

# Expérience d'un déploiement Asterisk dans une entreprise française

**Auteur :** Alexis de Lattre <[alexis\\_arobase\\_via.ecp.fr](mailto:alexis_arobase_via.ecp.fr)>

Vous avez le droit de copier, distribuer et/ou modifier ce document selon les termes de la [GNU General Public License](#) version 3 ou n'importe quelle version ultérieure, telle que publiée par la Free Software Foundation.

La version la plus à jour de ce document se trouve à l'adresse <http://people.via.ecp.fr/~alexis/asterisk/>

## Historique

Date	Changement
2008-01-13	Version initiale
2008-01-27	Relecture et ajout du diagramme (vive OpenClipart !)

## Introduction

Ce document a pour objectif de faire profiter les personnes envisageant un déploiement d'Asterisk dans leur entreprise de l'expérience que j'ai acquise ; il détaille l'architecture mise en place, les choix techniques que nous avons fait et la motivation de ces choix, les erreurs à éviter et il fourni des liens vers nos fichiers de configuration à titre d'exemple. C'est d'une certaine façon ma contribution à la communauté Asterisk francophone. Ce document n'est pas un tutoriel sur Asterisk ; il existe déjà suffisamment de tutoriels qui expliquent la configuration et le paramétrage d'Asterisk.

Ce document a été rédigé à l'issue du déploiement d'Asterisk dans mon entreprise en Août 2007. Quand nous avons réalisé ce déploiement, nous n'avions aucune compétence et aucune expérience en VoIP. Par contre, nous avons une solide expérience en administration Linux et en réseaux IP.

Pourquoi préciser dans le titre *dans une entreprise française* ? Cette documentation n'est pas spécifique à la France. Cependant, certains aspects, notamment l'utilisation de lignes RNIS, ne concernent que les pays utilisant ce type de ligne pour les installations téléphoniques en entreprise (ce qui est assez courant en Europe, mais pas forcément sur les autres continents).

Pré-requis :

- la lecture du livre *Asterisk, the Future of Telephony*, dont la version électronique est disponible gratuitement sur le [site d'Oreilly](#),
- de bonnes compétences en administration Linux,
- de bonnes compétences en réseaux.

## Installation précédente

Notre installation téléphonique précédente était centrée autour d'un PABX Alcatel (loué à France Telecom) qui était doté de :

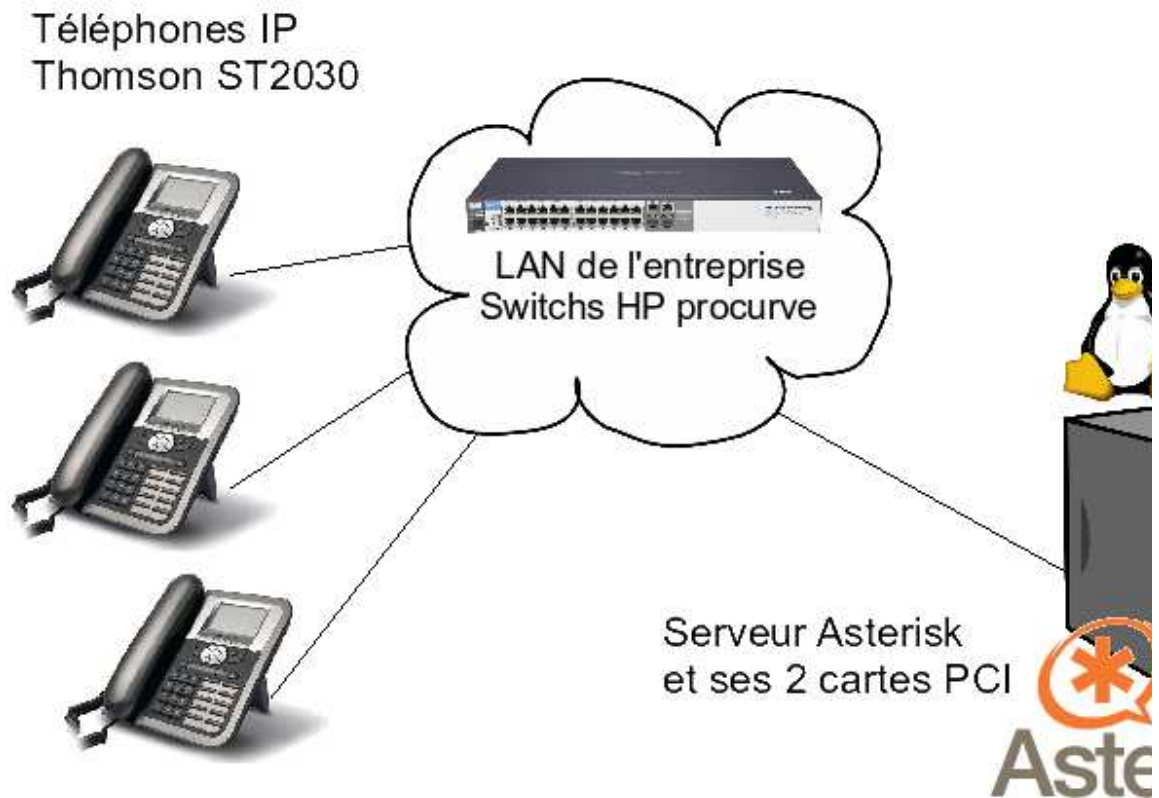
- 4 ports RNIS,
- 8 ports numériques, reliés à des téléphones Alcatel,
- 4 ports analogiques : 1 relié à notre fax et 3 reliés à des téléphones analogiques classiques.

Ce PABX était relié à 3 lignes RNIS T0 de France Telecom. Rappel : une ligne T0 permet d'avoir 2 conversations téléphoniques simultanées.

Ce PABX ne convenait plus à nos besoins car, avec la croissance des effectifs de l'entreprise, il n'y avait plus de ports libres pour brancher des téléphones supplémentaires !

## Installation VoIP avec Asterisk

### Schéma d'architecture



## Version des composants logiciels actuellement utilisés

Composant	Version du logiciel
Distribution Linux	Debian Etch 4.0
Noyau Linux	2.6.20.15
Zaptel	1.4.7.1
Driver mISDN + mISDNuser	1.1.7
libpri	1.4.3
Asterisk	1.4.17, patché avec la modification de chan_sip.c proposée par Thomson pour l'interception d'appels
IAXmodem	1.0.0
Hylafax	4.3.1 (package Etch)
Avantfax	3.0.5
Téléphones ST2030	firmware 1.58

## Mes bookmarks

Voilà un peu en vrac les sites Web actuellement présents dans la catégorie *Asterisk / VoIP* de mes bookmarks :

- [Asterisk.org](http://Asterisk.org), le site communautaire du projet Asterisk.
- [Asterisk France](http://Asterisk France) : site communautaire français sur Asterisk et

tout ce qui tourne autour (les téléphones IP notamment). Les forums qui sont assez actifs. J'en profite pour remercier tous les utilisateurs d'Asterisk France qui ont répondu à mes questions dans les forums !

- [Site de Digium](#), la société qui est derrière Asterisk et qui vend ses services et ses cartes téléphoniques.
- [Thomson ST2030 - page de téléchargement](#) : cette page contient en particulier les nouvelles versions des firmwares des téléphones.
- [mISDN.org](#) : le site communautaire des drivers RNIS.
- [Voip-info.org](#) : très gros Wiki communautaire qui parle des sujets relatifs à Asterisk et à la VoIP en général et qui donne aussi des infos intéressantes sur les téléphones IP.
- [The Asterisk Book](#) : la version actualisée du livre "Practical Asterisk 1.4", qui aborde tous les aspects pratiques de l'installation et du paramétrage d'Asterisk.
- [Asteriskvoipnews.com](#) : site qui traite l'actualité d'Asterisk et de la VoIP.
- [The Asterisk Guru](#) : site proposant un certain nombre de tutoriels sur Asterisk et ce qui tourne autour.

## Budget

Voilà le budget de notre déploiement :

Composant	Produit retenu	Acheté chez	Prix unitaire HT	Quantité	Total HT
Serveur	Dell Optiplex GX620	<a href="#">Dell</a>	700 €	1	700 €
Carte RNIS	Digium B410P	<a href="#">Opcom</a>	570 €	1	570 €
Carte pour le fax	Digium TDM400P avec 1 port FXS	<a href="#">Opcom</a>	117 €	1	117 €
Téléphone IP	Thomson ST2030	<a href="#">HL2D</a>	Inférieur à 100 €	30	< 3 000 €
Extension standard	Thomson ST2030e	<a href="#">HL2D</a>	Inférieur à 70 €	1	< 70 €

Note : je ne peux pas donner les prix exacts de HL2D car ils ne sont pas publics.

## Retour d'expérience global

Globalement, le déploiement s'est très bien passé. Notre nouvelle installation est :

- stable : même si c'est encore un peu tôt pour porter un jugement définitif, je peux dire que mes doutes sur la stabilité d'Asterisk sont levés.

- pleinement fonctionnelle : nous avons retrouvé toutes les fonctionnalités de notre PABX Alcatel sans exception et nous pouvons maintenant profiter des fonctionnalités supplémentaires apportées par Asterisk et la VoIP.
- identique à la précédente au niveau de la qualité audio.

Au niveau technique, les 3 choses que je voudrais dire à ceux qui se lanceraient dans le même type de migration sont :

1. Asterisk et la VoIP ne sont pas des technologies simples qui "tombent en marche" tous seuls : il faut s'y intéresser vraiment et s'y plonger à fond pour bien maîtriser tous les paramètres techniques... et ils sont nombreux ! Il faut bien mûrir les choix techniques à faire (ils sont listés ci-dessous) et cela prend du temps !
2. le plus long n'est pas de comprendre et d'écrire les fichiers de configuration d'Asterisk mais ceux des téléphones IP !
3. pour tout ce qui concerne la carte RNIS, il ne faut jamais se fier aux documentations ni aux drivers de Digium, mais toujours aux docs et aux drivers de [misdn.org](http://misdn.org), qui sont maintenus par la société allemande [Beronet](http://beronet.com) (par cette petite info, je vous évite de nombreuses heures d'errements et de tests infructueux !).

## Choix et motivation des choix

Les 3 choix les plus importants, sur lesquels vous devrez concentrer toute votre attention avant le déploiement, sont à mon avis :

- le choix des téléphones IP,
- le choix du "packaging" d'Asterisk,
- le choix du canal de communication avec l'extérieur.

### Téléphones IP

**Choix réalisé** : téléphone Thomson ST2030.

Le choix du téléphone IP est très important pour plusieurs raisons :

- il représente la plus grosse partie du budget,
- c'est le paramétrage des téléphones IP qui a pris le plus de temps, et non le paramétrage d'Asterisk,
- c'est ce que verront les utilisateurs de la nouvelle installation,
- il joue un rôle important dans la qualité audio.

Vous allez choisir un téléphone IP professionnel qui utilisera le protocole SIP. Ne cherchez pas sur les sites de vente grand public du type LDLC ou Rue du commerce, vous ne les trouverez pas. Il vous faudra plutôt consulter les sites des constructeurs pour avoir les spécifications et les sites de vente spécialisés dans la téléphonie ([One Direct](http://OneDirect.com), [Novacom](http://Novacom.com), [IP & Go](http://IP&Go.com)) pour avoir les prix. Les principaux constructeurs qui proposent des téléphones IP à des prix abordables (i.e. prix unitaires compris entre 90 et 130 euros HT) sont :

- Thomson : téléphones ST2030 et ST2022,
- Grand Stream, notamment le téléphone GXP2000,
- Budget Tone, qui est la marque entrée de gamme de Grand Stream,
- Aastra,
- Snom,
- Siemens, qui fait notamment des téléphones SIP DECT, comme par exemple le téléphone Gigaset C470,
- Linksys, dont la gamme SPA comporte maintenant des téléphones SIP (à chercher dans la catégorie *Products / Business / IP Communications / Linksys voice system...* sinon vous allez vous retrouver au milieu des téléphones Skype, USB...)

J'exclue Cisco de la liste ; leurs téléphones IP ont très bonne réputation mais sont très chers.

Les critères de choix sont à mon avis, par ordre d'importance :

1. la qualité du rendu et de la captation audio du combiné et du haut parleur éventuel,
2. la qualité physique du combiné et des touches,
3. le niveau de compatibilité avec Asterisk (dans mon expérience, c'est surtout la fonction d'interception des appels qui peut poser problème),
4. les fonctionnalités du téléphone : codecs supportés, PoE, haut parleur, fonction main-libre, fonction d'interception des appels, le nombre de voyants de supervision de lignes, possibilité d'avoir une "extension standard" qui permet de monitorer un grand nombre de lignes, possibilité d'autoconfiguration des téléphones via des fichiers de configuration partagés en TFTP ou en HTTP.

Les fonctionnalités suivantes sont proposées aujourd'hui par tous les téléphones IP : présentation du numéro, notification des appels de absence et des messages présents sur le répondeur, mise en attente d'un appel, transfert d'appel, possibilité d'avoir un carnet d'adresse.

Pour faire notre choix, nous avons décidé de faire une pré-sélection en utilisant les informations disponibles sur Internet puis d'acheter un téléphone de chaque modèle pré-sélectionné. Nous avons pré-sélectionné le téléphone Thomson ST2030 et le téléphone Dlink DPH120S. Mon idée était d'avoir un téléphone orienté professionnel dont j'avais entendu du bien (le Thomson) et un téléphone d'une marque orientée grand public (le Dlink), en apparence moins cher. Lors de nos tests, nous avons rapidement constaté que le firmware du téléphone Dlink n'était pas maintenu et souffrait de plusieurs bugs mineurs non corrigés. De l'autre côté, le téléphone Thomson donnait satisfaction et avait un firmware bien maintenu, nous l'avons donc retenu.

Si c'était à refaire aujourd'hui, je incluerai plus de téléphones dans la pré-sélection vu l'importance du choix. Je pense que ma short list contiendra au moins un téléphone Linksys (gamme SPA), un téléphone Grand Stream ou Budget Tone (selon le budget visé) et le

téléphone Thomson ST2030.

Au final, le téléphone Thomson ST2030 a pas mal de qualités :

- son de bonne qualité, tant au niveau du rendu que de la captation (le haut parleur est aussi de bonne qualité),
- bonne qualité du combiné et des touches,
- 10 touches de supervision en standard et possibilité d'ajouter des extensions pour superviser un grand nombre de lignes, ce qui est utile pour le poste qui sert de standard,
- toutes les fonctionnalités que j'ai cité plus haut, qui sont attendues sur un téléphone professionnel, sont bien présentes et fonctionnelles,
- le firmware est bien maintenu (une release tous les 2 ou 3 mois depuis début 2007) et l'accès aux nouveaux firmware est gratuit,
- aucun problème de compatibilité avec Asterisk sauf pour la fonctionnalité d'interception d'appel.

Ce que je reproche à ce téléphone :

- la mise en place de l'autoconfiguration par fichiers de configuration est laborieuse : il faut bien lire la doc pour ne pas "se faire avoir" par certaines subtilités. On a passé beaucoup de temps à faire du fine tuning des fichiers de configuration.
- il faut patcher Asterisk pour avoir la fonctionnalité d'interception d'appel (patch pour Asterisk 1.2 et Asterisk 1.4 fourni sur le site de Thomson).
- et un point anecdotique : la qualité de la langue anglaise utilisée dans la documentation ne fait pas honneur à Thomson !

## Canal de communication avec l'extérieur

**Choix réalisé** : conserver les lignes RNIS utilisées dans l'installation précédente pour les appels entrants et sortants.

Pour la sélection du canal de communication avec l'extérieur, trois choix s'offraient à nous. Leurs avantages et inconvénients respectifs sont résumés ci-dessous. Par la suite, quand je parle d'un lien xDSL, je désigne un lien ADSL ou SDSL.

Numéro	Lien entrant	Lien sortant	Avantages	Inconvénients
1	RNIS	RNIS	<ul style="list-style-type: none"> <li>• Rien à changer</li> <li>• Je n'ai jamais vu une ligne RNIS en dérangement (qui peut dire la même chose d'un lien ADSL ?)</li> </ul>	<ul style="list-style-type: none"> <li>• Conservation de la dépendance vis-à-vis de France Telecom pour le lien RNIS</li> </ul>

2	VoIP sur xDSL	VoIP sur xDSL	<ul style="list-style-type: none"> <li>• Vous pourrez vous vanter d'avoir <i>tout</i> migré en VoIP !</li> <li>• Vous pouvez bénéficier des tarifs agressifs sur les communications de certains opérateurs VoIP.</li> <li>• Vous évitez les tracas des drivers des cartes RNIS.</li> </ul>	<ul style="list-style-type: none"> <li>• Si choix d'un lien SDSL, ne pas négliger son coût. Si choix d'un lien ADSL, il faut prévoir une solution de secours quand le lien en down !</li> <li>• Si votre opérateur VoIP n'est pas le même que l'ISP qui vous fourni le lien xDSL, cela veut dire que vos flux VoIP passent par Internet, or il n'y a pas de garantie de qualité de service sur Internet, donc pas de garantie sur la qualité audio de vos communications vers l'extérieur.</li> <li>• Il faut que le portage des numéros de téléphones existants vers votre opérateur VoIP se passe bien !</li> </ul>
3	RNIS	VoIP sur xDSL	<ul style="list-style-type: none"> <li>• Vous bénéficiez de la fiabilité parfaite du lien RNIS pour les appels entrants.</li> <li>• Pas de portage des numéros de téléphone à réaliser.</li> <li>• Votre lien RNIS sert aussi de lien de backup pour les appels sortants.</li> <li>• Vous pouvez bénéficier des tarifs agressifs sur les communications de certains opérateurs</li> </ul>	<ul style="list-style-type: none"> <li>• Il faut payer à la fois l'abonnement pour le lien RNIS et l'abonnement pour le lien ADSL ou SDSL.</li> </ul>

			VoIP.	
--	--	--	-------	--

Nous avons choisi la première solution car c'était la solution la plus simple à mettre en oeuvre (moyennant les petits tracas liés aux drivers RNIS... mais ce document devrait vous permettre de les éviter) et qu'elle n'ajoutait aucun délai.

En ce qui concerne la deuxième solution, ma mini étude des offres disponibles sur le marché français ne m'a pas permis d'identifier un opérateur capable de proposer à la fois le lien ADSL ou SDSL avec ses propres DSLAM et les communications en VoIP via des comptes SIP, MGCP ou autre (dans le cas d'une interconnexion SIP, on appelle l'offre un *trunk SIP*). Aujourd'hui, seuls Orange, Free, Neuf, Alice et Comptel possèdent leurs propres DSLAM dans les NRAs. Or aucun de ces opérateurs ne propose aujourd'hui aux PME d'accès VoIP "en direct" (c'est-à-dire sans passer par les ports FXS de leurs box) pour relier un IPBX à leur réseau : Comptel et Orange proposent des offres VoIP qui utilisent des boîtiers qui intègrent un modem SDSL pour se relier à leur réseau... mais ces boîtiers ne proposent que des ports RNIS pour l'interfacage téléphonique ; Free ne propose qu'un seul compte SIP (donc une seule communication simultanée) dans ses abonnements ; Neuf ne propose pour le moment qu'un compte SIP (même si une offre est peut-être en préparation) et Alice ne propose rien à ma connaissance. Par contre, SFR Entreprise a lancé fin 2007 une offre baptisée "SFR One Solution" qui comprend la possibilité d'avoir une interconnexion VoIP en MGCP (un protocole alternatif à SIP) via une connexion ADSL ou SDSL de chez SFR, qui passe en fait par les DSLAMs de Neuf. Vu que Neuf a été racheté par SFR, on peut espérer que la collaboration technique entre les 2 entreprises est bonne... mais le commercial SFR que j'ai interrogé sur ce sujet en Octobre 2007 m'a répondu "Je ne connais pas encore bien l'offre car je n'aurai la formation sur cette nouvelle offre que la semaine prochaine et franchement je vous déconseille d'y souscrire tout de suite car vous serez en réalité les bêta-testeurs de ce nouveau système !"

Si vous voulez tout de même vous lancer dans la 2ème solution, l'opérateur Direct Centrex propose des trunks SIP (i.e. des interconnexions pour des IPBX via le protocole SIP) avec des coûts de communication très compétitifs ; il prétend avoir de très bon accords de peering avec un certain nombre d'ISP français. Cet opérateur propose le service de portage des numéros. Avant de souscrire à son offre, vérifiez que l'accès ADSL ou SDSL que vous avez prévu pour faire passer vos communications VoIP est chez un ISP avec lequel cet opérateur a une bonne interconnexion IP.

La 3ème solution, qui est apparemment souvent adoptée par les entreprises qui migrent vers un IPBX, est probablement un bon compromis en attendant les vraies offres d'interconnexion VoIP des opérateurs xDSL (par "vraies offres" j'entends des offres conçues pour

les entreprises avec une qualité de service garantie de bout en bout, une garantie de temps de rétablissement, un mécanisme de redondance avec un lien de backup, etc...) ce qui permettra d'adopter sereinement la deuxième solution. L'opérateur Direct Centrex est également un candidat potentiel pour assurer les appels sortants via un lien xDSL chez un ISP.

L'analyse des trois solutions présentées dans cette section n'est issue que de mon expérience assez limitée dans ce domaine et de mes petites recherches sur Internet et lors du salon IP Convergence. Si vous avez un point de vue différent ou une expérience réussie ou non avec un opérateur VoIP pour un déploiement en entreprise, alors je serai ravi de recevoir un mail de votre part, ce qui me permettra d'améliorer ce chapitre.

## Le packaging d'Asterisk

**Choix réalisé** : Asterisk et drivers compilés à la main sur une Debian stable.

Tout d'abord, il me faut clarifier ce que j'entends par "packaging" d'Asterisk. Asterisk, comme tous les logiciels libres importants, peut être installé soit via le package de votre distribution Linux habituelle (fichier de package `.rpm` ou `.deb`), soit en téléchargeant les sources sur le site Web du projet et en les compilant à la main (le fameux `./configure && make && make install`).

Dans le cas d'Asterisk, il existe en tout 4 solutions :

- utiliser le package Asterisk de votre distribution Linux habituelle et les drivers des cartes téléphoniques packagés avec votre distribution,
- sur votre distribution Linux habituelle, télécharger les sources d'asterisk sur le site [asterisk.org](http://asterisk.org) et les compiler "à la main" ainsi que les drivers des cartes téléphoniques,
- utiliser une distribution Linux "spéciale Asterisk", i.e. une distribution Linux dont l'objet est de contenir Asterisk, tous les outils susceptibles d'être utilisés avec Asterisk ainsi que les drivers des cartes téléphoniques. J'ai principalement entendu parler de 3 distributions "spéciale Asterisk" : Tribox, distribution la plus ancienne qui est proposé par la société Fonality, AsteriskNow, une distribution assez récente proposée par la société Digium (la société derrière Asterisk) et qui est encore en version bêta et Elastix, proposé par la société PaloSanto Solutions, nouvel acteur sur ce créneau et qui semble très dynamique.
- acheter Asterisk Business Edition auprès de Digium ou d'un de ses revendeurs : c'est une version payante d'Asterisk proposée par Digium. Cette version a passé des tests de qualité avant sa release et est accompagné d'un support technique. L'ordre de grandeur du prix est de 700 euros HT pour chaque tranche de 40 communications simultanées.

Je vais essayer de récapituler les avantages et inconvénients de

chaque approche :

Choix	Avantages	Inconvénients
<p>Asterisk packagé dans sa distribution Linux habituelle</p>	<ul style="list-style-type: none"> <li>• Vous avez déjà l'habitude de l'administration de cette distribution que vous connaissez bien</li> <li>• En cas de faille de sécurité, n'avez qu'à mettre à jour votre distribution selon la procédure habituelle (style <code>apt-get update</code> &amp;&amp; <code>apt-get upgrade</code> sous Debian/Ubuntu)</li> </ul>	<ul style="list-style-type: none"> <li>• Ce n'est pas la solution la plus courante, donc ce n'est pas celle sur laquelle vous pourrez trouver de l'aide le plus facilement</li> <li>• Les paquets Asterisk sont parfois vieux et pas toujours très bien maintenus (ça dépend des distribs)</li> <li>• Vous aurez probablement envie d'utiliser la dernière version officielle des drivers des cartes téléphoniques, or les packages de votre distribution proposent peut-être des drivers plus anciens</li> </ul>
<p>Asterisk compilé à la main sur sa distribution Linux habituelle</p>	<ul style="list-style-type: none"> <li>• Vous avez déjà l'habitude de l'administration de cette distribution que vous connaissez bien</li> <li>• Vous choisissez vous-même les versions d'Asterisk, des librairies et des drivers des cartes téléphoniques</li> <li>• C'est la solution qui est documentée dans la doc officielle d'Asterisk et dans la plupart des docs sur Asterisk</li> </ul>	<ul style="list-style-type: none"> <li>• Requier un minimum d'expérience de la compilation d'un logiciel libre et d'un driver Linux (rien de très compliqué !)</li> <li>• Il faut vous occuper vous-même des mises à jour de sécurité : recevoir les mails d'annonce des failles de sécurité, télécharger les versions corrigées si nécessaire, recompiler, etc...</li> </ul>
<p>Distribution dédiée Asterisk (Trixbox, AsteriskNow, Elastix)</p>	<ul style="list-style-type: none"> <li>• Vous êtes assuré d'avoir une combinaison "noyau + drivers de cartes téléphoniques + librairies + Asterisk"</li> </ul>	<ul style="list-style-type: none"> <li>• Il vous faut apprivoiser une nouvelle distribution Linux (Trixbox est basée sur CentOS)</li> </ul>

	<p>qui marche bien</p> <ul style="list-style-type: none"> <li>• Mises à jour de sécurité simplifiées</li> <li>• Dans le cas de Trixbox, c'est une solution répandue sur laquelle vous devriez pouvoir trouver du support de la communauté</li> <li>• Ces distributions proposent par défaut une jolie interface Web d'administration, ce qui doit en principe éviter de se familiariser avec les fichiers de configuration d'Asterisk</li> <li>• Ces distribution packagent en plus un certain nombre d'applications périphériques à Asterisk, comme par exemple des logiciels de fax2mail</li> </ul>	<ul style="list-style-type: none"> <li>• Vous n'avez pas la maîtrise complète des versions d'Asterisk, des librairies, des drivers, etc...</li> </ul>
Asterisk Business Edition	<ul style="list-style-type: none"> <li>• Inclus un support professionnel fourni par Digium pendant 1 an</li> <li>• Tests de qualité logicielle effectués sur chaque release</li> </ul>	<ul style="list-style-type: none"> <li>• Cette offre a évidemment un coût</li> <li>• On sort un peu du domaine de l'opensource, puisque cette version est sous une licence propriétaire et non sous licence GPL</li> </ul>

Dans notre cas, nous avons éliminé d'office la première solution car notre distribution habituelle est Debian et les packages Asterisk dans la distribution Debian stable sont vieux (la Etch contient Asterisk 1.2.13 et ne propose pas Asterisk 1.4, qui est par contre présent dans la Sid). Nous avons un peu testé Trixbox, mais nous n'avons aucune expérience des distributions CentOS ce qui ne nous a pas aidé à nous motiver pour explorer plus à fond cette piste.

Nous avons donc finalement décidé de partir sur la 2ème solution en installant une Debian stable avec un noyau Linux compilé à la main et l'ensemble "mISDN + mISDNuser + zaptel + libpri + asterisk" compilé également à la main. Nous avons choisi de ne pas installer d'interface Web d'administration comme par exemple [FreePBX](#), car nous avons déjà passé du temps à lire les docs d'Asterisk qui apprennent à se

servir des fichiers de configuration et éditer un fichier de configuration avec notre éditeur de texte préféré ne nous fait pas peur.

Si c'était à refaire, j'explorerai plus à fond la piste des distributions "dédiées Asterisk", pour me faire un avis plus précis sur cette option.

## Serveur

**Choix réalisé** : Ordinateur Dell Optiplex de bureau.

Nous cherchions un serveur rackable fiable avec 2 slots PCI 32 bit 33 Mhz pour accueillir la carte RNIS et la carte avec le port FXS pour le fax. Malheureusement, la contrainte d'avoir 2 ports PCI 32 bit 33 Mhz impose souvent des serveurs 2U voir 3 ou 4U, ce qui devint assez imposant et coûte généralement un peu cher. Nous avons donc finalement choisi de prendre un de nos ordinateurs de bureau Dell Optiplex GX620, qui a l'avantage d'être peu couteux et facilement remplaçable en cas de panne. Ses caractéristiques techniques sont les suivantes :

- processeur Pentium D 2,8 Ghz
- 1 Go de RAM
- 2 disques SATA, qui seront utilisés en RAID 1 (grâce au RAID logiciel de Linux)
- 2 slots PCI 32 bit 33 Mhz

Cette configuration matérielle est en réalité très surdimensionnée par rapport à notre utilisation. En effet, quand on ne fait pas de transcodage systématique des flux audio (tous nos téléphones utilisent le codec G711A) et qu'on n'utilise l'audioconférence que de façon ponctuelle, les besoins en RAM et en CPU sont modestes.

## Fax

**Choix réalisé** :

- pour la réception de fax : système type fax2mail avec archivage Web basé sur IAXmodem + Hylafax + Avantfax
- pour l'émission de fax : fax "traditionnel" connecté à la carte TDM400P ou utilisation de l'interface Web d'Avantfax (qui permet d'envoyer un fax à partir d'un document PDF)

Il n'existe pas aujourd'hui de fax IP à la façon d'un téléphone IP ; il en existera peut-être à l'avenir avec l'adoption par les constructeurs de la norme T38. Le choix doit donc se faire entre garder son fax traditionnel et mettre en place un système de type *fax2mail* pour la réception et de *print2fax* pour l'envoi.

Dans un souci de simplicité pour les utilisateurs et les administrateurs nous avons choisi de conserver notre fax traditionnel pour l'émission et

la réception lors du déploiement initial.

Ensuite, pour éviter les traditionnelles indisponibilités du fax liées au classique "*le rouleau du fax est fini et on a oublié d'en acheter un d'avance*", économiser le prix des rouleaux et éviter le gaspillage de papier lié au spam par fax, nous avons mis en place un simple système de *fax2mail* avec IAXmodem et Hylafax. A chaque fax reçu, un mail est envoyé vers une adresse mail donnée avec en pièce jointe un PDF contenant le fax.

Ce dernier système avait un petit inconvénient : le mail contenant le fax était adressé à deux personnes chargées de faire la distribution du fax, mais quand ces personnes étaient absentes, le reste du personnel n'avait pas accès aux fax. D'où l'idée d'avoir un archivage Web des fax reçus, qui est une fonctionnalité proposée par Avantfax. Avantfax permet aussi d'envoyer un fax à partir d'une interface Web : il suffit d'indiquer le numéro de téléphone du destinataire et de sélectionner un fichier PDF de son ordinateur (Avantfax propose aussi de générer facilement une page de garde).

Si c'était à refaire, je partirai probablement dès le début sur une solution fax2mail pour la réception de fax, car ce n'est pas compliqué à mettre en oeuvre, et, pour l'émission, je brancherai le fax sur une ligne analogique ou j'étudierai la possibilité de le relier à un boîtier du type Linksys PAP2T, qui assure la conversion entre une prise analogique et du SIP sur IP/Ethernet (et qui est d'ailleurs plus économique qu'une carte TDM400P !). Cela permettrait de se passer de la carte TDM400P, et on pourrait alors choisir un serveur rackable 1U avec 1 seul port PCI 32 bit 33 Mhz pour la carte RNIS.

## Choix des cartes téléphoniques

**Choix réalisé** : carte Digium B410P pour le RNIS et carte Digium TDM400P pour relier le fax.

Quand nous avons fait ce choix, notre motivation était la suivante : en tant qu'auteur principal d'Asterisk et développeur des drivers Linux des cartes, les cartes téléphoniques de Digium devaient sûrement être le meilleur choix.

En réalité, Digium est bien l'auteur principal du package zaptel qui fourni le driver pour la carte TDM400P utilisée pour avoir un port analogique pour le fax, mais Digium n'est pas impliqué dans le développement des drivers RNIS ! Les drivers RNIS sont maintenus par la société allemande Beronet qui publie les drivers sur le site [misdn.org](http://misdn.org). Cette société vend sur son site Web ses propres cartes RNIS en plus des cartes de Digium (les tarifs pour la carte RNIS 4 ports de Beronet sont du même ordre de grandeur que la carte B410P de Digium).

J'en profite pour rappeler une deuxième fois cette information importante : **pour tout ce qui concerne le RNIS, il ne faut pas se fier aux docs ni aux drivers de Digium, mais plutôt utiliser les drivers et lire les docs du projet misdn.org** . Pourquoi ? Parce que Digium est une société américaine et que le RNIS n'est quasiment pas utilisé aux Etats-Unis, d'où le peu d'intérêt de Digium sur ce point. Cela devrait vous éviter de nombreuses heures de galère. En effet, dans la doc de Digium sur la carte B410P, il est expliqué qu'il faut faire un *make b410p* dans le répertoire contenant les sources de *zaptel* pour compiler les drivers pour la carte ; en réalité, cette commande lance un *wget* sur un tarball mis en partage sur le site de Digium, sachant que ce tarball contient (ou en tout cas contenait à l'époque où j'ai fait la manip) une version assez vieille des drivers RNIS, qui ne fonctionnait pas bien du tout. Par contre, dès que j'ai installé la release stable des drivers misdn.org, tout s'est mis à marcher normalement.

Comme le l'explique dans le paragraphe précédent, si c'était à refaire, je me passerai de la carte TDM400P pour le fax.

## Réseau

**Choix réalisé** : garder la VoIP dans le même VLAN que tout le reste, et utiliser de la QoS au niveau IP (DiffServ, et plus précisément le champ DSCP) pour mettre les flux voix en priorité par rapport au trafic data. Pas de PoE.

Le choix est très lié à l'architecture réseau existante et au matériel réseau déjà en place. On vous conseillera sûrement de mettre les flux VoIP dans un VLAN dédié. Dans nos locaux, nous manquons de prises réseau dans un certain nombre de bureaux et nous avons donc recours à de petits switchs non administrables au niveau de ces bureaux pour raccorder les différents ordinateurs du bureau (et maintenant les téléphones également) au reste du réseau. Il y a donc de toute façon une cohabitation des flux data et voix au niveau de ces petits switchs. Ensuite, notre préoccupation n'est pas tant de séparer les flux voix des flux data que de s'assurer que les flux voix ne seront jamais perturbés par le volume des flux data ; par exemple, nous voulions que si un lien entre 2 switchs du réseau est saturé, les flux voix ne soient pas impactés. D'où le recours à la QoS. Nous avons choisi de faire de la QoS au niveau IP plutôt que de la QoS au niveau Ethernet car nos switchs supportent la QoS IP et les téléphones ST2030 sont capables de tagger les flux en QoS IP (et Asterisk également), ce qui simplifie la configuration des switchs, qui n'ont alors qu'à se baser sur les tags des paquets IP pour attribuer le niveau de priorité.

Attention, il ne faut pas perdre de vue l'essentiel : si on veut avoir une bonne qualité audio, il faut simplement un réseau où il n'y a *aucune* perte de paquet. En effet, les flux VoIP sont envoyés en UDP sans algorithme particulier de correction d'erreur ni de renvoi de paquet

perdu ; toute perte de paquet, si il s'agit d'un paquet transportant le son d'une conversation VoIP, va donc très probablement entraîner une dégradation ponctuelle de la qualité audio. Si vous avez du matériel réseau de bonne qualité (j'entends par là des switchs administrables qui tiennent un peu la charge, et pas des hubs non administrables même si ils sont de marque) et que vous ne l'utilisez que pour échanger des fichiers bureautique et des mails (c'est à dire que vous l'utilisez à 1% de ses capacités), alors vous n'aurez même pas besoin de mettre en place de QoS, la VoIP marchera très bien sans cela. Par contre, si vous avez des vieux hubs 10 Mb sur lesquels la diode de collision s'allume régulièrement, il faudra absolument faire quelque chose : mettre en place de la QoS ne vous apportera rien car votre hub n'a certainement pas de files de priorité, mais il faudra plutôt mettre votre vieux hub au placard et le remplacer par un vrai switch administrable qui tient la route.

Dans notre entreprise, nous avons un réseau composé uniquement de switchs administrables de bonne qualité (différents modèles de switchs HP Procurve) mis à part les petits switchs non administrables dans certains bureaux, mais certains traffics de données autres sont susceptibles de saturer des liens entre les switchs, d'où l'idée de mettre en place de la QoS.

La fonctionnalité PoE (Power over Ethernet) permet d'alimenter les téléphones en électricité par le câble réseau plutôt que par le bloc d'alimentation du téléphone. Il faut que cette fonctionnalité soit supportée par les téléphones IP (ce qui est le cas des téléphones Thomson ST2030) et par les switchs sur lesquels ils sont branchés (ce n'est pas une fonctionnalité courante sur les switchs ; il vous faudra probablement acheter de nouveaux switchs dans le but de supporter le PoE). L'avantage du PoE est de n'avoir qu'un seul câble utilisé pour les téléphones IP au lieu de deux, ce qui fait plus propre sur les bureaux ! Nous n'avons pas utilisé la fonctionnalité PoE pour notre déploiement.

## Fonction standard

Le PABX Alcatel de notre installation précédente proposait une solution intéressante pour la gestion du standard. Nous avions un poste dédié au standard, qui sonne quand on reçoit un appel de l'extérieur sur le numéro de téléphone du standard. En plus de cela, il était possible, sur les autres téléphones Alcatel, d'activer ou non une "fonction standard" par simple pression d'une touche dédiée : quand la fonction standard est activée sur un poste, ce poste sonne en plus du poste dédié au standard quand on reçoit un appel de l'extérieur sur le numéro du standard, et c'est le premier qui décroche qui obtient la communication. Cela permet de ne pas rater des appels au standard quand la standardiste est déjà au téléphone ou quand elle n'est pas à son poste.

Malheureusement, il n'existe pas d'équivalent de cette fonctionnalité prêt à l'emploi avec Asterisk et un téléphone IP. Dans le but de ne pas perdre cette fonctionnalité lors de la migration, ce qui n'aurait pas manqué d'alimenter les critiques des utilisateurs, nous avons développé nous-même cette fonctionnalité grâce à de simples scripts AGI. Malheureusement, nous n'avons pas trouvé le moyen de programmer une touche du téléphone Thomson ou de personnaliser son menu pour reproduire le fonctionnement de la touche dédiée à la fonction standard sur les téléphones Alcatel (il aurait fallu être capable d'afficher l'état d'une variable binaire et de la modifier, via un protocole quelconque). Nous avons donc implémenté le système suivant :

1. pour savoir si son poste est en "mode standard" ou non, l'utilisateur compose un numéro de téléphone qui l'amène à un menu vocal programmé dans Asterisk,
2. ce menu vocal va consulter une variable binaire dans AstDB (Asterisk Database) pour le poste en question. Cette variable binaire a été créée spécialement pour cette fonctionnalité : si elle contient 1 alors la fonctionnalité standard est activée, si elle contient zéro elle est désactivée. Le serveur vocal annonce donc à l'utilisateur si la fonctionnalité est activée pour le poste ou non (via des phrases pre-enregistrées par nos soins).
3. ce menu propose alors d'appuyer sur une touche pour modifier l'état de la variable ou de raccrocher si on ne veut pas la modifier.

Au final, cela donne par exemple une annonce vocale du style "La fonctionnalité standard de votre poste est actuellement désactivée. Si vous voulez activer la fonctionnalité standard, appuyez sur la touche 1, sinon, vous pouvez raccrocher". C'est un peu fastidieux, mais ça marche bien. Quand on reçoit un appel au standard :

1. un script AGI s'exécute et consulte l'état de la variable binaire dédiée à la fonction standard pour chaque poste. Ce script retourne la liste des postes pour lesquels la variable est à 1
2. Asterisk fait sonner, en plus du poste dédié au standard, tous les autres postes de la liste renvoyée par le script AGI
3. avec cette configuration, c'est le premier qui décroche qui prend la communication.

Cette solution marche bien mais n'est pas très pratique à l'utilisation. Nous ne l'avons finalement pas retenue et nous avons adopté une solution beaucoup plus radicale :

1. quand on reçoit un appel au standard, le poste dédié au standard sonne pendant 15 secondes ;
2. s'il n'a pas décroché au bout de ce temps là, tous les postes présents dans une liste définie par moi-même se mettent à sonner en plus du standard, pendant 15 secondes supplémentaires ;
3. si toujours aucun poste ne décroche, l'appel est basculé sur le répondeur du standard.

L'avantage de cette solution est qu'il permet de pallier au problème suivant : les utilisateurs avaient tendance à ne pas activer la fonction

standard sur leur poste pour ne pas assumer la corvée de prendre des appels du standard quand la standardiste n'est pas disponible pour prendre l'appel. L'inconvénient est qu'il faut faire appel à l'administrateur de l'IPBX quand on veut modifier la liste des postes déclarés "solidaires" du standard après les 15 premières secondes.

## Société de service

Pour nous accompagner en cas de besoin pendant le déploiement et pour assurer la maintenance du système dans les cas où nous ne saurions pas résoudre les problèmes techniques nous-même, nous avons choisi de recourir aux prestations d'une société de service. Cela permet aussi de s'entourer de gens expérimentés sur Asterisk et la VoIP et d'avoir une personne qui peut intervenir sur le système quand je suis absent de l'entreprise.

Nous avons mis en concurrence deux sociétés :

- Proformatique, qui est à ma connaissance la seule société française spécialisée sur les solutions VoIP basées sur Asterisk. Elle propose soit de vendre une *appliance Asterisk*, i.e. un IPBX basé sur un PC et des cartes téléphoniques sur lequel tourne Asterisk et dont ils maîtrisent intégralement le logiciel (solution dénommée *Xivo*); soit d'accompagner leurs clients dans le déploiement d'Asterisk sur leur propre plateforme matérielle et logicielle.
- Morea, société de service informatique pour les petites entreprises, qui a de fortes compétences sur Asterisk. Elle propose soit d'installer leur propre version améliorée de Trixbox soit d'accompagner leurs clients dans le déploiement d'Asterisk sur leur propre plateforme matérielle et logicielle.

Les deux sociétés sont à mon avis un bon choix. Nous avons finalement choisi la société Morea car nous connaissions un des associés de la société, qui nous avait prouvé son sérieux sur des projets antérieurs. Le choix le plus important consiste à opter pour une solution packagée proposée par la société ou pour une solution déployée sur sa propre plateforme matérielle et logicielle. Ces sociétés vous inciteront à opter pour la première solution, qui est plus sécurisante pour eux car ils connaissent parfaitement l'ensemble des composants matériels et logiciels de l'IPBX puisqu'ils ont déjà eu l'occasion de déployer le système à l'identique chez de nombreux autres clients. La deuxième solution a l'avantage pour nous d'être plus indépendant de la société de service retenue, mais elle nécessite de se plonger beaucoup plus profondément dans les aspects techniques d'un déploiement Asterisk, ce qui prend du temps.

## Installation et configuration du serveur Asterisk

Comme je l'ai dit dans le chapitre précédent, j'ai choisi d'installer Asterisk et les drivers "à la main" sur une distribution Debian stable dans un PC Dell Optiplex GX620 doté d'une carte RNIS Digium B410P et d'une carte Digium TDM400P avec 1 port FXS.

## Préparation du hardware

Tout d'abord, il convient de préparer la configuration hardware. La carte B410P a besoin d'être configurée : chaque port RNIS peut être dans un de ces deux modes :

- mode *NT*, qui signifie *Network Termination*, qu'il faut utiliser si vous raccordez le port RNIS à un téléphone RNIS par exemple ;
- mode *TE*, qui signifie *Terminal Equipment*, qu'il faut utiliser si vous raccordez le port RNIS à la prise RNIS de France Telecom.

Référez-vous à la documentation de la carte pour savoir reconnaître et changer le mode d'un port. J'ai donc vérifié que tous les ports de ma carte étaient bien en mode TE (la carte nous a été livrée dans cette configuration).

J'ai ensuite installé chaque carte dans un emplacement PCI. Attention, pour que les ports FXS de la carte TDM400P puissent fonctionner, la carte a besoin d'être alimentée comme un disque dur (probablement parce que l'alimentation du bus PCI ne lui suffit pas) : il m'a fallu acheter une rallonge d'alimentation Molex car les câbles de l'alimentation n'étaient pas assez longs.

J'ai enfin installé 2 disques durs SATA pour pouvoir ensuite les configurer en RAID 1 (le RAID 1 est le mode *miroir* : il assure la redondance des données et permet de continuer l'exploitation si un des deux disques durs est en panne), en utilisant le RAID soft de Linux. Si vous avez un peu de temps, je vous conseillerai de réaliser un test SMART sur les 2 disques durs pour vérifier qu'ils ne sont pas en mauvais état avant de commencer.

## Installation Debian Etch

J'ai installé une distribution Debian stable, qui est actuellement la version 4.0 dite *Etch*. J'ai fait un partitionnement très simple avec une partition de swap de même taille sur chaque disque et une partition racine en RAID 1 (évidemment de même taille sur chaque disque). A la fin de l'installation, je n'ai sélectionné aucun des ensembles de packages proposés, pour ne pas me retrouver avec un serveur graphique et des logiciels inutiles. J'ai fait le choix d'une machine dédiée à Asterisk ; je ne veux donc avoir logiciel installé non nécessaire au fonctionnement d'Asterisk.

A la fin de la procédure d'installation, j'ai supprimé les packages inutiles, installé mon shell et mon éditeur de texte préféré ainsi qu'un

serveur SSH pour pouvoir accéder à la machine à distance. C'est le moment d'installer également, si ce n'est pas déjà fait, les outils nécessaire à la compilation d'un nouveau noyau, des drivers et d'Asterisk : vous aurez besoin des packages *gcc*, *make*, *flex*, *libncurses5-dev* et probablement d'autres encore...

## Installation du nouveau noyau

J'ai choisi d'installer un noyau 2.6.20.15, car c'était au moment du déploiement la dernière version de la branche 2.6.20, qui avait déjà acquis un peu de maturité à l'époque. Je mets en partage mon fichier `.config` pour ceux que ça intéresse.

Après avoir rebooté sur le nouveau noyau et vérifié que tout fonctionnait correctement (RAID soft en état synchronisé, carte réseau fonctionnelle, ...), je suis passé au coeur du sujet !

## Installation des drivers et des logiciels

Il ne s'agit pas ici de détailler toute la procédure d'installation d'Asterisk... elle est déjà expliquée dans de très nombreux tutoriels, mais seulement de mettre en évidence les problèmes que j'ai rencontrés.

J'ai procédé dans cet ordre :

1. installation des drivers Zaptel pour supporter la carte TDM400P.
2. installation des drivers RNIS pour supporter la carte B410P : installation de `mISDN` (les modules noyau) et `mISDNuser` (la partie userspace).
3. après l'installation des drivers, j'ai redémarré la machine pour vérifier qu'ils se chargeaient bien au démarrage et reconnaissaient les deux cartes... et j'ai eu la mauvaise surprise de voir que ça ne se passait pas bien ! Après avoir un peu fouillé, j'ai fini par m'apercevoir que le module `netjetpci.ko` fourni par `mISDN` reconnaissait la carte TDM400P alors que ce sont les drivers Zaptel (en l'occurrence le module `wctdm`) qui doit assurer le pilotage de la carte. Au final, j'ai supprimé le fichier `/lib/modules/2.6.20.15/extra/netjetpci.ko` et j'ai ajouté `wctdm` dans le fichier `/etc/modules`.
4. installation de `libpri`, qui est une librairie utilisée par Asterisk pour certaines fonctions liées au RNIS.
5. installation d'Asterisk, en vérifiant qu'il reconnaît bien l'installation de `mISDN` lors du `./configure` (sinon il ne compilera pas `chan_misdn.so`, qui est le module d'Asterisk qui gère l'accès aux lignes RNIS). J'ai appliqué sur les sources d'Asterisk le patch proposé par Thomson pour faire marcher le `call pickup` (l'interception d'appel en français) via les touches de fonction des téléphones. Ce patch est disponible sur [le site de Thomson](#) (fichier ZIP *Asterisk 1.4 patch for Call Pickup* du 18/10/2007), et consiste à modifier deux fonctions du fichier `channels/chan_sip.c`. Pour le moment, ce patch ne m'a malheureusement pas permis de faire marcher la fonction qu'il est censé faire marcher !

Pour l'installation du fax2mail, j'ai procédé en deux temps :

1. Installation d'IAXmodem et de Hylafax en suivant [ce tutoriel](#). Comme proposé dans le tutoriel, j'ai installé la dernière version d'IAXmodem en prenant les sources sur le site du projet et j'ai utilisé les packages Debian de la Etch pour installer Hylafax.
2. Installation d'Avantfax en suivant [la doc d'installation](#) (il faut avoir un peu de temps devant soi !)

## Déploiement et configuration des téléphones ST2030

Je ne vais pas réexpliquer ici ce qui est expliqué dans la documentation de Thomson, mais plutôt vous faire part de mon expérience dans ce domaine.

Vu que la taille de notre parc (plusieurs dizaines de téléphones), il est hors de question de les configurer un par un via leur interface Web. Nous avons donc opté pour la configuration en "provisionnement TFTP". Le plus long et fastidieux a été de mettre au point les fichiers de configuration, car il y a énormément de paramètres, qui n'ont pas tous la même importance... Je vous fournis mes fichiers de configuration, en espérant que ça vous fera gagner du temps.

Premier piège à éviter : quand on fait une modification dans un fichier de configuration du téléphone en partage TFTP, si on veut que le téléphone daigne tenir compte de la modification à son prochain reboot, il y a souvent une manipulation supplémentaire à faire, qui est détaillée dans le tableau ci-dessous.

Nom du fichier	Appellation en anglais dans la doc Thomson	Rôle	Mes modifications	En cas de modification d'un paramètre du fichier
ST2030S_MACtelephone.inf	Provisioning INF	Lien vers les autres fichiers de configuration	Personnalisation simple : changement des noms des fichiers	Rien à faire. Comme ce fichier contient les noms de presque tous les autres fichiers, il faut bien sûr le mettre à jour à chaque fois qu'un autre fichier change de nom
ST2030S_MACtelephone.txt	MAC-specific config file	Fichier contenant les paramètres spécifiques à chaque téléphone	Personnalisation complète	Changer le paramètre <i>config_sn</i> du fichier

ComConf2030S_version.txt	Common Config	Fichier contenant les paramètres communs à tous les téléphones	Personnalisation complète	Changer le nom du fichier
TelConf2030SG_version.txt	Telephone Config	Paramètres "bas niveau" du téléphones et paramètres des codecs	1 seul paramètre modifié : <code>vad</code> mis à <code>off</code> pour le codec G711A	Changer le nom du fichier
v2030SG.version.zz	Firmware	Fichier contenant le micrologiciel du téléphone	Fichier binaire, aucune modification possible	-
Melodies.txt	Melody	Fichier contenant les musiques d'attente (je crois)	Aucune	Changer le nom du fichier
Sys_Ringtones.txt	System Melodies	Fichier contenant les sonneries	Aucune	Changer le nom du fichier
Bellcore_CW.txt	Call Waiting tones	Fichier contenant les tonalités d'attente	Aucune	Changer le nom du fichier

Au niveau de mon serveur TFTP, tous les fichiers *ST2030S\_MACtelephone.inf* sont des liens symboliques vers le même fichier que j'ai appelé *ST2030S\_all.inf*.

J'ai mis en ligne mes fichiers ComConf2030S, ST2030S\_all.inf et un de mes fichiers ST2030S\_MACtelephone.txt ; vous les trouverez à l'adresse <http://people.via.ecp.fr/~alexis/asterisk/st2030/>. J'ai mis un maximum de commentaires à l'intérieur des fichiers. Vous êtes libres de les utiliser... j'espère que cela vous sera utile !

Deuxième piège : il faut toujours partir du fichier *TelConf* fourni par Thomson. Je veux dire qu'à chaque fois que vous voulez mettre à jour vers un nouveau firmware, il ne faut pas garder le fichier TelConf que vous utilisiez avec le firmware précédent (qu'il ait été personnalisé ou non), mais toujours prendre le fichier TelConf fourni avec le nouveau firmware, et le re-personnaliser si besoin. Dans mon cas, la seule modification que j'apporte au fichier TelConf de Thomson est la désactivation de l'auto-détection de la voix (paramètre VAD comme Voice Auto Détection) pour le codec que j'utilise (codec G711A). Dans la section G711A du fichier TelConf, j'ai donc :

```
set coding 0 vad off
```

Truc et astuce : pour rebooter facilement le téléphone, il suffit d'appuyer simultanément sur les touches répertoire, haut-parleur et

bis (touche avec une double flèche vers le haut).

## Changement du paramétrage réseau

Comme je l'ai expliqué dans la section *Réseau* du chapitre *Choix et motivation des choix*, nous avons choisi d'utiliser de la QoS au niveau IP, appelée DiffServ, pour mettre le trafic VoIP en priorité par rapport à tous les autres traffics. La méthode retenue a été de tagger les paquets IP avec un champ DSCP particulier et de déclarer au niveau du matériel réseau que les flux ainsi taggés ont une priorité maximale.

Tout d'abord, je vous conseille la lecture de [l'article sur DiffServ](#) dans Wikipedia. Pour vérifier que vous avez bien tout compris, voilà une petit résumé : dans l'entête des paquets IP, il y a 8 bits réservés au champ TOS (Type of Service), aussi appelé DiffServ. Les 6 premiers bits du champ TOS constituent le champ DSCP. Par convention, on met ce champ à la valeur *101 110* pour les flux de priorité maximale ; cela s'appelle la classe DSCP "Expedited Forwarding" (EF). Les 2 derniers bits forment le champ ECN (Explicit Congestion Notification), qui ne nous intéressent pas dans notre cas.

Nous voulons donc que les flux VoIP soient taggés ainsi :

Champ	Binaire	Hexadécimal	Décimal
DSCP	101 110		
TOS/DiffServ	101 110 00	0xB8	46

Pour mettre en place la QoS, il faut :

- que le téléphone tagge les flux IP qu'il envoie, ce qui se configure dans le fichier "Common config" des téléphones ST2030 (paramètres *DSCPflag=2*, *DSCP1=46* et *DSCP2=46*),
- que Asterisk tagge les flux IP qu'il envoie, ce qui se configure dans le fichier *sip.conf* (paramètres *tos\_sip=ef* et *tos\_audio=ef*),
- que les switches donnent la priorité maximale aux flux ainsi taggés... c'est ce qui nous reste à faire !

Sur les switches HP Procurve modèles 2510, 2810, 2824 et 2848 (la fonctionnalité de QoS n'est pas disponible sur les Procurve 2524), cela se configure de la façon suivante :

```
switch(config)# qos type-of-service diff-services
```

Pour vérifier que la fonctionnalité QoS DiffServ a bien été activée :

```
switch# show qos type-of-service
Type of Service [Disabled] : Differentiated Services
Codepoint DSCP Policy | Priority
```

```

-----+-----
000000 | No-override
[...] |
101101 | No-override
101110 | 7
101111 | No-override
[...] |

```

Il faut dire au switch que les flux dont le champ DSCP est 101 110 doit être traité en priorité maximale, qui est la priorité 7. En fait, c'est le réglage par défaut, comme on peut le voir ci-dessus.

## Les imperfections qu'il reste à corriger

Voilà la liste des points qu'il me reste à résoudre pour être parfaitement satisfait de mon déploiement VoIP avec Asterisk :

- l'interception d'appels sur les téléphones ST2030 en appuyant sur la touche de monitoring qui clignote, sachant que le pick-up "manuel" réalisé en composant \*8<numero du poste qui sonne> fonctionne bien, cf [mon post sur ce sujet](#) dans le forum d'Asterisk France.
- la présentation du numéro à l'extérieur : le numéro de téléphone qui est présenté aux correspondants externes qu'on appelle n'est parfois pas bon, cf [mon post sur ce sujet](#) dans le forum d'Asterisk France.
- la présentation du numéro en interne : le poste A reçoit un appel de l'extérieur et voit le numéro du correspondant qui s'affiche sur son écran ; il décroche puis transfère l'appel vers le poste B ; le poste B est alors en ligne avec le correspondant extérieur mais le numéro qui est présenté sur son écran est celui du poste A.
- il y a souvent des erreurs de transmission lors de l'envoi de fax via le fax traditionnel connecté à la carte TDM400P. Pour essayer de résoudre le problème, j'ai récemment désactivé la fonction d'echo cancel dans la configuration de Zaptel (`echocancel=no` dans `zapata.conf`), après avoir lu qu'il ne fallait pas activer ce genre de fonction pour l'envoi de fax, et j'attends de voir les résultats. J'ai aussi lu le paragraphe intitulé *Zap and Digium card issues* de la page [Asterisk and fax calls](#) de [voip-info.org](#), où ils disent qu'il peut y avoir un certain nombre de problèmes mais que ça marche généralement bien.

Si vous connaissez la solution à l'un de ces problèmes, je serai ravi de recevoir un mail de votre part !

## Commandes de diagnostic

### pour les téléphones

Voici une liste de commandes que j'utilise pour diagnostiquer un problème sur un téléphone, après m'être connecté en telnet sur le poste :

- Pour voir la version du firmware :

```
# version
```

- Pour voir la version des différents fichiers de configuration, ce qui me permet de vérifier que le téléphone est bien à jour avec la dernière version de des fichiers de configuration :

```
# autoprovision show
```

- Après, quand je veux vérifier qu'un certain paramètre du fichier de configuration a bien été pris en compte, je cherche dans le document *ST20XXS\_Config File Syntax\_version.pdf* (fourni dans le ZIP du firmware de Thomson) la commande permettant d'afficher la valeur du paramètre et je la tape dans la ligne de commande.

## pour Asterisk

En cas de problème avec Asterisk, j'ai le réflexe d'utiliser les commandes suivantes au prompt d'Asterisk ou au prompt du shell du serveur :

- Pour voir les entités SIP :

```
asterisk*CLI> sip show peers
```

- Pour les lignes mISDN :

```
asterisk*CLI> misdn show stacks
BEGIN STACK_LIST:
* Port 1 Type TE Prot. PMP L2Link DOWN L1Link:DOWN Blocked:0 Debug:2
* Port 2 Type TE Prot. PMP L2Link DOWN L1Link:DOWN Blocked:0 Debug:2
* Port 3 Type TE Prot. PMP L2Link DOWN L1Link:DOWN Blocked:0 Debug:2
* Port 4 Type TE Prot. PMP L2Link DOWN L1Link:DOWN Blocked:0 Debug:2
```

Remarque : quand une ligne RNIS n'est pas occupée, c'est normal que le paramètre *L2Link* et *L1Link* soit à *DOWN*, car les lignes RNIS ne sont alimentées que quand il y a un appel en cours ou en attente.

- Pour la carte TDM400P :

```
asterisk*CLI> zap show status
Description                               Alarms    IRQ      bpviol   CRC4
Wildcard TDM400P REV I Board 1           OK        0        0        0
```

et au prompt du shell :

```
# ztcfg -vv
```

- Quand je veux voir les échanges SIP entre un téléphone et Asterisk, je suis parfois amené à utiliser *tcpdump* :

```
# tcpdump -n -A -s 65535 host 10.1.2.3 and port 5060
```

où 10.1.2.3 est l'IP du téléphone.

XHTML 1.1 et CSS valides