gigue : variation de délai (nécessite un buffer de resynchronisation en bout de chemin).

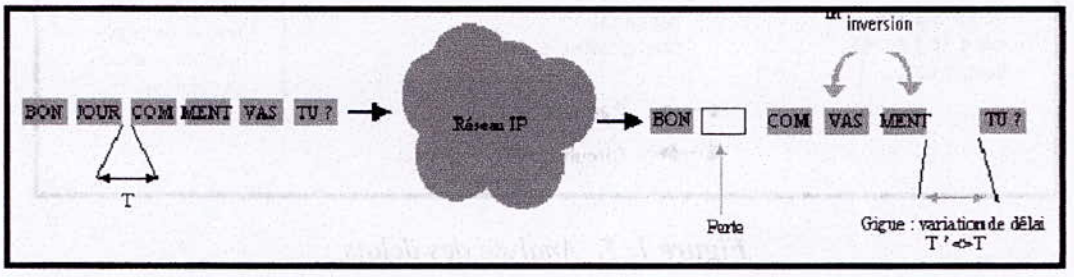


Figure 1. 6. *Difficultés associées à la VoIP*

1.3.3.1. Analyse des délais

Quantifier le délai de transmission sur le réseau de manière fiable est quasi impossible, car il y a trop d'inconnues : Table de routage, congestion, pannes, files d'attente… Cependant sur le chemin que prendrait une transmission de voix, on peut détailler certains délais inhérents au réseau :

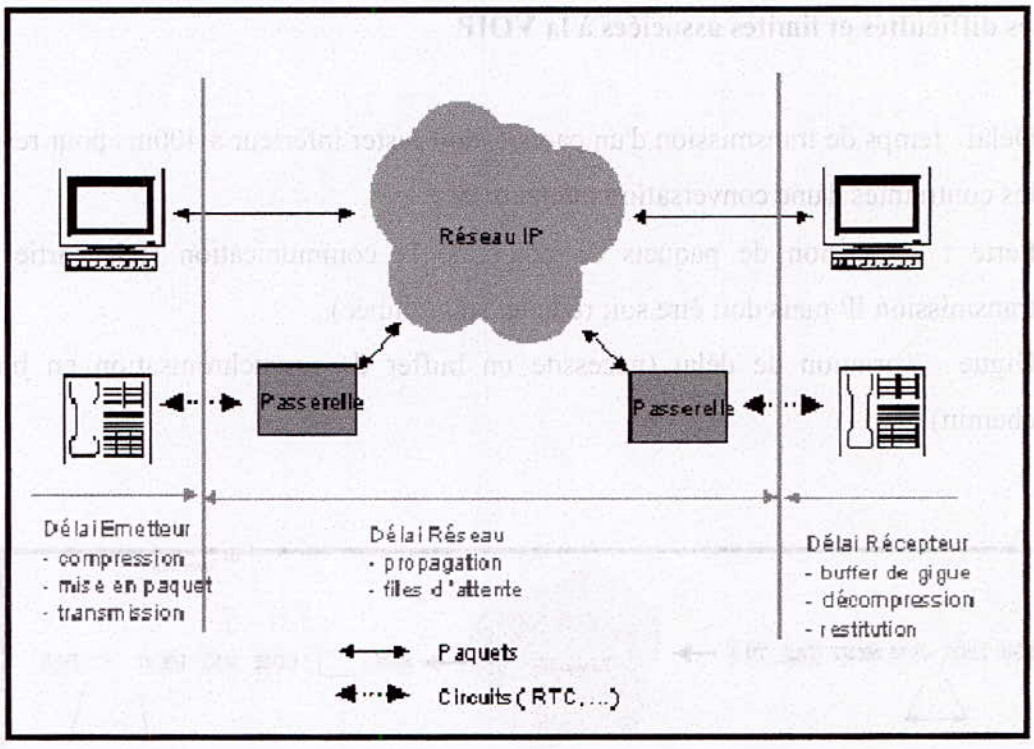


Figure 1. 7. *Analyse des délais*

1.3.3.1.1. Délais de l'émetteur

Numérisation et codage : temps mis par une carte son ou une passerelle pour numériser et coder un signal initialement analogique.

Compression qui se décompose en trois parties :

- Délai de trame : contrairement à la numérisation d'un signal qui se fait en continue, la compression porte sur une certaine longueur de données. Attendre ces informations induit un temps non nul de traitement
- Délai d'encodage : la compression par synthèse s'appuyant sur la prédiction, ce délai est nécessaire à l'encodeur pour savoir, pendant qu'il est en fonctionnement comment évolue le signal.
- Délai de traitement : temps mis par l'algorithme pour compresser une trame. Il dépend du processeur et de l'algorithme utilisé.

Mise en paquets : intervalle de temps pendant lequel l'application constitue un paquet (création de l'en-tête, remplissage des données).

- **Transmission** : ce temps dépend de la configuration dans laquelle on se trouve. A savoir soit on est relié par modem soit par accès direct sur un LAN-WAN. [1][2]

1.3.3.1.2. Délais réseau :

-Propagation : sur un réseau filaire, la vitesse de propagation est de 200000 km/s, cela induit un temps de propagation non nul.

- Commutation et files d'attente : suivant la nature du réseau différents temps peuvent être indexés.

1.3.3.1.3 Délais du récepteur (ce sont les mêmes que pour l'émetteur) :

- Réception.
- Buffer de gigue : cette mémoire tampon permet de resynchroniser les paquets arrivant avec des délais variables. Elle sert donc à compenser les décalages et remettre en ordre les paquets.
- Dépaquetisation.
- Décompression.
- Décodage et conversion numérique analogique.

Jusqu'à présent les mesures effectuées avec une solution téléphone à téléphone (via IP), sur un réseau bien géré et surdimensionné (en bande passante), montrent un délai total de 200 ms.

1.3.3.2. Analyse des pertes

La perte d'un paquet occasionne un manque d'information lors de la réception du signal audio. Suivant le nombre de paquets perdus, la qualité sonore en bout de ligne peut s'en ressentir. Dans la philosophie IP, cette perte de paquet fait partie intégrante du concept. En effet les routeurs sont obligés (avec l'algorithme Random Early Detection) de détruire des paquets afin d'anticiper d'éventuelles congestions. [3]

Il existe principalement quatre causes de perte de paquet :

- Durée de vie épuisée (TTL = 0).
- Retard à la réception supérieur au buffer de gigue.
- Destruction par un module congestionné.
- Invalidité du paquet due à des défauts de transmission.

1.3.3. Scénarios de téléphonie IP

Selon le type de terminal utilisé, on distingue trois scénarios possibles de téléphonie sur IP.

1.3.3.1. Téléphonie de PC à PC

Dans ce scénario, les deux correspondants utilisent un PC rattaché au réseau Internet par l'intermédiaire d'un fournisseur d'accès Internet. Cette technique nécessite des participants à la communication d'avoir un PC muni d'un modem, d'une carte réseau, d'un microphone, d'un haut-parleur et d'un logiciel de téléphonie IP compatible de chaque côté. La voix est comprimée et décomprimée par un logiciel de compression. Ce mode de fonctionnement nécessitait auparavant que les correspondants se fixent un rendez-vous préalable sur Internet ou soient connectés en permanence. De nos jours, des protocoles de signalisations ont été élaborés pour pallier à ce genre de problème.

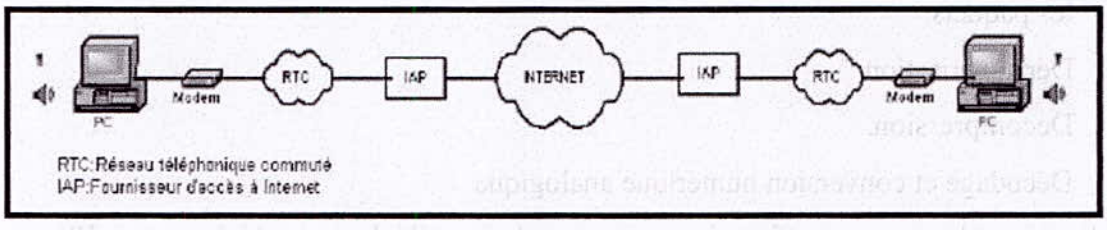


Figure 1. 8. Téléphonie de PC à PC

1.3.3.2. Téléphonie entre PC et poste téléphonique et vis-versa

Dans ce scénario, l'un des correspondants utilise un PC rattaché au réseau Internet par un fournisseur d'accès Internet, l'autre correspondant utilise un téléphone rattaché au réseau téléphonique commuté. Une passerelle est nécessaire ente les deux réseaux pour rendre possible cette technique et faire la conversion entre réseaux (dans ce cas elle fait la conversion Internet-RTC et vis versa).

Elle se charge également de l'appel du correspondant et de l'ensemble de la signalisation relative à la communication téléphonique du côté du correspondant demandé. Du côté PC, une signalisation d'appels est nécessaire pour établir une communication et négocier les paramètres de communication multimédia.

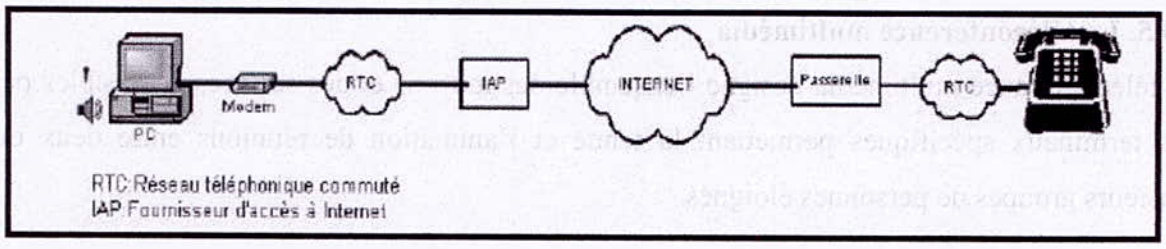


Figure 1. 9. *Téléphonie entre PC et poste téléphonique*

1.3.3.3. Téléphonie entre postes téléphoniques

Dans ce cas les deux correspondants utilisent un téléphone conventionnel via le réseau téléphonique commuté. Une passerelle est utilisée de chaque côté entre ce réseau et le réseau Internet pour convertir la voix IP en voix et vis-versa. Le réseau Internet est utilisé pour la connexion longue distance.[11]

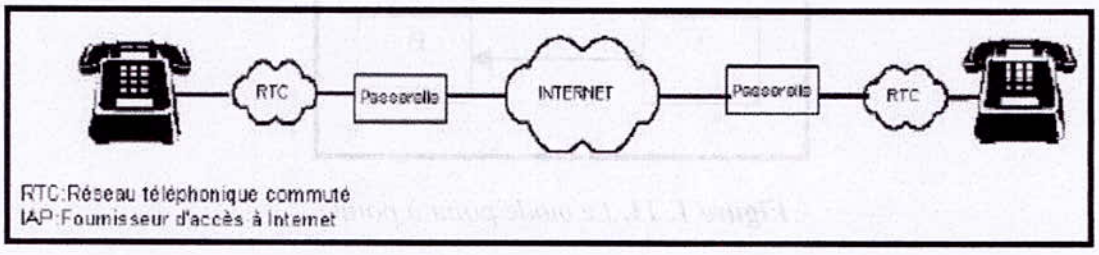


Figure 1. 10. *Téléphonie entre postes téléphoniques*

1.3.4. Qualité de service et capacité – QoS

La qualité de service est un facteur déterminant dans la téléphonie et, à ce titre, occupe le centre du débat sur la téléphonie IP. La qualité peut se définir sur plusieurs plans : fiabilité, débit, sécurité.

Néanmoins, c'est parce que la qualité de transmission de la téléphonie caractérisant actuellement l'Internet public est perçue comme médiocre que la téléphonie sur Internet est rarement considérée comme un service de qualité. En règle générale, pour améliorer la qualité, on peut soit mettre en application des critères de qualité de service, soit accroître la capacité disponible. [11]

1.3.5. La téléconférence multimédia

La téléconférence multimédia désigne l'ensemble des moyens et des services accessibles par des terminaux spécifiques permettant la tenue et l'animation de réunions entre deux ou plusieurs groupes de personnes éloignés.

Toutes les applications de téléconférence utilisent un ou plusieurs médias : la voix, l'image fixe ou animée et/ou des données en temps réel.

1.3.6. Les modes de communication

1.3.6.1. Le mode point à point

Le mode point à point correspond à une communication entre deux participants.
C'est le mode de fonctionnement le plus simple à mettre en œuvre.

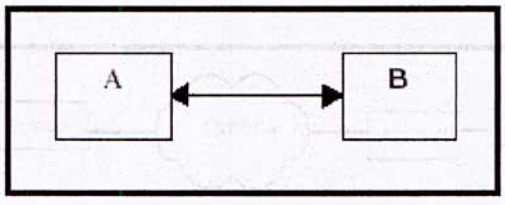


Figure 1. 11. *Le mode point à point*

1.3.6.2. Le mode multipoint pleinement interconnectés

Ce mode correspond au cas où une téléconférence met en relation plus de deux participants simultanément. Ces participants sont interconnectés en multicast.

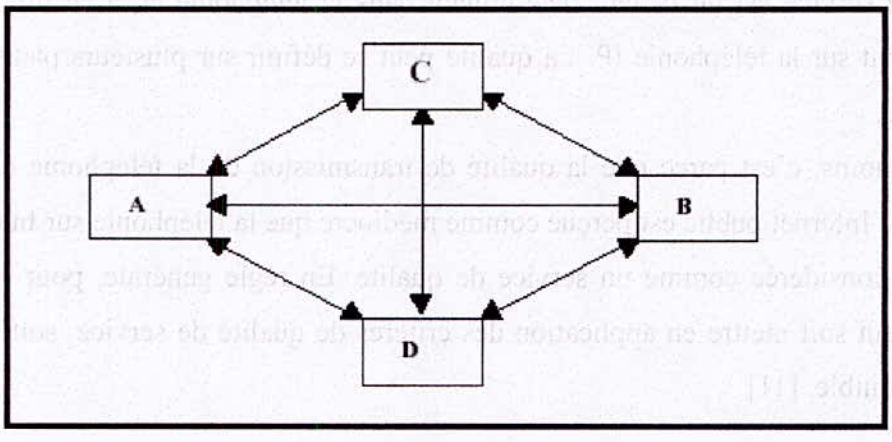


Figure 1. 12. *Le mode multipoint pleinement interconnecté*

1.3.6.3. Le mode multipoint via un MCU

Dans ce mode plusieurs participants sont connectés via une unité de contrôle MCU (Multipoint Control Unit). Celle-ci gère les procédures d'appels entrants et sortants (entre le MCU et les terminaux). [11]

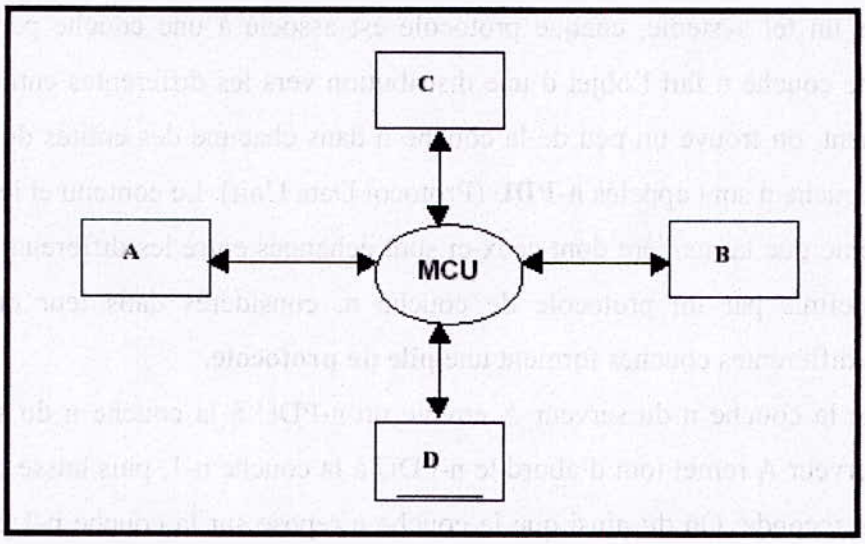


Figure 1. 13. *Le mode multipoints via un MCU*

1.3.7. Les protocoles

Le transfert de données sur Internet s'effectue par paquets de données. Chaque document, qu'il s'agisse de texte, image ou voix, est numérisé puis réparti en paquets. Chacun de ces paquets est alors envoyé sur Internet indépendamment des autres et essaie de prendre le chemin le plus rapide pour parvenir à sa destination. Ceci est réalisé en fonction de l'encombrement d'une partie ou de l'autre du réseau au moment où le paquet est expédié. La segmentation de l'information permet une plus grande flexibilité dans l'utilisation des ressources puisque la communication ne monopolise pas une ligne donnée.

1.3.7.1. Un protocole de réseau

Dans un protocole réseau, les entités échangeant des messages et les différentes actions sont le fait de composant matériels ou logiciels d'un appareil électronique (ordinateur, routeur ou tout autre appareil doté de capacité de mise en réseau).Un protocole définit le format et l'ordre des messages échangés entre deux entités interlocutrice ou plus, ainsi que les actions générées au moment de la transmission ou réception d'un message ou de tout autre

avènement. Toute activité au sein de l'Internet impliquant plusieurs entités interlocutrice distantes est régie par un protocole.

1.3.7.2. Les couches de protocoles et leurs modèles de services

Afin de réduire la complexité de la conception d'un réseau, les architectes de réseau ont choisi d'organiser les protocoles, ainsi que les matériels et logiciels qui en dépendent, en différentes **couches.** Dans un tel système, chaque protocole est associé à une couche particulière. Le protocole d'une couche n fait l'objet d'une distribution vers les différentes entités du réseau qui le constituent, on trouve un peu de la couche n dans chacune des entités du réseau . ces fragments de couche n sont appelés **n-PDU** (Protocol Data Unit). Le contenu et le format d'un n-PDU, de même que la manière dont ceux-ci sont échangés entre les différents éléments du réseau, sont définis par un protocole de couche n. considérés dans leur ensemble, les protocoles des différentes couches forment une **pile de protocole**.

Lorsque la couche n du serveur A envoie un n-PDU à la couche n du serveur B, la couche n du serveur A remet tout d'abord le n-PDU à la couche n-1, puis laisse celle ci livrer le n-PDU à la seconde. On dit ainsi que la couche n repose sur la couche n-1 pour envoyer son n-PDU à la couche n du serveur B. dans ce contexte, un concept à saisir absolument est celui de **modèle de service**. On considère en effet que la couche n-1 procure des **services** à la couche n. cette dernière peut par exemple garantir le bon acheminement du n-PDU sans erreurs et dans un laps de temps donné ou se contenter d'assurer que celui ci arrive bien à destination, sans prendre d'engagement concernant les éventuelles erreurs. [6][13]

Pile de protocole Internet

La pile de protocole Internet est constituée de cinq couches successives :

- la couche physique
- la couche liaison
- la couche réseau
- la couche transport
- la couche application.

Plutôt que d'avoir recours au terme n PDU pour évoquer les unités de données de protocoles de chacune de ces couches, nous utiliserons des noms qui sont à savoir respectivement :

- 1-PDU
- trame

- data gramme
- segment
- message.

La pile de protocole Internet et les noms correspondants sont représentés à la figure suivante :

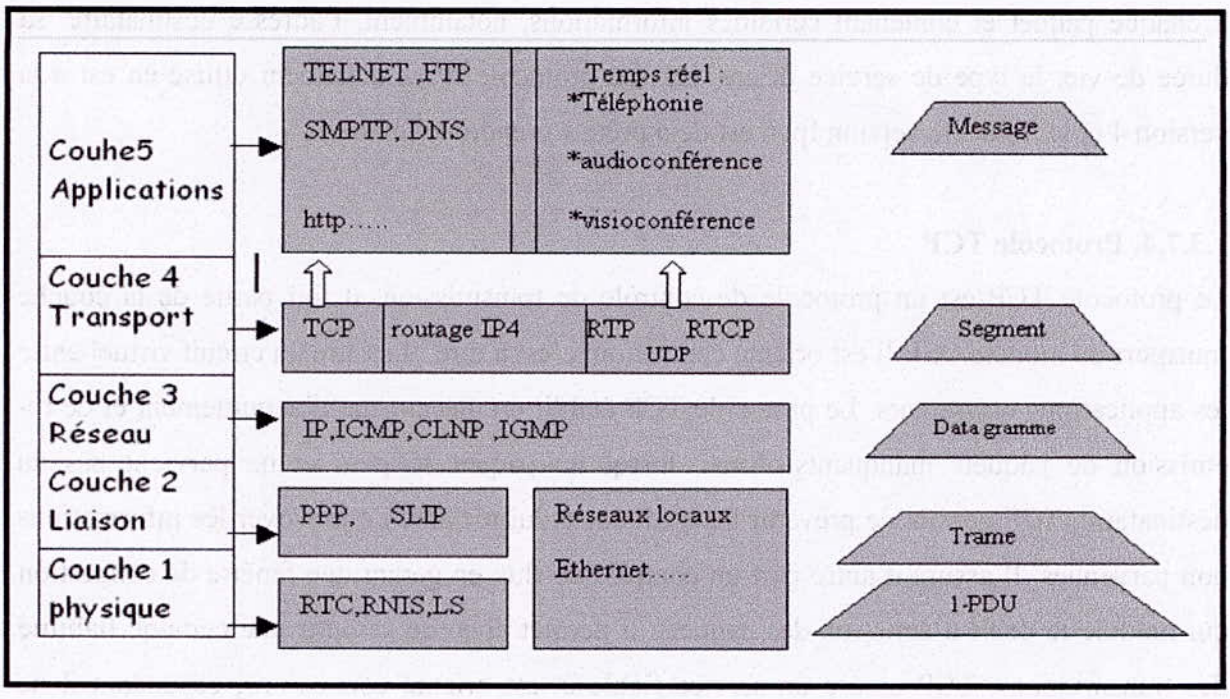


Figure 1. 14. *La pile de protocole Internet et les unités de données de protocole*

FTP : File Transfer Protocol

TCP : Transport Control Protocol

SMPTP : Simple Mail Transmission Protocol

RTP : Real Time Transport Protocol

DNS : Domain Name System

RTCP : Real Time Transport Control Protocol

PPP : Point to Point Protocol

SLIP : Serial Line Internet Protocol

IGPM: Internet Group Management protocol

LS : algorithme de routage par état de lien

1.3.7.3. Protocole IP

Le protocole IP est au centre du fonctionnement de l'Internet. Il fait partie de la couche Internet de la suite de protocoles TCP/IP. Il assure sans connexion un service non fiable de

délivrance de paquets IP. Le service est non fiable car il n'existe aucune garantie pour que les paquets IP arrivent à destination. Certains paquets peuvent être perdus, dupliqués ou remis en désordre. On parle de remise au mieux. Le protocole IP permet aux paquets de se déplacer sur le réseau Internet, indépendamment les uns des autres, sans liaison dédiée. Chacun d'entre eux, envoyé sur le réseau, se voit attribuer une adresse IP. Cette dernière est un en-tête accolé à chaque paquet et contenant certaines informations, notamment, l'adresse destinataire, sa durée de vie, le type de service désiré, etc. Le protocole IP actuellement utilisé en est à la version 4 et la nouvelle version Ipv6 est déjà prête à prendre le relais.

1.3.7.4. Protocole TCP

Le protocole TCP est un protocole de contrôle de transmission, il fait partie de la couche transport du modèle OSI. Il est orienté connexion, c'est à dire, il assure un circuit virtuel entre les applications utilisateurs. Le protocole TCP établit un mécanisme d'acquittement et de re-émission de paquets manquants. Ainsi, lorsqu'un paquet se perd et ne parvient pas au destinataire, TCP permet de prévenir l'expéditeur et lui réclame de renvoyer les informations non parvenues. Il assure d'autre part un contrôle de flux en gérant une fenêtre de congestion qui module le débit d'émission des paquets. Il permet donc de garantir une certaine fiabilité des transmissions. TCP assure un service fiable et est orienté connexion , cependant il ne convient pas à des applications temps réel à cause des longs délais engendrés par le mécanisme d'acquittement et de retransmission .[10]

1.3.7.5. Protocole UDP

Le protocole de datagramme utilisateur (UDP) est le protocole de transport sans confirmation. UDP est un protocole simple qui permet aux applications d'échanger des datagrammes sans accusé de réception ni remise garantie. Le traitement des erreurs et la retransmission doivent être effectués par d'autres protocoles. UDP n'utilise ni fenêtrage, ni accusés de réception, il ne reséquence pas les messages, et ne met en place aucun contrôle de flux. Par conséquent, la fiabilité doit être assurée par les protocoles de couche application. Les messages UDP peuvent être perdus, dupliqués, remis hors séquence ou arriver trop tôt pour être traiter lors de leurs réception. UDP est un protocole particulièrement simple conçu pour des applications qui n'ont pas à assembler des séquences de segments. Son avantage est un temps d'exécution court qui permet de tenir compte des contraintes de temps réel ou de limitation d'espace mémoire sur un processeur, contraintes qui ne permettent pas l'implémentation de protocoles beaucoup plus lourds comme TCP. Dans des applications temps-réel, UDP est le plus

approprié, cependant il présente des faiblesses dues au manque de fiabilité. Des protocoles de transport et de contrôle temps-réel sont utilisés au dessus du protocole UDP pour remédier à ses faiblesses et assurer sa fiabilité. Ces protocoles sont RTP et RTCP et sont détaillés dans le paragraphe suivant. [10]

1.3.7.6. Les protocoles de transport temps réel

1.3.7.6.1. Le protocole RTP

Le groupe de l'IETF a développé en 1993 le protocole de transport en temps réel (RTP, RFC 1889) dont le but est de transmettre sur Internet des données qui ont des propriétés temps réel (audio, vidéo …). C'est un protocole de la couche application du modèle OSI et utilise les protocoles de transport TCP ou UDP, mais, généralement, il utilise UDP qui est mieux approprié à ce genre de transmission.

La figure 1.15 représente l'architecture du protocole RTP.

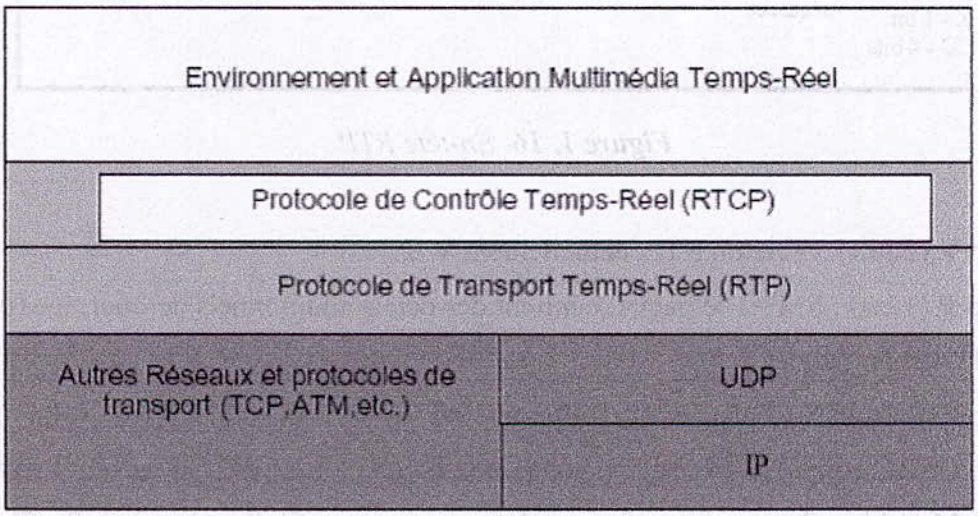


Figure 1. 15. *Architecture RTP*

Le rôle principal de RTP consiste à mettre en œuvre des numéros de séquences de paquets IP et des mécanismes d'horodatages (timestamp) pour permettre de reconstituer les informations de voix ou vidéo. Plus généralement, RTP permet :

- D'identifier le type de l'information transportée;
- D'ajouter des indicateurs de temps (horodater) et des numéros de séquence à l'information transportée;

➢ De contrôler l'arrivée à destination des paquets.

L'en-tête RTP (figure 1.16) indique le type, la source, et les caractéristiques de contrôle du temps des données encodées. Ce standard inclut en effet un horodateur (Timestamp) des paquets permettant au destinataire d'utiliser la numérotation des séquences pour reconstituer l'ordre original des paquets et de détecter les paquets perdus. [9]

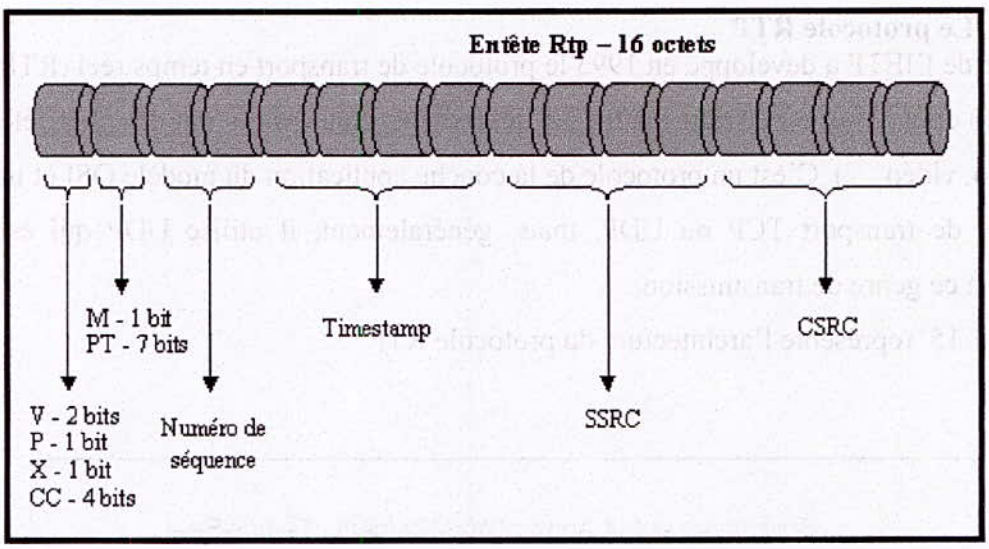


Figure 1. 16. *En-tête RTP*

· Le champ **V** (2 bits) : Version RTP, actuellement V=2.

· Le champ **P** (1 bit) : Si P=1 le paquet contient des octets additionnels de bourrage (padding) pour finir le dernier paquet.

· Le champ Extension **X** (1 bit) : Si X=1, l'en-tête est suivie d'un paquet d'extension.

· Le champ **CSRC** count **CC** (4 bits) : Il contient le nombre de CSRC qui suivent l'en-tête.

· Le champ **M** (1 bit) : Son interprétation est définie par un profil d'application (Profile).

· Le champ **Données** type **PT** (7 bits) : Ce champ identifie le type de données (audio, vidéo, image, texte…).

· Le champ **numéro de séquence** (16 bits) : Sa valeur initiale est aléatoire et il s'incrémente de 1 à chaque paquet envoyé, il peut servir à détecter des paquets perdus.

· Le champ **timestamp** (32 bits) : Ce champ reflète l'instant où le premier octet du paquet RTP a été échantillonné.

· Le champ **SSRC** (32 bits) : Ce champ identifie de manière unique la source de synchronisation. Sa valeur est choisie de manière aléatoire par l'application.

· Le champ **CSRC** (32 bits) : Ce champ identifie les sources participantes. RTP supporte différents types de codage et de compression des données. Beaucoup de formats standardisés sont acceptés (GSM, G.723.1, G.729 pour l'audio et MPEG,
H261 pour la vidéo).

1.3.7.6.2 Le protocole RTCP

RTCP (Real Time Control Protocol, RFC 1889) est un protocole de contrôle utilisé, conjointement avec RTP pour contrôler les flux de données et la gestion de la bande passante.
RTCP permet de contrôler le flux RTP, et de véhiculer périodiquement des informations de bout en bout pour renseigner sur la qualité de service de la session de chaque participant à la session. Des quantités telles que le délai, la gigue, les paquets reçus et perdus sont très importants pour évaluer la qualité de service de toute transmission et réception temps réelles.
C'est le protocole sous-jacent (UDP par exemple) qui permet grâce à des numéros de ports différents et consécutifs (port pair pour RTP et port impair immédiatement supérieure pour RTCP) le multiplexage des paquets de données RTP et des paquets de contrôle RTCP. Le protocole RTCP remplit trois fonctions :

· L'information sur la qualité de service : RTCP fournit, en rétroaction des informations sur la qualité de réception des données transmises dans les paquets
RTP. Cette information est utilisée par la source émettrice pour adapter le type de codage au niveau des ressources disponibles.

· L'identification permanente : RTCP transporte un identificateur original de la source RTP c'est à dire la provenance du flux, appelé CNAME (Canonical name).
Cet identificateur permet une identification permanente de chacun des flux multimédia entrants.

· Le calibrage de la fréquence d'émission : La réception des feed-back et la connaissance du nom permanent servent à ajuster la fréquence d'envoi des paquets à la bande passante mise à la disponibilité de l'utilisateur situé à l'autre extrémité.

Chaque paquet RTCP contient un rapport émetteur ou récepteur suivi d'une description de la source (source description). [9]

Il existe cinq types de paquets de contrôle. Chaque paquet commence par un en-tête fixe suivi d'éléments structurés qui peuvent être de longueur variable selon le type de paquet.

Messages RTCP

RR	Rapport Émetteur
SR	Rapport Récepteur
SDES	Description de Source
BYE	Fin de session
APP	Nouveau champ

Figure 1. 17. *Les paquets de contrôle RTCP*

RR : contient le rapport de la qualité de la livraison des données des participants passifs (récepteur) incluant le nombre de paquets reçus, le nombre de paquets perdus, la gique et l'horodatage qui permet le calcul du délai total de transmission entre les deux parties.

SR : contient le rapport de la qualité de livraison des données des participants actifs (émetteurs). Il contient les champs du RR, et des informations sur l'émetteur, la synchronisation (pour synchroniser deux sources de données), un compteur cumulatif de paquets et le nombre d'octets envoyés.

SDES : contient l'information concernant les émetteurs comme le nom canonique (CNAME), et le nom d'usager (NAME), le numéro de téléphone (PHONE) et d'autres informations concernant les participants.

BYE : indique la fin de la participation d'une des parties.

APP : Dans le cas des applications spécifiques, les données peuvent être transmises dans des paquets spécifiques de type APP.

La **figure 1.15** détaille l'en-tête ce message commun à tous les paquets RTCP :

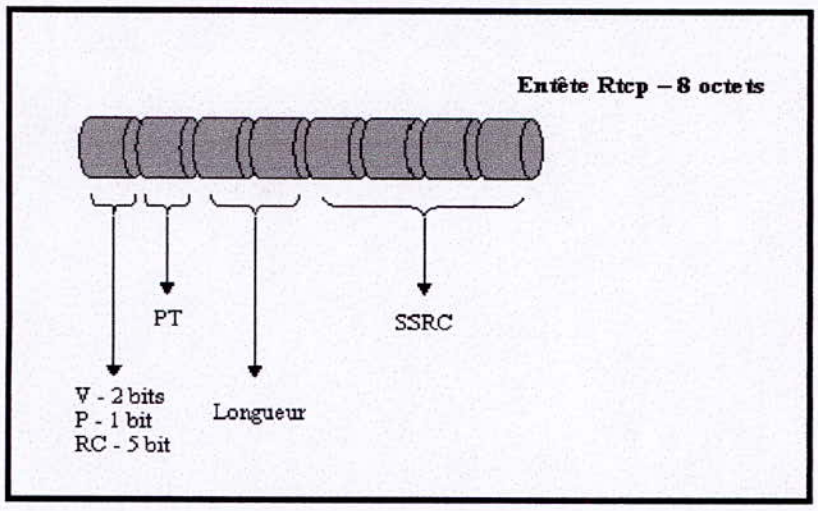


Figure 1. 18. *En-tête RTCP*

Avec :

· Le champ **V** (2 bits) : Indique la version de RTP, actuellement V=2.

· Le champ **P** (1 bit) : Si P=1 le paquet contient des octets additionnels de bourrage (padding) pour finir le dernier paquet.

· Le champ **RC** (5 bits) : Il contient le nombre de rapports contenus dans le paquet (un paquet pour chaque source)

· Le champ **PT** (8 bits) : Il donne le type de rapport du paquet (PT=SR=200, PT=RR=201, PT=SDES=202, PT=BYE=203, PT=APP=204).

· Le champ **Longueur** (16 bits) : Il indique la longueur du paquet.

1.4. Conclusion

Dans ce chapitre trois points importants sont traités :

- Les réseaux téléphoniques publics et privés
- L'évolution des PABX suivant l'évolution de l'informatique
- La ToIP et les différents protocoles utilisés par celle-ci

Avec l'évolution de l'informatique et de l'électronique, les installations téléphoniques privées sont passées vers une nouvelle technologie qu'est la VoIP.

L'**IP PBX** (serveur d'appel ToIP et serveur d'application) est un élément indispensable pour le passage vers une infrastructure tout IP.

Chapitre 2

Le Protocole SIP
« Session Initiation Protocol »

2.1. Introduction

Les premières technologies de VoIP étaient propriétaires, donc très différentes les unes des autres. La nécessité de standardisation afin d'assurer une compatibilité entre les produits, a conduit à l'apparition de protocoles standards, comme le H323 et le protocole SIP que nous allons étudier dans ce chapitre.

2.2. Origine et objectif de conception du protocole SIP

Le concept de session est introduit initialement dans la RFC 2727 (Session Description Protocol, SDP, protocole de description de session), et défini comme un ensemble de flux transportant divers types de media entre des émetteurs et des récepteurs. Tous les échanges suivants correspondent à cette définition de session :

- Une conversation téléphonique ;
- Une vidéo conférence ;
- Une prise de contrôle de PC à distance ;
- Deux usagers qui échangent des données ;
- Un échange de messages instantanés.

Le protocole de session SIP (Session Initiation Protocol) fut défini pour la première fois dans la RFC 2543 établie par le groupe de travail MMUSIC (Multiparty Miltimedia Session Control ; contrôle de session multimédia à plusieurs participants de l'IETF.Ce groupe MMUSIC vise à définir un cadre technologique complet pour les communications multimédias, fondé sur les protocoles suivants :

- Le protocole SDP et le protocole d'annonce de sessions SAP (Session Announcement Protocol, RFC 2974) ;
- Le protocole de transmission de données temps réel RTSP (Real Time Streaming Protocol, RFC 2326) pour contrôler les serveurs d'informations en « temps réel », des données isochrones comme des films ;
- Le protocole SIP

Ces protocoles en complètent d'autres également définis à l'IETF, comme le protocole RTP (RFC 1889)qui vient du groupe de travail AVT(Audio/Video Transport)et qui est utilisé pour le transport de données isochrones, ou encore RSVP qui provient du groupe de travail INTRSERV (INTegrated SERVices)et qui permet pour la réservation de bande passante.

Le protocole SIP a maintenant son propre groupe de travail qui se coordonne avec le groupe MMUSIC, principalement sur les travaux de ce dernier concernant les améliorations apportées au protocole SDP qui est utilisé intensivement dans SIP.
Un des objectifs principaux du protocole SIP est de rester aussi simple que possible, et dans ce but des principes « classiques » d'ingénierie de protocoles réseaux, comme l'isolation entre couche protocolaires ou la séparation entre blocs fonctionnels (par exemple la syntaxe et la sérialisation des messages, la retransmission...) ont été passés par pertes et profits, considérées sans doute un peu rapidement comme des lourdeurs inutiles. La RFC d'origine visait à définir en 150 pages tous les détails techniques requis pour la gestion des sessions, couvrant la fiabilité des transmissions de messages de contrôle, le transport, la sécurité, et définissant un ensemble de primitives génériques pour les fonctions suivantes :

- La localisation des usagers, c'est-à-dire la détermination des paramètres techniques (exemple : adresses IP) nécessaire pour joindre le terminal à utiliser pour la communication. Ce point couvre aussi l'association entre usagers et terminaux ;
- La disponibilité des usagers, qui comprend leur accessibilité, et leur intention d'accepter ou non une communication ;
- Les capacités des terminaux, permettant de déterminer le type de média et leurs paramètres utilisable pour la communication ;
- L'établissement de session, pour « sonner » le dispositif distant, et établir la session média entre l'appelant et l'appelé ;
- La gestion de la session, y compris le transfert des flux en cours de communication, de relâchement de la communication, la modification des paramètres de session, et l'invocation de services. Le périmètre de SIP à été volontairement restreint aux conférences à contrôle « lâche » (loose control), c'est-à-dire que les fonctions comme le contrôle de droits de parole ou l'élection de l'orateur sorte du cadre de SIP tel que défini actuellement. Ces fonctions pourront éventuellement être rajoutées par un protocole de niveau supérieur encapsulé dans les messages SIP. [2][16]

2.3. De la RFC 2543 à la RFC 3261

Le protocole SIP est longtemps reste sous forme de document de travail, ce que l'IETF appelle un « draft » , avant d'être finalement publié sous forme de RFC portant le numéro 2543 en mars 1999. Cette première spécification publiée porte déjà le numéro de version « 2.0 ». Cette première version de la RFC SIP contenait de nombreuses erreurs en conséquences ne pouvait être considérée comme une spécification. Les premiers réseaux SIP

utilisés donc tous leur propre « version » de SIP, avec des corrections et des extensions apportées par chaque fabriquant à la RFC originale.

Avec l'expérience accumulée au cours de ces essais la RFC a été mise à jour dans neuf versions intermédiaires (« bis »), jusqu'à ce que tous les changements soient fusionnés dans une nouvelle RFC en juin 2002 : la RFC 3261. Les aspects les plus importants de la RFC originale sont maintenant sépares dans plusieurs RFC plus spécifiques. La RFC 3261 est une nouvelle version majeure du protocole, et n'est pas compatible avec l'ancienne version, en dehors des scénarios d'appels les plus simples. [2]

2.4. L'architecture SIP

2.4.1. Architecture en couche de SIP

L'architecture en couches de SIP, telle que la présente le modèle OSI, incorpore les protocoles : RTP, RSVP, RTCP, SAP et SDP.

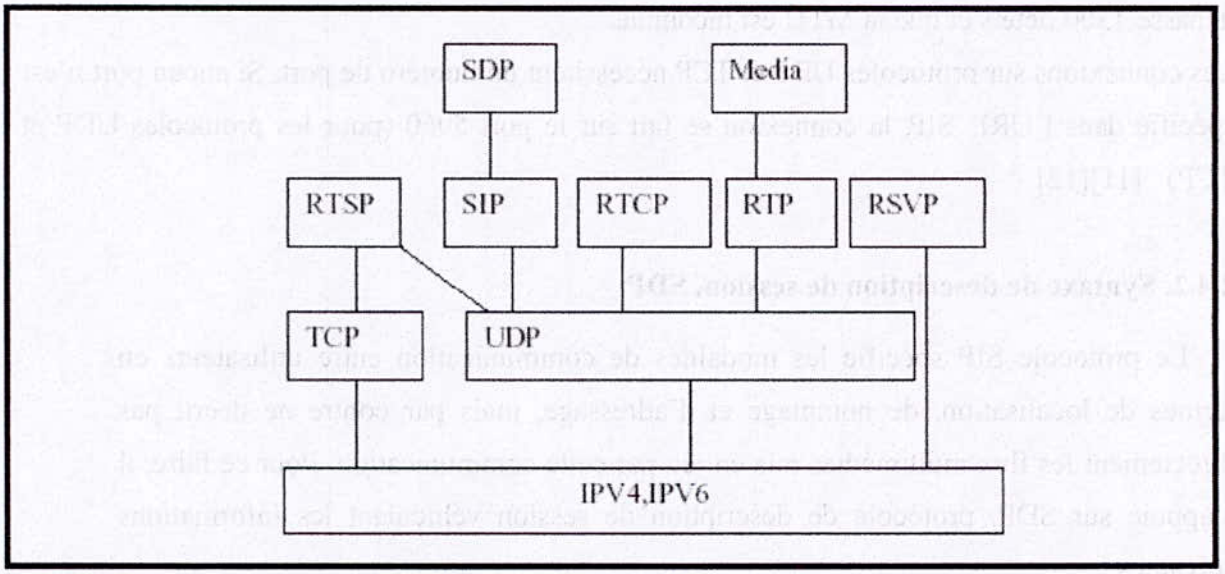


***Figure 2.** 1. Architecture en couche SIP*

- **RSVP** est un protocole utilisé pour réserver les ressources réseaux sur IP avec une excellente qualité de service (QoS) ;
- **R.T.P.** (Real-time Transport Protocol) pour transporter des informations en temps réel avec une excellente qualité de services ;
- **R.T.C.P.**(Real-Time streaming Control Protocol) pour assurer le contrôle de flux des données multimédia ;

- **S.A.P.** (Session Announcement Protocol) pour préciser si les sessions multimédia ouvertes ;
- **S.D.P** (Session Description Protocol) est un protocole de description des sessions multimédia.

Les messages échangés par un client et un serveur SIP sont indépendants du protocole de transport sous-jacent. La RFC d'origine n'imposait que le support du protocole de transport UDP. La RFC 3261 impose maintenant le support des protocoles UDP et TCP, mais le protocole UDP est le plus utilisé car il donne un meilleure contrôle sur la retransmission et les délais de latence. Le seul problème qui se pose avec UDP est qu'il ne peut pas transporter beaucoup d'information sans provoquer de fragmentation (SIP ne définit aucun mécanisme de fragmentation de niveau applicatif). La RFC 3261 recommande donc d'utiliser TCP, si la taille de la requête approche de moins de 200 octets la taille du plus long élément d'information (MTU, Maximum Transmit Unit) le long du chemin entre émetteur et récepteur, ou bien si elle dépasse 1300 octets et que la MTU est inconnue.
Les connexions sur protocoles UDP et TCP nécessitent un numéro de port. Si aucun port n'est spécifié dans l'URI SIP, la connexion se fait sur le port 5060 (pour les protocoles UDP et TCP). [11][12]

2.4.2. Syntaxe de description de session, SDP

Le protocole SIP spécifie les modalités de communication entre utilisateurs en termes de localisation, de nommage et d'adressage, mais par contre ne décrit pas directement les flux multimédias mis en jeu par cette communication. Pour ce faire, il s'appuie sur SDP, protocole de description de session véhiculant les informations suivantes :

- le nom de la session de communication ;
- son but (ou son objet) ;
- les dates et heures d'activité de cette session ;
- les divers flux audio ou vidéo qui la composent ;
- tout paramètre caractérisant ces flux (adresses, ports, formats, etc.) ;
- de façon optionnelle, de l''information additionnelle, précisant par exemple la bande passante requise ou une information de contact de la personne responsable.

Une description de session SDP est une information textuelle, consistant en lignes de caractères de la forme type = valeur.

La description de session est structurée en une section qui s'applique à toute la session (commençant par v = ...), et plusieurs sections de description de média (commençant par m=...). Les paramètres dans les sections média remplacent les paramètres par défaut du niveau session. [2][14]

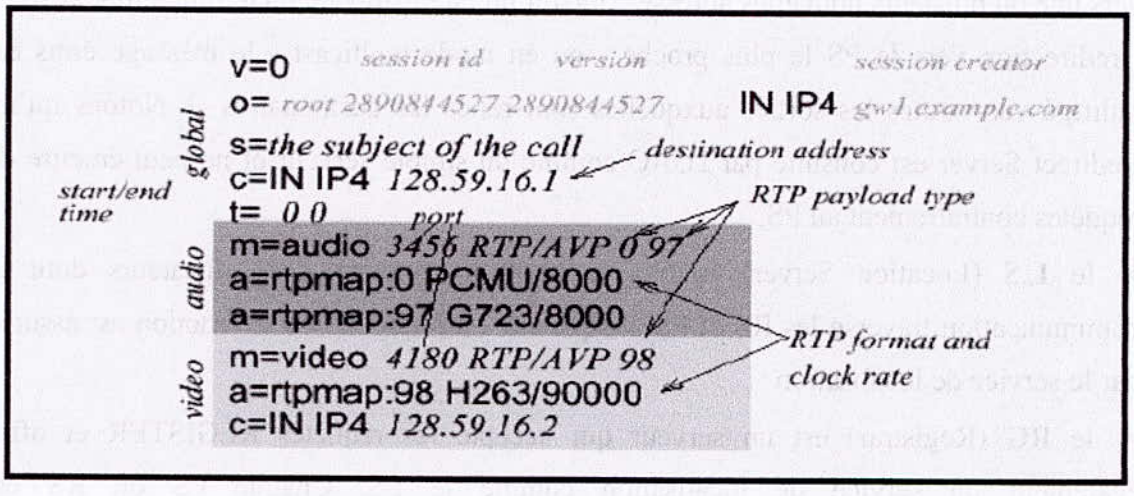


Figure 2. 2. *Exemple de SDP pour la téléphonie IP*

2.4.3. Utilisation de SIP

L'ouverture d'une session à l'aide du protocole SIP peut s'effectuer de façon directe entre deux User Agents qui jouent le rôle de client et de serveur ou de façon indirecte au travers d'un serveur proxy. Dans ce dernier cas, le serveur à en charge la localisation des utilisateurs dont l'adresse est passé dans le message INVITE. Dans le cas de changement de localisation, le serveur proxy est renseigné sur l'adresse de l'utilisateur à l'aide du serveur de localisation. Et le serveur proxy adresse un message 302 MOVE TEMPORARILY avec les nouvelles coordonnées de localisation

2.4.4. Topologies

Dans un système SIP on trouve deux composantes, les users agents (U.A.S et U.A.C) et un réseau de serveurs.

- l'**U.A.S** (User Agent Server) : représente l'agent de la partie appelée, c'est une application de type serveur qui contacte l'utilisateur lorsqu'une requête SIP est reçue.
  Et elle renvoie une réponse au nom de l'utilisateur

- l'**U.A.C** (User Agent Client) : représente l'agent de la partie appelante, c'est une application de type client qui initie les requêtes
- le **relais mandataire** ou **P.S**. (Proxy Server) : auquel est relié un terminal fixe ou mobile (lors de son déplacement, le terminal est relié au PS le plus proche et change constamment de PS) agit à la fois comme client et serveur. Un tel serveur peut interpréter et modifier les messages qu'il reçoit avant de les retransmettre
- le **R.S** (Redirect Server) : réalise simplement une association (**mapping**) d'adresses vers une ou plusieurs nouvelles adresses (lorsqu'un client appelle un terminal mobile - redirection vers le PS le plus proche - ou en mode multicast - le message émis est redirigé vers toutes les sorties auxquelles sont reliés les destinataires -). Notons qu'un Redirect Server est consulté par l'UAC comme un simple serveur et ne peut émettre de requêtes contrairement au PS;
- le **L.S** (Location Server) fournit la position courante des utilisateurs dont la communication traverse les RS et PS auxquels il est rattaché : cette fonction est assurée par le service de localisation ;
- le **RG** (Registrar) est un serveur qui accepte les requêtes REGISTER et offre également un service de localisation comme le LS. Chaque PS ou RS est généralementrelié à un Registrar.

La figure ci-dessous illustre la liaison entre les différents types de serveurs SIP :

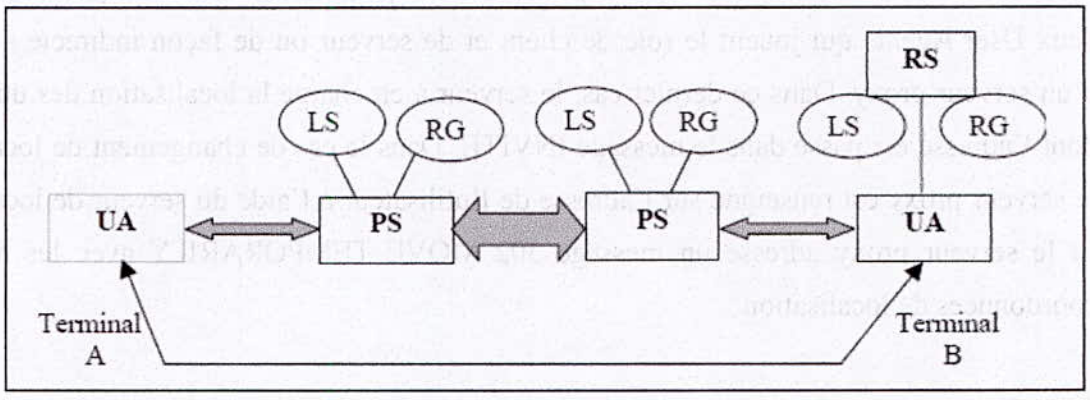


Figure 2. 3. *Mode Client/Serveur et les serveurs SIP -*

L'appel entre les deux terminaux A et B (UA) peut s'effectuer directement ou bien en adressant une requête à un serveur SIP (figure 2.3).

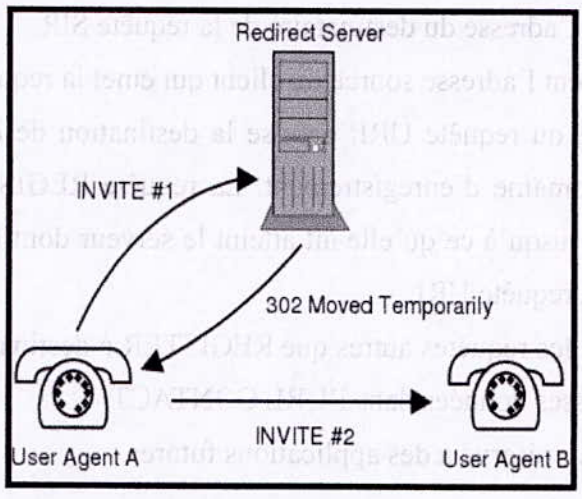


Figure 2. 4. *REDIRECT server*

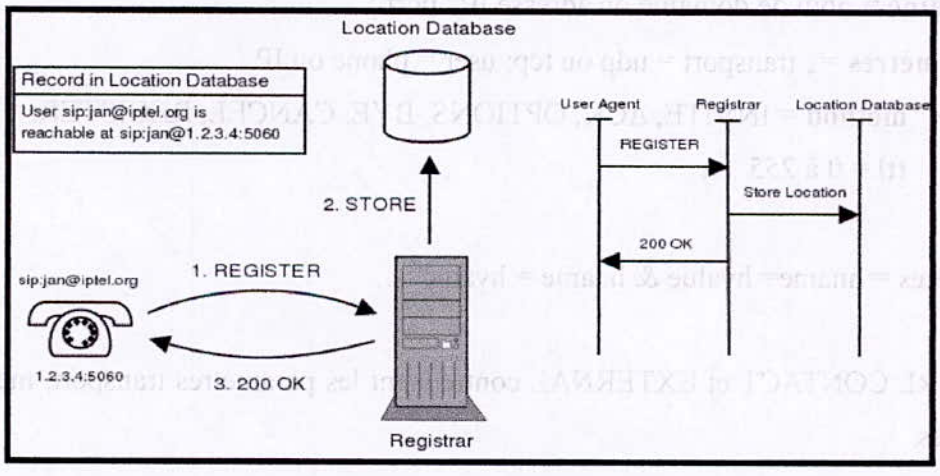


Figure 2. 5. *REGISTRAR server*

2.5. URL SIP

Cinq types de format d'appellation universel SIP (URL SIP) sont définis qui sont : (FROM, COURANTE, TO, CONTACT et EXTERNAL).

- URL TO : contient l'adresse du destinataire de la requête SIP.
- URL FROM: contient l'adresse source du client qui émet la requête SIP.
- URL COURANTE ou requête URI: précise la destination de la requête REGISTER c'est à dire son domaine d'enregistrement. La requête REGISTER est transmise de serveur en serveur jusqu'à ce qu'elle ait atteint le serveur dont le domaine correspond à celui listé dans la requête URI.
- URL CONTACT : les requêtes autres que REGISTER à destination de l'URL TO sont redirigés aux adresses données dans l'URL CONTACT.
- URL EXTERNAL : réservé à des applications futures.

La structure de l'URL est la suivante :

sip :informations_utilisateur@domaine paramètres en-têtes avec :

- **Informations_utilisateur** = (nom de l'utilisateur : mot de passe) ou (numéro de téléphone si user = phone) ;
- **domaine** = nom de domaine ou adresse IP : port ;
- **paramètres** = ; transport = udp ou tcp; user = phone ou IP ;
  - **method** = INVITE, ACK, OPTIONS, BYE, CANCEL, REGISTER:
  - **ttl** = 0 à 255 .....
- **en-têtes** = hname= hvalue & hname = hvalue ....

Seules les URL CONTACT et EXTERNAL contiennent les paramètres transport, method, ttl, et des en-têtes.

Nous illustrons par un exemple une requête SIP par l'exemple d'une personne nommée mfga dont l'url est mfga@ecole.dz qui invite une personne Bill ayant pour url Bill@ele.enp.edu.dz

La requête INVITE sera la suivante :

INVITE sip :ele.enp.edu.dz
Via: SIP/2.0/UDP enp.edu.dz
From: sip:mfga@uqam.dz

To: sip:Bill@ele.enp.edu.dz
CallID: 1468@enp.edu.dz
Cseq: 1 INVITE

2.6. Les messages SIP

2.6.1. Syntaxe des messages SIP

Les messages SIP sont codés en utilisant la syntaxe de message http/1.1(RFC 2068). Le jeu de caractères utilisé est défini dans la norme ISO 10646 et utilise le codage UTF-8(RFC 2279). Les lignes sont terminées par les caractères CR LF (Retour Chariot, Nouvelle Ligne), mais les récepteurs doivent également accepter de recevoir les codes CR ou LF seuls.
Il existe deux types de messages, les requêtes et le réponses, structurées comme l'indique la Figure2.6 (SP étant l'abréviation de « Single sPace », un seul caractère espace).[2]

Ligne de depart :	SIP –version SP Status-Code SP Reason phrase CRLF (pour les réponses) Méthode SP Request URI SP SIP-version CRLF(pour les requêtes)
En-têtes :	en-tête général, ou de réponse, ou d'entité (pour les réponses) ou de requête, ou d'entité(pour les requêtes)
Ligne vide	
Corps du message	

Figure 2. 6. *Format de message SIP*

Corps du message : dans le cas des requêtes, un corps est ajouté ou non selon la méthode utilisée (par exemple : le corps du message d'une requête INVITE contient des information indiquant la progression de la requête). Dans le cas des réponses, le corps du message est obligatoire (par exemple : la réponse à une requête INVITE contient dans le corps du message, une description de la session ; protocole SDP).

2.6.2. Les en-têtes

2.6.2.1. En-tête général

Certains des champs d'en-tête sont présents à la fois dans les requêtes et les réponses, et forment l'en-tête général « general header » :

- Call-ID, Cseq, From, To, Via, Encryption, Content type et Content length

2.6.2.2. En-têtes de requête

En plus des champs obligatoires de l'en-tête général, les requêtes peuvent comporter des champs d'en-tête supplémentaires, formant l'en-tête de requête (request header) :

- Accept, Accept-Language, Expires, Priority, Record-Route et Subject

2.6.2.3. En-tête de réponse

Le champ d'en-tête de réponse autorise le serveur à ajouter des informations concernant sa réponse, qui ne peuvent pas être placées dans la ligne d'état, sur lui- même et sur l'accès à la ressource identifiée par la requête URI :

- Allow ou Proxy-Authorization ou Retry-After ou Server ou Unsupported ou Warning ou WWW-Authenticate.

2.6.2.4. En-tête d'entité

Un champ qui définit le type d'informations contenues dans le Corps du message ou la ressource identifiée par la requête en l'absence du Corps du message :

- Content-Encoding ou Content-Lenght ou Content-Type

Dans le tableau suivant on donnera le détail d'utilisation de ces différents champs :

Champs	Utilisations
Accept	Utilisé dans les messages INVITE, OPTIONS et REGISTER qui permet d'indiquer les types de média qui seront acceptés dans la réponse à ce message.
Allow	Indique les méthodes valides supportées par les entités identifiées par la requête URI.

Call-ID	Identifie une invitation précise ou tous les enregistrements d'un client particulier.
Contact	Champ pouvant apparaître dans les requêtes INVITE, ACK et REGISTER ou dans les réponses de codes 1xx, 2xx, 3xx et 485. Il fournit en général l'URL où l'utilisateur pourra être contacté
Content-Length	Il indique simplement la taille du Corps du message envoyé, en nombre décimal d'octets.
Cseq	Chaque requête doit obligatoirement contenir un numéro de séquence Cseq (entier non signé de 32 bits). Le Cseq initial est choisi arbitrairement par celui qui envoie la requête INVITE mais doit toujours être inférieur à 2^31. Le numéro de séquence s'incrémente d'une unité pour chaque nouvelle requête envoyéedans un dialogue (à l'exception des requêtes ACK et CANCEL).
From	indique la personne à l'origine du message
Max-forwards	utilisé pour limiter le nombre de PS ou passerelles que la requête peut traverser jusqu'au prochain serveur dans le sens de l'UAC vers l'UAS (downstream).
Server	Il contient les informations sur les softwares utilisés par les UAS
To	C'est l'adresse du destinataire. Ce champ est bien sûr obligatoire. Liste quelles configurations ne sont pas supportées par le serveur.

Via	Contient les adresses des serveurs (PS) que traverse la requête. Permet de distinguer les deux versions de SIP (RFC 2543 et RFC 3261). Selon la RFC 3261 le champ d'en-tête <via> contient un paramètre « branch » commençant par « z9hG4bK ».

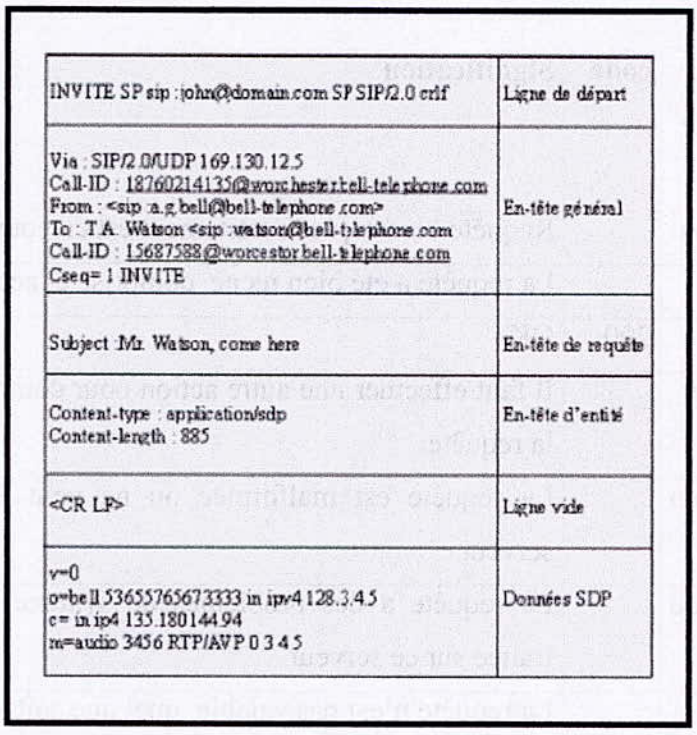
Tableau 2. 1. *Exemples de champs d'en-têtes*

2.6.3 Les requêtes SIP

Les requêtes SIP sont envoyées d'un client SIP à un serveur SIP. Les méthodes suivantes sont définies :

- **ACK** : Cette requête est envoyée d'un client SIP à un serveur SIP pour confirmer qu'il a reçu une réponse finale d'un serveur, comme 200 OK, pour la requête INVITE ;
- **BYE** : Cette requête est envoyée par l'appelant ou l'appelé pour relâcher un appel ;
- **CANCEL** : Cette requête sert à annuler une requête précédente, tant que le serveur n'a pas envoyé de réponse finale ;
- **INFO** : Définie dans la RFC 2976, cette requête est utilisée pour transporter de l'information qui ne change pas l'état de l'appel. Certains vendeurs l'utilisent pour transporter des tonalités DTMF ;
- **INVITE** : Cette requête est utilisée pour commencer un nouvel appel ;
- **MESSAGE**: Définie dans la RFC 3428, cette requête permet l'envoi de messages instantanés ;
- **NOTIFY** : Définie dans la RFC 3265, cette requête sert à envoyer les notifications d'évènements ;
- **OPTIONS** : Un client envoie une requête OPTIONS à un serveur pour s'acquérir de ces capacités. Le serveur doit renvoyer la liste des méthodes supportées, et peut également dans certains cas préciser les capacités particulières de l'utilisateur précisé dans l'URL, et indiquer comment il aurait répondu à un éventuel message INVITE ;
- **PRACK** : Définie dans la RFC 3262, cette requête implémente le mécanisme spécial de sécurisation des réponses provisoires ;
- **PREFER** : Définie dans la RFC 3515, cette requête permet la redirection d'appels ;

- **REGISTER** : Cette requête permet aux clients d'enregistrer leur localisation sous forme d'une ou plusieurs adresses auprès d'un serveur d'un serveur particulier « registrar » ;
- **SUSCRIBE** : Définie dans la RFC 3265 pour demander une notification d'évènements ;
- **UPDATE** : Cette méthode supplémentaires est définies dans la RFC 3311, utilisée pour la mise à jour des paramètres média avant la réponse finale au premier INVITE (phase : early dialog).



INVITE SP sip :john@domain.com SP SIP/2.0 crlf	Ligne de départ
Via : SIP/2.0/UDP 169.130.12.5 Call-ID : 187602141351@worchester.bell-telephone.com From : <sip :a.g.bell@bell-telephone.com> To : T.A. Watson <sip :watson@bell-telephone.com Call-ID : 15687588@worcestor.bell-telephone.com Cseq= 1 INVITE	En-tête général
Subject :Mr. Watson, come here	En-tête de requête
Content-type : application/sdp Content-length : 885	En-tête d'entité
<CR LF>	Ligne vide
v=0 o=bell 53655765673333 in ipv4 128.3.4.5 c= in ip4 135.180144.94 m=audio 3456 RTP/AVP 0 3 4 5	Données SDP

Figure 2. 7. *Format d'une requête SIP*

2.6.4. Les réponses SIP

Un serveur SIP répond à une requête SIP au moyen de une ou plusieurs réponses, dont les codes sont de la forme 2xx, 3xx, 4xx, 5xx et 6xx, sont des réponse « finales » et terminent la transaction courante. Les réponses de la forme 1xx sont des réponses provisoires (provisional), et ne terminent pas la transaction courante.

Le format général d'une réponse SIP est illustré par la figure2.8.

La première ligne d'une réponse contient toujours un code numérique, suivi d'une courte explication textuelle. La majorité de l'en-tête de la réponse est copiée à partir de la requête initiale. Selon le code de réponse, il peut également y avoir des champs d'en-tête supplémentaires, et le contenu du message peut être vide, ou bien contenir une description de session SDP, ou encore un texte explicatif.

Six catégories de codes sont définies, classées selon la valeur du premier digit. Le tableau suivant montre la différence entre les réponses 1xx, 2xx, 3xx, 4xx, 5xxet 6xx .

Famille De codes	**Type De réponse**	**code**	**Signification**
1xx	Information		Requête bien reçue,le traitement est en cours
2xx	Succès		La requête a été bien reçue, comprise et acceptée
		200	OK
3xx	Redirection		Il faut effectuer une autre action pour compléter le traitement de la requête
4xx	Erreur du client		La requête est malformée ou ne peut être exécutée par ce serveur
5xx	Erreur du serveur		La requête a des problèmes de syntaxe ou ne peut pas être traitée sur ce serveur
6xx	Problème global		La requête n'est pas valable, quel que soit le serveur

Tableau 2. 2. *Les codes SIP*

Cette classification rend plus aisée l'ajout de nouveaux codes de réponses : dans le cas où un terminal ne comprend pas un code Cxx, il doit le traiter comme un code C00. Ainsi même les terminaux obsolètes sont susceptibles de réagir « intelligemment » face à une réponse de code inconnu. Ces terminaux peuvent également fournir les informations textuelles contenues dans la courte explication textuelle.

La RFC 3326 a rajouté un nouvel en-tête Reason qui permet notamment de véhiculer de manières transparentes les codes utilisés par le réseau téléphonique.

SIP/2.0 302 Moved temporarily	Ligne d'état
From : sip : user@caller.com To : sip :user1@domain.com Call-ID :27145@caller.com location : sip :bob@domain.comm expire : Wed, 29 jul 2005 :00 :00 GMT cseq : 1 INVITE	En-têtes
< >	Ligne vide
Données de réponses	Corps du message

Figure 2. 8. *Format des réponses SIP*

2.7. Dialogue, transaction, et retransmission de messages

Un dialogue SIP est identifie par la combinaison des tags « From » et « To », du « Call-ID » et du paramètre « branch » de l'en tête « Via » (figure suivante). Une fois le dialogue établi, toutes les requêtes et les réponses du dialogue doivent inclure ces champs d'en tête. Ceci l'identification du dialogue concerné pour le « user agent » qui reçoit le message.

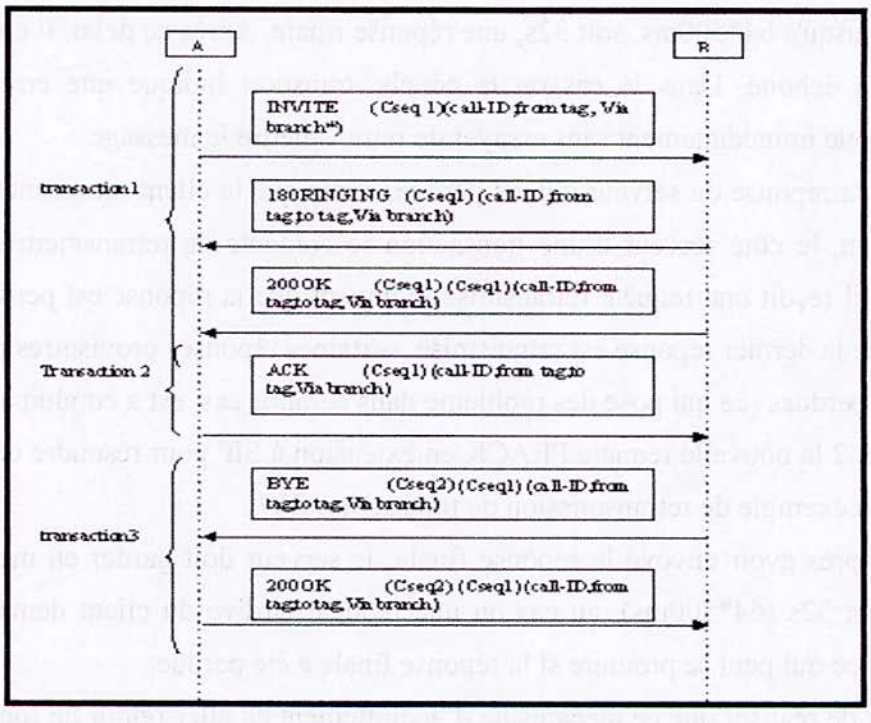


Figure 2. 9. *Exemple de dialogue SIP*

Chaque transaction est identifiée par la valeur commune de l'en-tête « Cseq » (le nom de méthode et le numéro de séquence doivent être identique). Les valeurs de l'en-tête « Cseq » doivent être distincts pour des transactions distinctes. Les seules exceptions sont pour la méthode ACK et la transaction CANCEL.

La transaction CANCEL utilise le même identificateur de séquence « Cseq » que la transaction qu'elle tente d'annuler, mais le nom de la méthode est « CANCEL ».
La transaction ACK n'est utilisée qu'en relation avec une transaction précédente de type INVITE, qui utilise un mécanisme de connexion différent de toutes les autres transactions SIP qui n'utilisent qu'un aller-retour par transaction.

2.8. Les transactions

2.8.1. Transactions autres que INVITE

Lorsque SIP est utilisé sur des protocoles de transport non fiables, la fiabilité des transactions autres que INVITE se fonde sur la retransmission de message.

L'émetteur d'une requête retransmettra un message s'il ne reçoit pas la réponse provisoire ou définitive en moins de 500 ms[1] (ou une meilleure estimation du temps d'aller-retour si le »user agent » la calcule). Il continuera à la transmettre jusqu'à ce qu'il reçoive une réponse, en doublant l'intervalle de retransmission à chaque essai, jusqu'à atteindre 4s. Le client attendra jusqu'à 64*500ms, soit 32s, une réponse finale. Après ce délai, il considéra que la transaction a échoué. Dans le cas ou la couche transport indique une erreur, alors la transaction échoue immédiatement sans essayer de retransmettre le message.

Si c'est la réponse du serveur qui est perdue, alors que le client retransmet la requête. Pour cette raison, le côté serveur d'une transaction se contente de retransmettre sa réponse chaque fois qu'il reçoit une requête retransmise indiquant que la réponse est perdue. A cause du fait que seule la dernier réponse est retransmise, certaines réponses provisoires peuvent être définitivement perdues (ce qui pose des problème dans certains cas, est a conduit à introduises dans la RFC 3262 la nouvelle requête PRACK en extension à SIP pour résoudre ce point). La figure 4.4 est un exemple de retransmission de transaction BYE.

Même après avoir envoyé la réponse finale, le serveur doit garder en mémoire cette dernière pendant 32s (64*500ms), au cas ou une requête tardive du client demanderait une retransmission, ce qui peut se produire si la réponse finale a été perdue.
Il est important de réaliser que ce mécanisme d'acquittement en aller-retour ne fonctionne que si la réponse finale à la requête arrive rapidement après que la requête ait été envoyée .SIP s'attend à ce que les transactions autres que INVITES s'exécutent en quelques secondes. Si ce

mécanisme d'acquittement en deux coups était utilisé pour des transactions qui prennent plus de temps à s'exécuter, alors la requête initiale serait envoyée plusieurs fois, ce qui bien sur serait très efficace. De ce fait, un mécanisme différent est requit pour les transactions de type INVITE. [2][12][16]

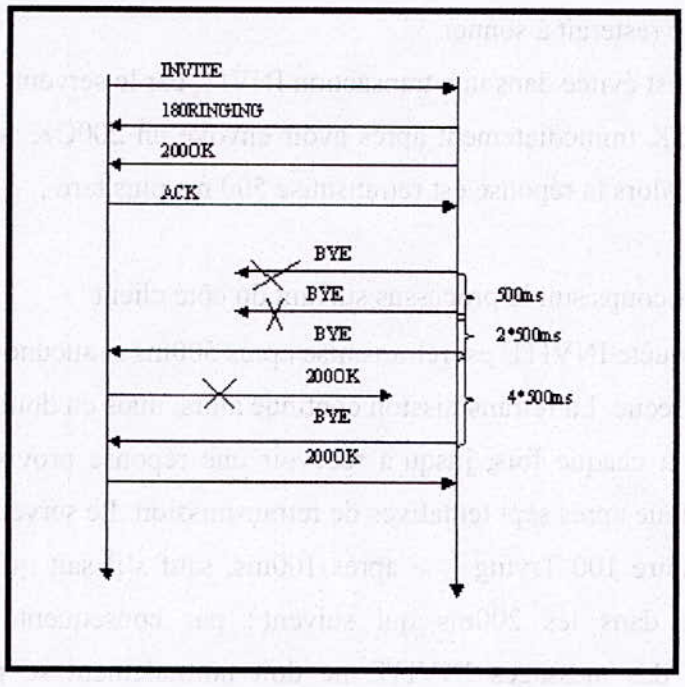


Figure 2. 10. *Retransmission de transaction BYE*

2.8.2. Transaction INVITE

A) Acquittement en trois coups

Le traitement de la requête INVITE est totalement différent du traitement des autres transactions. Il s'agit d'un des aspect les plus complexes du protocole SIP .

Les transactions en deux coups utilisées pour les requêtes autres que INVITE présenteraient les problèmes suivants :

- La transaction initiale est renvoyée toutes les 4 s jusqu'à ce qu'une réponse finale arrive. Dans une application téléphonique, le message 200 OK signalant que l'utilisateur a décroché peut arriver jusqu'à 3 minutes après l'envoi du INVITE (c'est la durée généralement laissée a l'état de sonnerie avant de relâcher une tentative d'appel dans le réseau), le message 200OK peut même ne jamais arriver .Afin d'éviter

des retransmissions inutiles, la retransmission du message INVITE s'arrête dés qu'une réponse provisoire ou finale est reçue ;

- Si la réponse 200OK est perdue, elle ne serait retransmise que 4 secondes plus tard dans le schéma de retransmission décrit plus haut , ce qui serait bien entendu inacceptable car aucun flux média bidirectionnel ne pourrait être établi entre l'appelant et l'appelé .Au lieu de cela ,si l'appelant a reçu une réponse provisoire 180RINGING , l'état de la ligne resterait à sonner.

  Cette situation est évitée dans une transaction INVIT, car le serveur s'attend à recevoir une requête ACK immédiatement après avoir envoyé un 200OK .si le message ACK n'est pas reçu , alors la réponse est retransmise 500 ms plus tard .

L'acquittement en trois coups suit le processus suivant du côté client :

- Sur UPD, la requête INVITE est retransmise après 500ms si aucune réponse provisoire ou finale n'est reçue .La retransmission continue alors, mais en doublant l'intervalle de retransmission à chaque fois, jusqu'à recevoir une réponse provisoire ou finale. La transaction échoue après sept tentatives de retransmission. Le serveur doit envoyer une réponse provisoire 100 Trying >> après 100ms, sauf s'il sait qu'il va envoyer une réponse finale dans les 200ms qui suivent : par conséquent le mécanisme de retransmission des messages INVITE ne doit normalement se produire que si le message est perdu, et non pas si le côté serveur est trop lent ;
- sur TCP ou sur un mécanisme de transport fiable, le message INVITE n'est jamais retransmis.

Le côté serveur d'une transaction INVITE sur UDP se contente de retransmettre la dernière réponse provisoire s'il reçoit un message INVITE retransmis (ce qui signifie qu'une ou plusieurs réponses provisoires ont été perdues) . La réponse finale est traitée différemment :

- dans la RFC2543, elle est retransmise après 500ms dans le cas où une requête ACK n'a pas été reçue, puis contenue à être retransmise avec des intervalles de temps qui doublent jusqu'à réception d'un message ACK. La retransmission de la réponse finale est arrêtée après sept essais infructueux, ou si une réponse BYE est reçue pour ce dialogue, ou bien , dans le cas des réponses3xx ,4xx ,5xx ,si une requête CANCEL est reçue .Si la retransmission abandonnée était celle d'un 200OK ,alors le serveur doit générer une requête BYE , au cas où la réponse 200OK serait arrivée au client ;

- dans la RFC 3261 qui spécifie plus proprement la couche de transaction, la retransmission des réponses de type 3xx sur les couches transport fiables n'est plus nécessaire. La réponse 200OK doit par contre être retransmise sur toutes les couches de transport même fiables car elle est susceptible d'être routée par plusieurs Proxy , qui peuvent tout à fait la retransmettre sur UPD est non fiable la réponse 200OK peut être perdue , et SIP n'autorise pas aux Proxy de participer au processus de retransmission des réponses 200OK , lequel est assuré de bout en bout seulement (voir le paragraphe suivant sur la requête ACK).Par conséquent si le serveur ne retransmet pas cette réponse 200OK ,celle-ci peut se perdre dans le réseau .La RFC 2543 étendait ce comportement à toutes les requêtes dans et non seulement les réponses finales 200OK ,afin d'éviter d'ajouter un cas d'exception de plus à une spécification déjà complexe . Dans la RFC 3261, qui définit de manière plus formelle les transactions, seules les réponses 200OK sont retransmises sur les couches de transport fiables. [2][12][16]

B) La requête ACK

La requête ACK termine le mécanisme de fiabilité en trois coups.
Les éléments constituants la requête ACK :

- la majorité des en-têtes doivent être identique aux en-têtes du INVITE originel ;
- l'en-tête « to » contient un « tag » ajouté par les serveurs ;
- l'en-tête « Cseq » contient un numéro identique à celui de la requête INVITE. Ce qui permet de corréler le message ACK à la transaction INVITE correspondante.

2.9. Établissement d'appel SIP

Dans ce cas deux clients établissent l'appel par l'intermédiaire d'un serveur proxy en mode point à point. Le diagramme de séquences suivant illustre ce cas de figure :

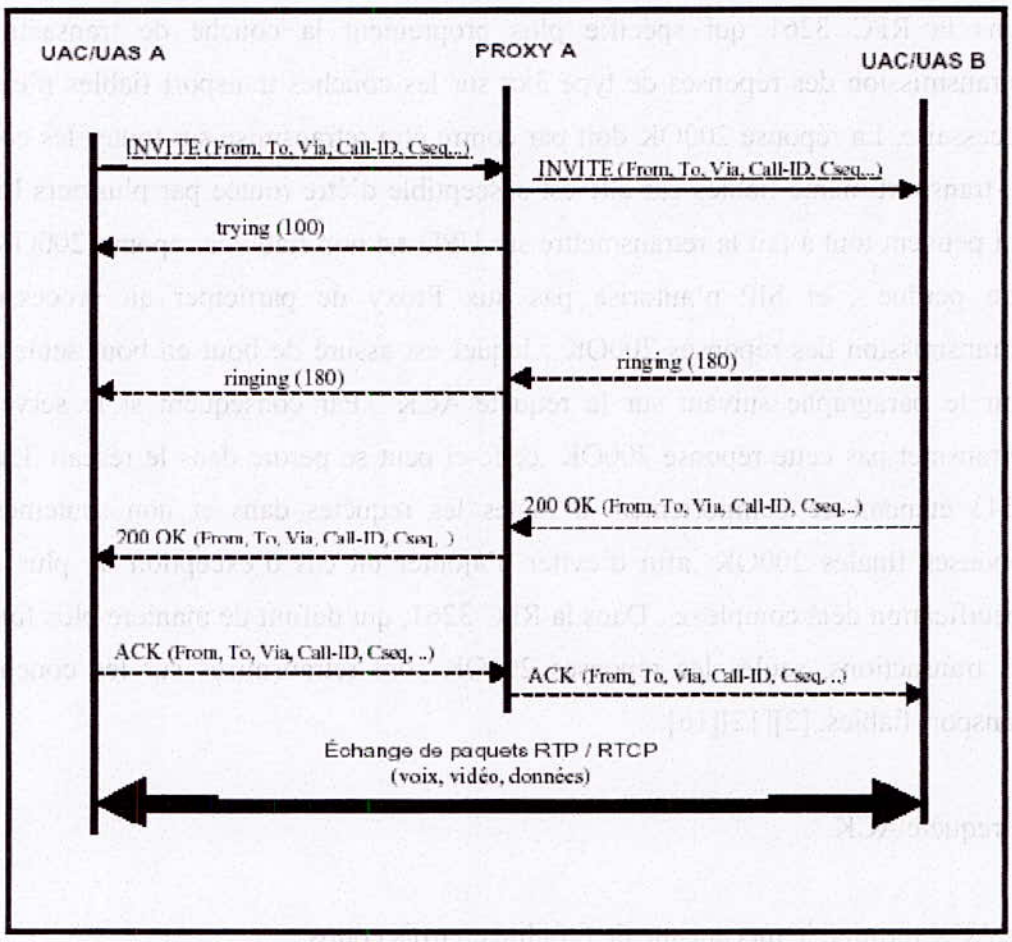


Figure 2. 11. *Etablissement d'appel SIP*

2.10. Transmission d'informations DTMF

Une fois le dialogue SIP établi, de nombreuses situations vont nécessiter la transmission d'informations de contrôle au milieu de l'appel. Pour les communications en temps réel, les requêtes au cours d'un dialogue concernent principalement la transmission de tonalités DTMF.

Les tonalités DTMF sont générées par les touches de téléphones modernes (ou le clavier). Les téléphones plus anciens à cadran rotatif génèrent d'autres signaux : de courtes interruptions, « flashhook », du circuit codent chaque chiffre du numéro composé. [2]

2.10.1. Transport de signaux téléphoniques sur les codeurs de parole bas débit

les codeurs de voix à bas débit(en pratique en dessous de 32bit/s) ne sont en général pas en mesure de transporter correctement les tonalités DTMF qui sont composées de deux

fréquences qui ne se trouvent pas dans la parole. La majorité des codeurs bas débit fonctionnent en modélisant les éléments fondamentaux de la parole, ne transmettant que les paramètres du modèle, et rendent tout à fait impossible la modélisation de fréquences pures.

Les tonalités DTMF codées puis décodées par de tels codeurs ne sont pas reconnues de manière fiable par les systèmes pilotés pas tonalités DTMF. Le *flashhook*, qui n'est pas un son, n'est pas transporté non plus par les codeurs de parole.

L'avantage clé de la voix sur IP comparé aux autres techniques de téléphonie est la capacité de contrôler un appel sans jamais se trouver sur le chemin du flux média. C est cette propriété qui permet de construire des sofswitchs (commutateurs logiciels), par opposition à la téléphonie traditionnelle qui nécessite des matrices de commutation sur matériel spécifique pour router les flux média.

2.10.2. La RFC 2833

Pour permettre la résolution rapide du problème de la transmission de tonalités DTMF et d'autres événements sur des communications utilisant des codeurs bas débit, la RFC 2833 proposa en mai 2000 un format spécifique de paquet RTP pour représenter les tonalités DTMF et d'autres événements téléphoniques .

La RFC2833 exige que les équipements de bordure de réseau implémentent un algorithme de détection de tonalités DTMF pour tous les flux média qu'ils génèrent. Ceci est trivial pour un téléphone IP qui connaît évidemment qu'elle touche est appuyée , mais pour les passerelles connectées au réseau téléphonique ceci demande d'implémenter des algorithmes de détection de fréquence dans les flux G.711 reçus du réseau classique .

L'idée est d'envoyer l'information DTMF dans le flux RTP sous forme d'un événement nommé , et non pas d'un signale audio codé .Lorsque le flux RTP résultant est reçu par une autre passerelle du réseau téléphonique , cette dernière doit régénérer l'information DTMF sous forme de signal audio .La transmission des événements DTMF dans le flux RTP ,avec le même numéro de séquence et la même référence temporelle que le reste du flux RTP ,assure une synchronisation parfaite entre la tonalité DTMF et le flux média , et évite la duplication éventuelle de signaux DTMF au niveau des passerelles (l'un reçu dans le flux média , l'autre sous une forme codée par la RFC2833) .

La figure 2.12 montre le format d'un événement téléphonique codé dans un paquet RTP. Ce type d'événement doit être généré dés qu'une tonalité spécifique de plus de 50ms est détectée .Chaque paquet est envoyé trois fois pour assurer une certaine redondance , en incrémentant le numéro de séquence RTP,et les autres champs restant identiques. Les tonalités les plus courtes peuvent être codées sous forme de paquet unique en positionnant le bit « end » (fin) .Les tonalités plus longues peuvent être transmises en répétant périodiquement des paquets de tonalités plus courtes, ou bien avec deux paquets dont l'un indique le début de l'événement et l'autre la fin .Cette méthode évite d'attendre la fin de la tonalité pour envoyer l'information sous forme de paquet RTP, ce qui causerait un délai inacceptable. [2][16]

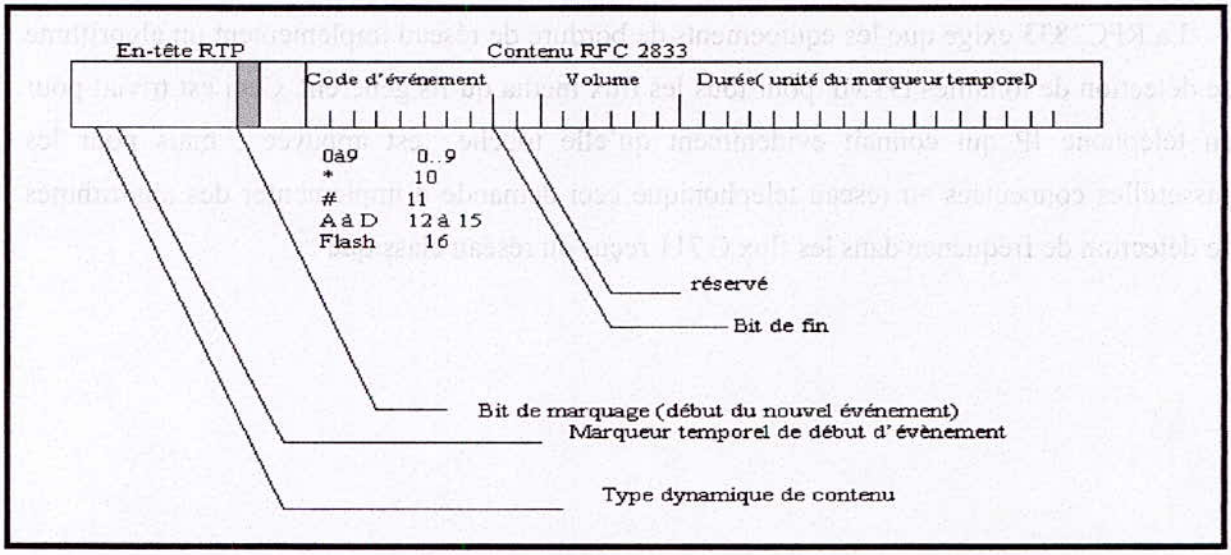


Figure 2. 12. *Format d'un évènement téléphonique codé dans un paquet RTP*

Comme le protocole SIP utilise le format SDP pour décrire le types de média attendus et leur codage, il a été nécessaire de créer un nouvel identificateur de format SDP pour indiquer les types d'événements que peut comprendre un récepteur.

Exemple : la ligne « m » est utilisée pour recevoir des événements téléphoniques :

```
m=audio 44143 RTP/AVP 110
a=rtpmap: 110 telephone-events/8000
```

De plus, le sou paramètre « fmtp » peut servir à indiquer les événements traités par le récepteur. Son format est :

```
a= fmtp :<format> <lis of values>
```

Par exemple un récepteur qui comprend tous les événements définis dans la figure 2.12, sauf A, B, C et D, et utilisant l'identificateur dynamique de type de contenu RTP 100, déclarera :

```
a=fmtp : 100 0-11,16
```

Toutes les implémentations de la RFC2833 doivent savoir gérer les événements 0 à 15, et la ligne « fmtp »est optionnelle.

Un des avantages du codage des informations DTMF sous forme d'événement est que les algorithmes d'analyse de fréquence sont effectués par les dispositifs de bordure (passerelles, téléphone IP) ce qui facilite l'implémentation des serveurs vocaux interactifs IP.

2.11. Exemple de transactions SIP

À l'aide d'un scanner (sniffer) de réseau " Ethereal " , il est possible d'analyser les trames envoyées lors d'une communication. Ceci en choisissant un filtre qui est tout simplement le protocole qu'on veut analyser.

Pour notre exemple :

Filtre = SIP et SDP (pour analyser le corps du message)

A) La requête

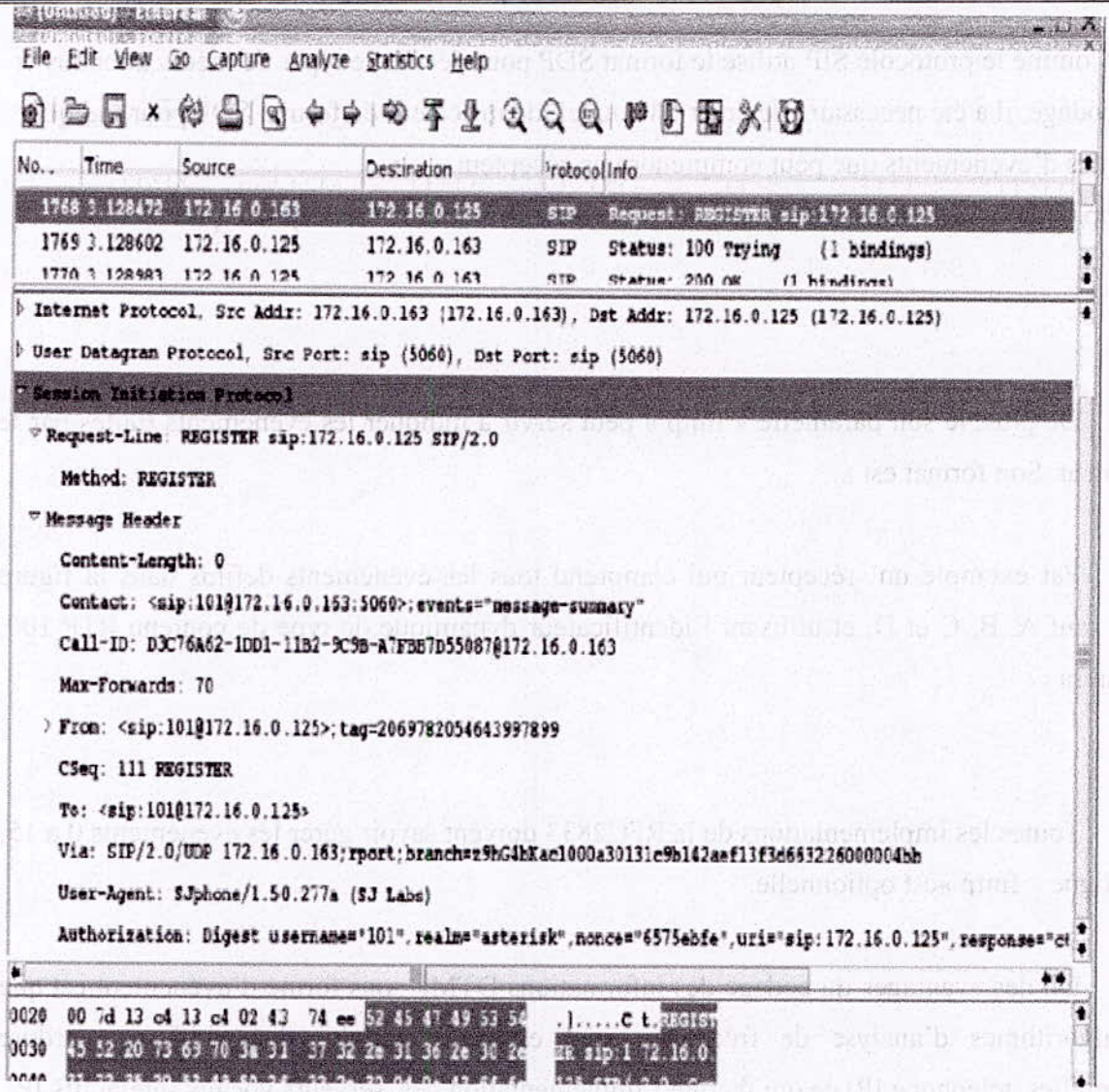


Figure 2. 13. *Analyse d'une requête REGISTER*

Dans cette exemple, une requête REGISTER est envoyé, comme indiqué d'un poste '172.16.0.163' vers un serveur '172.16.0.125'.

Un utilisateur lance une requête d'enregistrement avec un nom d'utilisateur « 101 » comme indiqué dans le champ d'en-tête contact.

Le message SIP est détaillé :

- ligne de requête « Request line »
  - ➢ la methode : REGISER, le protocole : SIP, l'adresse destinataire : 172.16.0.125, la version SIP : 2.0

- les en-têtes du message « message header »

- ➢ en-tête User-Aagent : Sjphone/1.50.177a(SJLabs) représente l'application qui a lancé la requête vers le serveur ; un softpohne .
- ➢ le protocole de transport utilisé est défini dans l'en-tête (UDP) via
- ➢ le port utilisé est de fini dans l'en-tête contact
- ➢ l'en-tête Cseq = 111, un numéro arbitraire

- comme on a vu précédemment, pour les requêtes le corps du message peut être ajouté ou pas selon la méthode utilisée. Pour la methode REGISTER, il n'est pas ajouté.

B) Les réponses

Réponse provisoire

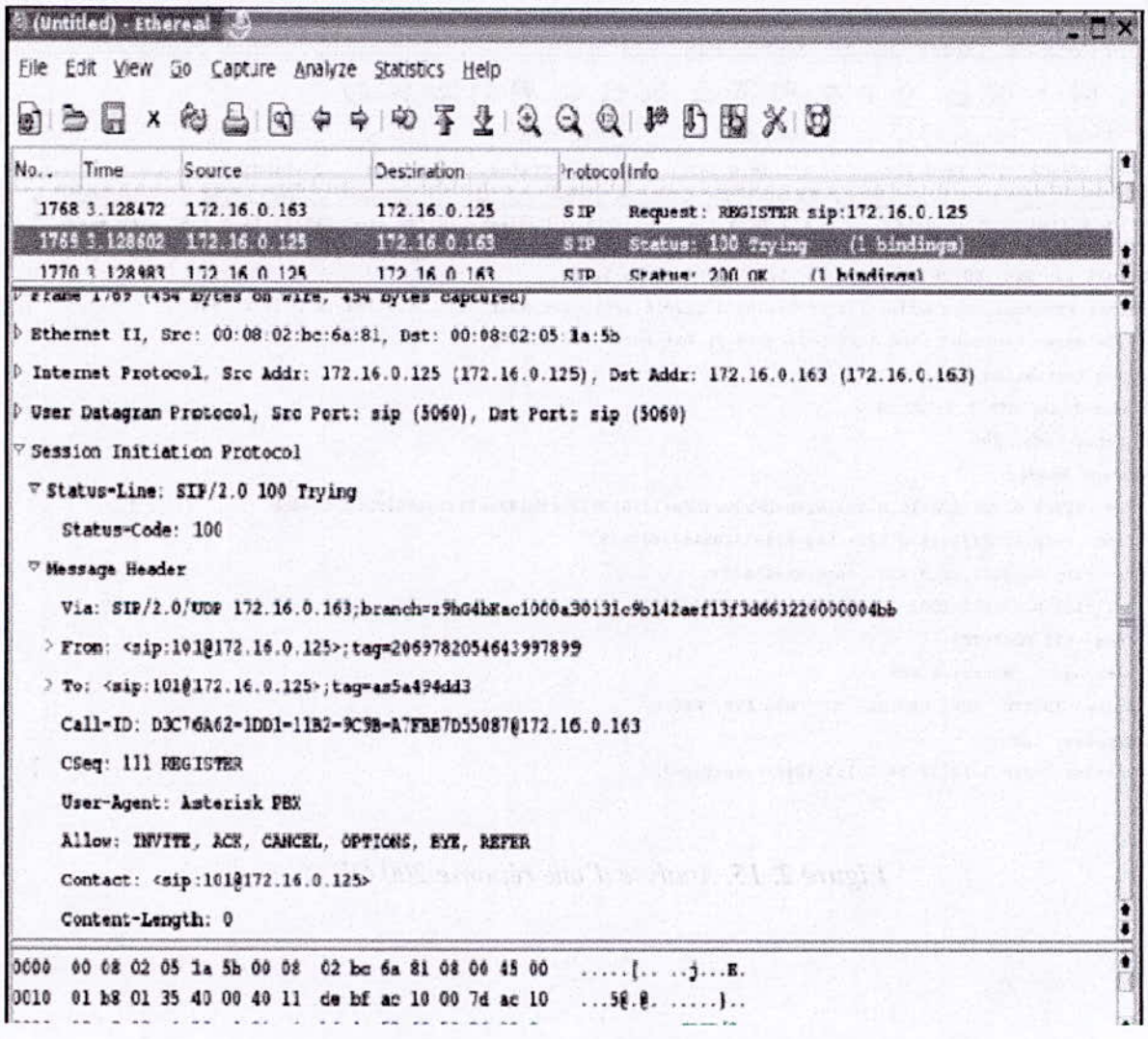


Figure 2. 14. *Analyse d'une réponse 100 Trying*

pour cette réponse on voit bien que la source =172.16.0125 , et que la destination=172.16.0.163 .

- ligne d'état « Status Line »
  - version sip : 2.0 code d'état = 100 « trying », tentative d'enregistrement
- message header
  - la plupart des en-têtes sont identiques à ceus de la requête
  - Cseq =111 il s'agit de la même transaction donc le Cseq n'est pas incrémenté
  - User Agent :Asterisk PBX ; le serveur qui envoie les réponse
- Le corps du message « message body »
  - Une description SDP

Réponse finale

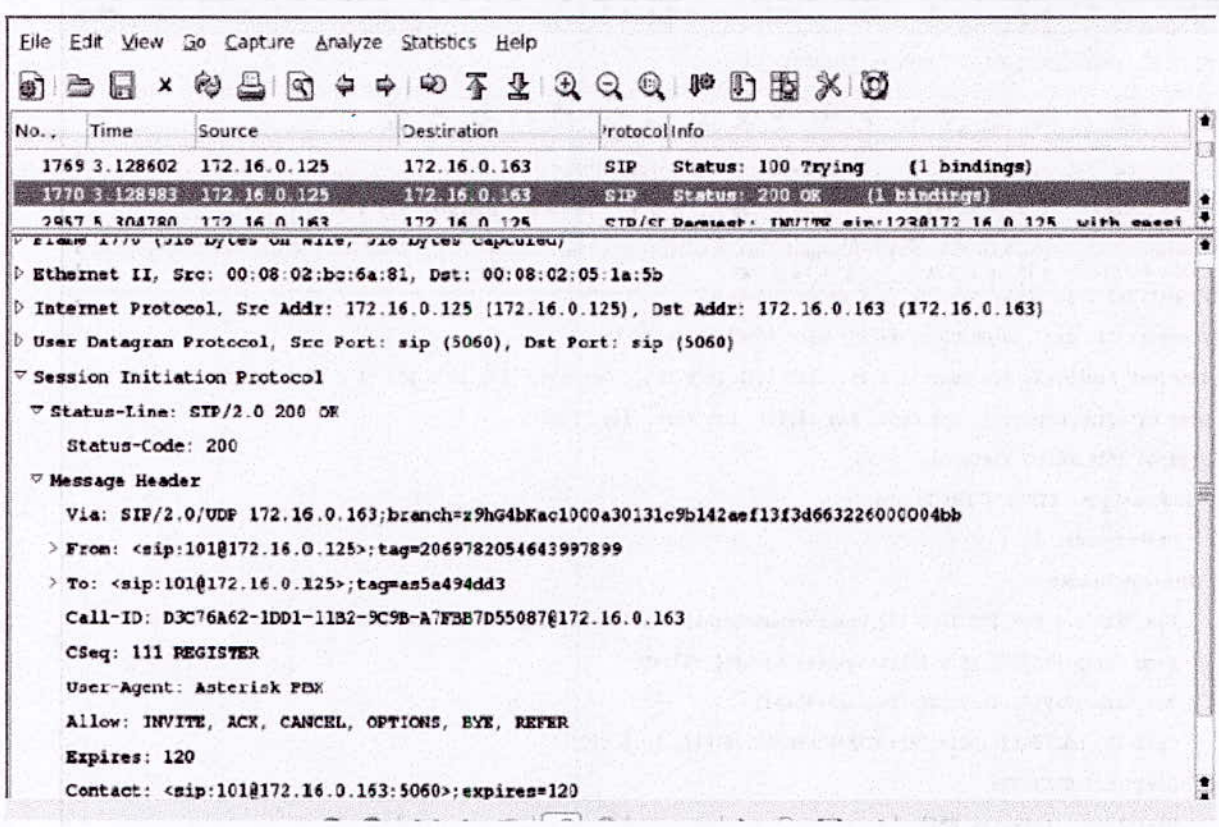


Figure 2. 15. *Analyse d'une réponse 200 OK*

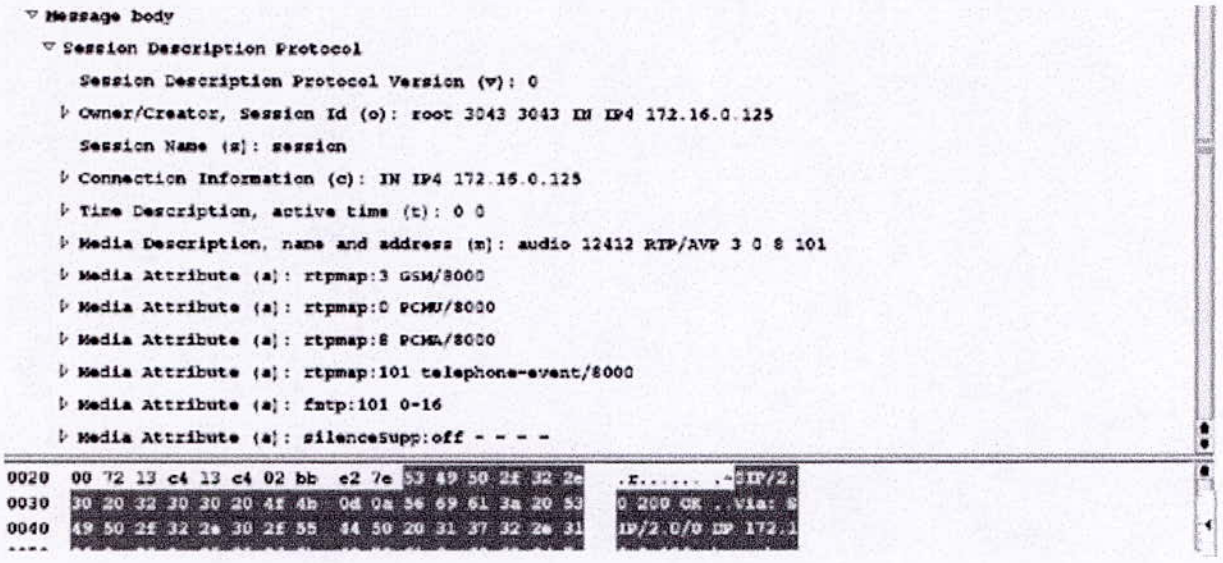


Figure 2. 16. *Description SDP pour une réponse 200OK*

- Le code d'état 200 OK envoyé par le serveur « dans ce cas un REGISTRER » termine la transaction <=> confirmation de l'enregistrement au prés du proxy.
- Le corps du message est une description SDP

2.12. Conclusion

SIP est un protocole de signalisation pour les applications de téléphonie et la visioconférence sur Internet. Sa simplicité et son intégration au monde IP, fait qu'il vient bousculer le mastodonte H.323 déjà bien implanté. Il intervient aux différentes phases de l'appel. Pour initialiser un appel ou y mettre fin.

Chapitre 3

APPLICATION : Etude et mise en œuvre d'un IPBX Asterisk

3.1 Introduction

Dans le présent chapitre nous exposerons notre application qui consiste à étudier et mettre en œuvre un IPBX. Nous avons choisi pour cela l'IPBX logiciel Asterisk.
En première partie de cette application nous procéderons à l'étude de celui-ci en abordant son architecture, sa configuration et puis son intégration dans les réseaux.
En seconde partie nous exposerons les fonctions d'Asterisk que nous avons pu mettre en application pour programmer des services.

3.2. Qu'est Ce qu'ASTERISK

Asterisk est un PBX logique complet, il se compile sous le système d'exploitation Linux et fournit toutes les fonctionnalités d'un PBX traditionnel ou plus. Il permet l'accès à des applications de téléphonie aux éléments d'un réseau. Son développeur est Mark SPENCER de Digium, Inc ; et son nom vient du symbole (*) qui dans les environnements DOS, LINUX et UNIX désigne un caractère générique.

Astérisk est un PBX hybride puisqu'il supporte différents protocoles et technologies de la VoIP, et peut interagir avec plusieurs standards de téléphonie à l'aide de matériel spécifique. Néanmoins, Asterisk n'a besoin d'aucun matériel additionnel pour la voix sur IP. Pour l'intercommunication avec l'équipement numérique et analogue de téléphonie, Astérisk doit être muni d'un certain nombre de dispositifs câblés construits par les commanditaires d'Asterisk ''Digium''.

3.3. Architecture d'ASTERISK

Asterisk est caractérisé par une architecture relativement simple, il comprend un PBX central représentant le noyau du système (central PBX core system) entouré d'APIs. Ce dernier se charge donc de manipuler et de diriger les interconnections internes du PABX en faisant abstraction du type de protocole, de codec ou de technologie utilisé, cette caractéristique offre à Asterisk la possibilité d'utiliser différentes technologies.
Le noyau d'Asterisk (Asterisk Core) manipule les modules suivants:
PBX Switching- commutation du PBX- le noyau de commutation relie de manière transparente les utilisateurs arrivants sur différentes interfaces de matériel et de logiciel.

Application Launcher – lanceur d'application- lance les applications qui fournissent des services aux utilisateurs, on cite la boite vocale, les salles de conférence, lecture de messages, ...

Codec Translator – traducteur de codecs- utilise les modules des codecs pour coder et décoder les différents formats de compression audio utilisés dans l'industrie de la téléphonie. Un certain nombre de codecs est disponible pour palier aux divers besoins et pour arriver au meilleur équilibre entre la qualité audio et l'utilisation de la bande passante.

Scheduler and I/O Manager- planificateur manager des entrées/sorties - Il traite la planification des tâches de bas niveau et la gestion du système pour une performance optimale dans toutes les conditions de charge.

Quatre APIs (Interface de programmation d'applications) sont définies pour les modules chargeables, facilitant l'abstraction du matériel et du protocole. En utilisant ce système de modules chargeables le noyau d'Asterisk s'affranchit des détails de connexion d'un appelant, des codecs utilisés, etc.

Les APIs qui entoure le noyau d'Asterisk sont : (Figure 4.1)

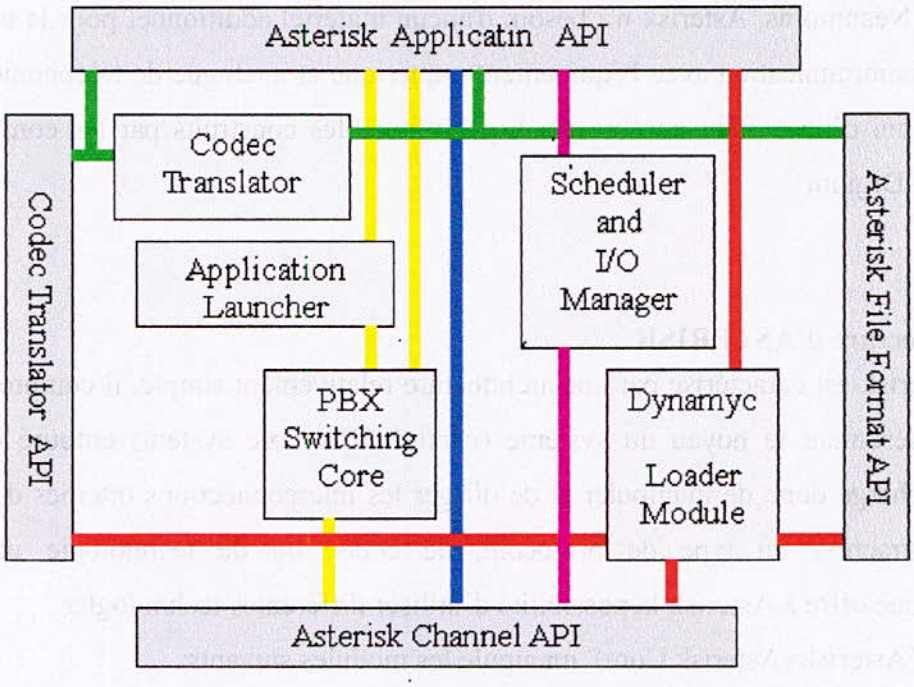


Figure3. 1 Architecture du PABX Asterisk

Channel API – charge le type de connexion sur laquelle l'utilisateur arrive (VoIP, ISDN, PRI). Les modules dynamiques se chargent pour gérer les détails qui caractérisent cette connexion.

Application API – API application- Elle autorise différents modules de tâches à être lancé pour exécuter les diverses fonctions du PABX : conférence, boite vocale, parkage, …..

Codec Translator API –API traducteur de codecs- charge les modules des codecs pour supporter le codage et décodage des différents formats audio (GSM, MP3,…..).

File Format API –gère l'écriture et la lecture des différents formats de fichiers et sauvegarde les données dans les fichiers systèmes.

En utilisant ces APIs Asterisk réalise une abstraction complète entre ses fonctions noyau de serveur PBX et les diverses technologies existantes dans le domaine de la téléphonie. Sa particularité modulaire permet à Asterisk d'intégrer de façon continue le matériel de commutation téléphonique, et les technologies de Voix par paquet. La capacité de charger les modules de codec permet à Asterisk d'être compatible avec le codec extrêmement compact nécessaire à la VoIP sur des connexions lentes comme un modem téléphonique tout en maintenant une haute qualité audio sur des types de connexion moins étroites. L'API d'application assure une utilisation en souplesse des routines d'application pour exécuter n'importe quelle fonction avec souplesse et à la demande, et reste ouverte au développement de nouvelles applications pour répondre aux besoins et situations spécifiques. De plus, en chargeant toutes les applications sous forme de routines, on permet ainsi l'accès à un système flexible, permettant à l'administrateur de concevoir aux appelants, le meilleur des chemins appropriés sur le système PBX et de modifier des chemins d'appel pour répondre aux besoins évolutifs en communication de l'entreprise.

3.4. Les protocoles de la VoIP supportés par ASTERISK

Asterisk supporte trois protocoles de la VoIP ; dont deux normalisés –normes de l'industrie- et un autre spécialement développé pour Asterisk et par conséquent celui qu'Asterisk supporte le mieux.

Inter-Asterisk Exchange (IAX) : IAX représente l'échange d'Inter Astérisk et a été développé comme alternative aux protocoles existants. Il existe actuellement 2 versions de IAX, où IAX2 est le plus commun. IAX n'est soumis à aucune norme, mais il a été adopté par différents fabricants de soft phones et hard phones. Son principal avantage est qu'il utilise un seul port UDP et cela facilite la liaison NAT.

Le protocole SIP : (session initiation protocol) est une norme de l'IETF pour la VoIP. De nombreux fournisseurs utilisent Asterisk avec le canal SIP, on en cite : SNOM et Cisco.

Le protocole H.323 : H.323 est une norme de l'ITU-T (International Telecommunication Union Stanardization Sector) il est utilise pour la teleconference (voix et Video). Il existe deux implémentations pour le protocole H.323 permettant son utilisation avec Asterisk: asteriskoh323 et chan_h323.

3.5. ASTERISK en réseau

Un réseau de téléphonie IP local doit intégrer : un serveur de téléphonie IP et des terminaux permettant la communication qui peuvent être soit des téléphones matériel IP ; soit des téléphones analogiques munis d'un adaptateur qui les transforme en téléphones IP ou même des téléphones logiciel (softphone) installés sur ordinateur.

Asterisk peut assurer le routage des appels, les services et toutes les fonctions d'un serveur VoIP, en plus de la possibilité de se connecter à différents réseaux téléphoniques existants.

Un serveur Asterisk peut en effet être connecté à un réseau téléphonique :

- soit à un réseau téléphonique commuté (RTC)
- soit à un réseau RNIS (ISDN)

Ces connexions ne peuvent être possible qu'après avoir équiper le serveur Asterisk de cartes spécialisées disponibles selon le nombre de lignes à connecter. Des cartes qu'il faut, bien évidemment configurer au niveau des fichiers de configurations d'ASTERISK.

3.5.1. Réseau local

Dans un réseau local, il existe trois communications possibles :

- d'un téléphone IP à un téléphone IP ;
- d'un téléphone IP à un téléphone analogique (POT) ;
- d'un téléphone analogique (POT) à un téléphone IP ;
- d'un téléphone analogique (POT) à un téléphone analogique (POT).

Les appels entre deux téléphones IP sont routés par le serveur Asterisk :

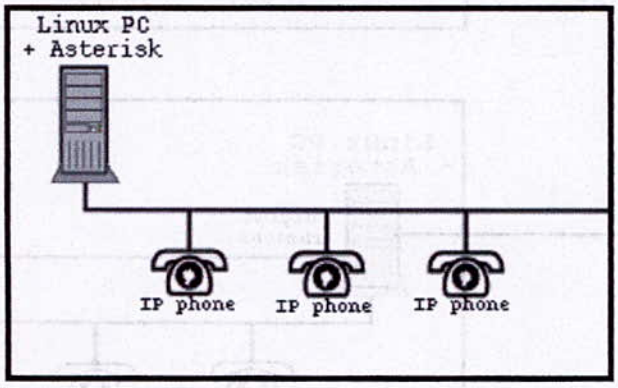


Figure3. 2 Communication de téléphone IP à téléphone IP

Les téléphones analogiques peuvent également être appelés par un téléphone IP. Pour cela, le serveur Asterisk doit être équipé d'une carte FXS telle que la carte Digium TDM400P avec les options correspondant au nombre de lignes : TDM10B, TDM20B, TDM30B, TDM40B.

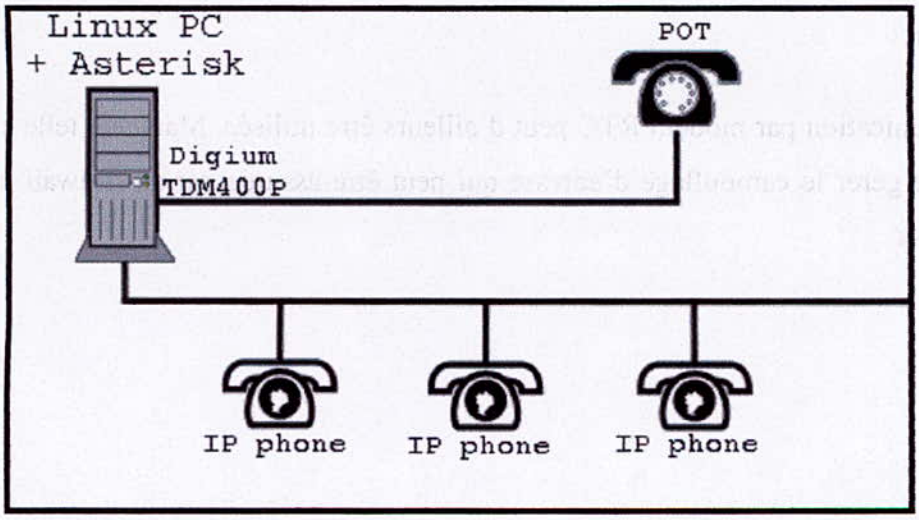


Figure3. 3 Communication de téléphones IP et POT(S)

3.5.2. Connexion de réseaux distants

Dans cette architecture, deux sites distants échangent leurs communications téléphoniques via internet.

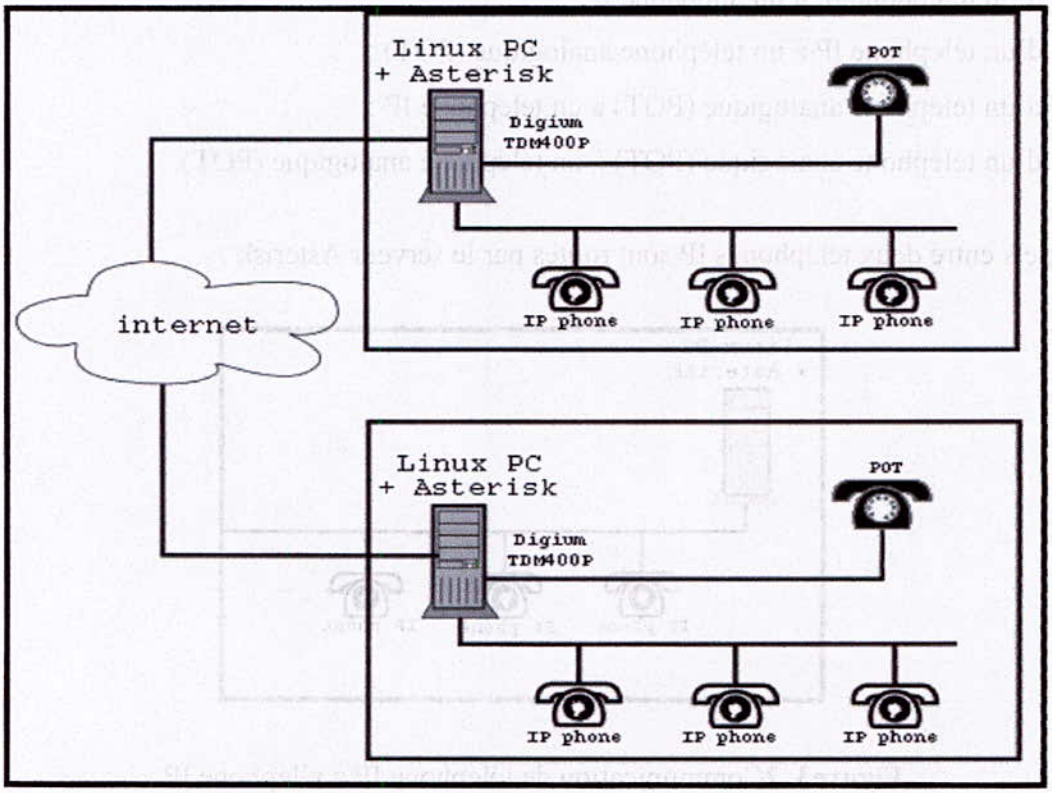


Figure3. 4 Communication entre deux réseaux distants

La connexion Internet peut être une ligne ADSL qui doit présenter un bon niveau de qualité de service et de régularité de la bande passante. La largeur de la bande passante n'a pas besoin d'être très important car les Codecs pour la voix sur IP atteignent de très forts taux de compression.

Une communication par modem RTC peut d'ailleurs être utilisée. Mais une telle architecture nécessite de gérer le camouflage d'adresse qui peut être assurée par un firewall en parallèle avec Asterisk

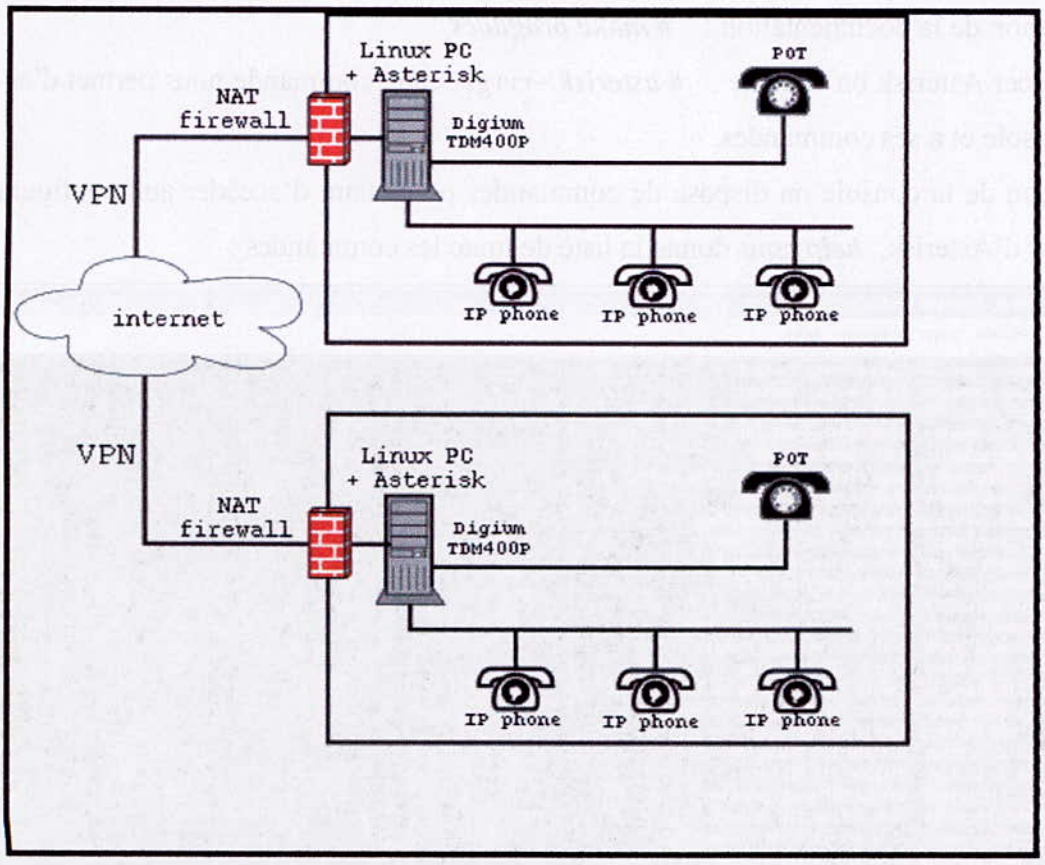


Figure3. 5 Architecture sécurisée entre deux sites

Les deux sites distants sont reliés par téléphonie IP sécurisée sur un réseau VPN et sont protégés par un firewall.

NAT (Network Adress Translation) est un traducteur d'adresses réseau, il est utilisé pour protéger les réseaux privés des intrusions d'Internet. Plusieurs entreprises utilisent les adresses privées dans leur réseau interne ce qui rend la communication à travers Internet pratiquement impossible. La technique de traduction NAT intervient pour faire correspondre à chaque adresse privée une adresse publique et à ce moment là la communication à travers Internet peut être établie.

4.6. Configuration d'ASTERISK

Asterisk est un logiciel open source, son installation se fait sous linux au niveau du Shell à partir du répertoire ou il se trouve en exécutant les commandes suivantes :

Compilation: ***# make***

Installation: ***# make install***

Installation des exemples : ***# make samples***

Installation de la documentation : ***# make progdocs***

Pour lancer Asterisk on exécute : ***# asterisk –vvvgc***, cette commande nous permet d'accéder à la Console et à ses commandes.

Au niveau de la console on dispose de commandes permettant d'accéder aux configurations actuelles d'Asterisk, ***help*** nous donne la liste de toute les commandes :

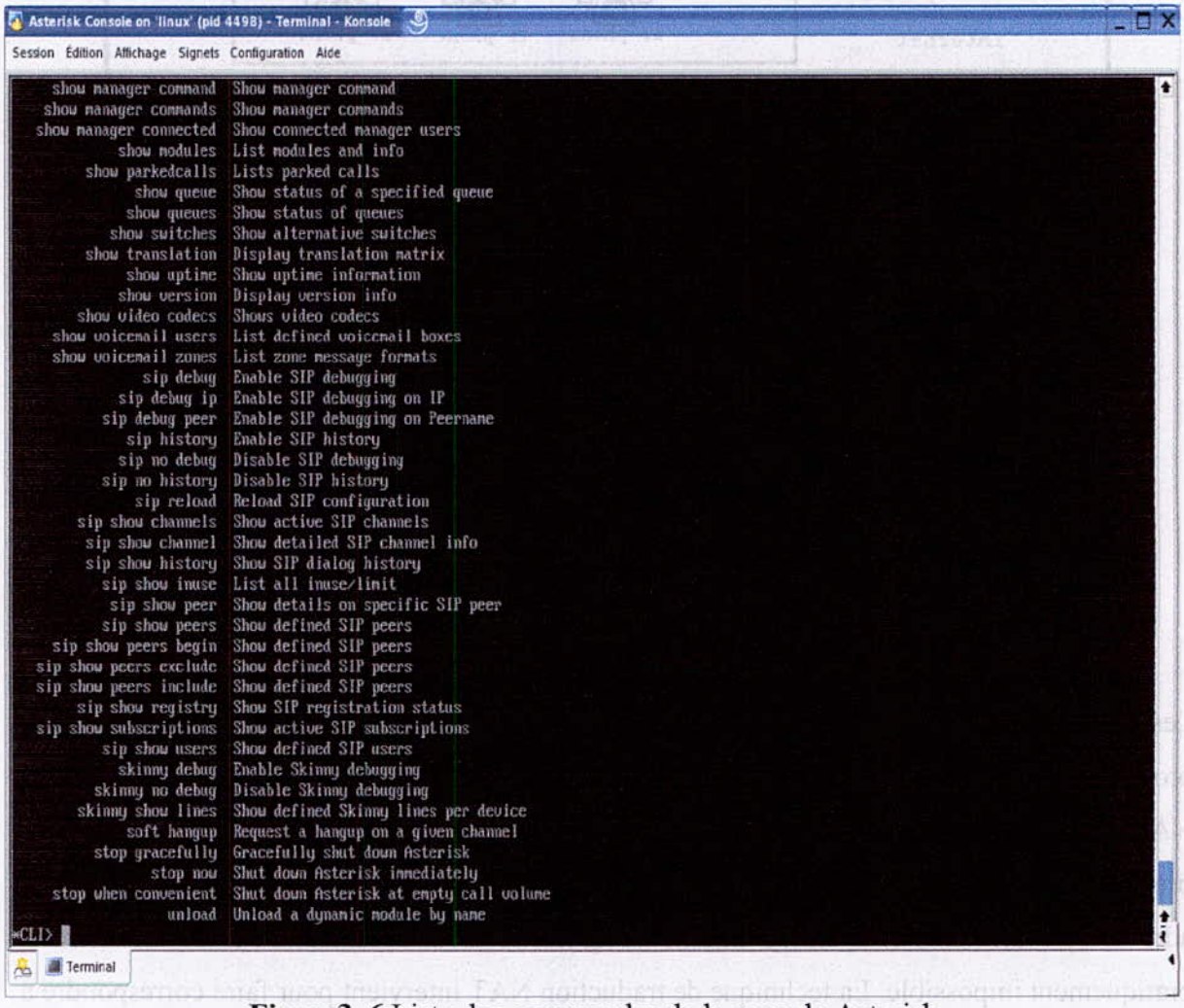
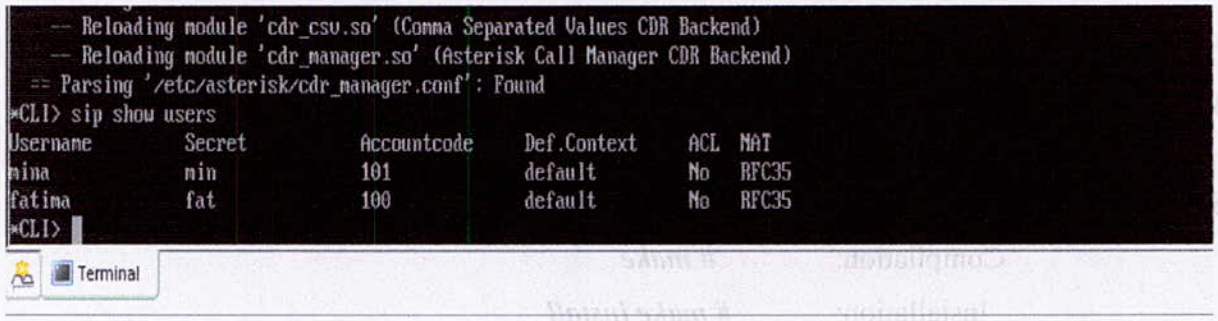


Figure3. 6 Liste des commandes de la console Asterisk

Par exemple ***sip show users*** nous affiche les informations concernant les utilisateurs de SIP :



La commande ***# asterisk*** permet de lancer Asterisk sans ouvrir la Console ; et si Asterisk est déjà lancé la commande ***# asterisk –r*** permet d'ouvrir sa Console.

Asterisk se configure à partir des fichiers de configurations qui se chargent pendant l'installation dans le répertoire ***'etc/asterisk',*** on dispose d'une structure spécifique qui nous permet de les éditer.

Les fichiers de configurations peuvent généralement être divisés en trois catégories:

Configuration d'interface - ce type de fichiers configure typiquement des interfaces de canal, ce qui est directement lié au matériel tel : alsa.conf, modem.conf, oss.conf, phone.conf, et zapata.conf.

Groupes simples - ce type de fichiers définit l'existence de diverses entités simples, comme les boîtes vocales, les salles de conférence, c à d configure les applications d'Asterisk par exemple : extensions.conf, meetme.conf, musiconhold.conf, parking.conf, et voicemail.conf.

Différentes entités - ces fichiers détaillent un certain nombre d'entités typique indépendantes, comme les clients et les serveurs. Ces fichiers sont utilisés le plus souvent pour des services de VoIP. On cite : iax.conf, oh323.conf, et sip.conf.

Nous allons voir dans ce qui suit les fichiers de configurations les plus usuels.

3.6.1. Le Dialplan

Le Dialplan est la partie exécutive des systèmes Asterisk, dans laquelle ce fait le routage des appels entrants vers les différentes applications d'Asterisk. C'est aussi grâce au Dialplan que le client peut accéder aux menus lui permettant de choisir telle ou telle application, ce choix se fait à travers les IVR (Interactive Voice Response) dont dispose Asterisk.

Le Dialplan est édité dans le fichier de configuration ***extension.conf*** qui se situe dans le répertoire ***etc/asterisk***.

Globalement, le Dialplan est un programme principal ayant une structure spécifique: il englobe des contexts et chaque context est une liste d'extensions qui permettent d'exécuter des applications.

Nous apporterons plus de détails concernant cette structure dans les paragraphes qui suivent.

3.6.1.1. Les contexts et les extensions

Le context est l'élément qui permet de composer le Dialplan et d'organiser le routage des appels. Puisque il offre à son administrateur de router les appels différemment selon leurs destinations, leurs sources, la technologie qui les supportent ou même l'heure à laquelle ils

arrivent. Donc définir plusieurs contexts est un moyen de sélectionner et diversifier le traitement des appels.

Un context est défini par son nom mis entre crochets : ***[local]***, et par une liste d'instructions qui traduisent des fonctions. Le début d'un context marque la fin du précèdent.

Les extensions sont les instructions que contient un context; elles assurent le déroulement d'un appel en exécutant des applications suivant l'ordre de leurs priorités.

.La syntaxe d'une extension est donnée comme suit :

exten => extension, priorité, application

Extension : représente le nom ou le numéro de l'extension

Priorité : représente la priorité puisque une extension peut être définie en plusieurs lignes qui s'exécutent selon leurs priorité.

Application : est l'application à exécuter.

Example d'extension:

exten => 100, 1, Answer ()

exten => 100, 2, playback (welcome)

exten => 100, 3, Hangup ()

Certains noms d'extensions sont réservés à des situations générales :

s : est l'extension de "début". Un appel qui n'a pas d'extension attribuée ; commence à cette extension.

t : est l'extension d'"arrêt". Quand un client est dans un menu de voix et n'entre aucun des chiffres du menu, l'extension d'arrêt est exécutée.

i : est l'extension "invalide". Quand un client compose une extension invalide, l'extension 'i'est exécutée.

o : est l'extension de l'"opérateur".

h : est l'extension "hangup". Cette extension est exécutée à la fin de l'appel.

Les noms des extensions peuvent contenir des nombres, des lettres, ou les symboles *, et #.

Si l'extension est précédée par "_", le nombre est considéré comme un 'pattern match' où :

'N' : représente n'importe quel chiffre 2-9,

'X' : représente n'importe quel chiffre 0-9,

'.' : représente toutes les extensions.

Exemple d'extension pattern match :

```
[local]
exten => _1XX, 1, Dial (SIP/100)
exten => _1XX, 2, Voicemail (b100)
exten => _1XX, 3, Hangup
```

Le context local permet de router tous les appels à destination des numéros ayant 3 caractères et commençant par 1 (123, 100,...). Cette option nous permet de définir un context et des services pour chaque département.

3.6.1.2. Les Macros

Comme nous l'avons vu, la logique des extensions s'avère très flexible mais elle peut présenter des redondances lors de création d'extension similaires pour différentes variables. Il existe une possibilité de réduire ces redondances en utilisant les Macros.

Les macros sont des contexts dont le nom commence par [macro-nom] ; ce sont des fonctions auxquelles on fait appelle dans un context en fonction d'arguments suivant la syntaxe :

exten => 100, 1, Macro (nom ${ARG1})

Exemple de macros :

```
[macro-message]
exten => s, 1, Voicemail (b${ARG1})
exten => s, 2, playback (vm-goodbye)
exten => s, 3, Hangup
[local]
exten => _1XX, 1, Dial (SIP/${EXTEN})
exten => _1XX, 2, Macro (message, ${EXTEN})
exten => _1XX, 3, Hangup
```

3.6.2. Configuration du canal SIP

Sip.conf est le fichier de configuration dans lequel est définit tous ce qui se rapporte au protocole SIP. A commencer par le port utilisé, les utilisateurs SIP et les Proxy SIP.

Il comporte une section [general] où sont définis les configurations générales :

Port : est le port que réserve Asterisk pour les connexions SIP qui arrivent par défaut sur le port 5060.

Bindaddr : est l'adresse IP sur laquelle Asterisk devrait être à l'écoute des requêtes SIP arrivant sur la même adresse.

Contexte : est le contexte par défaut ou tout client est placé.

Register : Enregistre un client dans un autre serveur SIP ou un SIP Proxy.

Allow : autorise un SIP codec

Disallow : exclut l'utilisation d'un SIP codec.

```
[general]
port = 5060
bindaddr = 172.16.0.1
context = default
disallow = g729
allow = ulaw
allow = gsm
register => 1234@mysipprovider.com/1234
register => 2345@myothersipprovider.com
```

Après avoir défini les paramètres généraux on passe a la définition de sections concernant les utilisateurs qui se divisent en trois catégories selon le type.

Type : détermine la classe de la connexion du client, pouvant être soit :

Peer : une entité vers laquelle Asterisk route des appels, un SIP provider par exemple.

User : une entité qui peut seulement placer des appels a travers Asterisk.

Friend : une entité pouvant être peer et user au même temps.

Host : l'adresse IP correspondante à l'utilisateur, elle peut etre statique – une valeur fixe- ou bien dynamique c à d que l'utilisateur peut s'enregistré avec n'importe quelle adresse du réseau et cette option convient parfaitement au réseaux DHCP.

Defaultip : Cette option peut être employée quand l'option ***host = dynamic***, pour router des appels vers un client qui ne s'est pas enregistré au niveau du serveur Asterisk.

Username : le nom de l'utilisateur.

Context : router les appels reçus vers le context choisi pour cette option.

Dtmfmode : sélectionne le mode de transmission des réponses DTMF : inbande, rfc2833, info.

Mailbox : ou est définie le numéro de la boite vocale ; plusieurs numéros peuvent êtres définis.

Secret : est le mot de passe de l'utilisateur servant à l'authentification des ces derniers lors de l'ouverture de leurs sessions.

Nat : sert lorsque les utilisateurs de type peer – SIP Proxy, Serveur- sont entres des pare-feu.

Exemple de configuration de sip.conf :

```
[general]
port=5060
bindaddr=172.16.0.10
context=default
register => 1234@mysipprovider.com
[directeur]
type=user
secret=direc
host=dynamic
defaultip=172.16.0.5
mailbox=2345,1234
context=local
```

4.7. ACTOS (Asterisk Configuration Tool Open Source

Le fonctionnement d'Asterisk est contrôlé par de nombreux fichiers de configuration. Le paramétrage d'Asterisk s'effectue en modifiant ces fichiers, seulement cette tache s'avère fastidieuse vu le nombre important de paramètres qu'englobe chacun d'eux. Afin de pallier à cette difficulté, une interface graphique ACTOS a été développée permettant ce paramétrage sans avoir à connaitre la syntaxe et les options de chaque fichier.

La version actuelle d'ACTOS (2.1) permet d'éditer quelques fichiers de configurations ;

- la configuration des paramètres des protocoles : iax.conf ; sip.conf ; agents.conf.
- le dialplan en ajoutant des contextes, des extensions, des variables,… extensions.conf
- le mode conférence et les salles de conférence; meetme.conf
- la file d'attente –en cours de développement- : queue.conf

ACTOS peut importer une configuration existante, ou exporter une configuration dans le format standard ASTERISK.

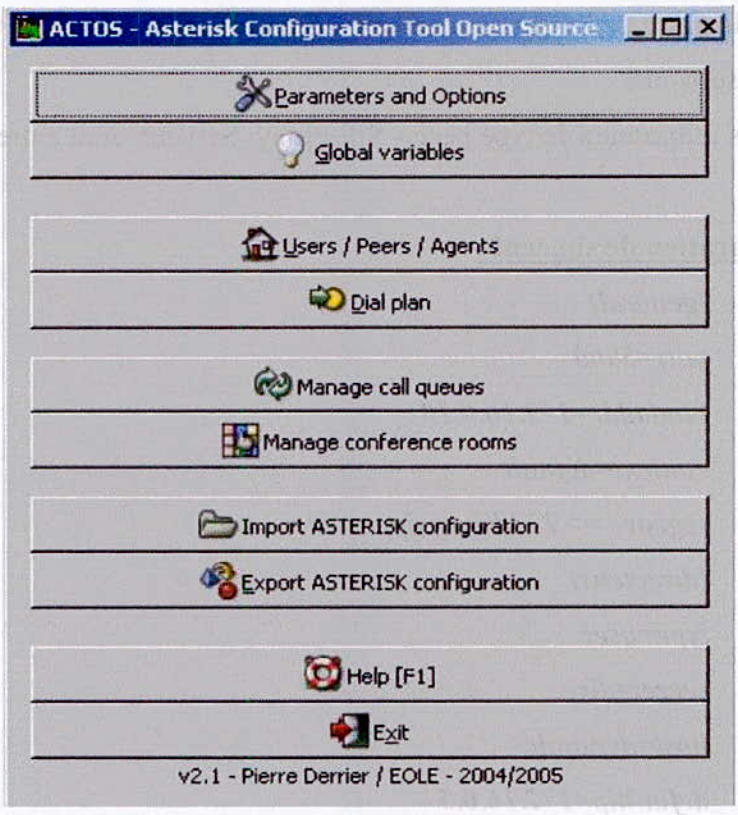


Figure3. 7 L'interface ACTOS

Pour exporter et mettre à jour la configuration depuis une machine distante il suffit d'installer ACTOS_SERVER sur une machine ASTERISK

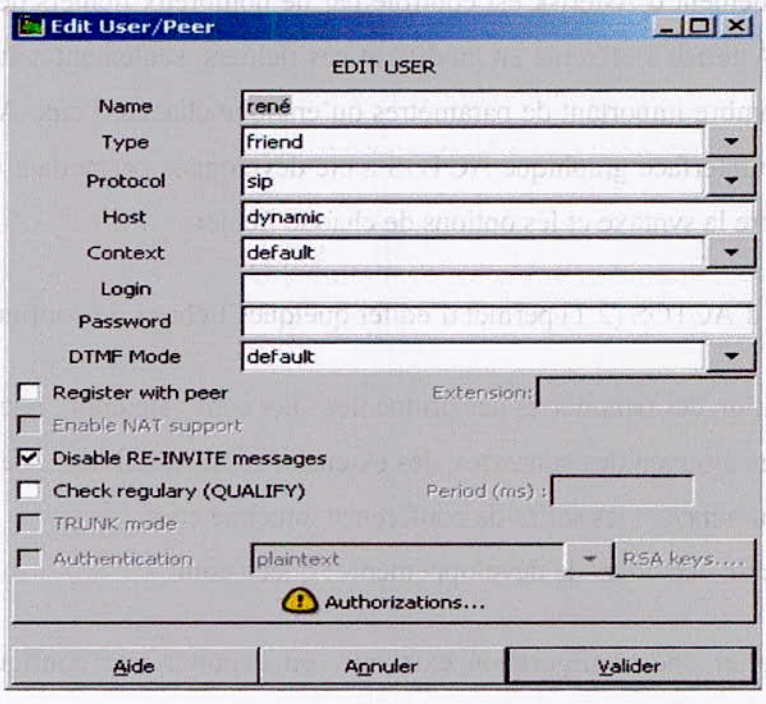


Figure3. 8 Editer des utilisateurs Asterisk

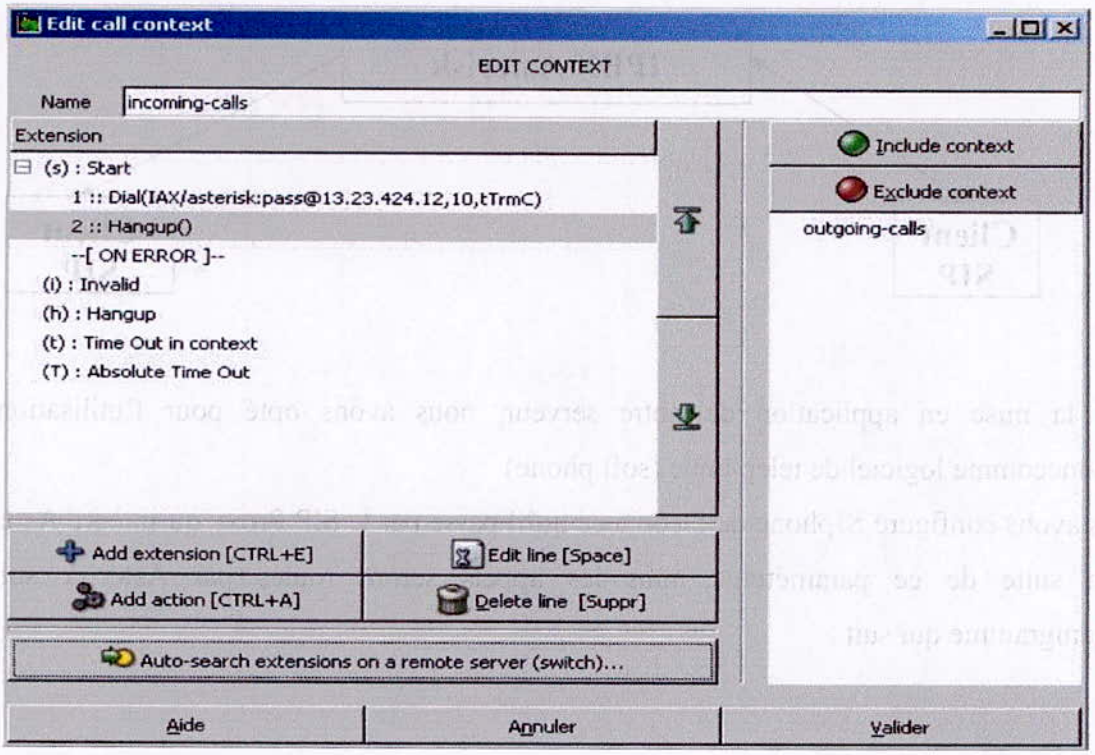
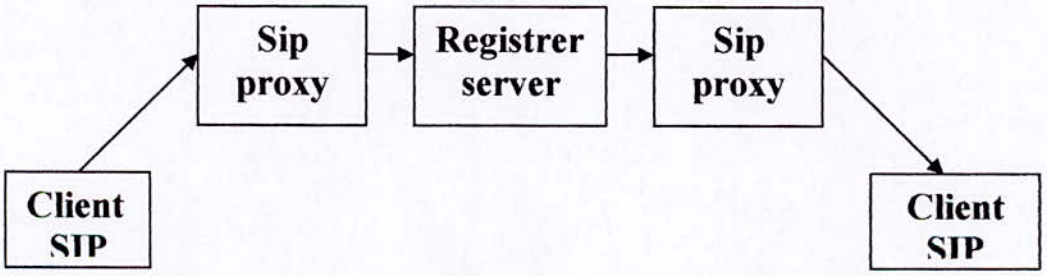


Figure3. 9 Editer le Dialplan.

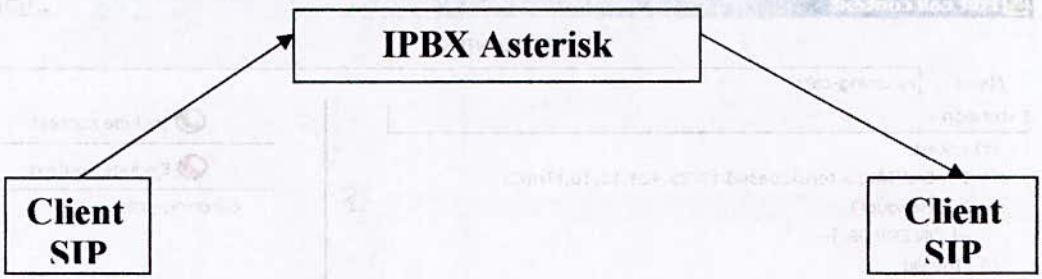
3.8. Mise en œuvre d'un IPBX ASTERISK

Notre application consiste en la mise en œuvre d'un IPBX Asterisk dans un réseau local utilisant le type de connexions supporté par le protocole SIP.

Pour assurer la communication entre utilisateurs SIP dans un réseau, il faut que celui-ci dispose d'un SIP proxy et d'un registrer server. Le proxy permettant l'aiguillage vers le registrer qui enregistre le client puis aiguille l'appel vers la destination à travers le proxy ; selon le schéma :



Dans notre cas notre réseau se limitera au serveur Asterisk puisqu'il intègre les fonctions d'un register et d'un proxy. Selon le schéma :



Pour la mise en application de notre serveur, nous avons opté pour l'utilisation de SJphonecomme logiciel de téléphonie (soft phone)

Nous avons configuré SJphone de façon à ce qu'il passe par le SIP Proxy qu'intègre Asterisk.

A la suite de ce paramétrage, tous les appels seront routés par Asterisk suivant l'organigramme qui suit :

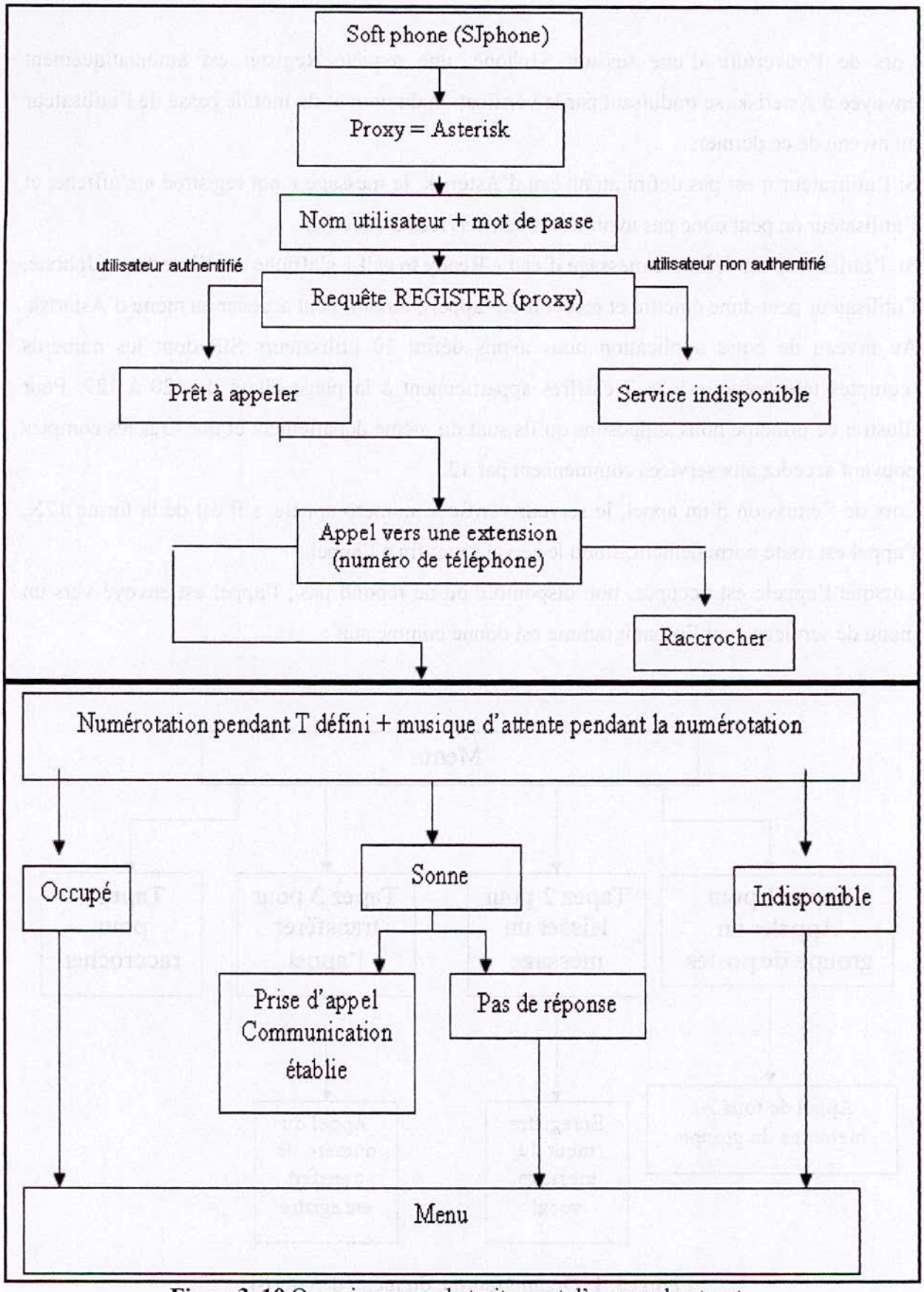


Figure3. 10 Organigramme de traitement d'un appel entrant.

Lors de l'ouverture d'une session SJphone, une requête Register est automatiquement envoyée à Asterisk, se traduisant par la vérification du nom et du mot de passe de l'utilisateur au niveau de ce dernier.

Si l'utilisateur n'est pas défini au niveau d'Asterisk, le message « not registred » s'affiche, et l'utilisateur ne peut donc pas avoir accès aux services d'Asterisk

Si l'utilisateur est défini, le message d'état « Ready to call » s'affiche sur l'interface SJphone, l'utilisateur peut donc émettre et recevoir des appels, aussi il peut accéder au menu d'Asterisk

Au niveau de notre application nous avons défini 10 utilisateurs SIP dont les numéros (comptes téléphoniques) de 3 chiffres appartiennent à la plage allant de 120 à 129. Pour illustrer ce principe nous supposons qu'ils sont du même département et que tous les comptes pouvant accéder aux services commencent par 12.

Lors de l'émission d'un appel, le serveur vérifie le numéro appelé, s'il est de la forme 12X, l'appel est routé normalement, sinon le serveur met fin à l'appel.

Lorsque l'appelé est occupée, non disponible ou ne répond pas ; l'appel est envoyé vers un menu de services dont l'organigramme est donné comme suit :

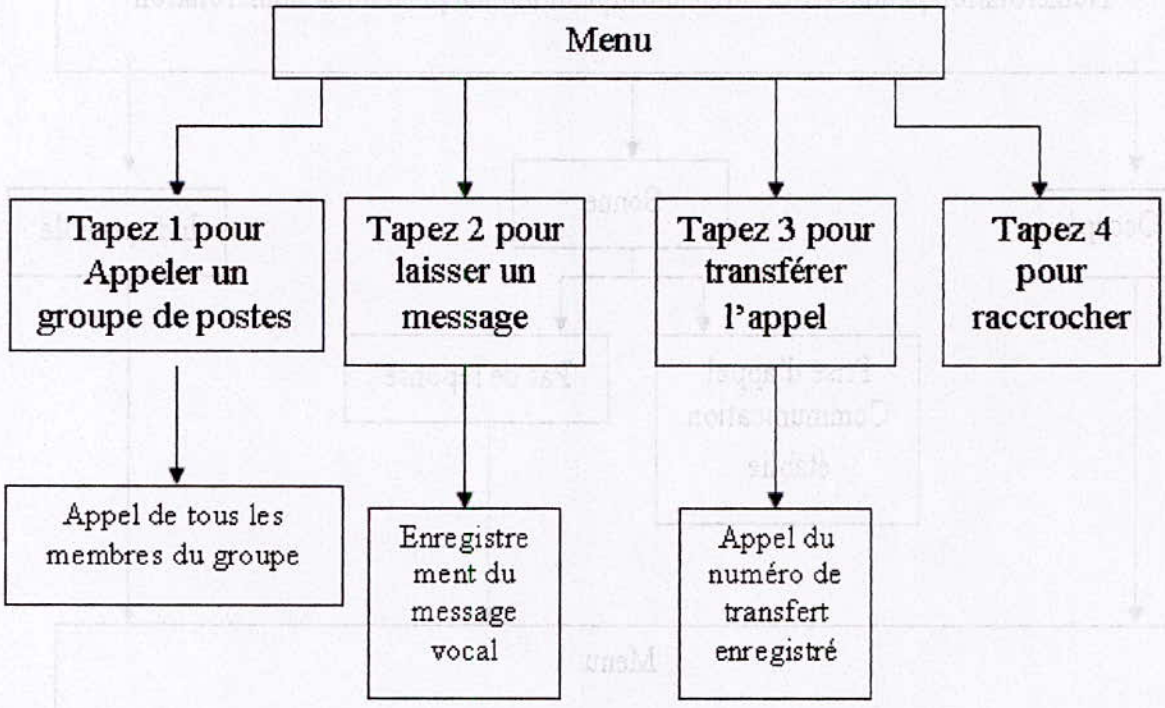


Figure3. 11 Organigramme du menu d'Asterisk

Le menu offre des services : boite vocale, file d'attente,transfert d'appel..., que l'utilisateur peut sélectionner en transmettant les chiffres ou extensions correspondant au service choisi, on parle alors de la notion d'interactivité client serveur.

En tapant 1, il est routé vers un groupe de postes défini, à ce moment là l'appel est routé vers tous les membres de ce groupe. Pour créer ces groupes nous nous sommes basés sur le fait que le groupe rassemble les employés d'un même bureau.

. Sinon s'il choisit d'être mis en file d'attente en tapant le 2, l'appel est routé vers la boite vocale de l'appelé dans le but d'y laisser un message.

Il a la possibilité de transférer l'appel vers un numéro que l'appelé a enregistré à travers un service précis que nous détaillerons plus loin dans ce chapitre.

L'utilisateur peut mettre fin à son appel en tapant 4, ou en raccrochant tout simplement.

Le service de consultation de la boite vocale (8)

L'utilisateur consulte sa boite vocale en appelant le 8 ; Asterisk répond à l'appel suivant les étapes de l'organigramme ci-dessous :

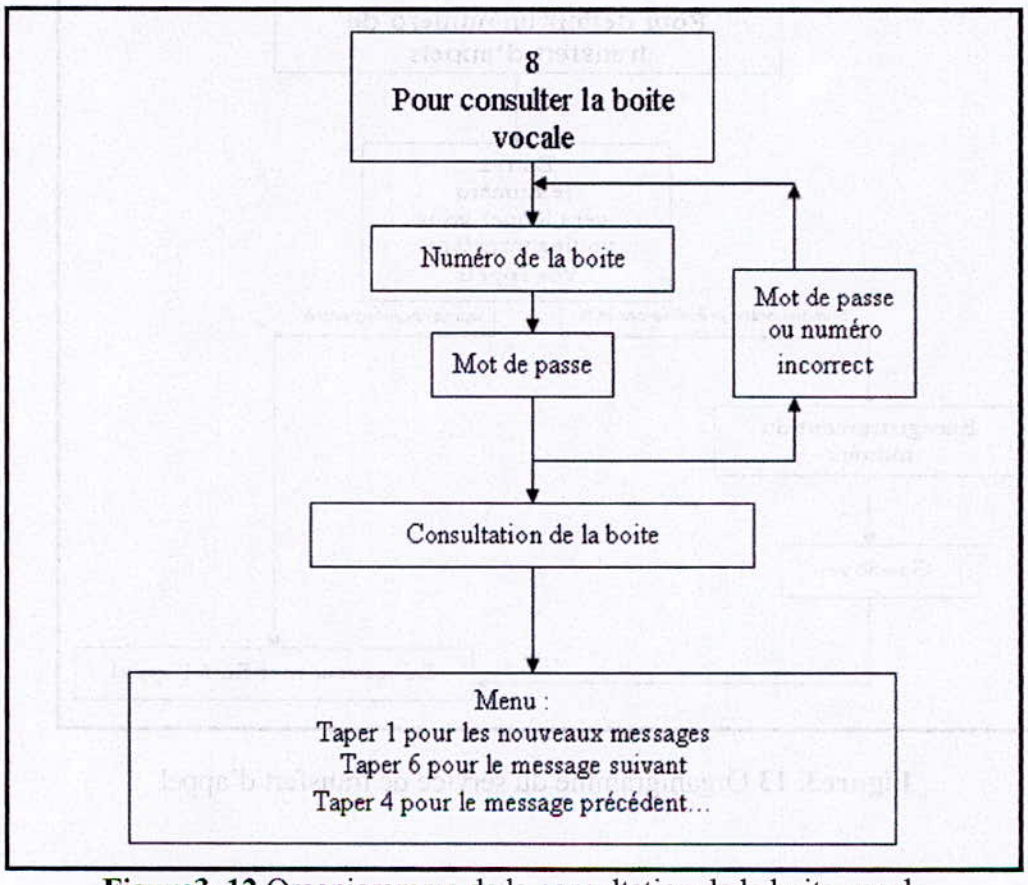


Figure3. 12 Organigramme de la consultation de la boite vocale.

La première étape est l'authentification : le serveur demande le numéro et le mot de passe de la boite vocale, s'ils sont validés l'utilisateur peut choisir d'écouter, de supprimer ou de réécouter les messages en suivant les instructions du menu.

Et si le numéro ou le mot de passe est incorrect, le serveur offre de refaire l'authentification une deuxième fois, si c'est toujours incorrect le serveur met fin à l'appel.

Pour créer les boites vocale des utilisateur il a fallu les définir dans le fichier de configuration se rapportant à l'application voicemail (voicemail.conf).

Les messages vocaux sont sauvegardés sous format gsm, il est cependant possible de les sauvegarde sous d'autres formats : gsm.

Le service de transfert d'appel (7)

Cet IPBX offre un autre service permettant aux utilisateurs de fournir un numéro vers lequel seront transférés tous les appels entrants, dans le cas ou l'appelant choisit de transférer l'appel. Ce numéro de transfert doit être précéder d'un '0' sinon il n'est pas pris en compte.

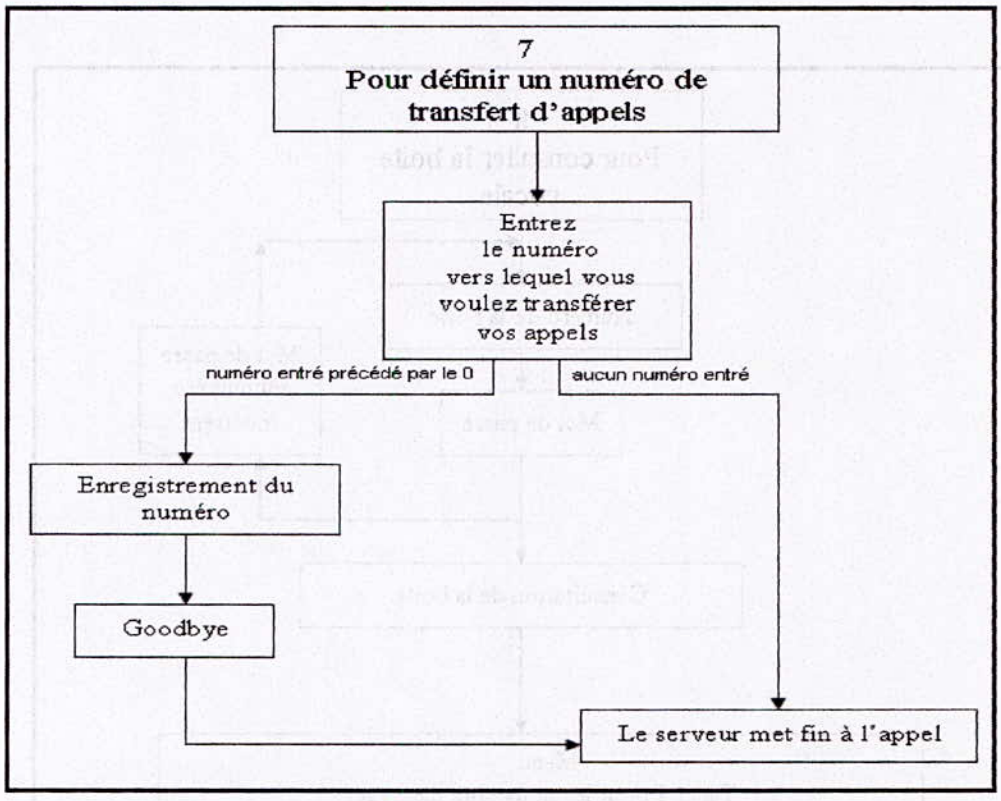


Figure3. 13 Organigramme du service de transfert d'appel

3.9. Conclusion

A travers cette application nous avons pu voir en premier lieu l'architecture interne de l'IPBX Asterisk ; et les différentes architectures en réseaux qui peuvent l'intégrer. Nous avons retenu qu'il présente une flexibilité remarquable vu sa capacité à supporter différentes technologies. Nous avons abordé en suite les notions principales qui nous ont permis de le configurer et de programmer les services requis.

Nous avons vu qu'il présentait de nombreuses et diverses applications que nous avons utilisées pour mettre en œuvre quelques unes des fonctionnalités des PBX à savoir : la messagerie vocale, le transfert d'appel, la file d'attente…

Conclusion générale

Conclusion générale

Au cours de notre travail portant sur l'étude et la mise en œuvre d'un IPBX, nous avons été amenées à étudier différents thèmes liés à la téléphonie.

Nous nous sommes intéressées à l'évolution des réseaux traditionnels basés sur la commutation des circuits vers les réseaux basés sur la VoIP. Ceci est justifié par notre intérêt à comprendre la mutation des PABX traditionnels vers les IPABX. Ces derniers, dont l'architecture est une architecture supportant le protocole IP et basée sur des standards tels H.323 ou (et) SIP par exemple, sont la solution pour la migration des entreprises vers la ToIP.

Une des particularités des plus intéressantes de notre approche réside dans l'exploitation d'une nouvelle philosophie consistant en le partage et la diffusion de sources d'applications informatiques. On pense particulièrement au système d'exploitation Linux, un des moteurs du libre. Asterisk dérive de cette approche.

Concernant cette application à travers laquelle nous avons tenté d'exploiter les différentes fonctionnalités qu'offre Asterisk dans un réseau local, nous nous sommes restreints à l'utilisation d'un seul protocole de VoIP (SIP) pour l'exploration la plus complète possible des fonctions offertes par cette plate forme.

Avec l'évolution actuelle des systèmes d'information des entreprises vers des organes ou services fédérateurs de l'information (on pense aux annuaires dont les deux implémentations majeures OpenLdap et ActiveDirectory), une idée naturelle est de déporter la base utilisateurs vers ces dits annuaires. Cela évitera entre autre la redondance de l'information au sein d'une l'entreprise.

Il aussi envisageable d'équiper Asterisk de cartes spécifiques qui lui permettrait de gérer des appels téléphoniques d'un réseau RTC traditionnel.

Annexes

ANNEXE A

LES CODES D'ETATS SIP LES PLUS COURANTS

Famille De codes	**Type De réponse**	**code**	**Signification**
1xx	information		Requête bien reçue,le traitement est en cours
		100	« trying »,en cours de traitement
		180	« Ringing »,sonnerie en cours
		181	L'appel est transféré
		182	« Qeued »,mise en file d'attente
		183	« Session progress »,utilisé pour diffuser des annonces réseau, c'est un peu l'equivalent du message RNIS PROGRESS
2xx	Succès		La requête a été bien reçue, comprise et acceptée
		200	OK
3xx	Redirection		Il faut effectuer une autre action pour compléter le traitementde la requête
		300	« Multiple Choices » :les diverses adresses possibles sont dans l'en-tête « contact »
		301	« Moved Permanently » :l'utilisaeur ne se trouve plus à cette adresse, la nouvelle se trouve dans l'en-tête contact
		302	« Moved Temporarily »une adresse alternative se trouve dans le champ « contact »,eventuellement avec une periode de validité
		305	« Use Proxy » :la destination doit etre jointe à travers un proxy
		380	Reservé à une future utilisationpour décrire un servicealternatif indiqué dans le corps du message
4xx	Erreur du client		La requête est malformée ou ne peut être executée par ce serveur
		400	« Bad Request »,requête mal formée
		401	« Unauthorized »,probleme d'autorisation
		402	« Payment Requiered »,ce service est payant
		403	« forbidden »,cette methode ou ces parametres sont interdits

		404	« Not Found »,la destination n'a pas ete trouvée
		405	« Method not Allowed »,cette methode n'est pas autorisée
		406	« Not Acceptable »,cette methode ou ses parametre de session ne sont pas acceptables
		407	« Proxy Authentication Requierd »,authentification requise
		408	« Requiest Timeout »,la requête a expirée
		409	« Conflect »,un conflit ne permet pas de poursuivre le traitement
		410	« Gone »,la cible de la requête est partie
		411	« Length Requierd »,il manque une indication de longueur
		413	« Requiest Message Body Too Large », corps de message trop grand
		414	« Request-URI Too Large »,adresse de requête trop grande
		415	« Insupported Media Type » ,type de média non supporté
		420	« Bad Extension »,mauvaise extension
		480	« Temporarily not Available »,temporairement non disponible
		481	« Call/Leg Transaction Does Not Exist »,
		482	« Loop Detected »,boucle detectée
		483	« Too Many Hops »,trop de sauts
		484	« Adress Incomplete »,adresse incomplete
		485	« Ambiguous »,requête ambiguë
		486	« Busy Here »,occupé à cette adresse
		487	« Request Terminated »,requête terminée
		488	« Not Acceptable Here »,une offre de session inacceptablea été reçue, par exemple dans un message UPDATE
		491	Une offre a été reçue dans un message UPDATE alors qu'une autre etait déjà en cours
5xx	Erreur du serveur		La requête a des problèmes de syntaxe ou ne peut pas être traitée sur ce serveur
		500	« Internal Server Error », erreur interne du serveur, également utilisée si une requête UPDATE est reçue avant qu'une réponse n'ait été générée à une précédente requête UPDATE
		501	« Not Implemented », non implémenté

		502	« Bad gateway », mauvaise passerelle
		503	« Service invailable », service non diponible
		504	« Gateway tomeout », delai depassé sur passerelle. Egalement utilisé si un utilisateur est en cours de notification pour accepter ou non la communication proposée, mais reçoit une nouvelle requête UPDATE, ne permettant pas de générer une réponse immédiate
		505	« SIP version not supported », version de SIP non supportée
6xx	Problème global		La requête n'est pas valable, quel que soit le serveur
		600	« Busy Everywhere », occupé partout
		603	« Decline », refus
		604	« Does not exist anywhere », n'existe nulle part
		606	« Not aceptable », la requête n'est pas acceptable

Annexe B

LES CHAMPS D'EN-TETES SIP

Champs	Utilisations
Accept	Utilisé dans les messages INVITE, OPTIONS et REGISTER qui permet d'indiquer les types de média qui seront acceptés dans la réponse à ce message.
Accept-Encoding	Conditionne la réponse car il détermine quels codages text-based y seront acceptés.
Accept-Language	Il permet au client d'indiquer au serveur quel langage à utiliser dans le corps du message de la réponse au client.
Allow	Indique les méthodes valides supportées par les entités identifiées par la requête URI.
Authorization	Champ optionnel à inclure par l'utilisateur souhaitant s'authentifier vis à vis du serveur auquel il est relié
Call-ID	Identifie une invitation précise ou tous les enregistrements d'un client particulier.
Contact	Champ pouvant apparaître dans les requêtes INVITE, ACK et REGISTER ou dans les réponses de codes 1xx, 2xx, 3xx et 485. Il fournit en général l'URL où l'utilisateur pourra être contacté

Content-Encoding	Indique le code utilisé pour écrire et lire l'en-tête d'entité. Ainsi le serveur qui reçoit le message sait quel mécanisme de décodage appliquer pour lire le Content-Type décrit ci-dessus et peut connaître le type de média utilisé. Ce champ est très utile si l'on veut compresser les en-têtes sans perdre les informations précieuses qu'ils contiennent
Content-Length	Il indique simplement la taille du Corps du message envoyé, en nombre décimal d'octets.
Content-Type	Il indique les types de média utilisés dans le Corps du message envoyé.
Cseq	Chaque requête doit obligatoirement contenir un numéro de séquence Cseq (entier non signé de 32 bits). Le Cseq initial est choisi arbitrairement par celui qui envoie la requête INVITE mais doit toujours être inférieur à 2^31. Le numéro de séquence s'incrémente d'une unité pour chaque nouvelle requête envoyéedans un dialogue (à l'exception des requêtes ACK et CANCEL).
Date	donne la date d'émission de la requête ou de la réponse. Les retransmissions possèdent la même date que la requête ou réponse d'origine.
Encryption	Ce champ spécifie si le message est crypté et suivant quel cryptage.
Expires	donne la durée au-delà de laquelle le message expire.
From	indique la personne à l'origine du message

Hide	Le client utilise ce champ lorsqu'il veut que le chemin compris dans l'En- Tête VIA soit caché aux prochains Proxy Servers que traversera le message.
Max-forwards	utilisé pour limiter le nombre de PS ou passerelles que la requête peut traverser jusqu'au prochain serveur dans le sens de l'UAC vers l'UAS (downstream).
Organization	Précise le nom de l'organisation à laquelle l'entité dont émane la requête ou la réponse appartient
Priority	Indique le niveau d'urgence de la requête, tel qu'il est perçu par le client.
Proxy-Authenticate	Ce champ doit être rempli dans une réponse Proxy Authentification Required (code 407).
Proxy-Authorization	Permet au client de s'identifier dans sa requête en destination d'un PS le lui ayant demandé.
Proxy-Requiere	Indique quels champs d'en-tête le PS supporte.
Record-Route	Ce champ permet de mémoriser un chemin pour faciliter l'acheminement de la réponse. Chaque Proxy Serveur traversé ajoute son adresse dans ce champ en début de liste. Le serveur appelé copie cette liste, sans la changer, dans l'en-tête Route de sa réponse. La première entrée est ainsi l'adresse du serveur le plus proche de celui qui répond.

Response-Key	Le client utilise cet en-tête dans sa requête pour déterminer la clé a utilisé pour crypter la réponse.
Retry-After	Ce champ n'est utilisé que dans les réponses Service Unavailable(code 503), Not Found (code 404), Busy (code 600) ou bien Decline (code 603) pour indiquer à l'emetteur de la requête quand est-ce qu'un service "normal" pourra reprendre. Il contient une date ou un nombre en décimal de secondes.
Server	Il contient les informations sur les softwares utilisés par les UAS
Subject	Résumé ou nature de l'appel qui peut permettre de le filtrer sans avoir à lire la description de la session.
Timestamp	Précise l'instant (date) où le client a envoyé la requête au serveur.
To	C'est l'adresse du destinataire. Ce champ est bien sûr obligatoire. Liste quelles configurations ne sont pas supportées par le serveur.
Insupported	Contient des informations sur le terminal de l'utilisateur (UAC/UAS) à l'origine de la requête.
Via	Contient les adresses des serveurs (PS) que traverse la requête. Permet de distinguer les deux versions de SIP (RFC 2543 et RFC 3261). Selon la RFC 3261 le champ d'en-tête <via> contient un paramètre « branch » commençant par « z9hG4bK ».

Warning	Les avertissements sont contenus dans les réponses dans le langage le plus compréhensible pour l'utilisateur.
WWW – Authenticate	Doit être inclus dans une réponse Unauthorized (code 401).

ANNEXE C:

Asterisk Applications and commands

Asterisk applications/

AbsoluteTimeout: Set absolute maximum time of call

AddQueueMember: Dynamically adds queue members

ADSIProg: Load Asterisk ADSI Scripts into phone

AgentCallbackLogin: Call agent callback login

AgentLogin: Call agent login

AgentMonitorOutgoing: Record agent's outgoing call

AGI: Executes an AGI compliant application

AlarmReceiver: Provide support for receving alarm reports from a burglar or fire alarm panel

Answer: Answer a channel if ring

AppendCDRUserField: Append to the CDR user field

Authenticate: Authenticate a user

BackGround: Play a file while awaiting extension

BackgroundDetect: Background a file with talk detect

Busy: Indicate busy condition and stop

ChangeMonitor: Change monitoring filename of a channel

ChanIsAvail: Check if channel is available

CheckGroup: CheckGroup(max[@category])

Congestion: Indicate congestion and stop

ControlPlayback: Play a file with fast forward and rewind

Cut: Cut(newvar=varname|delimiter|fieldspec)

DateTime: Says a specified time in a custom format

DBdel: Delete a key from the database

DBdeltree: Delete a family or keytree from the database

DBget: Retrieve a value from the database

DBput: Store a value in the database

DeadAGI: Executes AGI on a hungup channel

Dial: Place a call and connect to the current channel

DigitTimeout: Set maximum timeout between digits

Directory: Provide directory of voicemail extensions

DISA: DISA (Direct Inward System Access)

EAGI: Executes an EAGI compliant application

Echo: Echo audio read back to the user

EnumLookup: Lookup number in ENUM

Eval: Eval(newvar=somestring)

Exec: Exec(Appname(arguments))

Festival: Say text to the user

ForkCDR: Forks the Call Data Record

GetCPEID: Get ADSI CPE ID

GetGroupCount: GetGroupCount([groupname][@category])

Goto: Goto a particular priority, extension, or context

GotoIf: Conditional goto

GotoIfTime: Conditional goto on current time

Hangup: Unconditional hangup

HasNewVoicemail: Conditionally branches to priority + 101

HasVoicemail: Conditionally branches to priority + 101

ICES: Encode and stream using 'ices'

LookupBlacklist: Look up Caller*ID name/number from blacklist database

LookupCIDName: Look up CallerID Name from local database

Macro: Macro Implementation

MailboxExists: Check if vmbox exists

Milliwatt: Generate a Constant 1000Hz tone at 0dbm (mu-law)

Monitor: Monitor a channel

MP3Player: Play an MP3 file or stream

MusicOnHold: Play Music On Hold indefinitely

NBScat: Play an NBS local stream

NoCDR: Make sure asterisk doesn't save CDR for a certain call

NoOp: No operation

Park: Park yourself

ParkAndAnnounce: Park and Announce

ParkedCall: Answer a parked call

Playback: Play a file

Playtones: Play a tone list

Prefix: Prepend leading digits

PrivacyManager: Require phone number to be entered, if no CallerID sent

Progress: Indicate progress

Queue: Queue a call for a call queue

Random: Conditionally branches, based upon a probability

Read: Read a variable

Record: Record to a file

RemoveQueueMember: Dynamically removes queue members

ResetCDR: Resets the Call Data Record

ResponseTimeout: Set maximum timeout awaiting response

Ringing: Indicate ringing tone

SayAlpha: Say Alpha

SayDigits: Say Digits

SayNumber: Say Number

SayPhonetic: Say Phonetic

SayUnixTime: Says a specified time in a custom format

SendDTMF: Sends arbitrary DTMF digits

SendImage: Send an image file

SendText: Send a Text Message

SendURL: Send a URL

SetAccount: Sets account code

SetAMAFlags: Sets AMA Flags

SetCallerID: Set CallerID

SetCallerPres: Set CallerID Presentation

SetCDRUserField: Set the CDR user field

SetCIDName: Set CallerID Name

SetCIDNum: Set CallerID Number

SetGlobalVar: Set global variable to value

SetGroup: SetGroup(groupname[@category])

SetLanguage: Sets user language

SetMusicOnHold: Set default Music On Hold class

SetVar: Set variable to value

SIPDtmfMode: Change the dtmfmode for a SIP call

SMS: Communicates with SMS service centres and SMS capable analogue phones

SoftHangup: Soft Hangup Application

StopMonitor: Stop monitoring a channel

StopPlaytones: Stop playing a tone list

StripLSD: Strip Least Significant Digits

StripMSD: Strip leading digits

SubString: (Deprecated) Save substring digits in a given variable

Suffix: Append trailing digits

System: Execute a system command

TestClient: Execute Interface Test Client

TestServer: Execute Interface Test Server

Transfer: Transfer caller to remote extension

TrySystem: Try executing a system command

TXTCIDName: Lookup caller name from TXT record

UserEvent: Send an arbitrary event to the manager interface

Verbose: Send arbitrary text to verbose output

VoiceMail: Leave a voicemail message

VoiceMail2: Leave a voicemail message

VoiceMailMain: Enter voicemail system

VoiceMailMain2: Enter voicemail system

Wait: Waits for some time

WaitExten: Waits for some time

WaitForRing: Wait for Ring Application

WaitMusicOnHold: Wait, playing Music On Hold

Zapateller: Block telemarketers with SIT

Consol commands for Asterisk

! Execute a shell command

abort halt : Cancel a running halt

add extension : Add new extension into context

add ignorepat : Add new ignore pattern

add indication : Add the given indication to the country

add queue member : Add a channel to a specified queue

agi debug : Enable AGI debugging

agi no debug : Disable AGI debugging

database del : Removes database key/value

database deltree : Removes database keytree/values

database get : Gets database value

database put : Adds/updates database value

database show : Shows database contents

debug channel : Enable debugging on a channel

dont include : Remove a specified include from context

dump agihtml : Dumps a list of agi command in html format

exit : Exit Asterisk

extensions reload : Reload extensions and *only* extensions

help Display : help list, or specific help on a command

iax2 debug : Enable IAX debugging

iax2 no debug : Disable IAX debugging

iax2 provision : Provision an IAX device

iax2 set jitter : Sets IAX jitter buffer

iax2 show cache : Display IAX cached dialplan

iax2 show channels : Show active IAX channels

iax2 show firmware : Show available IAX firmwares

iax2 show peers : Show defined IAX peers

iax2 show peers begin : Show defined IAX peers

iax2 show peers exclude : Show defined IAX peers

iax2 show peers include : Show defined IAX peers

iax2 show provisioning : Show iax provisioning

iax2 show registry : Show IAX registration status

iax2 show stats : Display IAX statistics

iax2 show users : Show defined IAX users

iax2 trunk debug : Request IAX trunk debug

include context : Include context in other context

init keys : Initialize RSA key passcodes

load : Load a dynamic module by name

local show channels : Show status of local channels

logger reload : Reopens the log files

logger rotate : Rotates and reopens the log files

mgcp audit endpoint : Audit specified MGCP endpoint

mgcp debug : Enable MGCP debugging

mgcp no debug : Disable MGCP debugging

mgcp reload : Reload MGCP configuration

mgcp show endpoints : Show defined MGCP endpoints

no debug channel : Disable debugging on a channel

quit : Exit Asterisk

reload : Reload configuration

remove extension : Remove a specified extension

remove ignorepat : Remove ignore pattern from context

remove indication : Remove the given indication from the country

remove queue member : Removes a channel from a specified queue

restart gracefully : Restart Asterisk gracefully

restart now : Restart Asterisk immediately

restart when convenient : Restart Asterisk at empty call volume

save dialplan : Save dialplan

set debug : Set level of debug chattiness

set verbose : Set level of verboseness

show agents : Show status of agents

show agi : Show AGI commands or specific help

show applications : Shows registered applications

show application : Describe a specific application

show audio codecs : Shows audio codecs

show channels : Display information on channels

show channel : Display information on a specific channel

show codecs : Shows codecs

show codec : Shows a specific codec

show config handles : Show Config Handles

show dialplan : Show dialplan

show file formats : Displays file formats

show image codecs : Shows image codecs

show image formats : Displays image formats

show indications : Show a list of all country/indications

show keys : Displays RSA key information

show manager command : Show manager command

show manager commands : Show manager commands

show manager connected : Show connected manager users

show modules : List modules and info

show parkedcalls : Lists parked calls

show queue : Show status of a specified queue

show queues : Show status of queues

show switches : Show alternative switches

show translation : Display translation matrix

show uptime : Show uptime information

show version : Display version info

show video codecs : Shows video codecs

show voicemail users : List defined voicemail boxes

show voicemail zones : List zone message formats

sip debug : Enable SIP debugging

sip debug ip : Enable SIP debugging on IP

sip debug peer : Enable SIP debugging on Peername

sip history : Enable SIP history

sip no debug : Disable SIP debugging

sip no history : Disable SIP history

sip reload : Reload SIP configuration

sip show channels : Show active SIP channels

sip show channel : Show detailed SIP channel info

sip show history : Show SIP dialog history

sip show inuse : List all inuse/limit

sip show peer : Show details on specific SIP peer

sip show peers : Show defined SIP peers

sip show peers begin : Show defined SIP peers

sip show peers exclude : Show defined SIP peers

sip show peers include : Show defined SIP peers

sip show registry : Show SIP registration status

sip show subscriptions : Show active SIP subscriptions

sip show users : Show defined SIP users

skinny debug : Enable Skinny debugging

skinny no debug : Disable Skinny debugging

skinny show lines : Show defined Skinny lines per device

soft hangup : Request a hangup on a given channel

stop gracefully : Gracefully shut down Asterisk

stop now : Shut down Asterisk immediately

stop when convenient : Shut down Asterisk at empty call volume

unload : Unload a dynamic module by name

Références bibliographiques

[1] Bill Douskalis (1999). *IP Telephony : The integration of robust VoIP Services.* Prentice Hall.

[2] Olivier Hersent, David Gurle, Jean-Pierre Petit(2004). *La VOIX sur IP*

[3] International Engineering Consortium. *The future of Voice/Data Consolidation : Markets, Technologies and Strategies*,

[4] International Engineering Consortium, *Accelerating the deployment of Voice Over IP (VoIP) and Voice Over ATM (VoATM)* , [En ligne] http://www.iec.org/online/tutorials/voip_voatm/index.html

[5] D.Black Uyless (1999). *Voice Over IP*, Prentice Hall,.

[6] Jean-François Susbielle (1996). *Téléphonie sur Internet*. Paris : Eyrolles.

[7] www.PABX-FR.com

[8] D.Minoli and E.Minoli (1998). *Delivering Voice Over IP Networks*, John Wilew,.

[9] H Schulzrinne, S. Casner, R.Frederick, V.Jacobson, (1996). RTP: A Transport Protocol for Realtime Applications, RFC: 1889, *Internet Engineering Task Force*.

[10] R. Stevens, G.R. Wright, TCP/IP Illustrated, Volume 2, Addison Wesley, 1995.

[11] Mourad El-Allia,
" DÉVELOPPEMENT D'UN ENVIRONNEMENT DE COMMUNICATION MULTIMÉDIA (VOIX ET VIDÉO) SUR INTERNET "
http://www.livia.etsmtl.ca/publications/2002/Memoire_Mourad.pdf

[12] Shweizer Laurent, (2001). *Tutorial sur le protocole VoIP, SIP* , [En ligne]..
http://www.tcom.ch/Etudiants/2001/lschweizer/sip.pdf

[13] Rapport de Jean-Claude Merlin, La téléphonie sur Internet, Avril 1999, *Conseil général des technologies de l'information* [En ligne].
http://www.telecom.gouv.fr/documents/merlin/rap_merlin0499_2.htm#d

[14] Henning Schulzrinne, (1999) .*Session Initiation Protocol (SIP),*. [En ligne].
http://www.cs.columbia.edu/~hgs/sip/

[15] Jonathan Davidson, Jim Peters(1999). *Voice Over IP Fundamentals, Macmillan*

[16] RFC 3261,RFC 2833

[17] Techniques de l'ingénieur ,article TE7630

[18] http://www.voip.org

[19] http://www.asterisk.org

Résumé

Ce projet a pour objet l'étude et la mise en œuvre d'un IPBX sous le système d'exploitation Linux. Pour cela l'étude des principes de la voix sur IP et de ses protocoles est fondamentale. L'utilisation d'Asterisk qui est un IPBX Open Source a permis la communication VoIP dans un réseau local selon un plan à définir, et l'offre de services aux utilisateurs tel : la messagerie vocale, le transfert d'appel,...

Mots clés : ***VoIP, SIP, IPBX, Asterisk.***

Abstract

The aim of this project is the study and the use of an IPBX under the Linux Operating System. For this, the knowledge of the VoIP fundamental principles and protocols. The use of the Open Source IPBX Asterisk, allows us the integration of VoIP communications in a local network according to a plan that we define. It also offers different services such: voice mail, call transfer.

Key words: ***VoIP, SIP, IPBX, Asterisk.***

الملخص

الهدف من هذا العمل يكمن في دراسة و استعمال IPBX تحت نظام التشغيل Linux . لهذا دراسة مبادئ إرسال الصوت عبر بروتوكول الأنترنات VoIP ضروري. استعمال Asterisk يسمح باتصالات VoIP في شبكة محلية وذلك تبعا لمخطط نعرفه، كم يسمح بمنح خدمات للمستعملين مثل: تحويل الاتصالات و البريد الصوتي...

الكلمات المفتاحية: ***VoIP, SIP, IPBX, Asterisk***