

Cycle de formation des ingénieurs en Télécommunications

Option :

Ingénierie des Réseaux

Rapport de Projet de fin d'études

Thème :

Réalisation d'une interface de communication entre PBXs via AstérisK

Réalisé par :

Rabiaa GUEMRI

Encadrants :

M. Rached HAMZA
M. Slah DABOUZIA
M. Ali ATALLAH

Travail proposé et réalisé en collaboration avec



Tunisian Telecom Electric

Année universitaire : 2006/2007

A tous ceux que j'aime et qui m'aiment...

Rabíaa

Avant propos

Ce document s'inscrit dans le cadre d'un projet de fin d'études en Ingénierie des Réseaux (IRES). Il présente mon travail réalisé au sein de Tunisian Telecom Electric (TTE).

Ce projet est intitulé « Réalisation d'une interface de communication entre PBXs via AstérisK».

Je tiens à remercier le personnel de Tunisian Telecom Electric (TTE), le siège de mon projet de fin d'étude, qui m'ont accueillie parmi eux et m'ont offert l'opportunité de travailler sur ce projet fort intéressant, en particulier à M. Ali ATALLAH et M. Slah DABOUZIA pour leur encadrement, leur disponibilité, leurs conseils généreux et pour le soutien constant qu'ils n'ont cessé de me prodiguer.

Je tiens également à remercier mon encadreur M. Rached Hamza qui a fait preuve d'un grand esprit de collaboration et d'initiative.

Je remercie infiniment tous les enseignants et les cadres administratifs de Sup'Com pour leur contribution à ma formation.

Table des matières

Introduction générale	1
Chapitre I : VoIP et protocoles associés	3
Introduction	3
I. Les avantages de la VoIP :.....	3
1. Réduction des coûts	4
2. Standards ouverts et interopérabilité multifournisseurs.....	5
3. Choix d'un service opéré.....	5
4. Consommation de bande passante moins importante	5
5. Un réseau voix, vidéo et données (Triple Play)	5
6. Un service PABX distribué ou centralisé.....	6
7. Intégration des services vidéo	6
II. Principe de fonctionnement de la voix sur IP	6
1. Mode de fonctionnement	6
2. Principaux codecs utilisés	8
III.Principaux protocoles de la VoIP.....	8
1. Le protocole H323	9
2. Le protocole SIP	10
3. Comparaison entre SIP et H323.....	16
4. Le protocole RTP	17
7. Le RTCP	18
Conclusion	19
Chapitre II : Etude et choix de la solution	20
Introduction	20
I. Les IPBX matériels	20
II. Les IPBX logiciels : Asterisk	22
1. Caractéristiques	23
2. Installation	24
3. Emplacement des fichiers de configuration.....	26
4. Mode de fonctionnement	26
III.Choix de la solution.....	29
Conclusion	33
Chapitre III : Interconnexion des PBX via AstérisK	34
Introduction	34
I. Configuration des terminaux	34
1. Configuration des postes IP ST2030.....	35
2. Softphones	36
3. Téléphone analogique avec adaptateur ATA PAP2 de linksys.....	37
4. Adaptateur téléphonique SN4528	39
II. Configuration d'AstérisK	40
1. Sip.conf	40
2. Extensions.conf.....	43
III.Interconnexion des ipLDK via Asterisk.....	45
1. Configuration de l'ipLDK.....	46
2. Configuration d'AstérisK	50
Conclusion	53

Chapitre IV : performances de la solution mise en place	54
Introduction	54
I. Configuration des services	54
1. Voicemail	54
2. Transfert d'appel.....	59
3. Conférence téléphonique.....	63
II. Tests et qualité de service	64
1. Indicateurs à évaluer	64
2. Tests sur un appel.....	66
3. Test sur 50 appels	69
Conclusion	72
Conclusion générale.....	73
Annexe : commandes de base d'astérisque.....	74
Bibliographie.....	77

Liste des figures

Figure 1. 1 : La VoIP : une infrastructure moins onéreuse.....	4
Figure 1. 2: Convergence du Triple Play : voix, données et vidéo.....	6
Figure 1. 3: Processus de la VoIP	7
Figure 1. 4 : Protocoles secondaires ou associés	10
Figure 1. 5 : Enregistrement d'un utilisateur	14
Figure 1. 6 : Etablissement d'une communication avec un proxy SIP.....	14
Figure 1. 7 : Etablissement d'une session SIP à travers un Proxy.....	15
Figure 2. 1: Diagramme de connexion du système.....	21
Figure 2. 2 : Configuration d'une interconnexion des IP-LDK via IP.....	29
Figure 2. 3 : Schéma descriptif de l'application à réaliser	32
Figure 3. 1: Interconnexion des différents types de terminaux via Astérisik	34
Figure 3. 2 : Configuration de base de ST2030	35
Figure 3. 3 : Configuration des paramètres réseau	36
Figure 3. 4 : Configuration de X-lite sous Linux	36
Figure 3. 5: Configuration de X-lite sous Windows.....	37
Figure 3. 6 : Configuration de base d'une ligne de L'adaptateur Linksys	38
Figure 3. 7 : Configuration des paramètres réseau de Linksys	38
Figure 3. 8: Les types d'utilisateurs.....	42
Figure 3. 9: Enregistrement d'un client sous Astérisik.....	43
Figure 3. 10: Exécution d'un appel sous Astérisik	45
Figure 3. 11 : Schéma de travail demandé.....	46
Figure 3. 12 : Disposition des différentes cartes sur le LDK.....	47
Figure 3. 13 : Configuration de la carte MPB	47
Figure 3. 14 : Configuration de la carte VOIB	48
Figure 3. 15 : Configuration du proxy sip	49
Figure 3. 16 : Déclaration des paramètres d'authentification	49
Figure 3. 17 : Déclaration de la plage de numérotation	50
Figure 3. 18 : Enregistrement des ipLDK au niveau d'Astérisik.....	51
Figure 3. 19: Déroulement de communication entre deux postes appartenant à des ipLDK différents	52
Figure 3. 20 : Déroulement de communication entre un poste IP et un poste d'un LDK... ..	52
Figure 4. 1: Déroulement du transfert aveugle.....	60
Figure 4. 2 : Tableau de bord du déroulement de l'appel.....	67
Figure 4. 3: Valeurs des indicateurs MOS et R.....	67
Figure 4. 4 : Taux de paquets RTCP perdus.....	68
Figure 4. 5 : Gigue des paquets RTCP.....	68
Figure 4. 6 : Etat du serveur astérisik à la réception de l'appel	69
Figure 4. 7: Interface de la pondeuse d'appels	69
Figure 4. 8: Déroulement de l'enregistrement des utilisateurs auprès d'astérisik	70
Figure 4. 9: Gigue des paquets RTCP	70
Figure 4. 10 : Temps de réponse moyen	71

Liste des tableaux

Chapitre 1

Tableau 1. 1: Principales motivations pour le choix de la VoIP	3
Tableau 1. 2: Principaux codecs de la VoIP	8
Tableau 1. 3: Comparaison de SIP et H.323	16

Chapitre 2

Tableau 2. 1: Capacité du système LDK-300	22
---	----

Chapitre 3

Tableau 3. 1 : Allocation des adresses aux ipLDKs.....	46
--	----

Chapitre 4

Tableau 4. 1: MOS et qualité de transmission de la voix	65
Tableau 4. 2: Coefficient R de la qualité de transmission de la voix	65
Tableau 4. 3 : Performance du la machine d'astérisque vs le nombre d'appels simultanés...	72

Liste des abréviations

A

ACD : Automatic Call Distributor

ATA : Analog Telephone Adaptor

C

CDR : Call Detail Record

CTI : Couplage Téléphonie Informatique

D

DECT: Digital Enhanced Cordless Telephone

DHCP: Dynamic Host Control Protocol

DNS: Domain Name Server

DSP: Digital Signal Processor

DTIB: Digital Terminal Interface Board

DTMF: Dual Tone Multi Frequency

F

FXO: Foreign eXchange Office

FXS: Foreign eXchange Subscriber

I

IAX: Inter-Asterisk Exchange

IP: Internet Protocol

IPBX: Internet Private Branch Exchange

ipLDK: internet protocol LDK

K

KSU: Key Service Unit

L

LAN: Local Area Network

LS: Location Server

M

MCU: Multipoint Control Unit

MGCP: Media Gateway Control Protocol

MOS: Mean Opinion Score

MPB: Main Process Board

MTA: Mail Transport Agent

N

NAT: Network Address Translator

O

OPS: Open Settlement Protocol

P

PABX: Private Automatic Branch eXchange

PBX: Private Branch Exchange

PCM: Pulse Code Multiplex

P2P: Peer to Peer

PS: Proxy Server

Q

QoS: Quality of Service

R

RAS: Remote Access Service

RG : ReGister

RNIS : Réseau Numérique à Intégration de Service

RR : Receiver Report

RS : Redirect Server

RTC : Réseau Téléphonique Commuté

RTCP: Real-time Transport Control Protocol

RTP: Real time Transport Protocol TCP

T

TCP: Transmission Control Protocol

TDM: Time Division Multiplex

ToIP: Telephony over IP

S

SDES: Source Description

SDP : Session Description Protocol

SIP: Session Initiation Protocol

SLIB: Single Line Telephone Interface Board

SMTP Simple Mail Transfer Protocol

SR : Sender Report

U

UAC : User Agent Client

UAS: User Agent Server

UDP: User Datagram Protocol

URI: Uniform Resources Identifier

URL: Uniform Resources Locator

V

VoIB: Voice over IP Interface Board

VoIBE: Voice over IP Interface Board Enhanced

VoIP: Voice over IP

VPN: Virtual Private Network

W

WAN: Wide Area Network

WTIB: Wireless Terminal Interface Board

Introduction générale

La voix sur IP (VoIP, 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 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).

Au lieu de disposer à la fois d'un réseau informatique et d'un réseau téléphonique commuté (RTC), l'entreprise peut donc, grâce à la VoIP, tout fusionner sur un même réseau. Ça part 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, jigue-jitter, Qos,...).

Comme toute innovation technologique qui se respecte, la VoIP doit non seulement simplifier le travail mais aussi faire économiser de l'argent. Les entreprises dépendent 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 de grand intérêt pour la voix sur un 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 multi-sites, les sondages montrent que le phénomène de migration vers les systèmes de téléphonie sur IP en entreprise est actuellement engagé.

La migration de la solution actuelle vers la Téléphonie sur IP s'effectue en douceur. Les solutions de téléphonie sur IP sont conçues pour dégager une stratégie de migration à faible risque à partir de l'infrastructure existante.

Le scénario vers lequel va s'orienter la téléphonie sur IP dépend beaucoup de l'évolution du réseau lui-même. En effet, si Internet reste à peu près dans sa configuration actuelle où il est essentiellement dimensionné en fonction d'une qualité de service moyenne pour la transmission des données, il est fort probable que la téléphonie sur IP restera un marché limité. Cependant, cette nouvelle technologie constitue une solution idéale pour le cas d'interconnexion de PBX d'entreprises.

En effet, les entreprises multi-sites cherchent à économiser des dépenses énormes dues aux communications intersites. Il serait donc envisageable de profiter de cette nouvelle technologie émergente afin d'assurer cette interconnexion des PBX.

Dans ce même contexte, et dans le cadre de mon projet de fin d'étude, je suis appelée à interconnecter les PBX de l'entreprise d'accueil qui a ressenti le besoin de profiter de cette nouvelle technologie peu coûteuse et à priori assez performante. Le cahier de charge de ce travail propose que l'interconnexion se fasse via le protocole SIP compte tenu des spécificités des PBX en question. Pour ce faire, il est nécessaire d'implémenter un serveur SIP. Une étude des besoins montre que le logiciel libre AstérisK pourrait assurer ce rôle de serveur SIP.

En effet, l'application demandée consiste essentiellement à :

- Adapter AstérisK pour assurer sa connexion avec l'ipLDK.
- Configurer AstérisK pour établir la communication entre tous les postes liés aux différents IP-LDK.
- Configurer astérisK de façon à pouvoir exploiter le matériel traditionnel de l'entreprise pour communiquer via SIP.

Chapitre I : VoIP et protocoles associés

Introduction

La téléphonie IP est devenue importante pour les entreprises. L'enjeu est de réussir à faire converger le réseau de données IP et le réseau téléphonique actuel. D'emblé, une étude bibliographique des notions de base de cette nouvelle technologie ainsi que ces différents protocoles se montre nécessaire avant de se lancer à implémenter la solution proposée. Pour ce faire, ce premier chapitre sera consacré à une étude de la VoIP : ses avantages, son principe et les protocoles mis en jeu.

I. Les avantages de la VoIP :

Différents sont les raisons qui peuvent pousser les entreprises à s'orienter vers la VoIP comme solution pour la téléphonie. Le tableau suivant, issu d'un sondage auprès de 100 décideurs IT, montre les principales motivations pour déployer la téléphonie sur IP au sein d'une entreprise.

Motivations	Pourcentage
Réduction de coûts	75 %
Nécessité de standardiser l'équipement	66 %
Hausse de la productivité des employés	65 %
Autres bénéfices de productivité	64 %
Hausse du volume d'appels à traiter	46 %
Autres facteurs	50 %

Tableau 1. 1: Principales motivations pour le choix de la VoIP [1]

En effet, la VoIP offre de nombreuses nouvelles possibilités aux opérateurs et utilisateurs qui bénéficient d'un réseau basé sur IP. Les avantages les plus marqués sont les suivants.

1. Réduction des coûts

Les coûts généraux de l'infrastructure de réseau sont réduits. En effet, le déploiement d'un unique réseau convergé voix et données sur tous les sites permet de réaliser des économies sur les investissements productifs. De plus, comme le téléphone et le PC partagent le même câble Ethernet, les frais de câblage sont réduits. Les frais d'administration du réseau sont également minimisés. Il est ainsi possible de réaliser des économies à court et à long terme sur de nombreux postes: administration d'un seul réseau, fournisseur d'accès unique, unique contrat de maintenance, câblage commun, gratuité des communications interurbaines, réduction de la complexité de l'intégration d'applications.

D'autre part, en déplaçant le trafic voix RTC vers le réseau privé WAN/IP les entreprises peuvent réduire sensiblement certains coûts de communications. Réductions importantes mises en évidence pour des communications internationales, ces réductions deviennent encore plus intéressantes dans la mutualisation voix/données du réseau IP intersites (WAN). Dans ce dernier cas, le gain est directement proportionnel au nombre de sites distants.

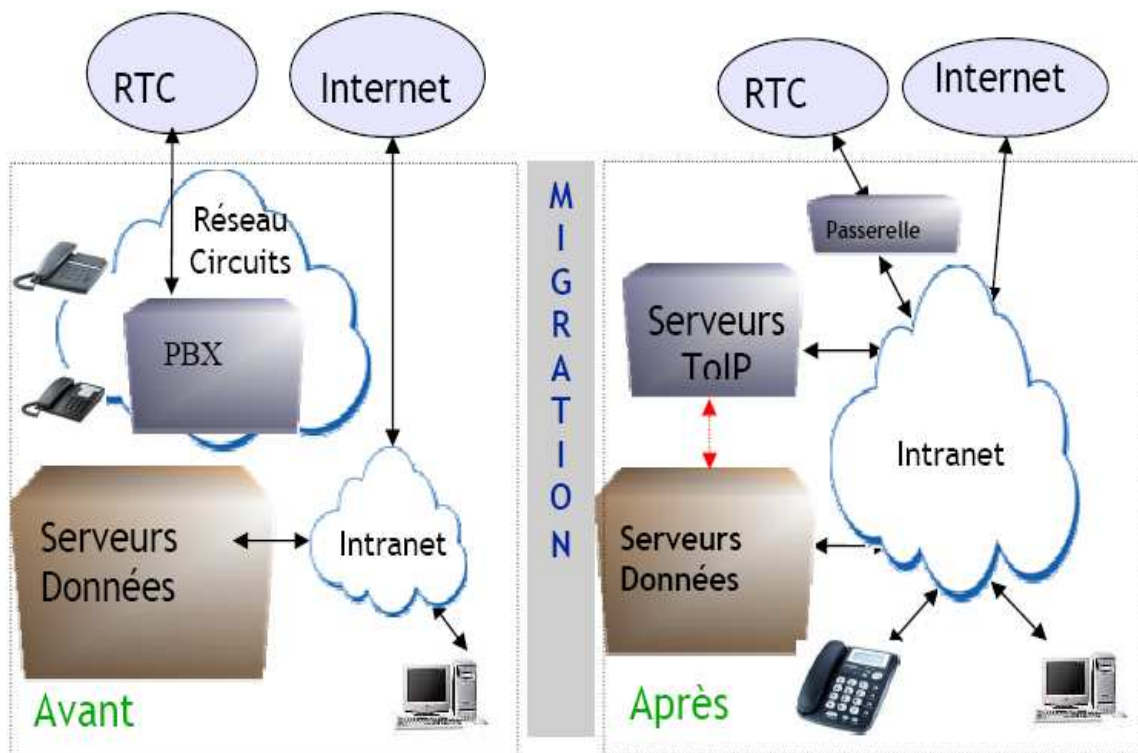


Figure 1. 1 : La VoIP : une infrastructure moins onéreuse [2]

2. Standards ouverts et interopérabilité multifournisseurs

Trop souvent par le passé les utilisateurs étaient prisonniers d'un choix technologique antérieur. La VoIP a maintenant prouvé tant au niveau des réseaux opérateurs que des réseaux d'entreprises que les choix et les évolutions deviennent moins dépendants de l'existant.

Contrairement à nos convictions du début, nous savons maintenant que le monde VoIP ne sera pas uniquement H323, mais un usage multi-protocoles selon les besoins de services nécessaires. Par exemple, H323 fonctionne en mode "peer to peer" alors que MGCP fonctionne en mode centralisé. Ces différences de conception offrent immédiatement une différence dans l'exploitation des terminaisons considérées.

3. Choix d'un service opéré

Les services opérateurs ouvrent les alternatives VoIP. Non seulement l'entreprise peut opérer son réseau privé VoIP en extension du réseau RTC opérateur, mais l'opérateur lui-même ouvre de nouveaux services de transport VoIP qui simplifient le nombre d'accès locaux à un site et réduit les coûts induits. Souvent les entreprises opérant des réseaux multi-sites louent une liaison privée pour la voix et une pour la donnée, en conservant les connexions RTC d'accès local. Les nouvelles offres VoIP permettent outre l'accès RTC local, de souscrire uniquement le média VoIP intersites.

4. Consommation de bande passante moins importante

Suivant l'algorithme d'encodage de la voix, la VoIP peut utiliser 10 fois moins de bande passante que les communications RTC, ce qui permet d'augmenter le nombre de canaux de communication.

5. Un réseau voix, vidéo et données (Triple Play)

En positionnant la voix comme une application supplémentaire du réseau IP, l'entreprise ne va pas uniquement substituer un transport opérateur RTC à un transport IP, mais simplifier la gestion des trois réseaux (voix, données et vidéo) par un seul transport. Une simplification de gestion, mais également une mutualisation des efforts financiers vers un seul outil. Concentrer cet effort permet de bénéficier d'un réseau de meilleure qualité, plus facilement évolutif et plus disponible, pourvu que la bande passante du réseau concentrant la voix, la vidéo et les données soit dimensionnée en conséquence.



Figure 1. 2: Convergence du Triple Play : voix, données et vidéo

6. Un service PABX distribué ou centralisé

Les PABX en réseau bénéficient de services centralisés tel que la messagerie vocale, la taxation, etc. Cette même centralisation continue à être assurée sur un réseau VoIP sans limitation du nombre de canaux. A l'inverse, un certain nombre de services sont parfois souhaités dans un mode de décentralisation. C'est le cas du centre d'appels où le besoin est une centralisation du numéro d'appel (ex : numéro vert), et une décentralisation des agents du centre d'appel. Difficile à effectuer en téléphonie traditionnelle sans l'utilisation d'un réseau IP pour le déport de la gestion des ACD (Automatic Call Distributor) distants. Il est ainsi très facile de constituer un centre d'appel ou centre de contacts (multi-canaux/ multi-médias) virtuel qui possède une centralisation de supervision et d'informations.

7. Intégration des services vidéo

La VoIP intègre une gestion de la voix mais également une gestion de la vidéo. Si nous excluons la configuration des "multicasts" sur les composants du réseau, le réseau VoIP peut accueillir des applications vidéo de type vidéo conférence, vidéo surveillance, e-Learning, vidéo on demand,..., pour l'ensemble des utilisateurs à un coût d'infrastructure réseau supplémentaire minime.

II. Principe de fonctionnement de la voix sur IP

1. Mode de fonctionnement

La voix sur IP (Voice over IP) caractérise l'encapsulation d'un signal audio numérique (la voix) au sein du protocole IP. Cette encapsulation permet de transporter la voix sur tout réseau compatible TCP/IP. Le transport de la voix sur un réseau IP nécessite au préalable sa numérisation. Il convient alors de récapituler les étapes nécessaires à la numérisation de la voix avant d'entrer dans les détails de la VoIP.

Le processus de la numérisation de la voix est schématisé par la figure suivante :

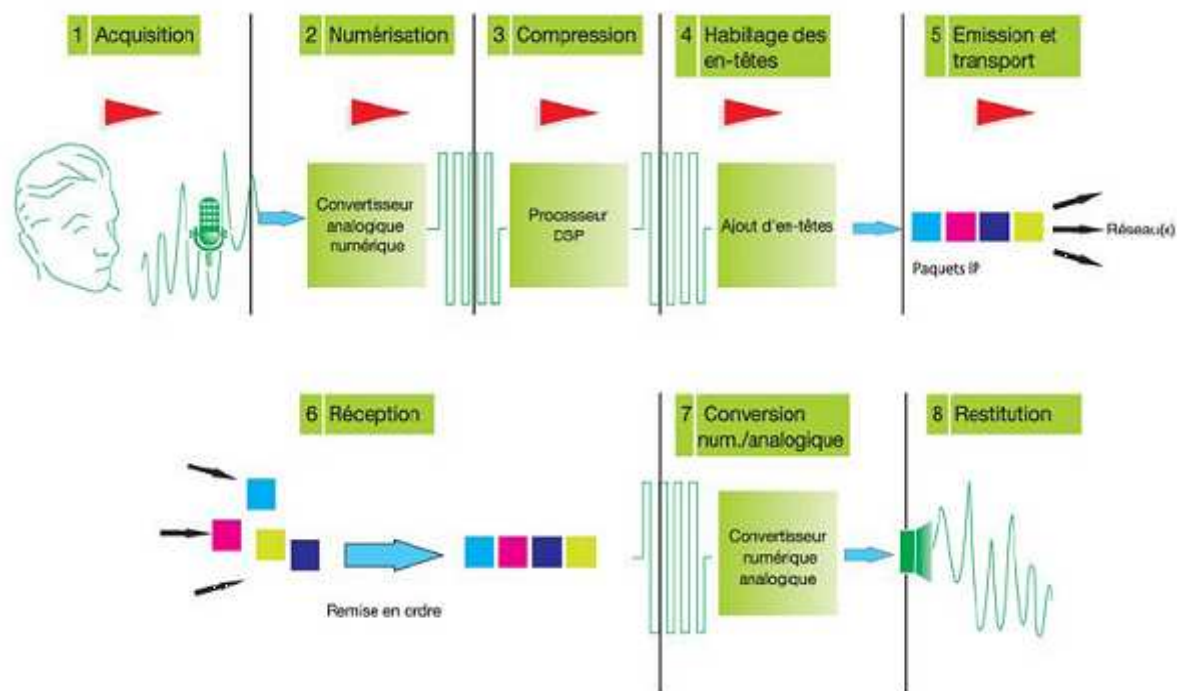


Figure 1. 3: Processus de la VoIP

a. Numérisation

Dans le cas où les signaux téléphoniques à transmettre sont sous forme analogique, ces derniers doivent d'abord être convertis sous forme numérique suivant le format PCM (Pulse Code Modulation) à 64 Kbps. Si l'interface téléphonique est numérique (accès RNIS, par exemple), cette fonction est omise.

b. Compression

Le signal numérique PCM à 64 Kbps est compressé selon l'un des formats de codec (compression / décompression) puis inséré dans des paquets IP. La fonction de codec est le plus souvent réalisée par un DSP (Digital Signal Processor). Selon la bande passante à disposition, le signal voix peut également être transporté dans son format originel à 64 Kbps.

c. Décompression

Côté réception, les informations reçues sont décompressées- Il est nécessaire pour cela d'utiliser le même codec que pour la compression - puis reconverties dans le format approprié pour le destinataire (analogique, PCM 64Kbps, etc.).

2. Principaux codecs utilisés

Le mot codec vient de 'codeur-décodeur' et désigne un procédé capable de compresser ou de décompresser un signal, analogique ou numérique, en un format de données. Les codecs encodent des flux ou des signaux pour la transmission, le stockage ou le cryptage de données. D'un autre côté, ils décodent ces flux ou signaux pour édition ou visionnage. Le but premier des codecs est de pouvoir traiter un maximum de données avec un minimum de ressources. [3]

Il existe une multitude de codecs, mais ils peuvent être divisés en deux grandes familles suivant leur manière de compresser les données. En effet, on distingue deux types de compression. La compression non-destructive (on dit aussi sans pertes ou lossless) qui permet de retrouver le signal initial tel qu'il était avant codage et la compression destructive (avec pertes ou lossy) qui prend en compte les caractéristiques des données à compresser et qui peuvent retirer des informations pouvant être considérées «non pertinentes».

Dans le monde de VoIP, les codecs sont employés pour coder la voix pour la transmission à travers des réseaux IP. Les codecs pour l'usage de VoIP sont désignés également sous le nom des vocodeurs, pour des « encodeurs de voix ». Quelques codecs soutiennent également la suppression de silence, où le silence n'est pas codé ou n'est pas transmis. Le tableau suivant présente les principaux codecs de la voix.

Codec	Débit binaire (Kbps)	Bande passante
ITU G.711	64	72,8
ITU G.726	24	32,8
ITU G.728	16	21,9
ITU G.729	8	16,8
ITU G.723	5.3 et 6.4	12,3

Tableau 1. 2: Principaux codecs de la VoIP [4]

III. Principaux protocoles de la VoIP

Le respect des contraintes temporelles est le facteur le plus important lorsque l'on souhaite transporter la voix. Il faut alors penser à implémenter un mécanisme de signalisation pour assurer la connexion entre les utilisateurs.

Plusieurs protocoles de VoIP ont vu le jour. Les deux protocoles les plus utilisés à nos jours sont H.323 et SIP. Le protocole H.323 a été élaboré dans le milieu des télécommunications à l'inverse du protocole SIP qui a été développé dans le milieu informatique. D'ailleurs, le matériel en question (les PBX en à interconnecter) supporte ces deux protocoles. On s'intéressera alors dans ce qui suit à étudier ces deux protocoles.

1. Le protocole H323

Ce fût en 1996 la naissance de la première version voix sur IP appelée H323. Issu de l'organisation de standardisation européenne ITU-T sur la base de la signalisation voix RNIS (Q931), ce standard regroupe un ensemble de protocoles de communication de la voix, de l'image et de données sur IP. Plus qu'un protocole, H.323 ressemble d'avantage à une association de plusieurs protocoles différents et qui peuvent être regroupés en trois catégories : la signalisation, la négociation de codec, et le transport de l'information.

a. Signalisation

Les messages de signalisation sont ceux que l'on envoie pour demander d'être mis en relation avec une autre personne, indiquant que la ligne est occupée, que le téléphone sonne... Cela comprend aussi les messages que l'on envoie pour signaler que tel téléphone est connecté au réseau et qu'il peut être joint. En H.323, la signalisation s'appuie sur le protocole RAS (Remote Access Service) pour l'enregistrement et l'authentification, et le protocole Q.931 pour l'initialisation et le contrôle d'appel.

b. Négociation

La négociation est utilisée pour se mettre d'accord sur la façon de coder les informations qu'on va s'échanger. Il est important que les téléphones (ou systèmes) parlent un langage commun s'ils veulent se comprendre. Il serait aussi préférable, s'ils ont plusieurs alternatives de langages qu'ils utilisent. Il peut s'agir du codec le moins gourmand en bande passante ou de celui qui offre la meilleure qualité. Le protocole utilisé pour la négociation de codec est le H.245.

c. Transport de l'information

Le transport de l'information s'appuie sur le protocole RTP (Real Time transport Protocol) qui transporte la voix, la vidéo ou les données numérisées par les codecs. On peut aussi utiliser les messages RTCP pour faire du contrôle de qualité, voire demander de renégocier les codecs si, par exemple, la bande passante diminue.

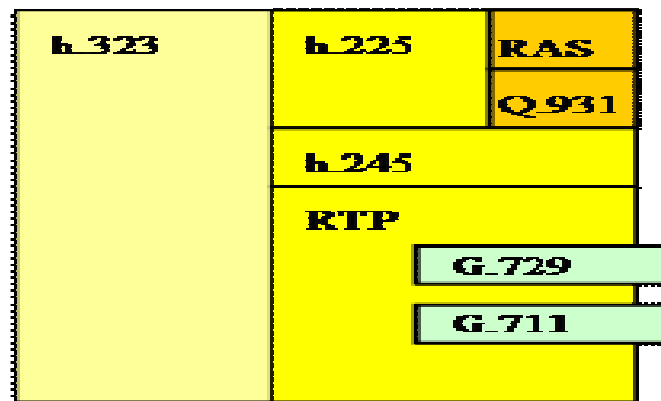


Figure 1. 4 : Protocoles secondaires ou associés [5]

2. Le protocole SIP

a. Définition

SIP est un protocole normalisé et standardisé par l'IETF qui a été conçu pour établir, modifier et terminer des sessions multimédia. Il se charge de l'authentification et de la localisation des multiples participants. Il se charge également de la négociation sur les types de média utilisables par les différents participants en encapsulant des messages SDP (Session Description Protocol). SIP ne transporte pas les données échangées durant la session comme la voix ou la vidéo. SIP étant indépendant de la transmission des données, tout type de données et de protocoles peut être utilisé pour cet échange. SIP remplace progressivement H323. Ceci est justifié par les différents atouts de ce standard. Il s'agit d'un protocole :[6]

- Ouvert : les protocoles et documents officiels sont détaillés et accessibles à tous en téléchargement.
- P2P : sur un LAN, SIP fonctionne complètement en P2P, ce n'est pas le cas pour l'établissement de sessions entre deux pairs séparés par un NAT
- Flexible : SIP est également utilisé pour tout type de sessions multimédia (voix, vidéo, mais aussi musique, réalité virtuelle, etc.)
- Points communs avec H323 : l'utilisation du protocole RTP et quelques codecs son et vidéo sont en commun.
- Simple : SIP est simple et très similaire à http. En effet, le client envoie des requêtes au serveur, qui lui renvoie une réponse.

b. Caractéristiques

Puisque le travail demandé sera effectué via le protocole SIP, on s'attardera un peu à expliquer les aspects et les caractéristiques qui font de ce protocole un bon choix pour l'établissement des sessions. Les principales caractéristiques du protocole SIP sont:

Fixation d'un compte SIP

Il est important de s'assurer que la personne appelée soit toujours joignable. Pour cela, un compte SIP sera associé à un nom unique. Par exemple, si en tant qu'utilisateur d'un service de voix sur IP, vous disposez d'un compte SIP et que chaque fois que vous redémarrez votre ordinateur, votre adresse IP change, vous devez cependant toujours être joignable. Votre compte SIP doit donc être associé à un serveur SIP (proxy SIP) dont l'adresse IP est fixe. Ce serveur vous allouera un compte et vous permettra d'effectuer ou de recevoir des appels quel que soit votre emplacement. Ce compte sera identifiable via votre nom (ou pseudo).

Changement des caractéristiques durant une session

Un utilisateur doit pouvoir modifier les caractéristiques d'un appel en cours. Par exemple, un appel initialement configuré en « voice-only » (voix uniquement) peut être modifié en « voix + vidéo ».

Différents modes de communication

Avec Sip, les utilisateurs qui ouvrent une session peuvent communiquer en mode point à point, en mode diffusif ou dans un mode combinant ceux-ci.

- Mode Point à point : on parle dans ce cas là « d'unicast » qui correspond à la communication entre 2 machines.
- Mode diffusif : on parle dans ce cas là de « multicast » (plusieurs utilisateurs via une unité de contrôle MCU - Multipoint Control Unit).
- Combinatoire : combine les deux modes précédents. Plusieurs utilisateurs interconnectés en multicast via un réseau à maillage complet de connexion.

Gestion des participants

Durant une session d'appel, de nouveaux participants peuvent rejoindre les participants d'une session déjà ouverte en participant directement, en étant transférés ou en étant mis en attente (cette particularité rejoint les fonctionnalités d'un PABX par exemple ou l'appelant peut être transféré vers un numéro donné ou être mis en attente).

Négociation des médias supportés

Cela permet à un groupe durant un appel de négocier sur les types de médias supportés. Par exemple, la vidéo peut être ou ne pas être supportée lors d'une session.

Adressage

Les utilisateurs disposant d'un numéro (compte) SIP disposent d'une adresse ressemblant à une adresse mail (sip:numéro@serveursip.com). Le numéro SIP est unique pour chaque utilisateur.

c. Méthodes et codes d'erreur

Principales Méthodes

Le protocole SIP repose sur un modèle Requête/Réponse. Les échanges entre un terminal appelant et un terminal appelé se font par l'intermédiaire de requêtes. La liste des requêtes échangées est la suivante :

- Invite : Cette requête indique que l'application (ou utilisateur) correspondante à l'Url Sip spécifié est invité à participer à une session. Le corps du message décrit cette session (par ex : média supportés par l'appelant). En cas de réponse favorable, l'invité doit spécifier les médias qu'il supporte.
- Ack : Cette requête permet de confirmer que le terminal appelant a bien reçu une réponse définitive à une requête Invite.
- Options : Un proxy server en mesure de contacter l'UAS (terminal) appelé, doit répondre à une requête Options en précisant ses capacités à contacter le même terminal.
- Bye : Cette requête est utilisée par le terminal de l'appelé à fin de signaler qu'il souhaite mettre un terme à la session.
- Cancel : Cette requête est envoyée par un terminal ou un proxy server à fin d'annuler une requête non validée par une réponse finale comme, par exemple, si une machine ayant été invitée à participer à une session, et ayant accepté l'invitation ne reçoit pas de requête Ack, alors elle émet une requête Cancel.
- Register : Cette méthode est utilisée par le client pour enregistrer l'adresse listée dans l'URL TO par le serveur auquel il est relié.

Codes d'erreurs

Une réponse à une requête est caractérisée, par un code et un motif, appelés respectivement code d'état et raison phrase.

Un code d'état est un entier codé sur 3 digits indiquant un résultat à l'issue de la réception d'une requête. Ce résultat est précisé par une phrase, textbased (UTF-8), expliquant le motif du refus ou de l'acceptation de la requête. Le code d'état est donc destiné à l'automate gérant l'établissement des sessions Sip et les motifs aux programmeurs. Il existe 6 classes de réponses et donc de codes d'état, représentées par le premier digit :

- 1xx = Information - La requête a été reçue et continue à être traitée.
- 2xx = Succès - L'action a été reçue avec succès, comprise et acceptée.
- 3xx = Redirection - Une autre action doit être menée afin de valider la requête.
- 4xx = Erreur du client - La requête contient une syntaxe erronée ou ne peut pas être traitée par ce serveur.
- 5xx = Erreur du serveur - Le serveur n'a pas réussi à traiter une requête apparemment correcte.
- 6xx = Echec général - La requête ne peut être traitée par aucun serveur.

d. Architecture

Dans un système Sip on trouve deux types de composantes, les Users Agents (UAS, UAC) et un réseau de serveurs :

Les User Agent

Le premier type de composant SIP est l'application de l'utilisateur final. Ce peut être, par exemple, un terminal de téléphonie ou de visioconférence sur IP, un serveur audio ou vidéo ou encore une passerelle vers un autre protocole. Ce type de composant est appelé User Agent (UA). Il se décompose en une partie cliente et une partie serveur. La partie cliente, appelée User Agent Client (UAC), envoie les requêtes SIP, et la partie serveur, appelée User Agent Server (UAS), les reçoit.

Le RG (Registrar)

Le Registrar est un serveur qui gère les requêtes REGISTER envoyées par les Users Agents pour signaler leur emplacement courant. Ces requêtes contiennent donc une adresse IP, associée à une URI, qui seront stockées dans une base de données. Les URI SIP sont très similaires dans leur forme à des adresses email : sip:utilisateur@domaine.com. Généralement, des mécanismes d'authentification permettent d'éviter que quiconque puisse s'enregistrer avec n'importe quelle URI.

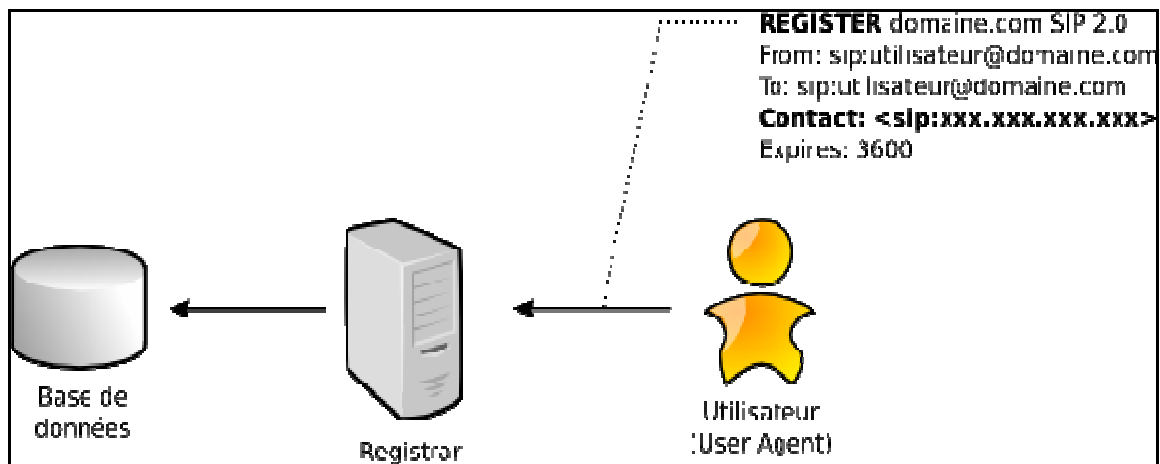


Figure 1. 5 : Enregistrement d'un utilisateur

Le Proxy SIP

Un serveur proxy sert d'intermédiaire entre deux User Agents qui ne connaissent pas leurs emplacements respectifs (adresse IP). En effet, l'association URI-Adresse IP a été stockée préalablement dans une base de données par un Registrar. Le Proxy peut donc interroger cette base de données pour diriger les messages vers le destinataire.

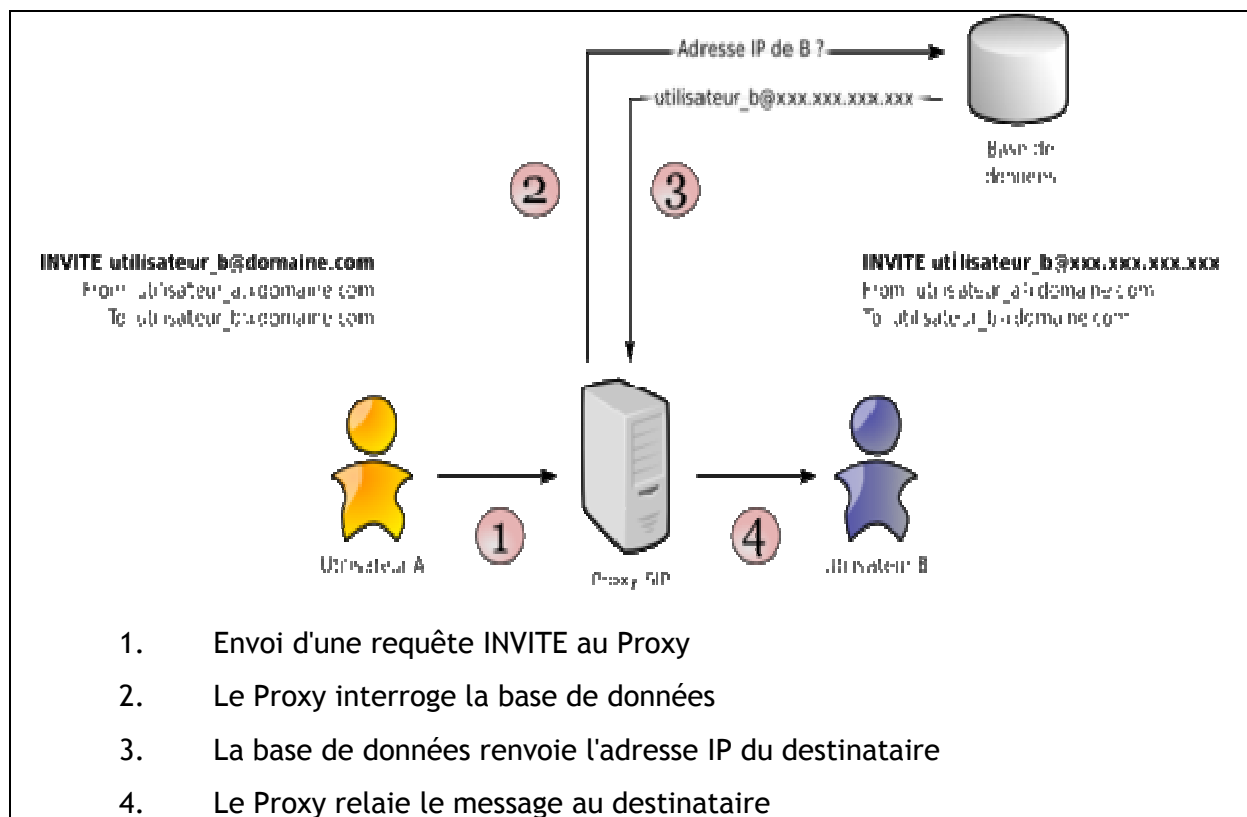


Figure 1. 6 : Etablissement d'une communication avec un proxy SIP

Le Proxy se contente de relayer uniquement les messages SIP pour établir, contrôler et terminer la session. Une fois la session établie, les données, un flux RTP pour la VoIP par exemple, ne transitent pas par le serveur Proxy. Elles sont échangées directement entre les User Agents.

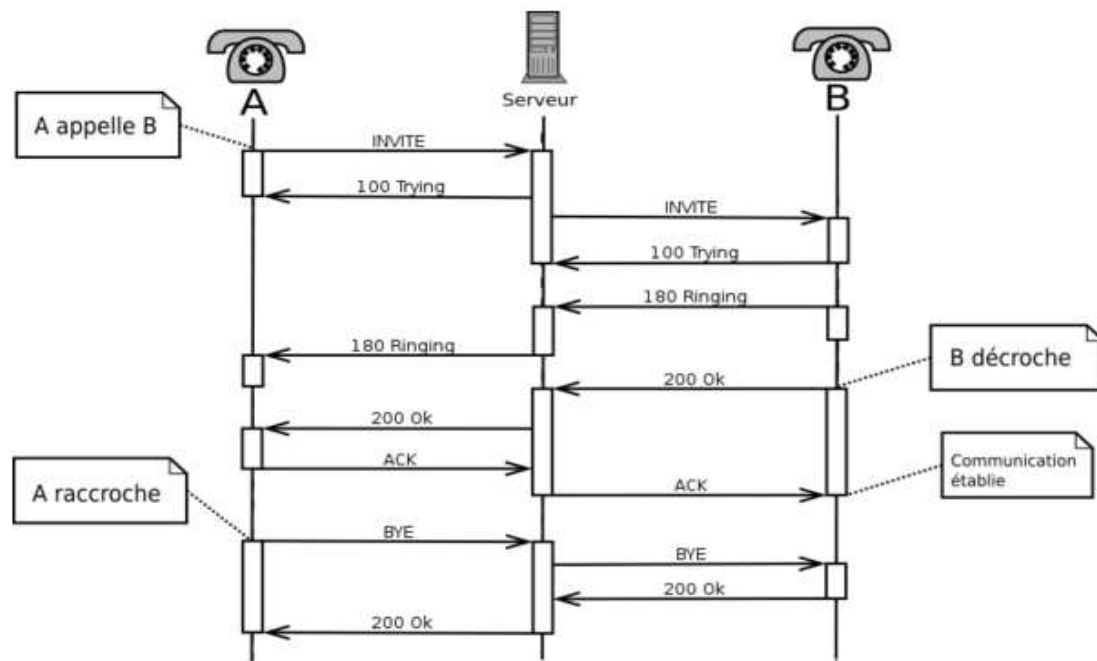


Figure 1. 7 : Etablissement d'une session SIP à travers un Proxy

🚦 Le RS (Redirect Server)

Les serveurs Redirect aident à localiser les User Agent SIP en fournissant une adresse alternative à laquelle l'utilisateur appelé peut être joint. Il réalise simplement une association (mapping) d'adresses vers une ou plusieurs nouvelles adresses. Notons qu'un Redirect Server est consulté par l'UAC comme un simple serveur et ne peut émettre de requêtes contrairement au PS.

🚦 Le LS (Location Server)

Lorsqu'une entité SIP souhaite joindre un correspondant à partir de son adresse SIP, elle est renseignée par le Location Server qui accède à la base d'information renseignée et tenue à jour par le Serveur Registrar. Donc, lorsqu'un serveur proxy recevra un message INVITE destiné à un destinataire, le Location Server lui indiquera l'adresse IP du destinataire et le Proxy Server routera le message vers l'adresse IP appropriée. On remarque dans les usages que le serveur Registrar est souvent associé au serveur Proxy et que le Location Server est, le plus souvent une entité logique.

3. Comparaison entre SIP et H323

Avantages du protocole H.323

- Il existe de nombreux produits (plus de 30) utilisant ce standard adopté par de grandes entreprises telles Cisco, IBM, Intel, Microsoft, Netscape, etc.
- Un niveau d'interopérabilité très élevé, ce qui permet à plusieurs utilisateurs d'échanger des données audio et vidéo sans faire attention aux types de média utilisés.

Avantages du protocole Sip

- Sip est un protocole plus rapide. La séparation entre ses champs d'en-tête et son corps du message facilite le traitement des messages et diminue leur temps de transition dans le réseau.
- Le nombre des en-têtes est limité (36 au maximum et en pratique, moins d'une dizaine d'en-têtes sont utilisées simultanément), ce qui allège l'écriture et la lecture des requêtes et réponses.
- De plus, il sépare les flux de données de ceux de la signalisation, ce qui rend plus souple l'évolution "en direct" d'une communication (arrivée d'un nouveau participant, changement de paramètres...).
- Sip est un protocole indépendant de la couche transport. Il peut aussi bien s'utiliser avec TCP qu'UDP.

Le tableau qui suit est un tableau récapitulatif des différentes caractéristiques de SIP et de H.323

	SIP	H323
Nombre échanges pour établir la connexion	1,5 aller-retour	6 à 7 allers-retours
Maintenance du code protocolaire	Simple par sa nature textuelle à l'exemple de Http	Complexe et nécessitant un compilateur
Evolution du protocole	Protocole ouvert à de nouvelles fonctions	Ajout d'extensions propriétaires sans concertation entre vendeurs
Fonction de conférence	Distribuée	Centralisée
Détection d'un appel en boucle	Oui	un appel routé sur l'appelant provoque une infinité de requêtes
Fonction de télé-services	Oui, par défaut	H.323 v2 + H.450

Tableau 1. 3: Comparaison de SIP et H.323

La simplicité, la rapidité et la légèreté d'utilisation, tout en étant très complet, du protocole Sip sont autant d'arguments qui pourraient permettre à Sip de nous convaincre s'utiliser ce protocole pour notre solution. De plus, ses avancées en matière de sécurité des messages sont un atout important par rapport à ses concurrents.

4. Le protocole RTP

a. Définition

RTP (Real time Transport Protocol) est un protocole qui a été développé par l'IETF afin de faciliter le transport temps réel de bout en bout des flots données audio et vidéo sur les réseaux IP, c'est à dire sur les réseaux de paquets. RTP est un protocole qui se situe au niveau de l'application et qui utilise les protocoles sous-jacents de transport Tcp ou Udp. Mais l'utilisation de RTP se fait généralement au-dessus d'Udp ce qui permet d'atteindre plus facilement le temps réel. Les applications temps réels comme la parole numérique ou la visioconférence constitue un véritable problème pour Internet. Qui dit application temps réel, dit présence d'une certaine qualité de service (QoS) que RTP ne garantie pas du fait qu'il fonctionne au niveau Applicatif. De plus RTP est un protocole qui se trouve dans un environnement multipoint, donc on peut dire que RTP possède à sa charge, la gestion du temps réel, mais aussi l'administration de la session multipoint.

b. Les fonctions de RTP

Le protocole RTP, Real Time Transport Protocol, standardisé en 1996, a pour but d'organiser les paquets à l'entrée du réseau et de les contrôler à la sortie. Ceci de façon à reformer les flux avec ses caractéristiques de départ. RTP est un protocole de bout en bout, volontairement incomplet et malléable pour s'adapter aux besoins des applications. Il sera intégré dans le noyau de l'application. RTP laisse la responsabilité du contrôle aux équipements d'extrémité. RTP, est un protocole adapté aux applications présentant des propriétés temps réel. Il permet ainsi de :

- Mettre en place un séquençement des paquets par une numérotation et ce afin de permettre ainsi la détection des paquets perdus. Ceci est un point primordial dans la reconstitution des données. Mais il faut savoir quand même que la perte d'un paquet n'est pas un gros problème si les paquets ne sont pas perdus en trop grands nombres. Cependant il est très important de savoir quel est le paquet qui a été perdu afin de pouvoir pallier à cette perte. Et ce par le remplacement par un paquet qui se compose d'une synthèse des paquets précédent et suivant.
- Identifier le contenu des données pour leurs associer un transport sécurisé.

- Reconstituer la base de temps des flux (horodatage des paquets : possibilité de resynchronisation des flux par le récepteur)
- L'identification de la source c'est à dire l'identification de l'expéditeur du paquet. Dans un multicast l'identité de la source doit être connue et déterminée.
- Transporter les applications audio et vidéo dans des trames (avec des dimensions qui sont dépendantes des codecs qui effectuent la numérisation). Ces trames sont incluses dans des paquets afin d'être transportées et doivent de ce fait être récupérées facilement au moment de la phase de dépaquetisation afin que l'application soit décodée correctement.

En revanche, ce protocole présente quelques limites. En effet, il ne procure pas de:

- Réserve de ressources sur le réseau
- Fiabilité des échanges (pas de retransmission automatique, pas de régulation automatique du débit)
- Garantie dans le délai de livraison (seules les couches de niveau inférieur le peuvent) et dans la continuité du flux temps réel.

7. Le RTCP

Le protocole RTCP (Real-time Transport Control Protocol) est fondé sur la transmission périodique de paquets de contrôle à tous les participants d'une session. C'est le protocole UDP (par exemple) qui permet le multiplexage des paquets de données RTP et des paquets de contrôle RTCP. Le protocole RTP utilise le protocole RTCP, qui transporte les informations supplémentaires suivantes pour la gestion de la session :

- Les récepteurs utilisent RTCP pour renvoyer vers les émetteurs un rapport sur la QoS. Ces rapports comprennent le nombre de paquets perdus, le paramètre indiquant la variance d'une distribution (plus communément appelé la gigue : c'est à dire les paquets qui arrivent régulièrement ou irrégulièrement) et le délai aller-retour. Ces informations permettent à la source de s'adapter, par exemple, de modifier le niveau de compression pour maintenir une QoS.
- Une synchronisation supplémentaire entre les médias. Les applications multimédias sont souvent transportées par des flots distincts. Par exemple, la voix, l'image ou même des applications numérisées sur plusieurs niveaux hiérarchiques peuvent voir les flots gérés suivre des chemins différents.

- L'identification car en effet, les paquets RTCP contiennent des informations d'adresses, comme l'adresse d'un message électronique, un numéro de téléphone ou le nom d'un participant à une conférence téléphonique.

- Le contrôle de la session, car RTCP permet aux participants d'indiquer leur départ d'une conférence téléphonique (paquet Bye de RTCP) ou simplement de fournir une indication sur leur comportement.

Le protocole RTCP demande aux participants de la session d'envoyer périodiquement les informations citées ci-dessus. La périodicité est calculée en fonction du nombre de participants de l'application. On peut dire que les paquets RTP ne transportent que les données des utilisateurs. Tandis que les paquets RTCP ne transportent en temps réel, que de la supervision. Les paquets de supervision fournissent aux nœuds du réseau les instructions nécessaires à un meilleur contrôle des applications temps réel. On peut détailler les paquets de supervision en 5 types:

- 200 - SR (Sender Report) : Ce rapport regroupe des statistiques concernant la transmission (pourcentage de perte, nombre cumulé de paquets perdus, variation de délai (gigue), ... Ces rapports sont issus d'émetteurs actifs d'une session.

- 201 - RR (Receiver Report) : Ensemble de statistiques portant sur la communication entre les participants. Ces rapports sont issus des récepteurs d'une session.

- 202 - SDES (Source Description) : Carte de visite de la source (nom, e-mail, localisation).

- 203 - BYE : Message de fin de participation à une session.

- 204 - APP : Fonctions spécifiques à une application.

Conclusion

Depuis quelques années, la téléphonie sur IP est annoncée comme le successeur de la téléphonie traditionnelle. Pour les entreprises, les avantages semblent évidents : au premier abord, coûts réduits et réseau local optimisé. L'intégration de la ToIP impose cependant des changements structurels et financiers qui peuvent être lourds de conséquences.

Faire le choix de la ToIP, c'est d'abord répondre à un certain nombre de questions, selon la taille et le budget de l'entreprise : Est-ce que l'entreprise dispose de plusieurs sites distants ? Peut-elle conserver son équipement existant (notamment ses téléphones) ? Quelle solution technologique privilégier ? Quelle stratégie de migration adopter ?

Chapitre II : Etude et choix de la solution

Introduction

La migration vers la voix sur IP n'est pas une étape simple et évidente. Il s'agit pour les entreprises de savoir comment effectuer ces migrations. Concrètement, cette migration doit se faire en douceur, c'est à dire bénéficier des avantages de la voix sur IP tout en exploitant le matériel existant au sein de l'entreprise.

L'intégration de la VoIP dans les entreprises consiste à remplacer les anciens PBX par des IPBX. Les IPBX sont les PABX dans le monde VoIP. Contrairement au monde de la téléphonie classique, les IPBX peuvent être matériels ou logiciels. Les IPBX logiciels peuvent être utilisés pour interconnecter les IPBX matériels. C'est sur cet aspect qu'on focalisera au cours de notre travail. Pour ce faire, nous commençons par étudier le PBX matériel en question. Dans une deuxième partie, nous nous intéresserons à l'étude du PBX logiciel choisi pour l'interconnexion et on finira par décrire la solution à mettre en place.

I. Les IPBX matériels

Tous les grands constructeurs de PABX se sont aujourd'hui tournés vers la production d'IPBX. Bénéficiant de tout le savoir-faire de la téléphonie, ces offres sont performantes et remplacent parfaitement les PABX. Elles sont stables et robustes. On distingue deux types d'IPBX matériels :

- Des PABX tout IP : Ce type de PBX ne supportent que les téléphones IP.
- Des PBX hybrides : Ce type de PBX peut fonctionner à la fois comme un PBX classique et comme un IPBX, il est possible de lui raccorder des postes IP ainsi que des postes numériques et analogiques. C'est ce type de commutateurs qui est plus utilisé au sein de la majorité des entreprises puisqu'il facilite la migration vers la VoIP. [7]

Dans notre cas d'étude, nous nous intéresserons au commutateur dont dispose l'entreprise d'accueil à savoir l'ipLDK 300/300E qui est un système téléphonique hybride de LG, complètement numérique désigné pour satisfaire les besoins des entreprises de taille moyennes.

Composant du LDK

Le LDK-300/300E est composé de 3 KSUs (Key Service Unit), Un KSU de base et deux KSUs d'extension. Chaque KSU contient plusieurs slots pour loger différents types de cartes. Les principales cartes supportées par le commutateur LDK-300 sont:

- **MPB:** Main Process Board, sans laquelle le KSU ne pourrait pas fonctionner. elle doit être installée sur le slot de MPB dans le KSU de base.
- **Slib:** Single Line Telephone Interface Board: cette carte permet de connecter des postes analogiques à l'ipLDK.
- **Dtib:** Digital Terminal Interface Board: cette carte permet de supporter jusqu'à 24 terminaux numériques.
- **Wtib:** Wireless Terminal Interface Board: Le système de téléphonie d'entreprise sans fil de l'ipLDK est utilisé via cette carte qui peut supporter jusqu'à 4 stations. L'ipLDK peut supporter jusqu'à 3 cartes WTIB, donc de connecter jusqu'à 192 terminaux DECT.
- **VOIBE:** Voice Over IP Interface Board et c'est la carte qui nous intéresse le plus au cours de ce travail. C'est cette carte qui va permettre de passer en IP, permettant ainsi aux postes numériques, analogiques et DECT de bénéficier de cette technologie. La carte supporte 12 canaux c'est à dire que le système avec une seule carte VoIBE peut supporter 12 communications simultanées. Il reste à noter que chacune des deux cartes MPB et VOIB dispose d'adresse IP.

Architecture du système

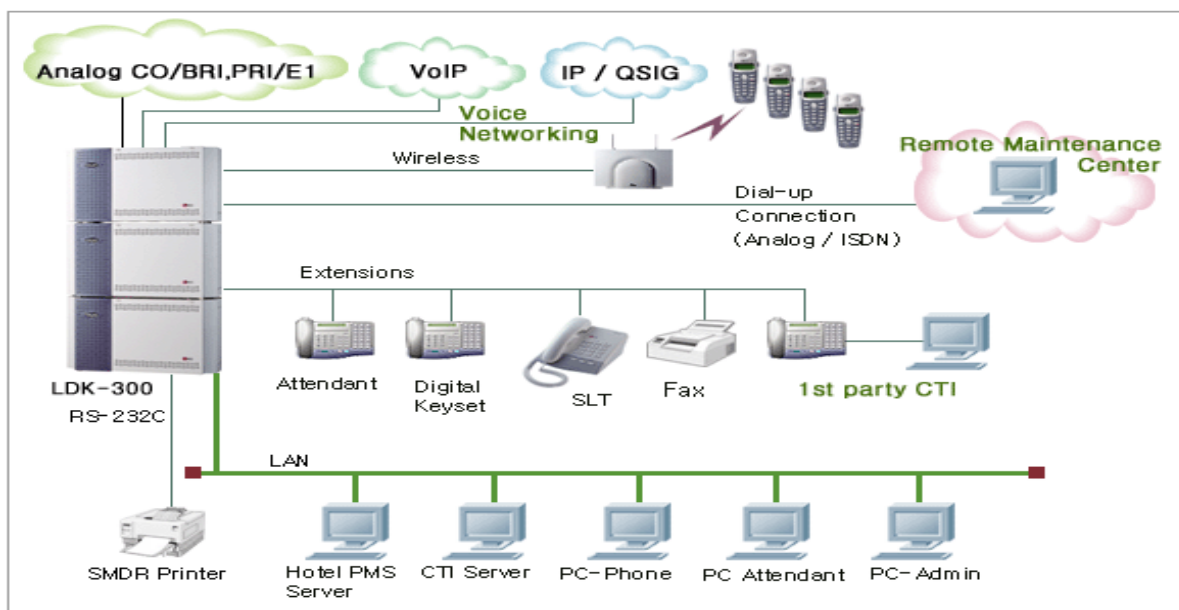


Figure 2. 1: Diagramme de connexion du système

 **Capacité du système LDK-300**

Time Slots (note A)	1 st KSU	144	Paging Zone	External	3 (MPB:1, MISB:2)
	2 nd KSU	144		Internal	35
	3 rd KSU	96	System Speed Dial		3000(24digits)
KSU		3	Station Speed Dial		100(24digits) / station
Serial Port(RS-232C)		4(MPB:2, MISB:2)	Last Number Redial		10
Alarm/Door bell input		3 (RAU:2, MISB:1)	Save Number Redial		1
External Control		7 (RAU:3, MISB:4)	DSS/DLS Console		7
Music Source Inputs		3 (MPB:1, MISB:2)	SMDR		5000
DTMF Receivers		80 (note B)	CO Line Group		72
Power Fail Circuit		90 (30/KSU)	Hunt/UCD Group		48
Intercom Group		15			

Tableau 2. 1: Capacité du système LDK-300

Les solutions propriétaires sont des solutions idéales pour de la téléphonie sur IP clef en main, elles s'avèrent néanmoins, très coûteuses. Etant complètement propriétaires, ces offres manquent d'adaptabilité notamment entre les différents protocoles et matériels (dans le cas de LDK, il ne supporte pas d'autres types de téléphones IP autres que ceux de LG Nortel).

De plus, le support n'est assuré que par la société qui vend l'offre : le client est totalement lié au constructeur. Cette philosophie ne dérangera pas les clients de la téléphonie, déjà habitués à cela avec les PABX. En revanche, le monde de l'informatique est gêné. Ainsi, de nombreux projets de logiciels réalisant les fonctions d'IPBX sont nés.

II. Les IPBX logiciels : Asterisk

Un équipement téléphonique a toujours été un matériel conçu pour répondre à un besoin particulier, que l'on nomme matériel ou logiciel dédié. Si les règles qui régissent la téléphonie sont relativement ouvertes, celles qui régissent le matériel ont été tenues secrètes. [8]

Les standards ouverts étaient l'une des conditions requises pour permettre à chacun de bâtir une infrastructure de téléphonie, en offrant la possibilité d'émuler ces systèmes coûteux (basés sur du matériel) avec un ordinateur et des logiciels. Une solution basée sur les standards ouverts et les logiciels libres est non seulement efficace techniquement, mais permet aussi une adaptation aisée aux besoins de chacun.

La question n'est pas de déterminer quelle technologie est la meilleure, mais quelle technologie permet aux communautés d'être propriétaires de leur processus de développement et de l'adapter à leurs besoins.

Aujourd'hui, la mini révolution dans le domaine des IPBX s'appelle AstérisK. Il y a encore quelques mois, ce mot était connu uniquement des puristes de VoIP au sein d'un cercle très fermé. Aujourd'hui, AstérisK est prononcé par toutes les langues. Il est devenu un système de téléphonie très populaire et de plus en plus utilisé au sein des entreprises (multi-sites ou non).

En effet, notre choix pour cette étude s'est porté sur cet IPBX. Dans ce qui suit, on s'intéressera à présenter cette solution open source qui émule le fonctionnement d'un central téléphonique.

1. Caractéristiques

AstérisK est un logiciel ipPBX avec toutes les fonctionnalités d'un PABX classiques. En d'autres termes, il s'agit de l'outil à la mode pour monter un IPBX.

AstérisK est un logiciel open source qui fonctionne sous les plateformes Unix, Mac OS X, OpenBSD, FreeBSD et Sun Solaris. Il est largement utilisé du fait qu'il reprend toutes les fonctionnalités des PABXs traditionnels, qui sont souvent associées à des technologies propriétaires coûteuses, en plus des nouvelles fonctionnalités qu'il intègre.

AstérisK est multi protocole. Il est compatible avec les différents protocoles de VoIP déjà existants tel que H323, MGCP, SIP, Skinny10, IAX2...

AstérisK, en plus d'être compatible avec l'essentiel des protocoles VoIP, supporte également tout type d'équipements VoIP. Il supporte également les protocoles TDM et peut s'intégrer au sein de tout type d'entreprise, quelque soit l'infrastructure.

Avec AstérisK, il est possible de disposer de plusieurs modèles de gestionnaires d'appels. Une entreprise peut intégrer un ou plusieurs AstérisK dans ses locaux ou bien bénéficier d'un tel système de façon distante. En effet, la grande force de cet outil qui devient de plus en plus incontournable réside dans le fait qu'il n'est plus nécessaire d'héberger l'équipement dans ses locaux.

Une telle solution permettrait de réduire les coûts pour toute entreprise soucieuse de disposer d'un PABX de qualité. De plus la migration vers cette solution VoIP peut être totalement transparente pour les employés de l'entreprise. Ils peuvent conserver leur numéro de téléphone.

L'utilisation d'un Astérisik permet d'accroître la mobilité des agents d'une entreprise. En effet, les communications se faisant via le réseau Internet, un agent peut travailler à son domicile, dans les locaux de l'entreprise et toujours être joignable sur son numéro de téléphone lié directement à Astérisik. Il suffit pour cela que l'agent dispose d'une connexion Internet sur le site où il se trouve.

2. Installation

Mise à jour du serveur

Nous avons choisi d'installer le logiciel Astérisik sous la plateforme Fedora Core 4. Une fois l'installation de Fedora est effectuée, une mise à jour du système est conseillée. Ceci est effectué par la commande :

```
# yum update
```

Vérification des paquetages installés pour le bon déroulement de l'installation du Service Asterisk

Puis il faut s'assurer de la présence de quelques packages d'installés. Pour vérifier si un paquetage est bien présent sur le serveur, il faut utiliser la commande suivante :

```
#rpm -q <nom_du_paquetage>
```

Les paquetages à vérifier sont : ncurses, ncurses-devel, openssl, openssl-devel, zlib, zlib-devel. Si un paquetage est manquant on peut l'installer à l'aide de la commande :

```
# yum install nom_du_paquetage
```


Téléchargement et décompression des packages nécessaires à l'installation

On commence par télécharger les packages nécessaires à l'installation d'Asterisk :

```
# cd /usr/src
# wget ftp://ftp.digium.com/pub/asterisk/asterisk-1.2.15.tar.gz
# wget ftp://ftp.digium.com/pub/asterisk/asterisk-addons-1.2.1.tar.gz
# wget ftp://ftp.digium.com/pub/asterisk/asterisk-sounds-1.2.1.tar.gz
# wget http://ftp.digium.com/pub/zaptel/zaptel-1.2.1.tar.gz
# wget http://ftp.digium.com/pub/libpri/libpri-1.2.1.tar.gz
```

Vu qu'il s'agit de fichier d'extension .tar.gz, il faut alors les décompresser à l'aide de la commande :

```
#tar zxvf <nom_du_fichier>.tar.gz
```

 Compilation et Installation.

▪ ZAPTEL

```
# cd zaptel-1.2/
# make clean
# make
# make config
# make install
# cd ..
```

▪ LIBPRI

```
# cd libpri-1.2
# make clean
# make
# make install
# cd ..
```

▪ ASTERISK

```
# cd asterisk-1.2.15
# make clean
# make
# make install
# make config
# make samples
# cd ..
```

▪ asterisk-addons

```
# cd asterisk-addons
# make clean
# make install
# cd..
```

▪ asterisk-sounds

```
# cd asterisk-sounds
# make clean
# make install
```

Ainsi Asterisk est installé il suffit maintenant de rebooter le serveur et de se connecter à la console CLI (Command Line Interface) via la commande :

```
# asterisk -r
```

3. Emplacement des fichiers de configuration

Une fois l'installation d'Astérisk est effectuée, plusieurs fichiers sont créés : [9]

- `/etc/asterisk/` : Contient tous les fichiers de configuration.
- `/usr/sbin/` : Contient le fichier binaire d'Astérisk (programme principal).
- `/usr/lib/asterisk/` : Contient les fichiers binaires qu'Astérisk utilise pour fonctionner.
- `/usr/lib/asterisk/modules/` : Contient les modules pour les applications, les codecs, et les drivers.
- `/usr/include/asterisk/` : Contient les fichiers d'entête requis pour la compilation des modules.
- `/var/lib/asterisk/agi-bin/` : Contient les scripts AGI utilisés par les applications AGI.
- `/var/lib/asterisk/astd/` : Contient la base de registre d'Astérisk.
- `/var/lib/asterisk/sounds/` : Contient les fichiers audio utilisés par Astérisk, par exemple pour les invites de la boîte vocale.
- `/var/run/` : Contient les processus en cours dans le système.
- `/var/run/asterisk.pid` : Fichier contenant le numéro du processus Astérisk en cours.
- `/var/spool/asterisk/outgoing/` : Contient les appels sortants d'Astérisk.
- `/var/log/asterisk/` : Contient les fichiers de logs d'Asterisk.
- `/var/spool/Asterisk/voicemail/` : Répertoire des boîtes vocales

4. Mode de fonctionnement

Le principe de fonctionnement d'Astérisk peut se résumer ainsi : des clients, avec différents paramètres, et des extensions, qui ne sont autres que des règles. C'est sous cette forme d'extension que l'on précise l'ensemble des instructions que suivra le traitement d'un appel à son arrivé pour aboutir à destination.

L'utilisation d'astérisk consiste à configurer un ensemble de fichiers de configuration. Ces fichiers de configuration sont créés sous `/etc/Asterisk`, à savoir :

- `adsi.conf`: Ce fichier contient la configuration d'Analog Display Services Interface.
- `adtranvofr.conf`: Ce fichier contient la configuration de l'utilisation d'Adtran's Voice sur le Frame Relay.
- `agents.conf`: Ce fichier contient la configuration de l'utilisation des agents, comme dans le cas d'un centre d'appel. Ceci nous permet de définir les agents et de leurs assigner des ID et des mots de passe.

- `alarmreceiver.conf`: Ce fichier configure l'application de récepteur d'alarme.
- `alarmreceiver.conf`: Ce fichier configure l'application de récepteur d'alarme.
- `alsa.conf`: Ce fichier contient les variables de configuration de la carte son de la console.
 - `asterisk.adsi`: Ce fichier contient le script ADSI par défaut d'Astérisque. Ceci va être exécuté à partir du téléphone si on utilisait du matériel ADSI.
 - `asterisk.conf`: Ce fichier définit certaines variables pour l'utilisation d'Astérisque. Il sert essentiellement à indiquer à Astérisque où chercher certains fichiers et certains programmes exécutables.
 - `cdr_manager.conf`: Ce fichier configure CDR (Call Detail Records) pour le Call Management.
 - `cdr_odbc.conf`: C'est le fichier de configuration des connexions avec les bases de données ODBC pour stocker les CDRs.
 - `cdr_pgsql.conf`: Ce fichier nous permet d'utiliser les bases de données PostgreSQL pour stocker les CDRs.
 - `cdr_tds.conf`: C'est un fichier de configuration pour l'utilisation de FreeTDS, permettant ainsi des connexions avec Microsoft SQL et Sybase.
 - `enum.conf`: Ce fichier configure l'utilisation d'ENUM, mécanisme permettant d'utiliser un numéro de téléphone comme clé de recherche dans le DNS pour trouver la manière de joindre une personne ou une autre entité.
 - `extconfig.conf`: permet de charger les files d'attente via les bases de données.
 - `extensions.conf`: Ce fichier configure le comportement d'Astérisque. C'est le fichier qui nous intéresse le plus dans ce travail.
 - `features.conf`: Ce fichier contient les options pour le call parking aussi bien que d'autres aspects divers.
 - `iax.conf`: Ce fichier configure les conversations VoIP en utilisant le protocole Inter-Asterisk Exchange (IAX).
 - `festival.conf`: Ce fichier définit les paramètres de Festival, qui est un programme open source qui permet à un serveur de parler un texte. Ceci est optionnel et on ne s'intéressera pas à le configurer puisque Astérisque contient déjà des enregistrements des phrases qu'on voudra que le serveur dise.
 - `iaxprov.conf`: Ce fichier permet de disposer de façon simple du Digium's S1011, souvent appelé IAXy.

- `indications.conf`: Ce fichier configure le comportement de notre système téléphonique, à savoir la cadence de la sonnerie et de la tonalité permettant de fournir aux utilisateurs des sons auxquels ils sont habitués.
- `logger.conf`: Ce fichier définit le type de logging qu'on va utiliser. Les paramètres par défauts sont convenables pour la majorité des utilisateurs.
- `manager.conf`: Ce fichier configure l'accès distant au call manager d'AstérisK.
- `meetme.conf`: Ce fichier de configuration permet d'établir des conférences simples.
- `mgcp.conf`: Ce fichier configure le protocole MGCP (Media Gateway Control Protocol). Ce protocole est utilisé par un certain nombre de matériels VoIP en particulier de Cisco.
- `modem.conf`: Ce fichier définit certaines variables pour nous permettre d'utiliser des modems avec AstérisK. Il faut noter qu'AstérisK ne supporte pas un grand nombre de modems et que puisque la majorité de modems fonctionnent en mode half duplex, ils ne seront pas trop performants.
- `modules.conf`: Ce fichier de configuration choisit quels modules d'AstérisK vont être initiés. On peut activer ou désactiver des fonctionnalités du PBX en changeant la configuration des paramètres dans ce fichier.
- `musiconhold.conf`: Ce fichier de configuration crée des échantillons de Music On Hold (musique d'attente) et définit quel musique jouer.
- `osp.conf`: On peut configurer le protocole OSP (Open Settlement Protocol) avec AstérisK à l'aide de ce fichier.
- `oss.conf`: Cette configuration est semblable à `alsa.conf`.
- `phone.conf`: Ce fichier nous permet d'utiliser quelques interfaces téléphoniques de Linux tel que `linejack`.
- `privacy.conf`: Ce fichier nous permet de configurer les options de confidentialité.
- `queues.conf`: Ce fichier de configuration nous permet de créer des files d'attente pour pouvoir gérer un grand nombre d'appels d'une façon intelligente.
- `res_config_odbc.conf`: C'est le fichier de configuration pour le stockage de la configuration sous une base de données ODBC.
- `res_odbc.conf`: C'est une autre partie de configuration de stockage sous une base de données ODBC.
- `rpt.conf`: Ce fichier permet d'utiliser un répéteur radio.

- rtp.conf: Ce fichier de configuration définit les ports à utiliser pour le protocole RTP (Real-Time Protocol). Il faut noter que les numéros listés sont des ports UDP.
- sip.conf: Ce fichier de configuration définit les utilisateurs du protocole SIP (Session Initiation Protocol) et leurs options. On peut aussi définir d'autres options globales pour SIP telles que, à quelle adresse IP rattacher, quels ports utiliser et les timeout qu'on va imposer. Nous focalisons sur ce fichier puisque notre solution est basée sur le protocole SIP.
- skinny.conf: Ce fichier configure le protocole skinny VoIP protocol.
- telecordia-1.adsi: C'est un autre échantillon de scripts ADSI.
- voicemail.conf: Ce fichier crée des utilisateurs du service voicemail et quelques options globales du Comedian Mail (système de messagerie vocale d'AstérisK).
- vpb.conf: Ce fichier configure le matériel de VoiceTronix.
- zapata.conf: Configure les paramètres de l'interface téléphonique Zapata.

III. Choix de la solution

Dans le cadre de ce projet, on cherche à interconnecter les différents LDK se trouvant sur les différents sites d'une entreprise entre eux via IP.

Le schéma suivant montre une architecture simplifiée de cette interconnexion :

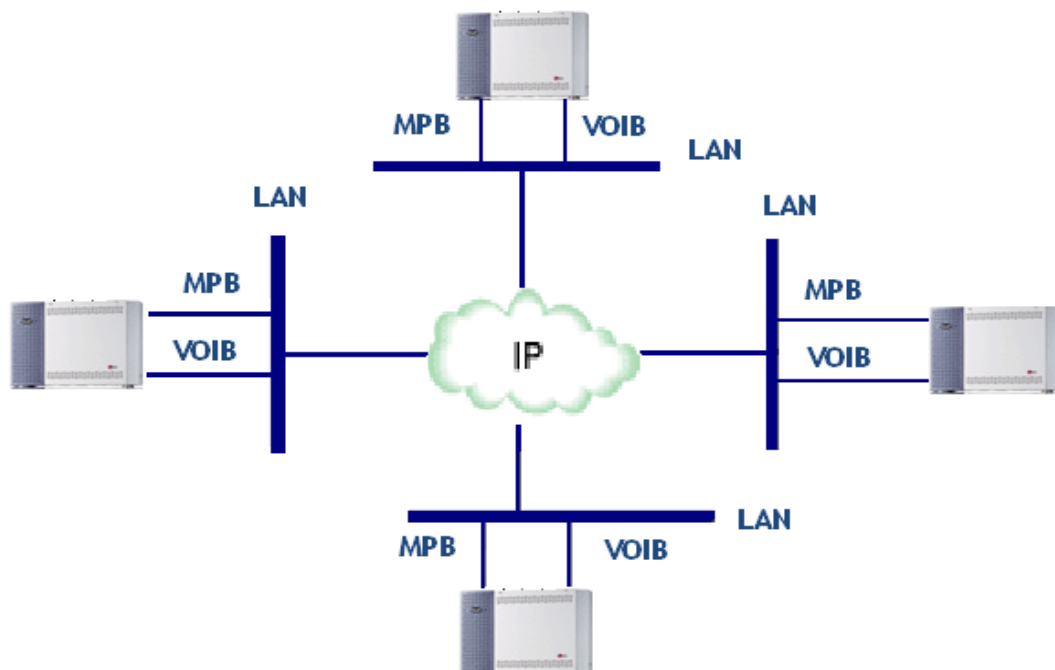


Figure 2. 2 : Configuration d'une interconnexion des IP-LDK via IP

Cette interconnexion sera mise en place via le protocole SIP vu sa simplicité et sa souplesse. L'entreprise devrait créer son propre domaine SIP et associer à ce domaine SIP des comptes SIP. Ainsi l'IPBX astérisk va jouer le rôle d'un serveur SIP centralisé au niveau du siège de l'entreprise. Les autres LDK, éparpillés sur les différents sites distants de l'entreprise, seront connectés à ce serveur SIP via Internet ou bien via des connexions VPN dont dispose généralement l'entreprise avec ses différents sites distants. Il s'agit alors de créer des trunks SIP entre le serveur Astérisk et les différents LDKs.

Cette solution semble à priori être une bonne orientation pour les entreprises multi-sites. En effet, Cette solution prend en considération les différents types de matériels dont peut disposer une entreprise ce qui n'est pas souvent le cas avec les PBX propriétaires. En effet, les ipLDK peuvent être interconnectés via d'autres protocoles tels que QSIG mais cette interconnexion ne prend en considération que le matériel du même constructeur (en particulier les téléphones IP). Avec Astérisk n'importe quel type de terminaux IP peut être ajouté au sein du réseau. D'autre part tout le matériel existant peut être utilisé avec cette solution sans nécessiter beaucoup de matériel supplémentaire. Les équipements nécessaires à l'implémentation complète de cette solution basée sur un serveur SIP sont : un serveur SIP (qui le serveur astérisk dans ce cas), des téléphones IP, des adaptateurs ATA pour interconnecter les postes analogiques, en plus des équipements traditionnels, à savoir les téléphones analogiques et numériques ainsi que les postes mobile (DECT).

Les téléphones IP : l'entreprise peut interconnecter n'importe quel téléphone IP qui supporte le protocole SIP, peu importe le fabricant. Dans notre cas d'étude nous avons utilisé des téléphones ST2030 de Thomson.

Les téléphones numériques : Dans notre cas d'études les postes numériques seront interconnectés à ce nouveau système via la carte DTIB qui joue le rôle d'un adaptateur téléphonie numérique-IP.

Les téléphones analogiques : Jusqu'il y a quelques années en arrière tous les téléphones en entreprise étaient analogiques. Cela implique que beaucoup d'entreprises utilisent encore ce type de téléphones aujourd'hui. En effet, si une entreprise ne désire pas changer complètement son infrastructure téléphonique, il est pratique d'utiliser des adaptateurs ATA qui permettent de convertir un signal analogique en signal IP et vice versa. Cette solution est peu coûteuse et permet de réutiliser les terminaux analogiques.

Les terminaux DECT : Au sein de l'entreprise des terminaux DECT sont utilisés grâce à l'utilisation de la carte WTIB sur le LDK, il serait alors nécessaire de continuer à les utiliser avec astérisk.

Les softphones : étant donné que c'est un logiciel, son intégration avec le CTI sera de plus en plus utilisée. Un softphone permet de faire des économies sur les terminaux, vu qu'ils sont dans la plupart des cas des logiciels gratuits. Pour des utilisateurs temporaires dans une entreprise cela convient parfaitement, car il évite le rachat de matériel supplémentaire. Il suffit de disposer d'un périphérique du type casque ou un combiné est indispensable.

Ainsi, les utilisateurs se trouvant sur les différents sites et disposant d'un matériels assez hétérogènes aussi bien de point de vue constructeurs que de point de vue technologie, peuvent communiquer les uns avec les autres au sein de ce système. Il suffit de disposer d'un compte (numéro de téléphone) SIP enregistré sur le serveur Asterisk.

Ce numéro de téléphone SIP est personnalisable et peut donc avoir la numérotation selon le choix du plan de numérotation de l'entreprise (à 5 numéros, 10 numéros, etc.).

Ces numéros de téléphone dit 'numéros SIP' ne sont visibles qu'à l'intérieur de cette entreprise. Autrement dit, seuls les numéros SIP associés au domaine de l'entreprise peuvent joindre directement les autres numéros SIP appartenant au même domaine.

Pour résumer, la solution proposée est alors expliquée dans le schéma ci-après :

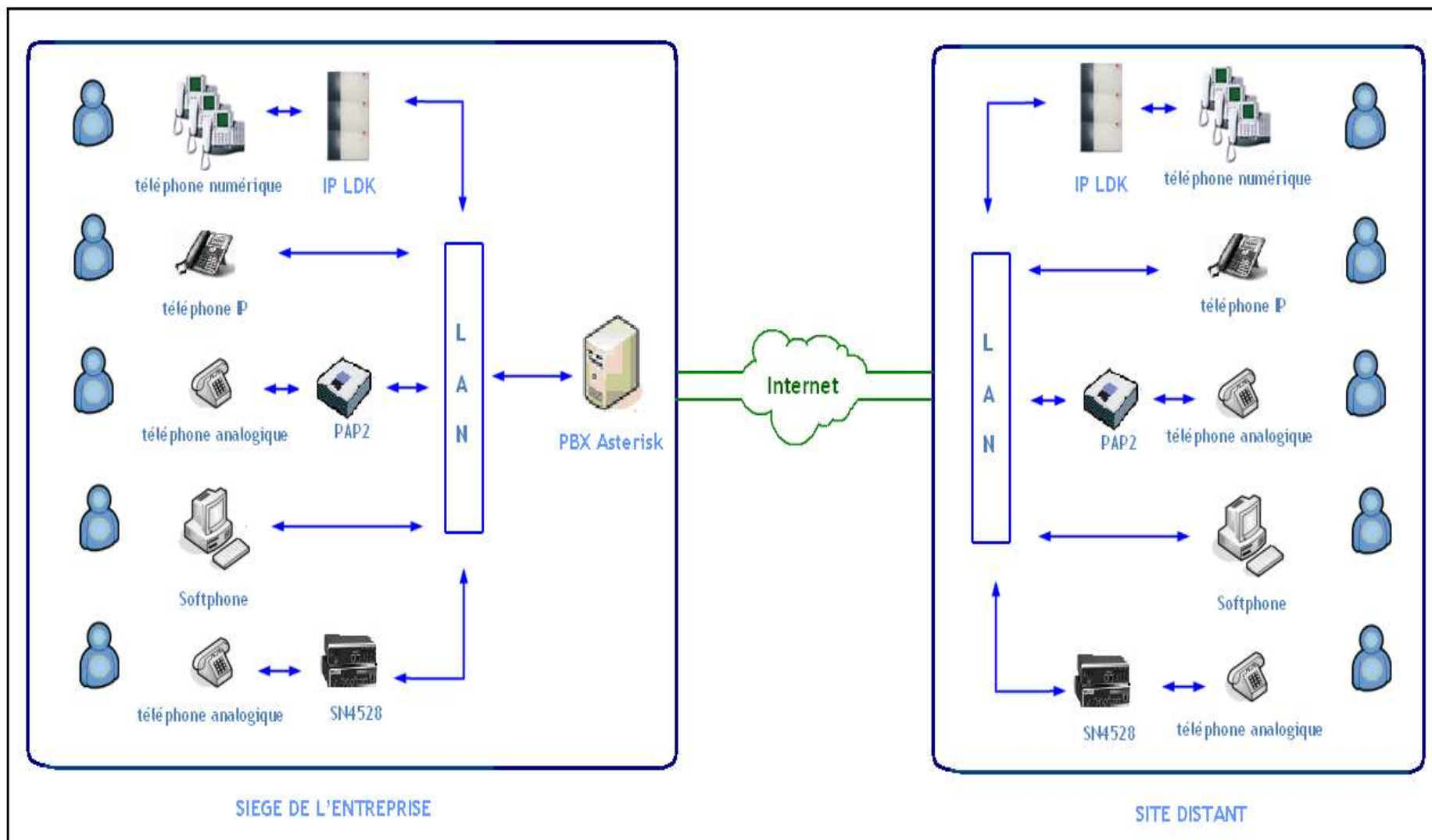


Figure 2. 3 : Schéma descriptif de l'application à réaliser

Conclusion

Dans ce chapitre, j'ai expliqué les différents avantages qui ont plaidoyé en faveur du choix d'AstérisK comme une solution de migration en VOIP et j'ai donné la nouvelle architecture à mettre en place pour cette migration. Dans le chapitre suivant, on s'intéressera à détailler les étapes d'intégration des ces différents terminaux dans le nouveau système.

Chapitre III : Interconnexion des PBX via Astérisik

Introduction

Astérisik a été créé pour remplacer les anciens PBX d'entreprises. Il est donc intéressant de configurer et de tester tous les types de terminaux les plus souvent utilisés en entreprise. Dans ce chapitre, plusieurs téléphones ont été testés et configurés. On s'intéressera à configurer 4 types différents de terminaux. Dans une deuxième partie, nous focalisons sur l'interconnexion des ipLDKs via astérisik.

I. Configuration des terminaux

Différents types de terminaux peuvent être utilisés au sein d'une entreprise. On s'intéressera dans ce qui suit à traiter les différentes configurations de ces différents postes pour pouvoir communiquer via Astérisik.

Le travail demandé dans cette partie dépend du type de matériels dont dispose l'entreprise. Dans ce cas de figure, les différents terminaux que nous configurerons sont schématisés dans la figure suivante:

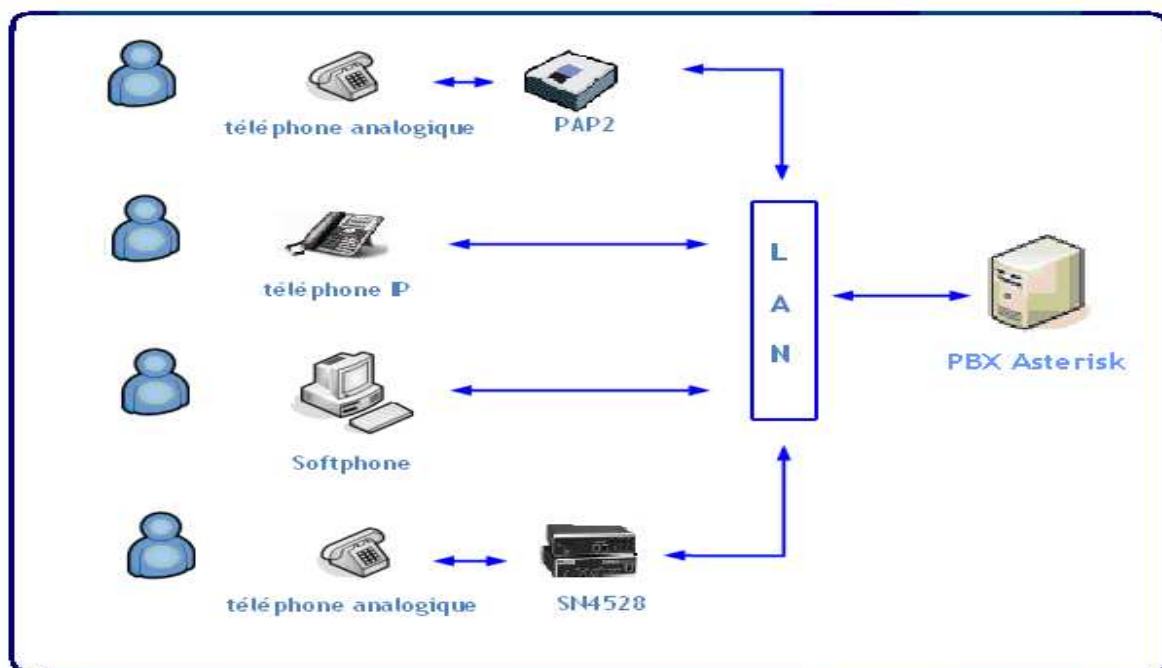


Figure 3. 1: Interconnexion des différents types de terminaux via Astérisik

1. Configuration des postes IP ST2030

Les téléphones IP seront un standard pour les entreprises. La progression technologique et leur avance dans le marché de la téléphonie montrent bien que ces terminaux seront le futur de la téléphonie d'entreprise.

Dans ce qui suit, on va s'intéresser à la configuration des postes IP ST2030 de Thomson. La configuration se fait à partir du navigateur web.

THOMSON
HOME SETUP ADVANCED UTILITY STATUS LOGOUT

Network Interface
Network Setup
VoIP Service
Basic Setup
Auto Provisioning
Basic Setup
APS Log
Auto Provisioning
Basic Setup
APS Log

Basic Setup

Profile Name :

Transfer to Voice Mail
Telephone Number:

Primary SIP Server :

SIP Unregister

SIP Transport UDP TCP Connect Reuse

Service Domain:

Registrar Server Address: port: (1025~49151)

Proxy Server Address: port: (1025~49151)

SIP Local Port: (1025~49151)

Registration Timer: (60~200,000)

Register Frequency: (60~1800 sec)

Ring Tone: ▼

Backup SIP Server :

SIP Unregister

SIP Transport UDP TCP Connect Reuse

Service Domain:

Registrar Server Address: port: (1025~49151)

Proxy Server Address: port: (1025~49151)

SIP Local Port: (1025~49151)

Registration Timer: (60~200,000)

Register Frequency: (60~1800 sec)

User Accounts :

Phone Number	Phone Name	Authentication ID	Password
<input style="width: 80px;" type="text" value="2002"/>	<input style="width: 80px;" type="text" value="2002"/>	<input style="width: 80px;" type="text" value="2002"/>	<input style="width: 80px;" type="text" value="****"/>

Figure 3. 2 : Configuration de base de ST2030

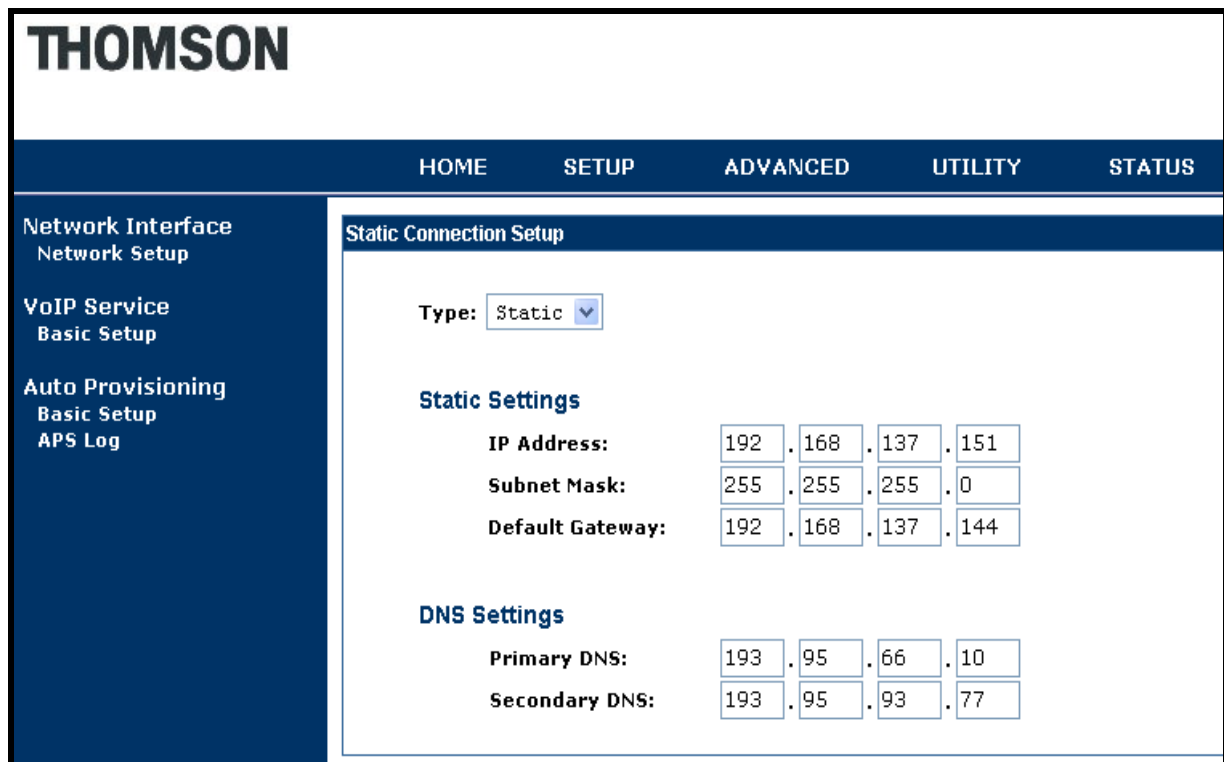


Figure 3. 3 : Configuration des paramètres réseau

2. Softphones

Dans ce cas d'étude, j'ai utilisé le logiciel X-lite comme exemple de softphone. En fait, il existe différentes versions de X-lite qui fonctionnent sous différents systèmes d'exploitation (Windows, Linux, MAC OS). Dans ce qui suit, j'expose deux exemples de configuration d X-lite sous Windows et Linux :

X-lite sous Linux



Figure 3. 4 : Configuration de X-lite sous Linux

X-lite sous Windows

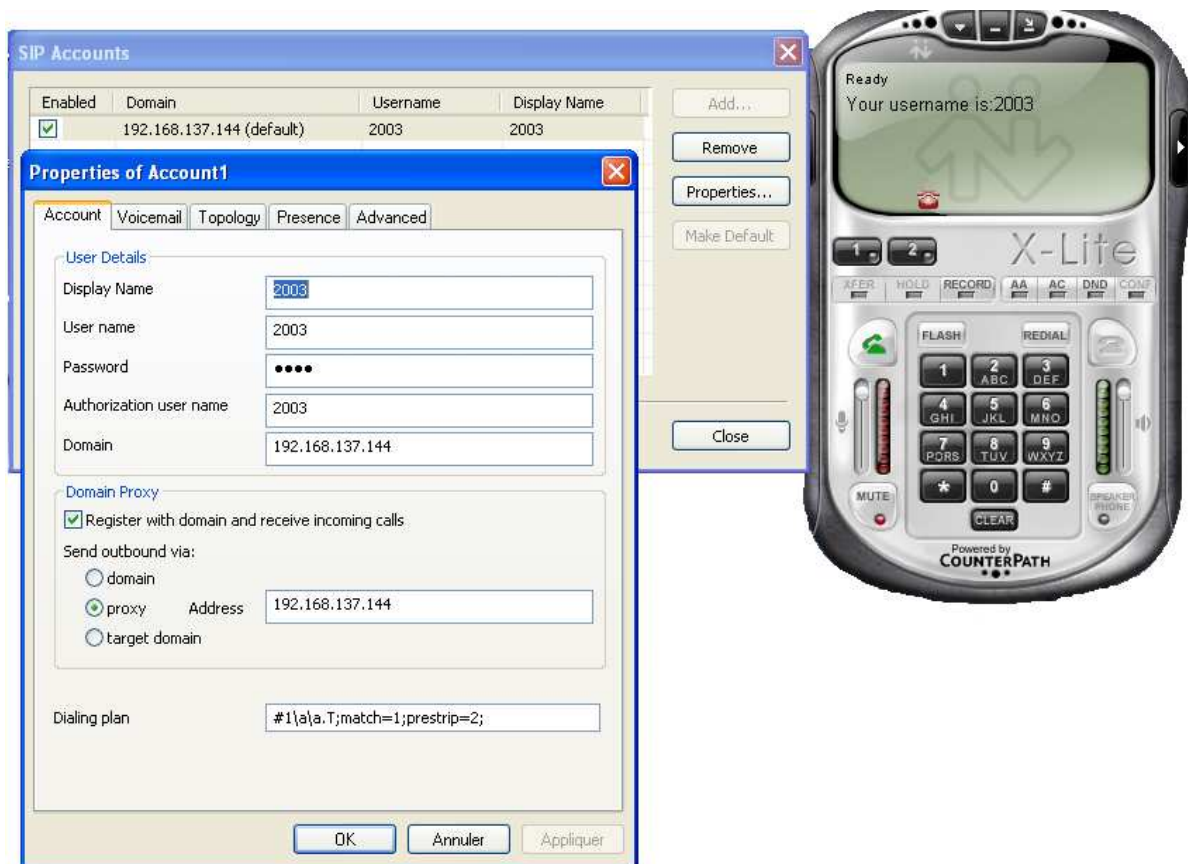


Figure 3. 5: Configuration de X-lite sous Windows

3. Téléphone analogique avec adaptateur ATA PAP2 de linksys

Pour pouvoir utiliser les postes analogiques en VoIP, on a recours aux adaptateurs ATA. Dans ce cas d'étude, on utilise des adaptateurs de type Linksys PAP2. Cet adaptateur possède deux ports téléphoniques qui permettent de raccorder des téléphones analogiques ou utiliser l'un des ports pour un télécopieur. Chaque port fonctionne indépendamment, avec un service téléphonique et des numéros de téléphone distincts, comme s'il s'agissait de deux lignes. Il est possible d'activer les deux lignes simultanément ou une seule ligne à la fois. La configuration des deux lignes est pratiquement identique. La configuration de l'adaptateur se fait depuis le navigateur Web.

LINKSYS
A Division of Cisco Systems, Inc. Firmware Version: 5.1.1(LS)

Phone Adapter with 2 Ports for Voice-Over-IP PAP2

Voice | Info | System | SIP | Regional | **Line 1** | Line 2 | User 1 | User 2

Basic View (switch to advanced view) User Login

SIP Settings	Line Enable: <input type="text" value="yes"/>		
	SIP Port: <input type="text" value="5060"/>		
Proxy and Registration	Proxy: <input type="text" value="192.168.137.144"/>	Register: <input type="text" value="yes"/>	
	Make Call Without Reg: <input type="text" value="yes"/>	Register Expires: <input type="text" value="3600"/>	
	Ans Call Without Reg: <input type="text" value="yes"/>		
Subscriber Information	Display Name: <input type="text" value="2000"/>	User ID: <input type="text" value="2000"/>	
	Password: <input type="password" value="*****"/>	Use Auth ID: <input type="text" value="yes"/>	
	Auth ID: <input type="text" value="2000"/>		
Supplementary Service Subscription	Call Waiting Serv: <input type="text" value="yes"/>	Block CID Serv: <input type="text" value="yes"/>	
	Block ANC Serv: <input type="text" value="yes"/>	Dist Ring Serv: <input type="text" value="yes"/>	
	Cfwd All Serv: <input type="text" value="yes"/>	Cfwd Busy Serv: <input type="text" value="yes"/>	
	Cfwd No Ans Serv: <input type="text" value="yes"/>	Cfwd Sel Serv: <input type="text" value="yes"/>	
	Cfwd Last Serv: <input type="text" value="yes"/>	Block Last Serv: <input type="text" value="yes"/>	
	Accept Last Serv: <input type="text" value="yes"/>	DND Serv: <input type="text" value="yes"/>	
	CID Serv: <input type="text" value="yes"/>	CW/CID Serv: <input type="text" value="yes"/>	
	Call Return Serv: <input type="text" value="yes"/>	Call Back Serv: <input type="text" value="yes"/>	
	Three Way Call Serv: <input type="text" value="yes"/>	Three Way Conf Serv: <input type="text" value="yes"/>	
	Attn Transfer Serv: <input type="text" value="yes"/>	Unattn Transfer Serv: <input type="text" value="yes"/>	
	MWI Serv: <input type="text" value="yes"/>	VMMI Serv: <input type="text" value="yes"/>	
Audio Configuration	Preferred Codec: <input type="text" value="G711u"/>	Silence Supp Enable: <input type="text" value="no"/>	
	Use Pref Codec Only: <input type="text" value="no"/>	FAX CED Detect Enable: <input type="text" value="yes"/>	
	DTMF Tx Method: <input type="text" value="Auto"/>		

CISCO SYSTEMS

Figure 3. 6 : Configuration de base d'une ligne de L'adaptateur Linksys

LINKSYS
A Division of Cisco Systems, Inc. Firmware Version: 5.1.1(LS)

Phone Adapter with 2 Ports for Voice-Over-IP PAP2

Voice | Info | System | SIP | Regional | **Line 1** | Line 2 | User 1 | User 2

Basic View (switch to advanced view) User Login

System Configuration	Enable Web Server: <input type="text" value="yes"/>	User Password: <input type="password"/>	
Internet Connection Type	DHCP: <input type="text" value="no"/>	Static IP: <input type="text" value="192.168.137.72"/>	NetMask: <input type="text" value="255.255.255.0"/>
	Gateway: <input type="text" value="192.168.137.144"/>		
Optional Network Configuration	HostName: <input type="text"/>	Domain: <input type="text"/>	
	Primary DNS: <input type="text" value="193.95.66.10"/>	Secondary DNS: <input type="text" value="193.95.93.77"/>	
	DNS Query Mode: <input type="text" value="Parallel"/>	Syslog Server: <input type="text"/>	
	Debug Server: <input type="text"/>	Debug Level: <input type="text" value="0"/>	

CISCO SYSTEMS

Figure 3. 7 : Configuration des paramètres réseau de Linksys

4. Adaptateur téléphonique SN4528

Le SN4528 est un adaptateur téléphonique qui permet de connecter jusqu'à 4 postes analogiques. La configuration de SN4528 se fait à partir d'une fenêtre HyperTerminal. La configuration consiste à :

a. Configuration de l'interface Ethernet

L'adresse IP qu'on veut attribuer à l'interface Ethernet eth0 est 192.168.137.73. Ceci est configuré via les commandes suivantes :

```
Node>enable
Node#configure
Node(cfg)#context ip router
Node(ctx-ip)[router]#interface eth0
Node(ctx-ip)[router][eth0]#ipaddress 192.168.137.73 255.255.255.0
```

b. Configuration de la Gateway SIP

La configuration de la Gateway SIP consiste à définir le nom de la gateway SIP puis la relier à l'interface Ethernet eth0. L'étape suivante consiste à définir le registre (ou le serveur SIP) sur lequel on va enregistrer les utilisateurs (Serveur Asterisk). On définit ensuite les différents utilisateurs avec leurs identifiants et leurs mots de passe. Dans ce cas, on a défini un seul utilisateur (user 5000 et mot de passe 0000).

```
192.168.137.73(cfg)#gateway sip sip
192.168.137.73(gw-sip)[sip]#bind interface eth0
192.168.137.73(gw-sip)[sip]#no shutdown
192.168.137.73(gw-sip)[sip]#registrar 192.168.137.144
192.168.137.73(gw-sip)[sip]#user 5000
192.168.137.73(gw-sip)[sip]#default-server 192.168.137.144
192.168.137.73(gw-sip)[sip]#authentication asterisk 5000 0000
```

c. Configuration de l'interface SIP

La configuration de l'interface SIP consiste à définir son nom puis à quelle Gateway et à quel serveur SIP elle va être liée.

```
192.168.137.73(cfg)#context cs
192.168.137.73(ctx-cs)[switch]#interface sip MySipIf
192.168.137.73(if-sip)[MySipIf]#bind gateway sip
192.168.137.73(if-sip)[MySipIf]#remote 192.168.137.144
192.168.137.73(if-sip)[MySipIf]#route call dest-table RT_FXS
```

d. Définition de la table de routage

Pour router les appels une table de routage doit être définie. La table est définie sous le nom RT_FXS et de type called-e164. Ensuite relier chaque utilisateur à l'interface fxs correspondante (00, 01, 02, 03).

On doit aussi définir une route par défaut (dans ce cas, l'interface par défaut est la fxo00).

```
192.168.137.73(ctx-cs)[switch]#routing-table called-e164 RT_FXS
192.168.137.73(rt-tab)[RT_FXS]#route 5000 dest-interface if_fxs00
192.168.137.73(rt-tab)[RT_FXS]#route default dest-interface if_fxo00
192.168.137.73(rt-tab)[RT_FXS]#route .... Dest-interface MySipIf
```

e. Définition des interfaces FXS

La définition des interfaces consiste à relier l'interface aux numéros des utilisateurs. Et ceci pour les différentes interfaces fxs00, fxs01, fxs02, fxs03.

```
192.168.137.73(ctx-cs)[switch]#interface fxs if_fws00
192.168.137.73(if-fxs)[if_fxs00]#route call dest-table RT_FXS
192.168.137.73(if-fxs)[if_fxs00]#subscriber-number 5000
```

f. Définition des ports FXS

```
192.168.137.73(cfg)#port fxs 0 0
192.168.137.73(prt-fxs)[0/0]#use profile fxs etsi
192.168.137.73(prt-fxs)[0/0]#encapsulation cc-fxs
192.168.137.73(prt-fxs)[0/0]#bind interface if_fxs00 switch
192.168.137.73(prt-fxs)[0/0]#no shutdown
```

II. Configuration d'Asterisk

Une fois la configuration des différents terminaux est faite, on passe dans une deuxième partie à la configuration des fichiers d'Asterisk. Dans cette partie du travail, puisque nous avons opté pour travailler avec le protocole SIP, nous focalisons le travail sur deux fichiers de configuration à savoir les fichiers sip.conf et extensions.conf.

1. Sip.conf

Le fichier sip.conf file contient les paramètres relatifs à la configuration de l'accès des clients au serveur Asterisk. Les clients doivent être configurés dans ce fichier avant de pouvoir émettre ou recevoir des appels via le serveur Asterisk.

Le fichier sip.conf commence par une section [général], qui contient les paramètres du canal (tel que le port en écoute, les codecs utilisés, etc.) et les options par défaut communes pour tous les utilisateurs définis dans sip.conf. Ces paramètres par défaut sont modifiables pour chaque utilisateur en déclarant les nouvelles valeurs au sein de la définition de chaque utilisateur.

Les principaux paramètres de cette section sont les suivants :

Port = port du protocole SIP, il est conseillé de laisser 5060 (valeur par défaut)

Bindaddr = Adresse IP qu'Asterisk accepte. Par défaut il accepte toutes les adresses IP.

Disallow = Rejette dans un premier temps tous les codecs.

Allow = Accepte dans l'ordre les codecs (ulaw, alaw...)

Localnet = adresse du réseau local et son masque.

Les sections suivantes définissent les paramètres des clients tels que le nom de l'utilisateur, le mot de passe, l'adresse IP. Les sections sont distinguées par des noms entre crochets suivis par les options du client. La partie la plus importante de la configuration des clients est le nom d'utilisateur et le mot de passe pour assurer l'authentification auprès du proxy SIP ainsi que l'adresse IP du client avec laquelle il va s'enregistrer.

Username: Numéro unique de l'extension

Secret : Mot de passe de l'utilisateur

Host : Il est possible de mettre une adresse fixe au client mais si les clients sont connectés derrière un serveur DHCP l'option dynamic est adéquate.

Context: le contexte du fichier extensions.conf où les appels sont gérés

Qualify : Contrôle si le client est joignable tous les x temps.

Callerid : Identification de l'appelant (généralement : "Nom <Num>")

Nat : Averti que le client se trouve derrière un NAT.

Dtmfmode: Il est possible de choisir les tons DTMF. Pour la plupart des clients, l'option rfc2833 est adéquate.

Canreinvite: Cette option est nécessaire pour certains terminaux comme le Cisco ATA 186, pour qu'Asterisk réinvite le client.

Type : Permet de donner les droits à l'utilisateur. On distingue 3 types d'utilisateurs à savoir : les peers, users et friends.

- Le type Peer peut seulement recevoir des appels mais il ne peut pas en émettre.
- Le type user peut seulement émettre des appels mais ne peut pas recevoir.
- Le type Friend regroupe un User et un Peer : il peut émettre et recevoir des appels.

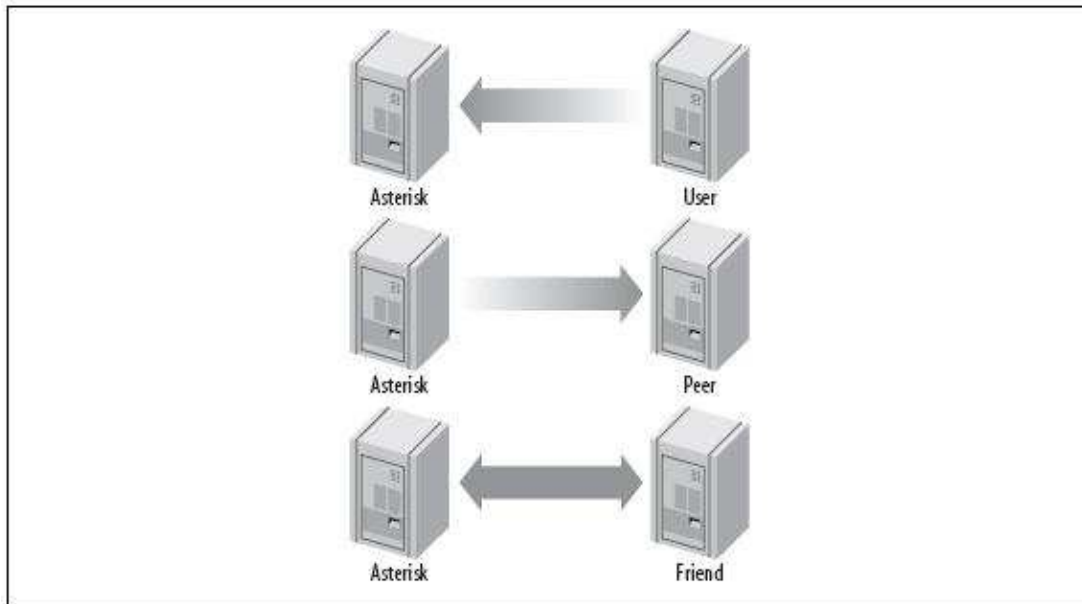


Figure 3. 8: Les types d'utilisateurs

Dans ce qui suit, nous exposons un exemple de configuration du fichier Sip.conf applicable aux différents types de terminaux cités dans la section précédente, on considère dans ce cas le client 2000.

```
[general]
port = 5060
srvlookup=yes
localnet=192.168.137.1/255.255.255.0
disallow=all
allow=alaw
allow=ulaw

[4000]
Username=4000
Secret=0000
Type=friend
Host=dynamic
Context=localtest
Qualify= yes
```

Avec cette configuration, un client SIP peut s'authentifier auprès du serveur AstérisK, mais ne peut pas recevoir d'appel. En effet, il faut lui associer une extension. C'est en quelque sorte une série de règles qu'il faut exécuter lorsque quelqu'un compose un numéro.

```
[root@tte ~]# asterisk -r
Asterisk 1.2.15, Copyright (C) 1999 - 2006 Digium, Inc. and others.
Created by Mark Spencer <markster@digium.com>
=====
Connected to Asterisk 1.2.15 currently running on tte (pid = 3631)
tte*CLI> sip show peers
Name/username          Host                Dyn Nat ACL Port      Status
4000/4000              192.168.137.72    D                5060     OK (11 ms)
1 sip peers [1 online , 0 offline]
tte*CLI> █
```

Figure 3. 9: Enregistrement d'un client sous AstérisK

2. Extensions.conf

Ce fichier définit le plan de numérotation d'astérisK à savoir les différentes règles à suivre pour traiter un appel. Il précise le traitement et le routage des appels entrants et sortants. Le contenu de ce fichier est subdivisé en des sections appelées contextes.

a. Contextes

Les contextes sont des groupes d'extensions. Les contextes sont dénotés en plaçant le nom du contexte entre crochets. Toutes les instructions qui suivent la définition du contexte font partie de ce contexte jusqu'à la définition du contexte suivant. Cette division du plan de numérotation permet d'empêcher différentes parties du plan de numérotation d'interagir entre elles. Une extension qui est définie dans un contexte est complètement isolée des extensions des autres contextes à moins qu'on permette spécifiquement l'interaction. [10]

Dans la pratique cette fonctionnalité sert essentiellement à regrouper les clients possédant les mêmes caractéristiques et ayant accès aux mêmes extensions. C'est en quelque sorte une manière de cloisonner la configuration et d'accroître la sécurité. En effet, en utilisant correctement les contextes, on peut accorder des utilisateurs des privilèges (accès à certains aspects qui ne sont pas accessibles à d'autres). Si le plan de numérotation n'est pas bien conçu, il est possible de permettre l'accès à d'autres utilisateurs qui peuvent utiliser frauduleusement le système.

b. Extensions

Au sein de chaque contexte on définit un ou plusieurs extensions. Une extension est une instruction qu'astérisque va suivre à la réception d'un appel ou à la composition d'un numéro. Les extensions spécifient la suite de traitements que subira un appel tout au long du plan de numérotation.

En téléphonie traditionnelle, les extensions ne peuvent être qu'une combinaison de chiffres qu'on compose pour faire sonner un autre poste. Avec astérisque c'est différent. En effet, une extension peut être désignée par une combinaison alphanumérique. Désigner les extensions par des noms est un concept révolutionnaire surtout qu'avec la VoIP on a tendance à désigner les utilisateurs par leurs noms ou par leurs adresses mail au lieu des numéros. Une extension complète est composée de 3 composants :

- a. Le nom (ou numéro) de l'extension
- b. La priorité (chaque extension peut comporter plusieurs étapes : le numéro de l'étape est appelé priorité)
- c. L'application (ou commande) qui exécute une action sur l'appel.

La syntaxe d'une extension est la suivante :

Exten => extension number (or name), command priority, command

Voici un exemple de la définition d'une extension. Le nom de l'extension est 123, la priorité est égale à 1, et l'application est Answer ().

Exten => 123, 1, Answer ()

c. Priorités

Chaque extension peut avoir plusieurs étapes appelées priorités. Les priorités sont numérotées à partir de 1. Chaque priorité exécute une application spécifique. Dans l'exemple suivant, l'extension répond à l'appel (priorité 1), puis raccroche (priorité 2):

exten => 123, 1, Answer ()

exten => 123, 2, Hangup ()

Il faut s'assurer que les priorités commencent à 1 et qu'elles sont toutes numérotées consécutivement. Si une priorité est sautée, astérisque ne va pas continuer par la suite. Si Astérisque ne suit pas toutes les priorités, il est alors conseillé de vérifier si une priorité n'a pas été sautée ou numérotée de façon erronée. Astérisque suit les priorités dans un ordre numérique.

d. Applications

Les applications sont la partie exécutive des plans de numérotation. Chaque application exécute une action spécifique sur le canal courant, tel que jouer un son, raccrocher la ligne etc...

Dans ce qui suit un exemple de configuration du fichier extensions.conf pour assurer le routage des appels entre les postes 4000 et 4001 ainsi que tester la qualité du son par un test d'écho.

```
[General]
Static=yes
[Localtest]
exten => 4000, 1, Dial (SIP/4000)
exten => 4001, 1, Dial (SIP/4001)
; Boucle d'échos servant de test
exten => 600, 1, Answer ()
exten => 600, 2, Echo ()
```

Une fois toute la configuration est faite, on passe au mode textuel, on passe au mode CLI (Command Line Interface) et on recharge toute la configuration pour prendre en considération les modifications apportées sur les deux fichiers de configuration sip.conf et extensions.conf.

```
# Asterisk -r
CLI> reload
```

Dès lors il est possible de router des appels entre ces deux postes. Le déroulement d'un appel du poste 4001 vers le poste 4000 est montré par la figure suivante :

```
-- Executing Dial("SIP/4001-09abfa90", "SIP/4000") in new stack
-- Called 4000
-- SIP/4000-09ac4fd0 is ringing
== Spawn extension (localtest, 4000, 1) exited non-zero on 'SIP/4001-09abfa90'
```

Figure 3. 10: Exécution d'un appel sous Asterisk

III. Interconnexion des ipLDK via Asterisk

Le travail demandé dans cette partie est d'assurer une communication entre deux postes appartenant à deux ipLDK différents via SIP. Cette tâche prend en considération les différents types de terminaux susceptibles d'être interconnectés à un ipLDK à savoir les postes analogiques, les postes numériques et les postes mobiles (DECT).

Le travail demandé est alors représenté par la figure suivante:

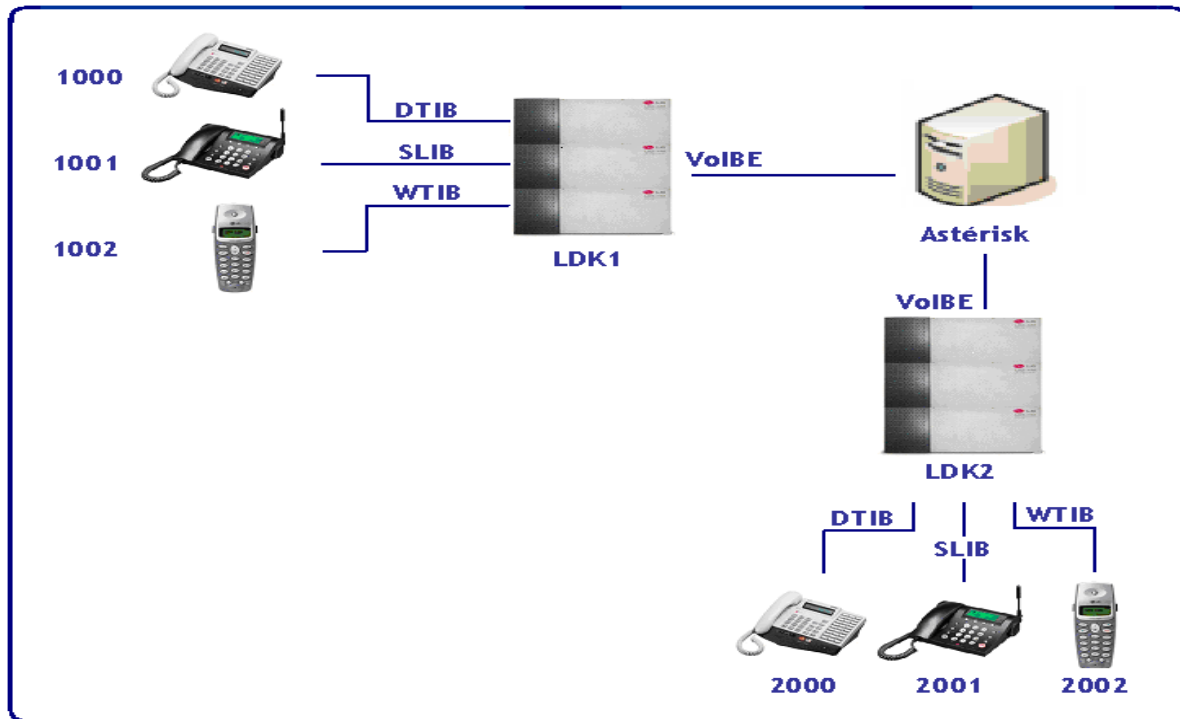


Figure 3. 11 : Schéma de travail demandé

Pour réaliser cette architecture et faire communiquer les différents postes entre eux, deux types de configuration doivent avoir lieu: celle du côté de l’ipLDK et celle du côté d’Astérisik.

1. Configuration de l’ipLDK

Pour commencer, on monte les différentes cartes nécessaires à ce travail, à savoir les cartes SLIB, DTIB, WTIB, et VOIB sur les slots de l’ipLDK (ceci sous entend que la carte MPB est déjà montée). Une fois les cartes sont insérées dans les slots correspondants, on passe à la configuration du l’ipLDK. L’ipLDK est configurable via un logiciel appelé LDK PCADMIN.

Vu qu’il s’agit d’interconnecter deux ipLDK, l’architecture est symétrique. Donc la configuration effectuée sur l’un des ipLDK est identique à celle de l’autre, exception faite au niveau des adresses IP de MPB et de la VoIB ainsi que la plage de numérotation.

	MPB	VOIB
ipLDK1	192.168.137.10	192.168.137.12
ipLDK2	192.168.137.11	192.168.137.13

Tableau 3. 1 : Allocation des adresses aux ipLDKs

Le schéma suivant montre la disposition des différentes cartes sur les slots de l'ipLDK2 :

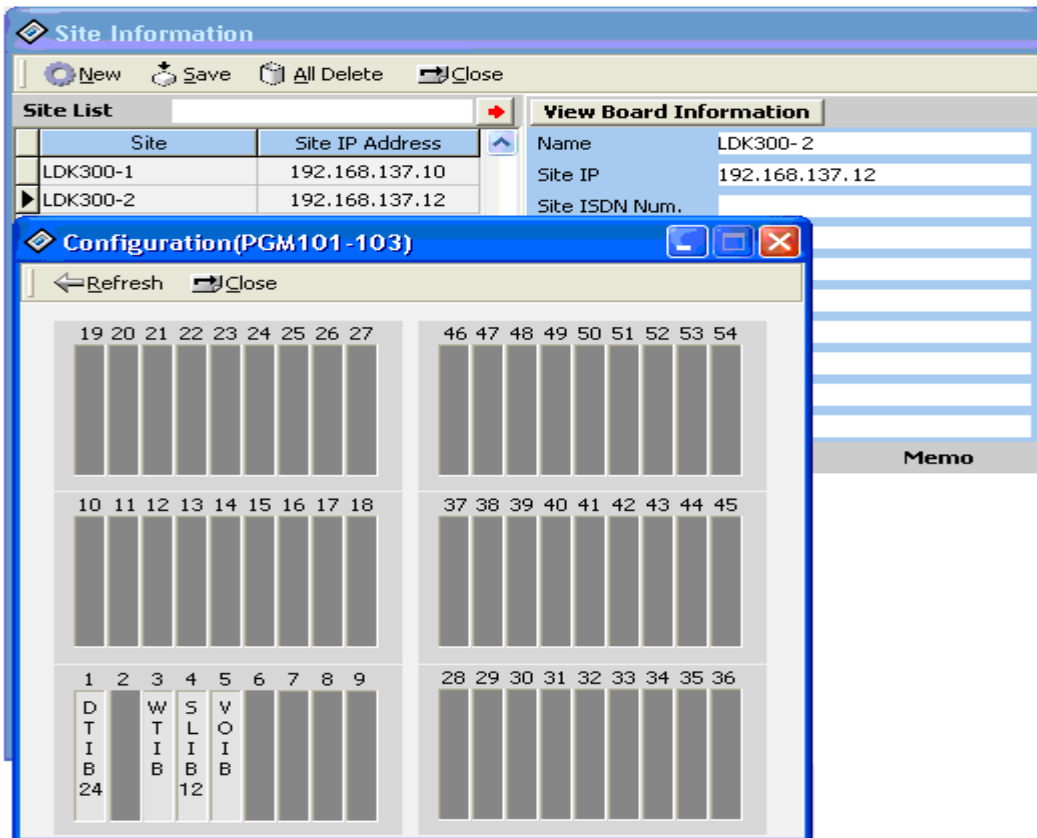


Figure 3. 12 : Disposition des différentes cartes sur le LDK

On passe par la suite à configurer les différents paramètres réseaux pour la carte MPB comme indiqué par la figure suivante :

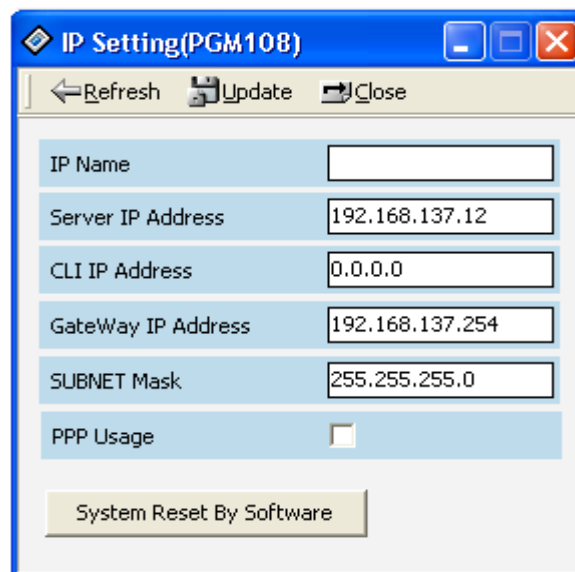


Figure 3. 13 : Configuration de la carte MPB

Maintenant, on passe à la configuration de la carte VoIB, il s'agit essentiellement de lui assigner une adresse IP, le codec utilisé, le protocole et le mode DTMF.

Figure 3. 14 : Configuration de la carte VOIB

La configuration du proxy sip auprès duquel la carte VoIB doit s'enregistrer est montrée par la figure qui suit. En effet, il s'agit en particulier de spécifier l'adresse IP du serveur Astérisik, de spécifier le port et le temps d'enregistrement.

Figure 3. 15 : Configuration du proxy sip

Par la suite on doit préciser les paramètres d’authentification du LDK auprès du serveur AstérisK, à savoir le l’identificateur de l’utilisateur, mot de passe. Il faut noter que l’identificateur doit être de la forme nom@adresseipd’Asterisk pour pouvoir s’enregistrer au près de ce dernier.

Index	User ID	Authentication User Name	Authentication User P	Contact Number	User ID Registr.	User ID Usage	Asc Stn.
1	MonPBX2@192.168.137.144				Register	ON	
2					Provision	OFF	
3					Provision	OFF	
4					Provision	OFF	

Figure 3. 16 : Déclaration des paramètres d’authentification

Ceci étant fait, il ne reste que de préciser la plage de numérotation des postes connectés sur le LDK. Pour le LDK2, la plage de numéro est comprise entre 2000 et 2299 vu que l'ipLDK peut connecter jusqu'à 300 postes.

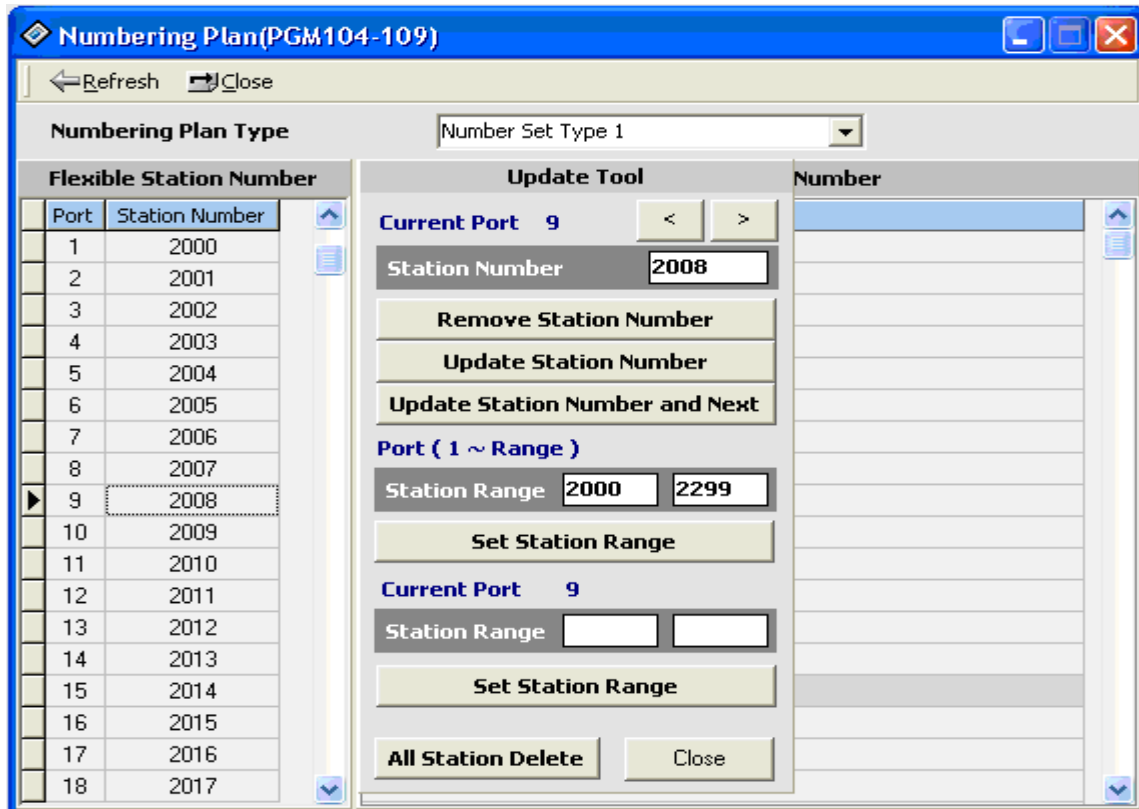


Figure 3. 17 : Déclaration de la plage de numérotation

Il est à noter que la configuration de l'autre ipLDK est similaire à celle-ci, exception faite sur le plan de numérotions qu'on a fixé à 1000 à 1299. Une fois la configuration des deux ipLDK est faite, on peut voir sur la console d'astérisque des requêtes d'enregistrement provenant des deux ipLDKs. Mais cet enregistrement ne peut être effectué qu'après avoir déclaré ces deux ipLDK au niveau d'Astérisque. Ceci est expliqué dans ce qui suit.

2. Configuration d'Astérisque

Comme c'est le cas de la première partie, la configuration minimaliste de l'ipLDK au niveau d'Astérisque doit se faire sur les deux fichiers de configuration sip.conf et extensions.conf.

a. Sip.conf

Pour pouvoir accepter les différents postes connectés sur le LDK (2000 à 2299 et 1000 à 1299) de communiquer via astérisque sans pour autant être obligé de les déclarer poste par poste , on a recours à la commande : 'Allowguest : yes'

Cette commande autorise la connexion au serveur Astérisik à n'importe quel poste appartenant à l'ipLDK qu'on déclarera.

Dans ce qui suit on peut voir un exemple de configuration du fichier Sip.conf. Toujours de la même façon, la configuration de l'autre LDK est similaire, il suffit d'ajouter un bloc qu'on appellera MonPBX2 dans ce même fichier.

```
[general]
Port = 5060
Srvlookup=yes
localnet=192.168.137.1/255.255.255.0
Allowguest=yes
Disallow=all
Allow=alaw
Allow=ulaw

[MonPBX1]
Type=friend
Host=dynamic
Context=loocaltest
Insecure=very
Qualify=yes
```

Une fois qu'on a déclaré les deux PBX au niveau du fichier SIP, leurs authentications avec astérisik prennent lieu. Les deux LDKs sont identifiés par l'adresse IP de la carte VoIP de chacun et non par celle de la carte MPB.

```
tte*CLI> sip show peers
Name/username      Host           Dyn Nat ACL Port      Status
MonPBX2            192.168.137.13 D           5060     OK (30 ms)
MonPBX1            192.168.137.11 D           5060     OK (14 ms)
```

Figure 3. 18 : Enregistrement des ipLDK au niveau d'Astérisik

Maintenant, pour pouvoir émettre et recevoir des appels, il faut ajouter les règles nécessaires au niveau du fichier extensions.conf.

b. Extensions.conf

Au lieu de répéter les mêmes commandes pour tous les postes sur la plage 2000 à 2299, il est possible d'agréger le plan de numérotation. Il s'agit de préfixer le nom de l'extension par '_', pour qu'il soit interprété comme un schéma d'extensions.

Quelques caractères ont une signification particulière :

- X - pour désigner n'importe quel chiffre compris entre 0-9
- Z - pour désigner n'importe quel chiffre compris entre 1-9
- N - pour désigner n'importe quel chiffre compris entre 2-9
- [1235-9] pour désigner n'importe quel chiffre compris entre les crochets.

Dans cet exemple, 1, 2, 3, 5, 6, 7, 8, 9).

Ainsi, pour désigner les postes compris entre 2000 et 2299, il suffit de mettre _20[0-2]X pour pouvoir appliquer cette extension pour n'importe quelle poste.

```
[general]
Static=yes
[loocaltest]
Exten => _10[0-2]X,1,Dial(SIP/${EXTEN}@MonPBX1)
Exten => _20[0-2]X,1,Dial(SIP/${EXTEN}@MonPBX2)
```

Après avoir fait cette configuration, les postes des deux LDK peuvent s'appeler entre eux et peuvent même appeler les autres postes qu'on a déjà configurés dans la première partie.

```
-- Executing Dial("SIP/MonPBX2-09abfa90", "SIP/1000@MonPBX1") in new stack
-- Called 1000@MonPBX1
-- SIP/MonPBX1-09ac4fd0 is ringing
-- SIP/MonPBX1-09ac4fd0 answered SIP/MonPBX2-09abfa90
-- Attempting native bridge of SIP/MonPBX2-09abfa90 and SIP/MonPBX1-09ac4fd0
== Spawn extension (loocaltest, 1000, 1) exited non-zero on 'SIP/MonPBX2-09abfa90'
```

Figure 3. 19: Déroulement de communication entre deux postes appartenant à des ipLDK différents

```
-- Executing Dial("SIP/MonPBX2-09ac7440", "SIP/4000") in new stack
-- Called 4000
-- SIP/4000-09aca060 is ringing
-- SIP/4000-09aca060 answered SIP/MonPBX2-09ac7440
-- Attempting native bridge of SIP/MonPBX2-09ac7440 and SIP/4000-09aca060
== Spawn extension (loocaltest, 4000, 1) exited non-zero on 'SIP/MonPBX2-09ac7440'
```

Figure 3. 20 : Déroulement de communication entre un poste IP et un poste d'un LDK

Conclusion

Dans ce chapitre, on a utilisé astérisk en tant que commutateur d'entreprise et en tant que moyen d'interconnexion des PBX matériels. Puisque Astérisk permet de remplacer un PABX matériel, il est nécessaire de noter qu'il offre la possibilité de bénéficier des services qu'offrent les commutateurs d'entreprises (musique d'attente, conférence, messagerie vocale, envoi des messages vocaux par email, etc..). Astérisk propose plusieurs milliers de fonctions normalement disponibles uniquement sur des autocommutateurs propriétaires. Dans le chapitre suivant, on s'intéressera à la configuration de quelques services sous astérisk et à quelques tests sur le réseau mis en place.

Chapitre IV : performances de la solution mise en place

Introduction

Puisqu'on a mis en place asterisk en tant que PBX, il est donc nécessaire de configurer les services qui sont généralement offerts par les PBX traditionnels et de tester la qualité de service assurée par ce dernier. Pour ce faire, on s'intéressera dans un premier temps à configurer les services de base que peuvent offrir un PBX à savoir le voicemail et le transfert d'appel. Dans le deuxième chapitre, on procédera à tester la qualité de service de la solution mise en place.

I. Configuration des services

1. Voicemail

a. Définition

La messagerie vocale (ou Voicemail) est un système centralisé de gestion des messages téléphoniques pour un grand nombre d'utilisateur d'un même groupe. Dans sa forme basique il imite les fonctions du répondeur téléphonique. Asterisk a un système de voicemail très flexible. Le service Voicemail d'Asterisk permet de gérer des messages vocaux lorsqu'un utilisateur n'est pas en mesure de répondre à un appel et permet à ce dernier d'accéder à ses messages depuis n'importe quel endroit. Les utilisateurs peuvent consulter leur boîte vocale depuis des postes téléphoniques ou depuis leurs comptes de messagerie (email). En effet, avec Asterisk le voicemail propose plusieurs services:

- Ecoute, sauvegarde et suppression des messages
- Création d'un menu d'accueil personnalisé
- Transfert des messages sur une autre boîte vocale
- Réception de message en pièce attaché sur un compte de messagerie.

b. Activation du voicemail sous astérisk

La mise en place de la forme basique du service voicemail revient principalement à modifier les trois fichiers de configuration :

- /etc/Astérisk/extensions.conf
- /etc/Astérisk/voicemail.conf
- /etc/Astérisk/features.conf

Pour chaque utilisateur désirant bénéficier de ce service, une ligne contenant ses informations personnelles doit être insérée dans le fichier voicemail.conf. En effet, pour posséder un voicemail, il suffit d'activer cette option lors de la création d'une extension. La syntaxe est la suivante :

<mailbox>=<password>,<name>,<email>

Exemple :

```
[general]
include vm_email.inc
include vm_general.inc
[default]
4001=>1234,4001,rabiaa.guemri@gmail.com
```

Au niveau du fichier de configuration extensions.conf, on montre la suite de règle que doit suivre l'appel pour passer à la messagerie vocale. L'acheminement de l'appel vers la messagerie vocale dépend de l'état de l'extension en question à savoir la valeur de la variable 'Exten- $\{\text{DIALSTATUS}\}$ ' : NOANSWER, BUSY, CONGESTION, CHANUNAVAIL. Selon cet état, un message d'accueil de la boîte vocale sera joué indiquant à l'appelant l'état de l'appelé et lui demandant de lui laisser un message.

```
exten => 4001,1,Dial(SIP/4001,60,Tt)
exten => 4001,2,Goto(4001- $\{\text{DIALSTATUS}\}$ ,1)
exten => 4001-NOANSWER,1,Voicemail(u4001@default)
exten => 4001-BUSY,1,Voicemail(b4001@default)
exten => 4001-CONGESTION,1,Voicemail(b4001@default)
exten => 4001-CHANUNAVAIL,1,Voicemail(u4001@default)
exten => 4001,3,PlayBack(vm-goodbye)
```

Pour pouvoir consulter ses messages vocaux, l'utilisateur doit posséder un numéro de boîte vocale. La déclaration de ce numéro se fait au niveau du fichier extensions.conf de la façon suivante :

```
exten => 8000,1,Answer()
exten => 8000,2,VoicemailMain(4001@default)
exten => 8000,4,Hangup()
```

Avec cette déclaration, lorsque l'utilisateur essaye de consulter sa boîte vocale en composant le numéro 8000, le système lui demande d'entrer son mot de passe qu'on a déjà déclaré dans le fichier voicemail.conf (qui est 1234 dans l'exemple ci-dessus).

Ce système de voicemail fournit un répondeur interactif pour la gestion et la consultation des messages vocaux. Par défaut la langue est l'anglais mais il est possible de la changer. En effet, il suffit de télécharger les packages de voix en langue française et de les compiler en effectuant les commandes suivantes :

```
wget http://www.sineapps.com/down/FrenchPrompts.tar.gz
tar -xvzf FrenchPrompts.tar.gz -C /var/lib/asterisk/sounds
```

Editer le fichier sip.conf et ajouter cette ligne au niveau du context [general] :

```
Language=fr
```

Editer dans le fichier indications.conf, et ajouter dans le context [general] :

```
Country=fr
```

Asterisk offre la possibilité de notifier les utilisateurs par mail de l'arrivée des messages vocaux. Le message vocal laissé peut être mis en pièce jointe avec l'email de notification envoyé. Pour pouvoir envoyer des notifications par mail aux utilisateurs qui viennent de recevoir des messages vocaux, deux étapes doivent être effectuées : la configuration de l'agent de transport de courrier (MTA), dans notre cas d'étude on a travaillé avec le Sendmail. Dans un second lieu, il faut créer les fichiers de configuration etc/asterisk/vm_email.inc et etc/asterisk/vm_general.inc.

c. Activation du Sendmail

Dans ce travail, j'ai utilisé Sendmail comme MTA pour pouvoir envoyer des notifications par mail. Il faut donc choisir Sendmail comme MTA par défaut (paramètres de système=>commutateur d'agent de transport de courrier=>Sendmail).

En premier lieu, il faut installer le package de Sendmail sur la machine sur laquelle est installé AstérisK. Ceci peut être réalisé par la commande suivante:

```
[root@localhost ~]# yum install Sendmail-cf
```

La deuxième étape consiste à éditer le fichier `/etc/mail/sendmail.mc` (par l'éditeur de texte `vi` ou `nano` ou autre par la commande `vi /etc/mail/sendmail.mc`). Je ne montre ici que la partie du fichier qui nous intéresse :

```
dnl # define(`SMART_HOST', `smtp.your.provider')
```

Maintenant, on change le nom serveur SMTP "smtp.your.provider" par le nom du serveur SMTP de la société, dans ce cas, "smtp.plante.tn". Ce qui donne comme alors :

```
define (`SMART_HOST', `smtp.plante.tn')
```

Pour que ces modifications prennent effet, il faut taper la commande suivante

```
[root@localhost mail]# make -C /etc/mail
```

Ceci étant fait, on passe par la suite à la configuration du fichier `/etc/hosts` qui contient par défaut cette configuration:

```
127.0.0.1    localhost
127.0.0.1    localhost.localdomain asterisk1.local
```

Il suffit de changer la configuration comme suit :

```
127.0.0.1    localhost
192.168.137.144  tte.com asterisk1.local
```

Où l'adresse 192.168.137.144 est celle de mon serveur AstérisK (à savoir celle du poste actuel) et tte.com est un nom de machine que j'ai attribué à ce poste.

Une fois ceci est fait, on sauvegarde les modifications, et on passe au fichier `/etc/resolv.conf` pour s'assurer de l'existence des deux serveurs DNS de la société.

```
nameserver 193.95.66.10 ; DNS primaire
nameserver 193.95.93.77 ; DNS secondaire
```

Une fois toute la configuration est faite, on redémarre le service réseau:

```
[root@localhost ~]# service network restart
Définition des informations IP pour eth0. Terminé.      [OK]
```

Pour tester que la configuration est bonne on peut essayer

Une fois la configuration du Sendmail est effectuée, on passe à la configuration d'asterisk à savoir les deux fichiers : vm_email.inc et vm_general.inc (qui se trouvent dans le dossier /etc/Asterisk).

d. Configuration de vm_email.inc et vm_general.inc

Le fichier vm_email.inc est le fichier de configuration de la mise en forme de l'email. En effet la mise en forme du mail prend en considération les valeurs de plusieurs variables :

- **VM_NAME:** Variable où est stocké le nom du destinataire
- **VM_MAILBOX:** Numéro du Voicemail du destinataire, le plus souvent ce numéro correspond à son extension.
- **VM_CALLERID:** si l'utilisateur est enregistré dans Asterisk le prénom, nom ainsi que le numéro de téléphone sont affichés. Si l'appelant n'est pas reconnu, seul le numéro de téléphone est affiché.
- **VM_DUR:** Variable où est stockée la durée du message en seconde.
- **VM_DATE:** La date de la réception du message vocal.

L'email peut être configuré comme bon nous semble, n'importe quel texte peut être inséré. Dans ma configuration, les coordonnées de TTE sont ajoutées dans l'email.

```
emailbody=${VM_NAME},\n\n Vous avez reçu un nouveau Voicemail sur
${VM_MAILBOX}:\n\n\t
From: \t${VM_CALLERID}\n\t
Durée:\t${VM_DUR} secondes\n\t
Date:\t${VM_DATE}\n\n
© TTE Tunisian Telecoms and Electronics \n
16. Rue des metiers charguiya 2 \n
```

Le deuxième fichier, vm_general.inc, permet de configurer le voicemail de façon plus technique. En effet, plusieurs variables doivent être déclarées :

- **Format:** Format du fichier audio.
- **Attach:** Autorise à joindre le fichier audio.
- **PBXskip:** Doit être activé si on veut omettre [PBX] de l'objet du mail.
- **Serveremail:** Le nom attribué au système lors de la réception de l'email.

- **Fromstring:** Texte affiché dans l'objet du mail.
- **Maxmessage:** Longueur maximale, en secondes, du message vocal.
- **Minmessage:** Longueur minimale, en secondes, du message vocal.
- **Maxsilence:** Temps d'attente, en secondes, après le message vocal.
- **Silencethreshold:** Ce que le système considère comme silence.
- **Skipms:** Le temps d'attente, en millisecondes, pour réécouter le message vocal.
- **Review:** Permet de réécouter le message si besoin.
- **Operator :** Permet à l'utilisateur de presser 0 pour sortir du voicemail.

```
Format=wav49|wav
Attach=yes
PBXskip=yes
Serveremail=Voicemail@tte.com.tn
Fromstring=Messagerie vocale
Maxmessage=180
Minmessage=3
Maxsilence=5
Silencethreshold=128
Skipms=3000
Review= no
Operator=yes
```

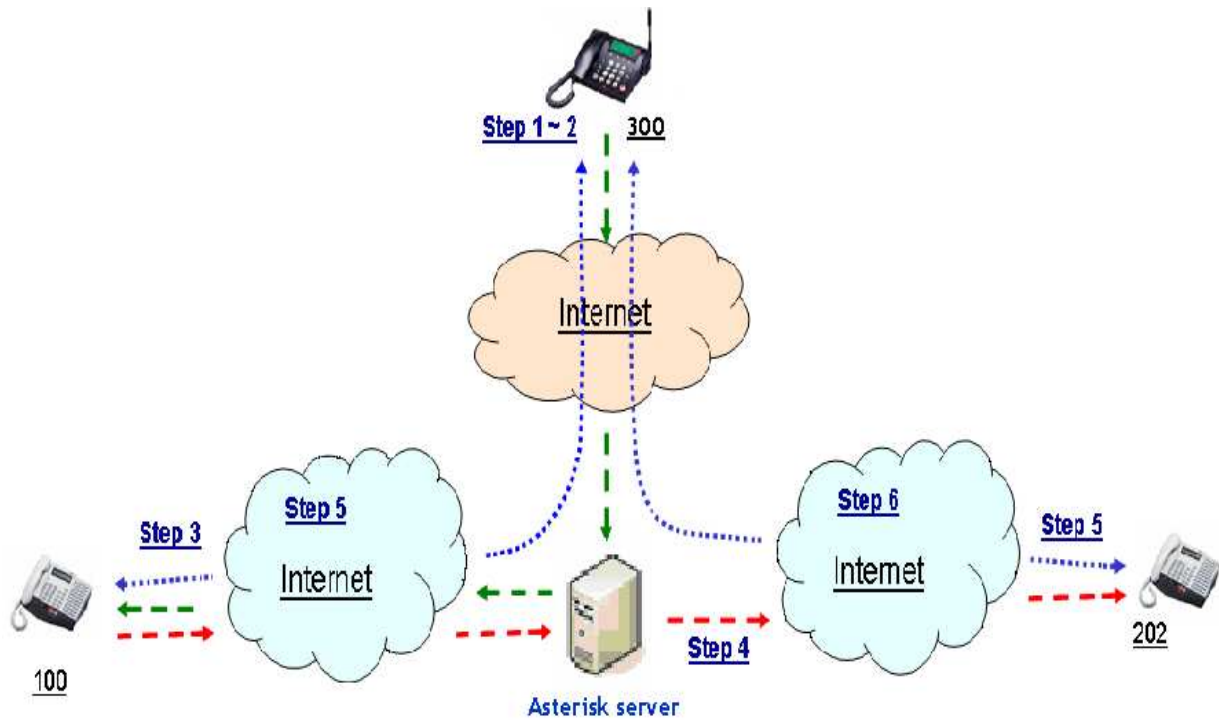
2. Transfert d'appel

a. Définition

Il est possible pour une station de transférer des communications vers une autre station qui peut être dans un système autre que celui auquel elle appartient. On distingue deux types de transfert de communication selon si le destinataire du transfert est au courant ou non du transfert :

✚ **Transfert supervisé:** lorsque la station qui a reçu l'appel entre en communication avec la station destinataire pour lui annoncer l'arrivée d'un appel avant d'effectuer le transfert de la communication.

✚ **Transfert aveugle:** lorsque la station qui a reçu l'appel effectue le transfert sans qu'elle n'entre en communication avec la station destinataire.



Etape	Description
1	l'appelant 300 appelle la station 100 du système.
2	L'appel arrive, la station 100 sonne.
3	La station 100 répond
4	La station 100 transfère l'appel vers la station 202, et raccroche
5	La station 202 sonne et répond
6	Conversation entre 300 et 202

Figure 4. 1: Déroulement du transfert aveugle

b. Activation du transfert d'appel

L'activation du transfert d'appel sous astérisque demande la configuration des deux fichiers de configuration extensions.conf et features.conf.

Concernant le fichier extensions.conf, il suffit d'ajouter les options t et T dans l'application Dial(). Ces options activent le transfert d'appels pour les deux postes appelant et appelé :

- t : permet à l'utilisateur appelé de transférer l'appel en composant la suite de digits définie dans le fichier features.conf.
- T : permet à l'utilisateur appelant de transférer l'appel en composant la suite de digits définie dans le fichier features.conf.

Exemple : exten => 4000, 1, Dial (SIP/4000, 60, Tt)

La configuration des paramètres du transfert se fait au niveau du fichier `features.conf`, il s'agit principalement de préciser la séquence à composer pour effectuer chaque type de transfert, et les différents timeouts ainsi que les sons à jouer:

- **blindxfer** : précise la séquence à composer pour effectuer un transfert aveugle
- **atxfer** : indique la séquence à composer pour effectuer un transfert supervisé.
- **Transferdigittimeout**: la durée en seconde à attendre entre la composition des différents digits lors du transfert.
- **Xfersound** : pour indiquer que le transfert est effectué.
- **Xferfailsound** : pour indiquer que le transfert a échoué.
- **Atxfernoanswertimeout**: durée à attendre avant d'annuler le transfert supervisé.
- **Parkedmusicclass**: précise le morceau de musique à jouer comme musique d'attente lors du transfert. La valeur par défaut jouera le morceau qui se trouve sous le dossier précisé par le fichier `musiconhold.conf`.

```
[general]
transferdigittimeout => 3
xfersound = beep
xferfailsound = beeperr
atxfernoanswertimeout = 15
parkedmusicclass=default
[featuremap]
blindxfer => #1
atxfer => *2
```

c. Activation du parcage d'appel

Le parcage d'appel (call parking) est un service dont le principe repose sur le transfert d'appel. Il s'agit d'un transfert supervisé sauf que pour le parcage, le transfert ne s'effectue pas vers des extensions réelles mais vers des extensions virtuelles qui ne sont pas déclarées au niveau du fichier `sip.conf`. L'utilisateur peut composer l'identifiant de ce poste de n'importe quelle extension déclarée sous astérisque et reprendre la communication mise en attente. La configuration du parcage d'appels est similaire à celle du transfert d'appels à l'exception de quelques configurations supplémentaires qui doivent être effectuées en particulier au niveau du fichier `features.conf`:

▪ **parkext**: indique la séquence de digit que doit composer l'utilisateur pour effectuer le parage d'appel. Quand l'utilisateur compose cette séquence, le système lui fournit le numéro du poste vers lequel l'appel a été transféré. Ces valeurs doivent être numériques puisque astérisque commence le parage des appels par la première valeur numérique et incrémente d'un pour chaque appel parqué.

▪ **Context** : précise le contexte où les appels parqués se trouvent.

▪ **Parkingtime** : précise la durée pendant laquelle l'appel parqué peut rester en attente avant d'être repris.

Ce qui suit est un exemple de configuration qu'on doit ajouter à la configuration du transfert aveugle et supervisé au niveau du fichier features.conf.

```
[general]
parkext => 700
parkpos => 701-720
context => parkedcalls
parkingtime => 45
```

Pour que les extensions déclarées au niveau du fichier sip.conf puissent effectuer le parage d'appel, il faut que le contexte indiqué au niveau du fichier features.conf soit inclus dans le même contexte que celui des postes en communication. Ceci peut être fait au niveau du fichier extensions.conf par la commande suivante :

```
Include => parkedcalls
```

d. Activation de la musique d'attente

Pour utiliser le service de musique d'attente lors du transfert ou du parage d'appel, il est nécessaire de configurer le fichier musiconhold.conf. Une configuration simpliste de ce fichier consiste à préciser le mode qui indique le type de fichier à jouer et le répertoire où sont placés les morceaux joués.

Dans le dossier précisé, on place le morceau de musique d'attente personnalisé sous le répertoire /var/lib/asterisk/mohmp3 afin d'être utilisé par défaut lors du transfert ou du parage.

```
[default]
mode=quietmp3
directory=/var/lib/asterisk/mohmp3
```

3. Conférence téléphonique

Astérisik offre la possibilité de créer des conférences téléphoniques entre les utilisateurs. Pour ce faire, il faut configurer les fichiers de configuration `meetme.conf` et `extensions.conf`. Concernant le fichier `meetme.conf`, on déclare les différentes conférences. La syntaxe de création d'une conférence est :

Conf => <confno>, <pin>

```
[general]
audiobuffers=32
[rooms]
conf => 400,1234
```

Pour le fichier `extensions.conf`, on peut préciser les différentes étapes à suivre pour accéder à une conférence.

La commande qui permet d'entrer dans une conférence téléphonique est :

Exten=> <roomnumber>, priority, MeetMe([confno],[options],[pin]])

Les options peuvent prendre une ou plusieurs des valeurs suivantes:

- 'M' – jouer la musique d'attente quand il y a un seul utilisateur dans la conférence.
- 'm' – active le mode monitor (écouter sans intervenir dans la conférence)
- 'p' – permet aux utilisateurs de quitter la conférence en appuyant sur '#'
- 'i' – annoncer l'entrée et la sortie des utilisateurs dans une conférence.
- 'q' – mode silencieux (ne pas jouer des alertes d'entrée sortie)

```
exten => 400,1,PlayBack(there-are)
exten => 400,2,MeetMeCount,400
exten => 400,3,PlayBack(personnes-conf)
exten => 400,4,MeetMe(400|1234)
exten => 400,5,Playback(vm-goodbye)
```

II. Tests et qualité de service

Lorsqu'on parle de téléphonie, les utilisateurs sous-entendent une qualité de service irréprochable. La téléphonie IP, n'est pas au niveau des 99,999% de la QoS de la téléphonie fixe, et ne l'atteindra probablement jamais.

Le terme QoS pour la téléphonie fait référence à la qualité du son. En effet, la qualité de service de la VoIP dépend de plusieurs facteurs. Il s'agit essentiellement du codec choisi. Le choix des codecs s'avèrera quand même important quand le nombre de communications simultanées devient important. Plus de compression? Plus de débit? Il n'y a pas de codec parfait, et si l'on réduit le débit, la qualité de la voix va probablement s'en ressentir. Le choix du codec est un compromis entre la qualité de service souhaitée et la capacité de l'infrastructure IP. Le paramètre le plus déterminant est la bande passante que l'on met en regard du nombre de communications simultanées à écouler.

Le choix du codec G.711 permet de bénéficier à réseau constant de la meilleure qualité de service, tandis que les compressions G.726, G.728, G.729 et G.723 apportent avec elles des diminutions initiales de la QoS.

Puisqu'on a testé la solution mise en place sur un LAN, il est évident qu'on dispose d'une bande passante assez suffisante pour assurer un nombre important d'appels simultanés, nous choisissons alors le codec qui privilégie la qualité à savoir le codec G.711.

Dans ce qui suit, on s'intéressera à l'étude de quelques indicateurs de QoS pour la solution mise en place. En premier lieu on testera les différents indicateurs pour un seul appel. Dans une deuxième partie, puisqu'il est difficile de mettre en place un grand nombre de terminaux SIP pour le test, on aura recours à des outils de test pour simuler un trafic RTP sur le réseau. Il s'agit d'utiliser une pondeuse d'appel pour générer des appels SIP vers le serveur AstérisK.

1. Indicateurs à évaluer

a. Echelle MOS

La première approche pour mesurer la qualité d'une restitution sonore est de faire appel à la méthode MOS. MOS est l'abréviation de Mean Opinion Score, ou Note d'opinion moyenne. Elle s'applique aussi bien à la téléphonie traditionnelle qu'à la ToIP. Il s'agit d'une note donnée à un codec audio pour caractériser la qualité de la restitution sonore. La note peut varier entre 1 (mauvais) et 5 (excellent, comparable à la version d'origine). Cette technique est issue de la téléphonie analogique.

Qualité de l'appel	score
Excellente	5
Bonne	4
Moyenne	3
Dégradée	2
Mauvaise	1

Tableau 4. 1: MOS et qualité de transmission de la voix

b. Facteur R

Ce facteur mesure la qualité de la voix par une valeur numérique entre 0 et 93. La valeur de ce champ prend en considération la perte de trame, la gigue et d'autres facteurs réseau. Le tableau suivant présente la correspondance entre les valeurs de R et la qualité de la voix transmise :

Qualité de la voix	score
excellente	90-93
bonne	80-90
moyenne	70-80
dégradée	60-70
mauvaise	0-60

Tableau 4. 2: Coefficient R de la qualité de transmission de la voix

c. La gigue

La gigue est la Moyenne de variation instantanée du retard pour les trames reçues. La gigue est une moyenne d'exécution de la valeur absolue de la variation. Plus encore que les délais eux-mêmes, la variation de ces délais impacte la qualité de la voix. L'oreille humaine est très sensible à ces problèmes de gigue. Au point que les conversations peuvent devenir inintelligibles. N'oublions pas que, pour des raisons d'efficacité, les protocoles d'échange de la VoIP se basent sur le couple RTP/UDP. Cela signifie que le réseau ne garantit absolument pas ni les délais d'acheminement, ni l'ordre, ni le même chemin pour tous les paquets, ni le bon acheminement... La responsabilité incombe aux différents équipements qui induisent un délai variable, dû par exemple, à des problèmes de congestion réseau ou tout simplement des problèmes de fiabilité des réseaux. La valeur de la gigue va de quelques ms à quelques dizaines de ms.

d. La perte de données

La transmission de la voix par paquets s'appuie sur le protocole RTP (real-time transport protocol). Ce dernier permet de transmettre sur IP les paquets de voix en reconstituant les informations même si la couche de transport change l'ordre des paquets. Il utilise pour cela des numéros de séquence et s'appuie sur UDP.

Les contraintes temps réel de délai de transit rendent inutile la retransmission des paquets perdus : même retransmis un datagramme RTP arriverait bien trop tard pour être d'une quelconque utilité dans le processus de reconstitution de la voix. En voix sur IP on ne retransmet donc pas les données perdues. Ces pertes de données VoIP sont dues aux congestions sur le réseau, qui entraînent des rejets de paquets tout au long du réseau, ou à une gigue excessive qui va provoquer des rejets de paquet.

Une perte de données régulière mais faible est moins gênante en voix sur IP que des pics de perte de paquets espacés mais élevés. En effet l'écoute humaine s'habitue à une qualité moyenne mais constante et en revanche supportera peu de soudaines dégradations de la QoS.

2. Tests sur un appel

Pour effectuer les différents tests et mesures de la QoS, on a opté pour le logiciel autoVoIP qui permet de capturer le flux de données circulant sur l'interface réseaux du poste sur lequel il est installé et de présenter les différentes mesures sous formes de graphes. Pour commencer, on effectuera les différents tests indiqués ci-dessus sur un seul appel. On devrait donc avoir recours à un softphone pour émettre l'appel du poste sur lequel est installé le logiciel de test. L'appel est effectué vers un poste connecté à un ipLDK. Cet appel émane du softphone dont l'identifiant est 4006 et dont l'adresse IP est 192.168.137.22 vers le poste 1000 appartenant à l'ipLDK MonPBX2.

Les deux figures suivantes montrent bien que l'appel s'est bien déroulé entre les deux postes qui sont bien enregistrés auprès du serveur Astérisk. Vu qu'il s'agit d'un seul appel, alors la bande passante est largement suffisante pour établir cet appel avec une qualité excellente. Ceci s'avère à partir des différents indicateurs déjà cités :

- R = 93
- MOS = 4.4

Outre la bande passante disponible ceci est aussi bien dû au codec utilisé (G.711) qui assure la meilleure qualité de service en VoIP.

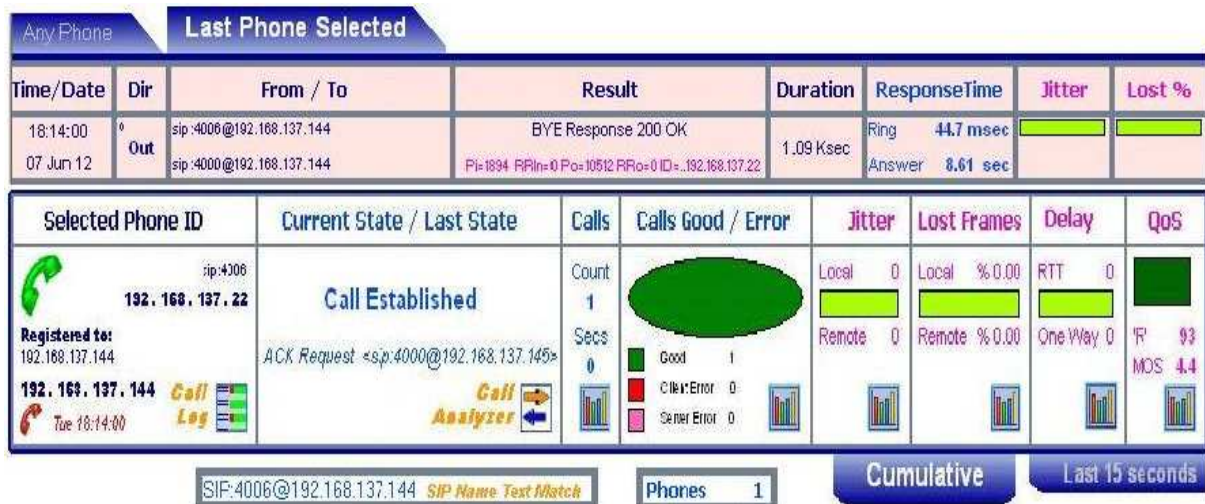


Figure 4. 2 : Tableau de bord du déroulement de l'appel

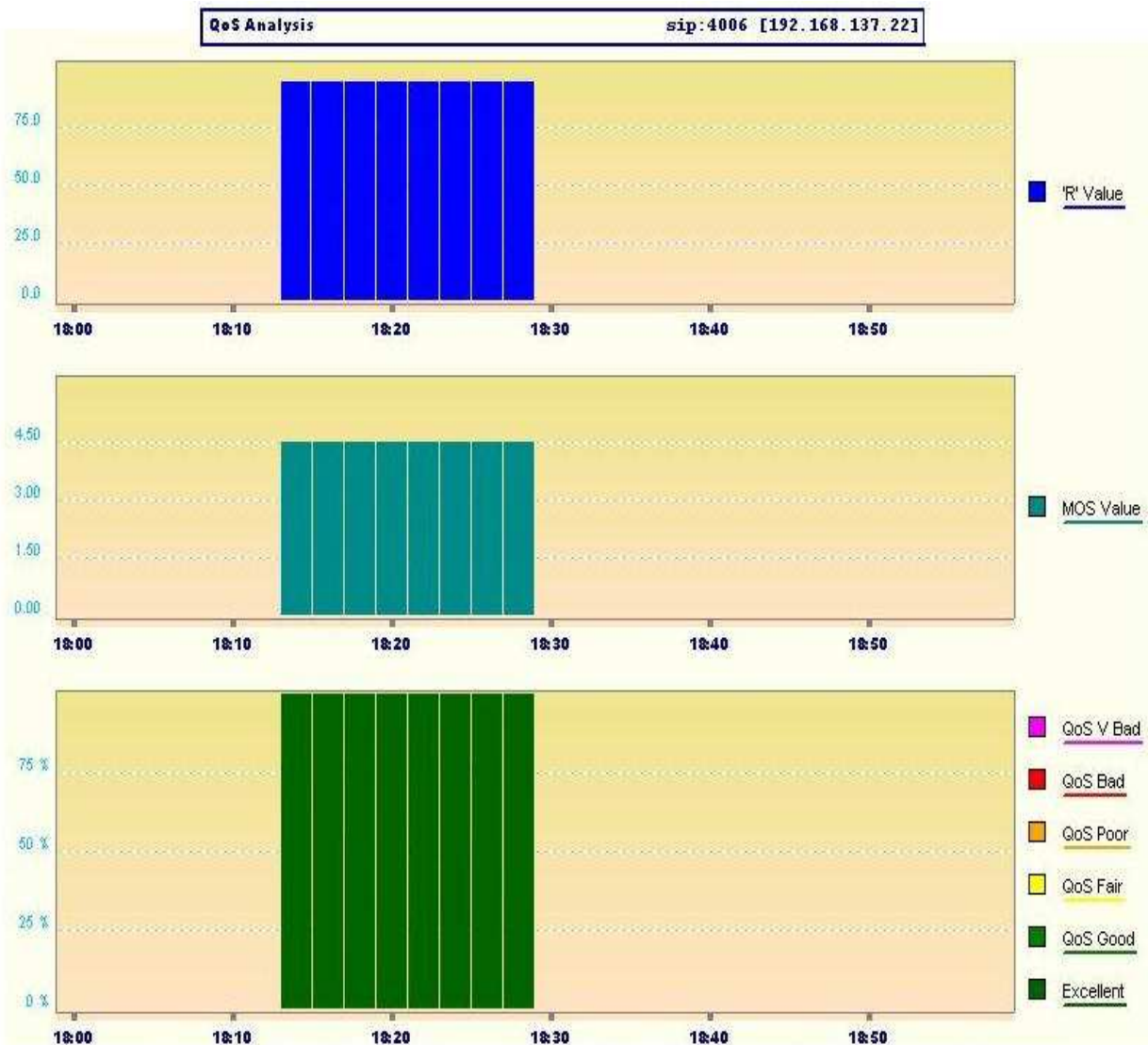


Figure 4. 3: Valeurs des indicateurs MOS et R

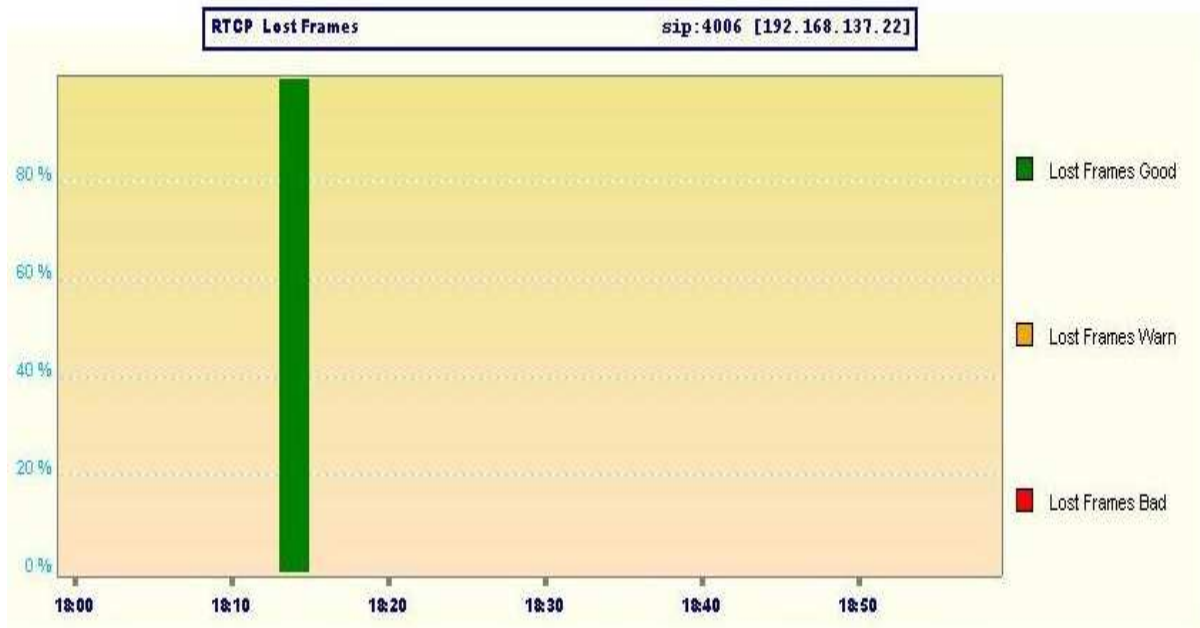


Figure 4. 4 : Taux de paquets RTCP perdus

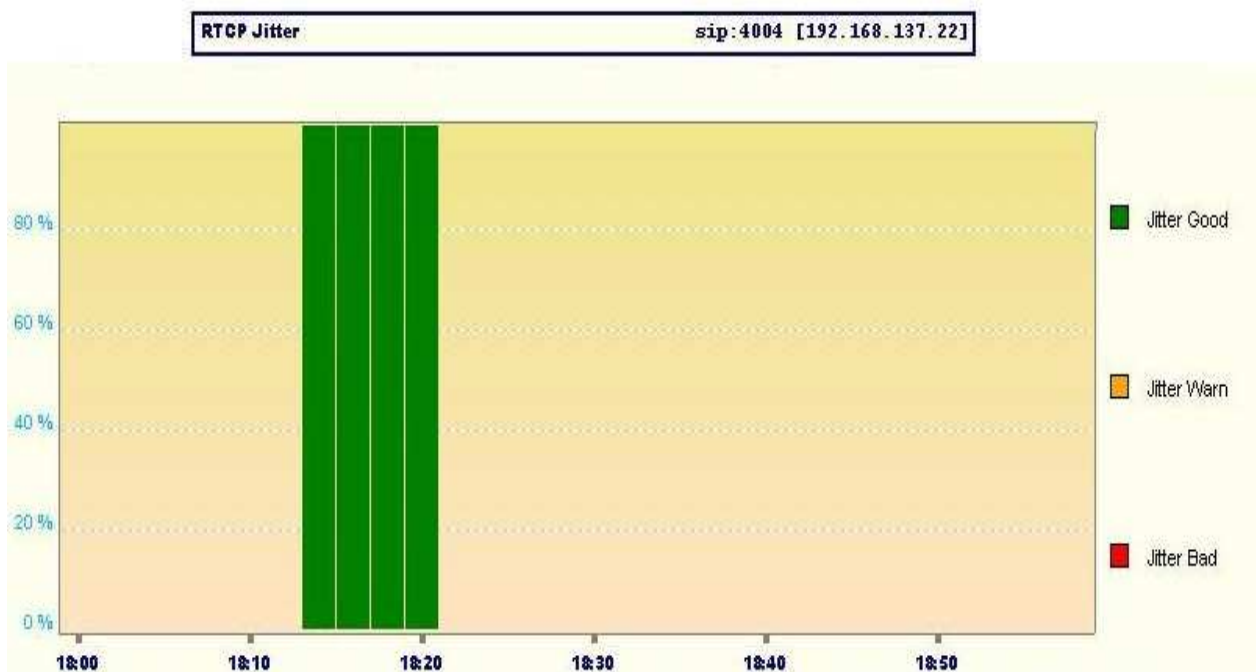


Figure 4. 5 : Gigue des paquets RTCP

On remarque que le logiciel montre que la qualité de l'appel est excellente de point de vue MOS et R qui sont presque égaux à leurs valeurs maximales. La bonne qualité de service s'avère aussi à partir de la valeur de la gigue qui est presque nulle. En outre le nombre de paquets perdus est trop faible voire même nul. La gigue des paquets de RTCP indique aussi de bonnes valeurs.

La figure suivante montre que le flux qui passe par le serveur astérisik est essentiellement un flux d'enregistrement. En outre les événements signalés sont tous des événements de redirection des appels.

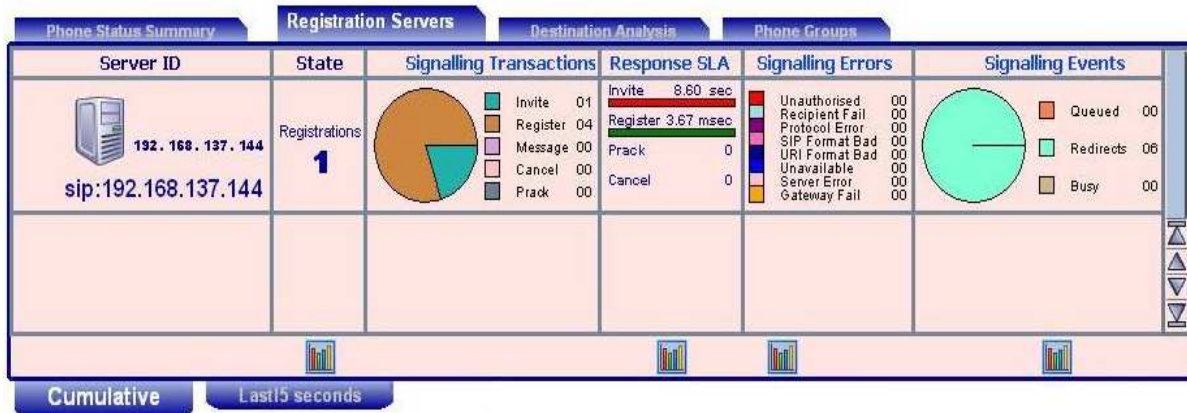


Figure 4. 6 : Etat du serveur astérisik à la réception de l'appel

3. Test sur 50 appels

Dans cette partie, on veut essayer de tester 50 appels simultanés gérés par le serveur Astérisik. Mais puisqu'on ne pouvait pas tester à la fois 50 terminaux physiques sur le réseau, on a alors recours à l'utilisation d'une pondeuse d'appel qui permet de générer des appels SIP. Pour ce faire, on a utilisé le logiciel Winsip comme pondeuse d'appel. Il fallait l'installer sur deux postes du réseau, un poste pour initier les appels et l'autre pour les recevoir.

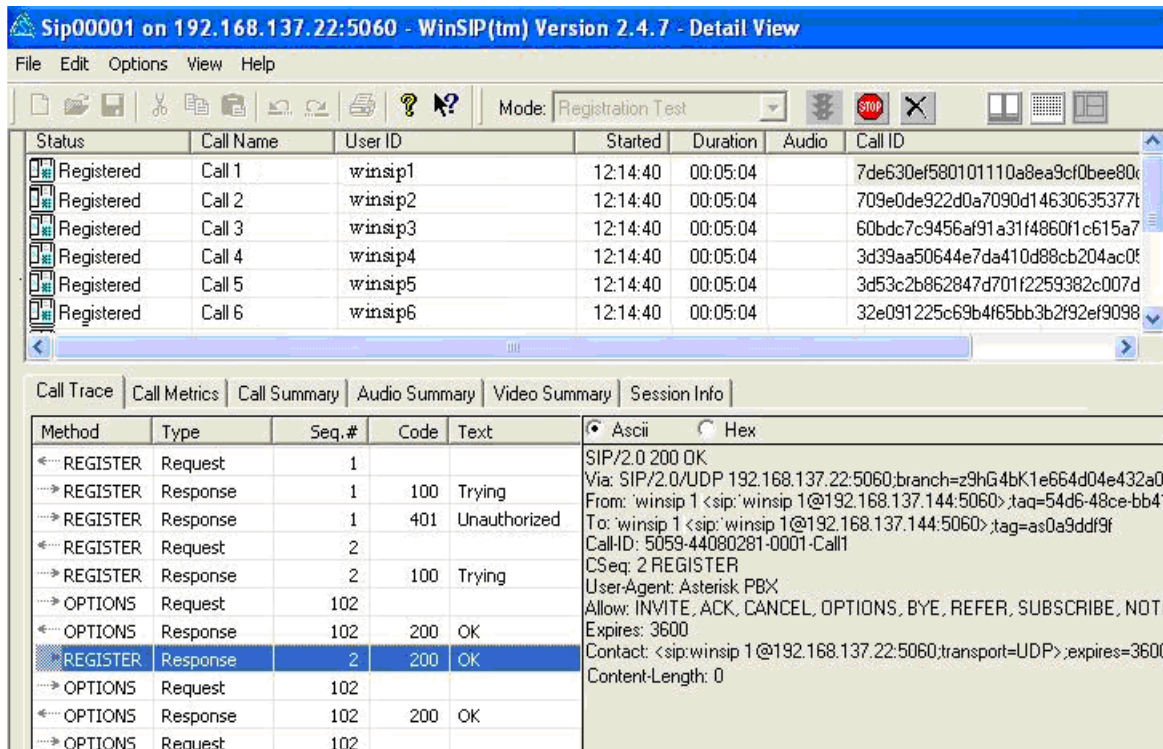


Figure 4. 7: Interface de la pondeuse d'appels

Une fois le logiciel installé, on declare les 50 utilisateurs virtuels de ce logiciel sous asterisk pour qu'il puisse les authentifier puis on initie des appels de l'un des postes sur lequel ce logiciel est installé. Les requêtes d'enregistrement émanant de ces différents clients arrivent auprès d'asterisk qui les traite une à une et envoie des réponses à chacun d'eux.



Figure 4. 8: Déroulement de l'enregistrement des utilisateurs auprès d'asterisk

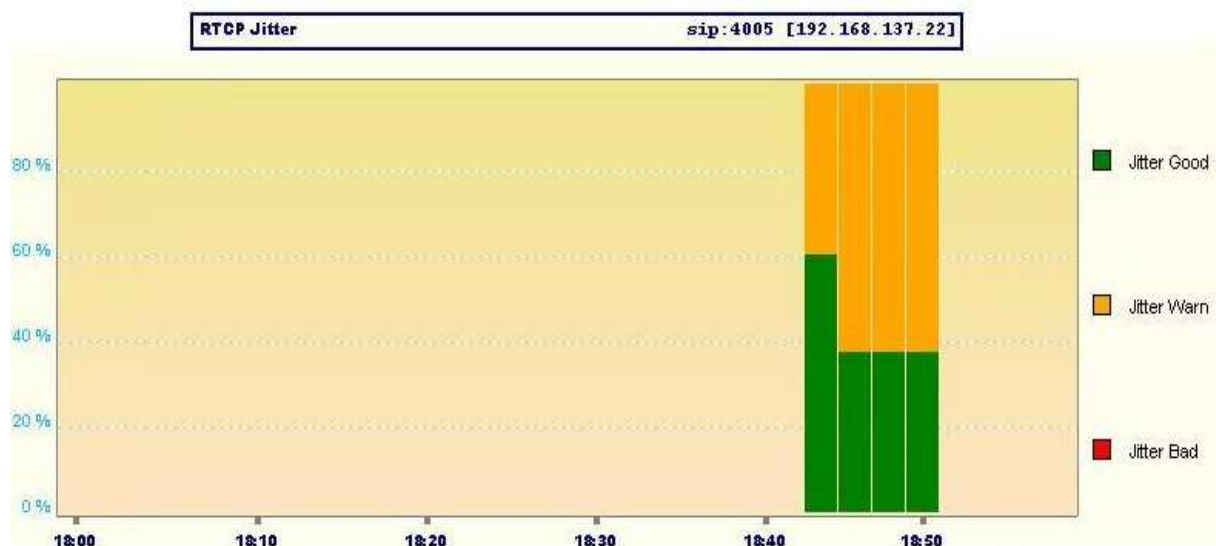


Figure 4. 9: Gigue des paquets RTCP

On remarque clairement que lorsqu'on a augmenté le nombre des appels simultanés, la gigue a augmenté de façon claire. Et pourtant plus que 30 % des paquets arrivent avec une valeur de gigue excellente alors que 60% arrive avec une gigue acceptable et dont les valeurs sont supportables.

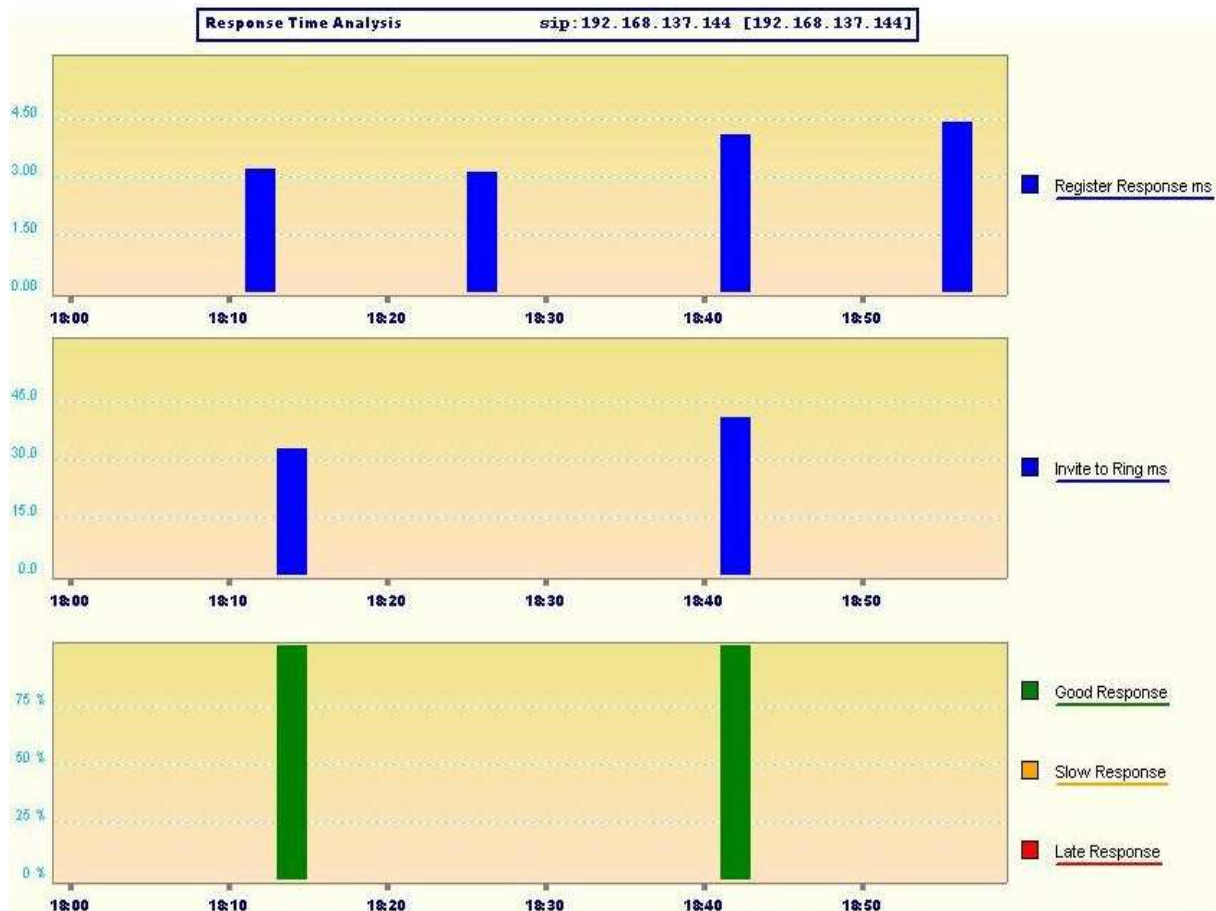


Figure 4. 10 : Temps de réponse moyen

De même, on remarque que l’augmentation du nombre d’appels simultanés a un effet sur les temps de réponse à savoir le temps d’enregistrement, et le temps nécessaire pour faire sonner le poste appelé. Malgré l’augmentation remarquée, ces valeurs sont négligeables puisqu’on garde toujours un temps de réponse moyen acceptable et qui ne peuvent pas nuire à la qualité du son.

En effet ces variations remarquées sont dues principalement à 3 facteurs :

- La bande passante : lorsque le nombre de communications simultanées est important, la consommation de la bande passante croît. Ce qui constitue une contrainte sur le nombre d’appels simultanés.
- Le codec utilisé : le choix du codec est essentiels. Selon lui, on définit d’une part la qualité de l’appel et la disponibilité de la bande passante.
- La performance de la machine sur laquelle est installé astérisik : en effet, pour pouvoir traiter grand nombre d’appels vers le serveur astérisik, la machine sur laquelle on a installé astérisik doit être suffisamment performante (RAM, CPU...). pour pouvoir effectuer le traitement nécessaire pour chaque appel. Le tableau suivant montre une correspondance entre le nombre d’appels simultanés et la configuration minimale nécessaire de la machine.

Nombre de canaux	Minimum recommandé
Entre 5 et 10	1-GHz x86, 512 MB RAM
Up to 15	3-GHz x86, 1 GB RAM
Plus que 15	des Dual CPUs

Tableau 4. 3 : Performance de la machine d'astérisque vs le nombre d'appels simultanés.

Conclusion

Dans ce chapitre on s'est intéressé dans la première partie à mettre en place les services de base qu'offre un PBX traditionnel sous l'IPBX astérisque. Dans une deuxième partie on a effectué quelques mesures sur la qualité de service des appels gérés par astérisque. Il s'est avéré clair que la qualité de service en VoIP avec astérisque est intimement liée à la bande passante disponible, aux codecs utilisés et à la configuration de la machine sur laquelle astérisque est installé.

Conclusion générale

La VoIP est une technologie émergente et qui tente plusieurs entreprises de l'exploiter vu les avantages qu'elle présente.

En Tunisie, cette technologie n'est pas encore très bien développée vu l'absence des fournisseurs de VoIP. Cependant, il est possible de déployer quelques applications de cette technologie au sein des entreprises multi-sites ce qui nous permettra de migrer les communications du réseau RTC vers le réseau IP.

Dans ce contexte, nous avons mis en place une solution qui permet de trunker de PBX appartenant à des sites distants via le logiciel l'IPBX open source Astérisik en utilisant le protocole SIP.

La solution déployée a été testé sur un réseau local. Les résultats des tests montrent que la solution présente de bonnes performances de point de vue MOS, gigue et taux de paquets perdus même en augmentant le nombre de communications simultanées.

La performance d'une solution VoIP dépend de plusieurs facteurs, principalement de :

- La bande passante : la bande passante doit être suffisante pour véhiculer un nombre important de communications. De ce fait, un bon dimensionnement de la bande passante sera nécessaire pour pouvoir véhiculer un grand nombre de communications simultanées tout en assurant une bonne qualité de son.
- Le codec utilisé : le choix du codec pour la VoIP est primordial. Il s'agit, en effet, de trouver un compromis entre la qualité du son et la consommation de la bande passante parce qu'un codec performant de point de vue qualité de son est gourmand en terme de bande passante.
- La capacité de la machine sur laquelle on a installé Asterisk puisque la performance de ce logiciel est intimement liée à la configuration matérielle du pc (RAM, vitesse, CPU...)

Commandes de base d'asterisk

Add extension: Ajoute une nouvelle extension à un contexte.
Add indication: Ajoute une indication donnée à un pays.
Add queue member: Ajoute un canal à une file d'attente.
Agi debug: Active le débogage AGI.
Agi no debug: Désactive le débogage AGI.
Bri debug span: Active le débogage BRI sur un span.
Bri intense debug span: Active le débogage BRI intense sur un span.
Bri no debug span: Désactive le débogage BRI sur un span.
Cdr mysql status: Affiche l'état de connexion de cdr_mysql.
Database del: Supprime les valeurs de la base de données.
Database deltree: Supprime les valeurs des arborescences de la base de données.
Database get: Afficher des valeurs de la base de données.
Database put: Ajoute ou met à jours les valeurs de la base de données.
Database show: Affiche le contenu de la base de données.
Debug channel: Active le débogage sur un canal.
Don't include: Supprime un include spécifique d'un contexte.
Exit: Quitte Asterisk.
Extensions reload: Recharge les extensions.
Help Display: Affiche l'aide sur Asterisk.
lax2 debug: Active le débogage IAX.
lax2 no debug: Désactive le débogage IAX.
lax2 show cache: Affiche le plan de numérotation cache de IAX.
lax2 show channels: Affiche les canaux IAX actifs.
lax2 show firmware: Affiche les firmwares IAX disponibles.
lax2 show peers: Affiche les peers IAX.
lax2 show peers begin: Affiche des peers IAX spécifiés.
lax2 show peers exclude: Affiche des peers IAX spécifiés.
lax2 show peers include: Affiche des peers IAX spécifiés.
lax2 show registry: Affiche l'état de registration IAX.
lax2 show stats: Affiche les statistiques IAX.
lax2 show users: Affiche des users IAX spécifiés.
lax2 trunk debug: Demande un débogage d'un trunk IAX.
Include context: Inclut un contexte dans un autre contexte.
Init keys: Initialise les clés RSA.
Local show channels: Affiche l'état d'un canal local.
Logger reload: Ouvre les fichiers logs.

Meetme: Exécute une commande sur une conference.

Mgcp audit endpoint: Affiche un terminal MGCP.

Mgcp debug: Active le débogage MGCP.

Mgcp no debug: Désactive le débogage MGCP.

Mgcp reload: Recharge une configuration MGCP.

Mgcp show endpoints: Affiche un terminal MGCP define.

No debug channel: Désactive le débogage sur un canal.

Pri debug span: Active le débogage PRI sur un span.

Pri intense debug span: Active le débogage PRI intense sur un span.

Pri no debug span: Désactive le débogage PRI sur un span.

Pri show span: Affiche les Information PRI.

Quit: Quitte Asterisk.

Reload: Recharge la configuration.

Remove extension: Supprime une extension spécifiée.

Remove indication: Supprime l'indication donnée du pays.

Remove queue member: Supprime un canal d'une fille d'attente donnée.

Restart gracefully: Redémarre Asterisk gracieusement.

Restart now: Redémarre Asterisk immédiatement.

Restart when convenient: Redémarre Asterisk quand le volume deviant nul.

Show audio codecs: Affiche les codecs audio.

Show channels: Affiche les informations d'un canal.

Show channel: Affiche les informations d'un canal spécifique.

Show codecs: Affiche les codecs.

Show codec: Affiche un codec spécifique.

Show conferences: Afficher l'état d'une conferences.

Show config handles: Affiche les Config Handles.

Show dialplan: Affiche le plan de numérotation.

Show file formats: Affiche le format des.

Show image codecs: Affiche les codecs image.

Show image formats: Affiche les formats d'image.

Show indications: Affiche une liste des pays ou des indications.

Show keys: Affiche les informations d'ela clé RSA.

Show manager command: Affiche les commande du manager.

Show manager commands: Affiche les commandes du manager.

Show manager connected: Affiche les utilisateurs du manager connectés.

Show modules: Liste les modules et les info.

Show parkedcalls: Liste les appels parqués.

Show queue: Affiche l'état d'une file d'attente.

Show queues: Affiche l'état des files d'attente.

Show switches: Affiche les switches alternatifs.

Show translation: Affiche la matrice de traduction.

Show video codecs: Affiche les codecs video.

Show voicemail users: Liste les boites vocals définies.

Show voicemail zones : Liste les formats de messages vocaux.

Sip debug: active le débogage SIP.

Sip debug ip: Active le débogage SIP sur une adresse IP.

Sip debug peer: Active le débogage SIP sur le nom d'un peer.

Sip history: Active l'historique du SIP.

Sip no debug: Désactive le débogage SIP.

Sip no history: Désactive l'historique du SIP

Sip reload: Recharge configuration SIP.

Sip show channels: Affiche les canaux SIP actifs

Sip show channel: Affiche les détails d'un canal SIP.

Sip show peer: Affiche les détails d'un peer SIP define.

Sip show peers: Affiche les peers SIP defines.

Sip show peers begin: Affiche les peers SIP defines.

Sip show peers exclude: Affiche les peers SIP defines.

Sip show peers include: Affiche un peer SIP define.

Sip show registry: Affiche l'état des enregistrements SIP.

Sip show subscriptions: Affiche les enregistrements SIP actifs.

Sip show users: Affiche les users de SIP.

Skinny debug: Active le débogage Skinny.

Skinny no debug: Désactive le débogage Skinny.

Soft hangup: Demande de raccrocher un canal donné.

Stop gracefully: Arrête asterisk gracieusement.

Stop now: Arrête Asterisk immédiatement.

Stop when convenient: Arrêter asterisk quand le volume d'appels est nul.

Zap destroy channel: Détruit un cannl.

Zap show cadences: Liste les cadences.

Zap show channels: Affiche les canaux zapata actifs.

Zap show channel: Affiche les informations d'un canal.

Bibliographie

[1] <http://www.frameip.com/voip>

[2] Jean-Pierre Lagasse, Téléphonie sur IP en entreprise : Evolution ou révolution dans le transport de la voix et accès à de nouvelles applications, Ed 3, Mai 2006

[3] <http://fr.wikipedia.org/wiki/Codec>

[4] <http://perso.orange.fr/wallu/VoIP.pdf>

[5] <http://fr.wikipedia.org/wiki/H.323>

[6] http://fr.wikipedia.org/wiki/Session_Initiation_Protocol

[7] Alberto Escudero Pascual, Louise BerthilsonVoIP, pour le développement un guide pour construire une infrastructure de télécommunication en région émergente, Décembre 2006

[8] Andy Powell, Getting Started With Asterisk, juillet 2005

[9] Jim Van Meggelen, Jared Smith, and Leif Madsen, Asterisk : The Future of Telephony, August 2005

[10] Building telephony systems with asterisk, David Gamillio, Barrie Dempster

Forums :

<http://www.asterisk-france.net/>

<http://groups.open-source.fr/>

<http://www.asterisk-support.ru/forums>

<http://www.frameip.com/forum/>

Résumé

Ce projet de fin d'étude a été réalisé au sein de Tunisian Telecom Electric (TTE).

Le but de ce projet est de réaliser une interface de communication VoIP entre des PBX via l'IPBX open source Astérisik. Cette interconnexion est faite en se basant sur le protocole d'initiation des sessions SIP.

Au début de ce projet nous avons présenté le principe de la VoIP ainsi que les protocoles associés. Ensuite nous avons expliqué la solution à mettre en place et justifié le choix d'astérisik pour assurer cette interconnexion.

Par la suite, nous avons expliqué la mise en place de la solution ainsi que l'intégration des différents équipements traditionnels de l'entreprise dans cette nouvelle architecture.

A la fin, nous avons intégré dans astérisik les services de base qu'offre un PBX et nous avons achevé le travail par des tests sur la qualité de service dans la solution mise en place.

Mots clés

VoIP, Asterisk, SIP, IPBX, ipLDK