

# SIP Communicator – Un outil open source de communication sur IP adapté à nos laboratoires et à nos universités

Emil Ivov

Université Louis Pasteur , LSIIT  
Pole API, 5 Bd Sébastien Brant  
67412 Illkirch, France  
emcho@sip-communicator.org

Jean-Marc Muller

LSIIT / CNRS  
Pole API, 5 Bd Sébastien Brant  
67412 Illkirch, France  
muller@lsiit.u-strasbg.fr

## Résumé

*SIP Communicator ( <http://sip-communicator.org> ) est une application open source (licence LGPL), dont le développement a été initié à l'Université Louis Pasteur, Strasbourg, et comprend aujourd'hui plus de 800 membres à travers le monde.. Ce logiciel permet d'établir des conversations audio et vidéo au travers de l'Internet via le protocole SIP. Il supporte également l'échange de messages instantanés en utilisant les protocoles les plus populaires tels Jabber, AIM/ICQ, MSN, Yahoo! Messenger, Bonjour ou IRC.*

*Ce document décrit l'histoire, les fonctionnalités, et l'architecture de SIP Communicator et énumère les avantages qui, selon nous, font de lui un outil bien adapté aux besoins des universités et laboratoires.*

## Mots clefs

CoIP, SIP, Jabber, Bonjour, Téléphonie, Messagerie Instantanée, VoIP, ToIP, CoIP

## 1 Introduction

De plus en plus d'universités et laboratoires prévoient une migration de leurs services de communication vers des architectures IPs. En plus de l'extensibilité et de la facilité de configuration qu'une telle migration donnerait à la téléphonie, elle permettrait également de supporter de nouveaux services tels que les appels vidéo, la messagerie instantanée ou le travail collaboratif, qui sont de plus en plus demandés dans les milieux de la recherche et de l'industrie.

En dehors de l'infrastructure qui doit être déployée pour assurer le support de ces services, se pose le choix des applications de télécommunications à utiliser. Il paraît évident que les services conventionnels tels que la voix peuvent en grande partie être assurés par des équipements dédiés comme les hard phones ou des adaptateurs VoIP par exemple. Cependant, pour profiter pleinement des

possibilités offertes par l'architecture IP et pouvoir utiliser tous les services disponibles il est indispensable d'installer des logiciels de communication.

SIP Communicator répond à l'ensemble des fonctionnalités que doit posséder un tel logiciel. Parmi ses qualités, on compte le support des conversations audio et vidéo avec SIP [1], un développement ouvert sous une licence libre, le support de plusieurs systèmes d'exploitation et plusieurs protocoles de communication. Il est compatible avec le protocole IPv6 et repose sur une architecture permettant une grande facilité de déploiement, de mise à jour et d'ajout de nouvelles extensions.

La suite de ce document est organisée de la manière suivante. La section 2 présente brièvement l'histoire de SIP Communicator. Les sections 3 et 4 décrivent ses fonctionnalités existantes ainsi que celles qui sont en cours de développement. La section 5 décrit l'architecture du logiciel. La section 6 expose les points qui, selon nous, font de SIP Communicator, un client adapté aux universités et laboratoires de recherche (et plus particulièrement des institutions françaises). La section 7 termine l'article par une brève conclusion.

## 2 Historique

Les travaux sur SIP Communicator ont débuté avec un projet de maîtrise informatique réalisé par Emil IVOV à l'université Louis Pasteur de Strasbourg. Cela faisait plusieurs années que ce dernier était passionné par la téléphonie IP, il en a donc profité pour se lancer dans la création d'un client VoIP.

À la fin de son projet, il a décidé de continuer à travailler sur le sujet, encouragé par son responsable, Thomas Noël, maître de conférences à l'ULP. Ensemble, ils ont décidé de rendre le projet public (Open Source) avec une licence libre. C'est ainsi que le projet SIP Communicator a été créé sur le site d'hébergement Java.NET, aujourd'hui disponible avec son propre nom de domaine : [www.sip-communicator.org](http://www.sip-communicator.org).

Cela fait bientôt quatre ans que l'application existe. Parmi ses utilisateurs, on retrouve des développeurs de plusieurs universités et des sociétés telles que France Telecom, British Telecom, Alcatel-Lucent, etc. En raison de son origine expérimentale, SIP Communicator est avant tout utilisé par les milieux de recherche et de développement et pour l'instant, il n'y a que très peu d'utilisateurs non professionnels.

Depuis novembre 2005, une nouvelle version a été créée, afin de rendre l'application stable, évolutive et facilement extensible. Cette dernière version a impliqué des changements fondamentaux aussi bien au niveau de l'architecture et des fonctionnalités, que de l'interface graphique de l'application.

SIP Communicator n'est plus le fruit du travail d'une seule personne : une communauté de développeurs gravite aujourd'hui autour de cette initiative. Le modèle open source retenu a permis d'enrichir très rapidement le produit par de nombreuses fonctionnalités indispensables pour une réelle percée aussi bien dans le monde professionnel que dans les foyers. Enfin, le support humain et matériel fourni par l'Université Louis Pasteur a également permis de pérenniser un certain nombre de travaux. Ce support se traduit encore actuellement par plusieurs transferts technologiques réalisés par l'équipe de recherche en Réseaux du LSIIT – Laboratoire d'informatique de l'Université Louis Pasteur de Strasbourg (UMR 7005 ULP/CNRS).

Après un peu plus d'une année de développement intensif (basé sur une nouvelle architecture modulaire utilisée également par de grands acteurs industriels comme Mercedes), SIP Communicator supporte les protocoles de messagerie instantanée et de téléphonie les plus connus, tels que Jabber, ICQ, AIM, GoogleTalk, MSN, Yahoo! Messenger, Bonjour et SIP. L'application inclut également un très grand nombre de fonctionnalités indispensables pour son utilisation. Il permet de gérer l'ensemble des contacts quel que soit le protocole utilisé. Il permet d'effectuer des appels audio et vidéo et d'enregistrer l'historique des appels et des messages. Mais SIP Communicator se veut bien plus qu'une simple application avec des services déjà bien connus. C'est avant tout une architecture logicielle modulaire développée avec l'ambition de supporter de nombreux nouveaux services. Ces services pouvant être intégrés sans pour autant remettre en cause un processus de développement complet et lourd. Enfin, SIP Communicator peut être vu comme un socle de support dynamique. Il sera capable d'intégrer à la volée de nouveaux services développés par des sociétés tierces.

Comme cela a été mentionné précédemment, il se développe en parallèle une communauté toujours grandissante de personnes qui contribuent au projet de différentes manières. Sur les 800 personnes que compte actuellement la communauté, une grande partie intervient en tant qu'observateur, participe aux discussions proposées sur les listes de diffusion, donne son avis sur le design, le développement ou le planning. D'autres personnes contribuent de manière ponctuelle en soumettant des rapports d'anomalies ou des corrections de bogues. Enfin

certaines membres s'impliquent davantage en participant activement au projet.

Cette année, SIP Communicator a été retenu pour participer au programme « Google Summer of Code » [8]. Ceci est pour nous une réussite et une véritable reconnaissance par la communauté Open Source au vu des autres projets acceptés dans le programme (e.g. Apache, Firefox, OpenOffice, Gimp ou Gaim).

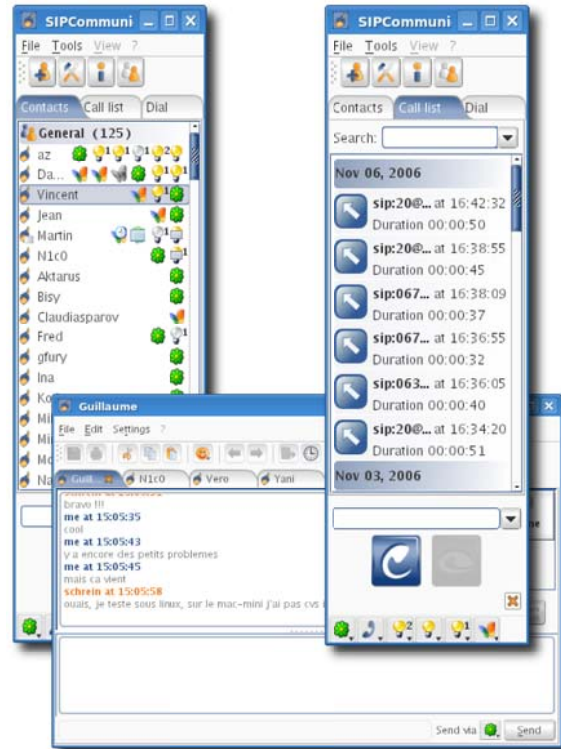


Figure 1 - Captures d'écran de SIP Communicator

### 3 Présentation des fonctionnalités existantes

Après deux années de développement actif, SIP Communicator a atteint un niveau de maturité où il peut être utilisé de manière journalière en tant que client de communication par défaut. Malgré son stade expérimental (version alpha) il intègre déjà la plupart des fonctionnalités qu'un utilisateur s'attend à trouver dans ce genre d'application.

#### 3.1 Conversations audio et vidéo

Depuis sa naissance en 2003, SIP Communicator fournit le support des conversations audio et vidéo via le protocole SIP. Ce protocole de téléphonie IP s'impose de plus en plus comme le standard de la voix sur IP. De nos jours une grande variété de produits (ex: soft phones) et de services commerciaux l'utilisent. SIP Communicator supportant le protocole SIP (comme son nom l'indique) est compatible avec l'ensemble de ces produits et donc utilisable avec la plupart des matériels existants.



Figure 2 - Session video avec SIP Communicator

Très récemment SIP Communicator a intégré en plus du protocole SIP, le support des appels audio avec l'extension de Jabber – Jingle [2].

### 3.2 Messagerie instantanée

Malgré sa popularité pour la téléphonie, le protocole SIP n'est pas très populaire pour la messagerie instantanée. SIP Communicator supporte de ce fait les principaux protocoles de messagerie instantanée:

- Jabber/GoogleTalk
- AIM
- ICQ
- MSN
- Bonjour
- Yahoo! Messenger

Il permet à l'utilisateur d'avoir accès simultanément dans un même environnement graphique à l'ensemble de ses comptes de messagerie instantanée en profitant des fonctionnalités de chacun.

### 3.3 Support multi plates-formes

SIP Communicator est écrit en JAVA et peut fonctionner sur la plupart des plates-formes majeures : Linux, BSD, Windows et Mac OS X. Des installateurs adaptés sont disponibles pour téléchargement sur <http://sip-communicator.org>

### 3.4 Support de multi-chat avec Jabber et IRC

Le chat à plusieurs est une application des plus populaires sur Internet depuis sa démocratisation. Aujourd'hui il représente beaucoup plus qu'un simple divertissement, il est utilisé dans les systèmes d'enseignement à distance ou le travail collaboratif.

A ce jour SIP Communicator permet à ses utilisateurs de participer à des sessions de multi-chat (ou chatrooms) avec les protocoles Jabber et IRC.

### 3.5 Gestion des plugins

SIP Communicator est bâti sur une architecture modulaire qui permet d'activer ou désactiver certaines fonctionnalités mais également d'en rajouter de nouvelles, non-fournies avec la version originale ou produite par des tiers. L'interface actuelle permet la gestion de l'ensemble de ses modules ainsi que l'installation des nouvelles fonctionnalités.

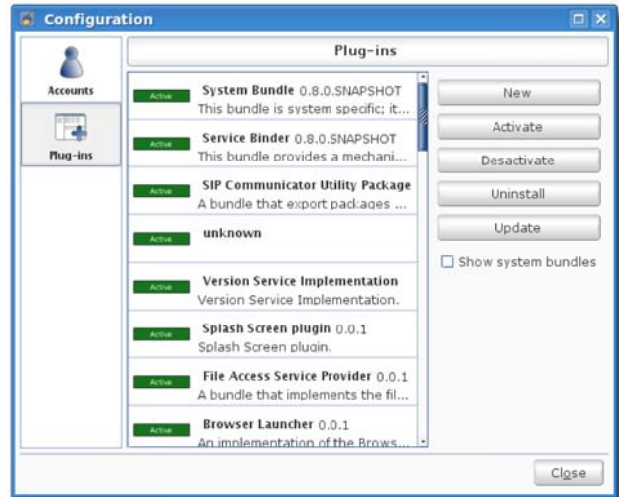


Figure 3 - Interface de gestion des modules

### 3.6 Et d'autres gadgets ...

L'architecture de SIP Communicator permet le développement et l'intégration de services qui ne sont pas directement liés aux communications.

Il est par exemple possible de s'abonner à des flux RSS et de les gérer comme s'ils étaient des contacts. Tout nouvel élément dans le flux est ensuite présenté comme un nouveau message instantané.

Une autre fonctionnalité implémentée pendant le Google Summer of Code 2007 permet de rajouter des machines dans sa liste de contacts. Les messages qui leurs sont envoyés sont interprétés comme des commandes SSH.

SIP Communicator supporte également ce qu'on appelle une *Meta Contact List*. Elle permet aux utilisateurs de regrouper sous un même contact toutes les adresses que possède une personne quelque soit le protocole de messagerie utilisé.

## 4 Fonctionnalités en cours de développement

La communauté de SIP Communicator est de plus en plus active, ce qui nous permet de nous concentrer sur un plus grand nombre de fonctionnalités. En voici quelques unes, actuellement développées par les différents contributeurs.

## 4.1 Tableaux blancs partagés.

Cette fonctionnalité qui propose un espace de dessin partagé et modifiable interactivement par plusieurs personnes, a été développée dans le cadre d'un projet étudiant dans l'édition 2007 de Google Summer of Code. Elle est actuellement en cours d'intégration et de test.

Les tableaux blancs se basent sur une extension XMPP [12] qui sert à transporter le contenu du tableau au format SVG.

## 4.2 Communications en P2P

Les services de communication tels que les appels audio et vidéo, les applications collaboratives et les téléconférences à plusieurs sont en général particulièrement coûteux en terme de ressources nécessaires à leur déploiement. C'est pour cette raison que l'équipe de SIP Communicator s'investi tout particulièrement dans la recherche et le développement de mécanismes permettant la distribution de ces services sur des réseaux d'overlay peer-to-peer.

Une telle architecture permettra de diminuer les coûts nécessaires au déploiement de services de téléphonie IP en utilisant de manière distribuée les ressources des machines utilisateurs.

## 4.3 Sécurité des communications

Dans la majorité des cas, les conversations téléphoniques sont confidentielles et leur contenu est sensé être protégé. C'est un des points auxquels la téléphonie sur IP ne répond pas toujours.

Pendant l'été 2007 nous avons commencé l'intégration du protocole SRTP dans SIP Communicator. Son utilisation permettra aux utilisateurs d'avoir le même niveau de sécurité lors des appels fait avec SIP Communicator qu'avec la téléphonie classique.

Le rôle de SRTP devient encore plus important dans un environnement peer-to-peer ou les flux média peuvent transiter par d'autres machines utilisateurs et un manque de cryptage compromettrait très sérieusement la confidentialité des appels.

## 4.4 Une connectivité fiable et robuste avec le protocole ICE

Lors des débuts de la téléphonie sur Internet, les protocoles et les implémentations existantes étaient soumis à beaucoup moins de contraintes qu'aujourd'hui. Les translateurs d'adresses (NATs) n'étaient pas communs et les machines des utilisateurs disposaient dans la majorité des cas d'une seule adresse IP (généralement publique).

De nos jours il n'existe quasiment plus d'utilisateurs avec des adresses IPv4 publique, et les NATs sont présents dans la majorité des installations. Il est de plus en plus commun pour une machine de disposer de plusieurs adresses IP: des adresses IPv4 et IPv6, des adresses VPN ou des tunnels Mobile IPv6 etc..

C'est pour cette raison que le protocole d'Etablissement Interactif des Connexions (ICE) [3] a été créé. Il permet une négociation beaucoup plus fiable des adresses qui vont être choisies pour l'échange des flux médias et garantit une transition transparente vers IPv6.

Le développement d'une pile ICE (le protocole étant assez récent, il n'en existe pas actuellement en Java) et son support dans SIP Communicator est une de nos priorités.

## 4.5 Encore d'autres protocoles de communication

En plus des protocoles actuellement supportés par SIP Communicator nous souhaitons en intégrer de nouveaux. Le protocole IAX (Inter-Asterisk eXchange)[4] par exemple prend de plus en plus d'ampleur dans le monde de la téléphonie grâce à sa simplicité et de sa capacité à s'adapter aux environnements NATs. Ce protocole permet la communication entre client et serveur ainsi qu'entre serveurs. Il a été conçu pour le contrôle et la transmission de flux multimédia, contrairement à SIP qui lui ne gère que la signalisation. Grâce à cette double fonctionnalité IAX permet une intégration relativement facile dans les réseaux NATés.

Malgré sa jeunesse et sa non-standardisation, IAX est de plus en plus utilisé ce qui rend son support dans SIP Communicator assez intéressant.

## 4.6 Géolocalisation

L'intégration de la géolocalisation permettra aux utilisateurs de SIP Communicator de visualiser la position géographique des membres de leur liste de contacts.

L'implémentation de cette fonctionnalité est réalisée selon l'extension XMPP – XEP-0080[5].

## 4.7 Des entrepôts centralisés

Comme nous l'avons déjà mentionné, SIP Communicator permet une gestion très fine des fonctionnalités en cours d'utilisation ainsi que l'installation et les mises à jour de nouveaux plug-ins. Nous travaillons actuellement sur l'installation et la maintenance d'un entrepôt officiel qui permettra les mises à jour automatique du logiciel de la même manière que Firefox ou Debian.

Cet entrepôt va également être utilisé pour regrouper des modules développés par des tierces parties et non inclus dans la version de base.

# 5 L'architecture

SIP Communicator est développé avec l'idée d'une extensibilité maximale. Dans ce but nous avons basé son développement sur un noyau OSGi [10] et nous utilisons l'implémentation Felix [11] créée par l'Apache Software Foundation.

Les différents modules dans l'architecture OSGi sont appelés des bundles (en français paquet). Un bundle peut exporter des services (des interfaces Java) et les rendre de cette manière disponibles aux autres parties de l'application.

Dans SIP Communicator, toute fonctionnalité existante est packagée et intégrée dans le logiciel en tant que bundle OSGi.

Cette segmentation apporte une grande souplesse et permet la création à moindre coûts de versions de SIP Communicator adaptées à un déploiement particulier. Il est possible d'imaginer par exemple que l'installation du logiciel dans un laboratoire particulier ne nécessite que le support des protocoles Jabber et SIP. Les administrateurs pourront donc facilement enlever les parties relatives aux protocoles tel que Yahoo! Messenger ou MSN pour ne pas surcharger l'interface graphique.

La suite de cette section décrit de manière plus technique les différents concepts dans l'architecture de SIP Communicator.

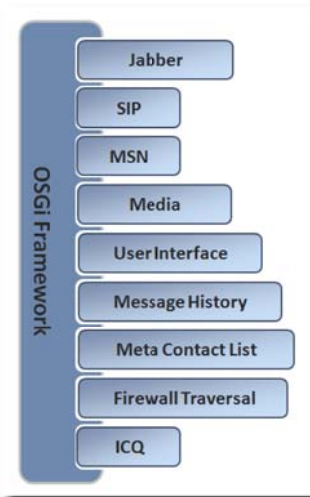


Figure 4 - Représentation simplifiée de l'architecture de SIP Communicator

## 5.1 Les différents services

Un service OSGi représente en fait un ensemble de fonctionnalités proposées par un bundle et mises à la disposition du reste de l'application. SIP Communicator définit plusieurs services tels que :

- **ProtocolProviderService** – permet l'utilisation des fonctionnalités d'un protocole de communication comme par exemple l'envoi et la réception de messages, le suivi des statuts de présence, les appels téléphoniques, etc.. Ce service est utilisé par l'interface graphique, les modules d'historique des messages et des appels ainsi que beaucoup d'autres plug-ins.
- **ConfigurationService** – Donne la possibilité aux autres parties du logiciel d'enregistrer de manière persistante des données de configuration comme par

exemple la taille des différentes fenêtres, les comptes de l'utilisateur, et d'autres préférences.

- **FileAccessService** – Ce service donne la possibilité aux autres modules de l'application et aux plug-ins externes de créer des fichiers temporaires, publics ou privés en accord avec la politique de SIP Communicator. Ceci permet par exemple de s'assurer que les différents plug-ins qui auront besoin de créer un fichier et d'y sauvegarder des détails spécifiques à un utilisateur, le placeront dans `/home/username/.sip-communicator` sous Linux et dans `C:\Documents and Settings\username\.sip-communicator` sous Windows sans que le développeur traite différemment les 2 cas.
- **NotificationService** – Gère la configuration et l'exécution de toutes les notifications suite à différents événements. Le service permet à l'utilisateur de personnaliser la manière d'être notifié à l'arrivée de nouveaux messages, appels ou autres événements et choisir entre une notification sonore, par pop-up dans la barre de notification système ou par exécution d'une commande.
- **MetaContactListService** – Donne accès à une liste commune des contacts disponibles dans tous les protocoles. En utilisant ce service, l'interface graphique ou d'autres modules, peuvent facilement afficher à l'utilisateur, l'ensemble de ses contacts et y effectuer des modifications quelque soit le protocole.
- **VersionService** – Permet à des modules comme le *splash screen* de démarrage ou la fenêtre du menu *Help->About* de consulter et afficher la version actuelle de SIP Communicator.
- **GeolocationService** – Donne des renseignements sur la position géographique actuelle de l'utilisateur. Selon les implémentations de ce service, cette position peut être récupérée d'un fichier de configuration, d'un équipement GPS ou d'un autre moyen de géolocalisation.
- **MediaService** – Donne accès à la gestion des périphériques de son ou de capture d'images ainsi que les flux médias utilisés lors d'un appel.
- **MessageHistoryService** et **CallHistoryService** – Ces services représentent des interfaces qui permettent de récupérer l'historique des messages et des appels. Ces services offrent aussi la possibilité d'effectuer des recherches basées sur des mots clés ou à des périodes précises.
- **UIService** – Ce service représente un moyen pour les plug-ins externes d'intégrer leurs composants graphiques à l'interface principale de SIP Communicator. Il permet aussi aux plug-ins de s'enregistrer et de recevoir des notifications pour certaines actions effectuées par l'utilisateur tel l'ouverture d'une fenêtre de chat avec un contact, la manipulation de la fenêtre principale etc..

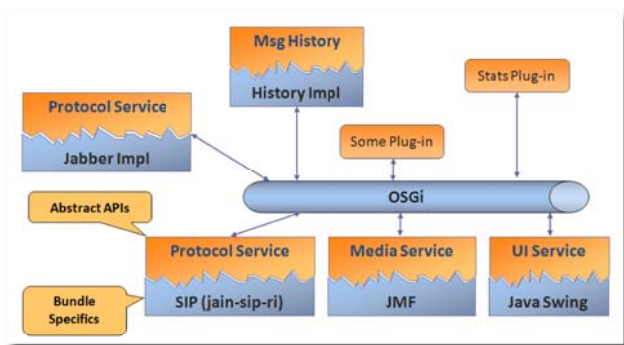


Figure 5 - Représentation des notions de service et implémentation dans SIP Communicator.

La réalisation d'un service OSGi se décompose en 2 parties:

- la définition du service même, de manière abstraite (souvent un service ne contient que des interfaces Java ou des classes abstraites);
- son implémentation.

Un service peut avoir plusieurs implémentations pouvant opérer de manière parallèle ou exclusive. Le ProtocolProviderService par exemple est implémenté pour chaque protocole supporté par SIP Communicator. Ainsi, une fois l'application lancée, ce service sera instancié autant de fois qu'il y a de comptes utilisateurs.

Pour d'autres services comme par exemple le GeolocationService, il est possible d'envisager des implémentations différentes basées sur des systèmes comme GPS, Galileo, ou Wi-Fi. Selon les équipements disponibles sur un poste utilisateur l'une ou l'autre sera utilisée.

En plus des services et de leur implémentation, on peut également avoir des bundles qui ne font qu'utiliser les services existants sans en définir de nouveaux. Dans SIP Communicator, il existe des bundles de ce type pour les assistants de création de compte, le splash screen ou l'utilitaire qui permet de visualiser l'historique des appels.

## 6 Les avantages pour les laboratoires et les universités

Comme nous l'avons déjà mentionné, SIP Communicator est codé en JAVA et s'appuie sur une architecture modulaire (OSGi) facilement extensible et permettant aisément l'adjonction de nouvelles fonctionnalités. Cette architecture facilite l'évolution du logiciel et sa personnalisation pour des besoins spécifiques d'un établissement. SIP Communicator est développé sur un modèle open source qui en plus de la garantie d'une exploitation complètement gratuite, répond également au critère important de sécurité car il enlève toute obscurité spécifique aux logiciels propriétaires. Le projet est entouré d'une communauté grandissante de développeurs qui réagit rapidement aux remontées de bugs importants et qui fait continuellement évoluer l'application.

Ces diverses qualités en font un candidat idéal au déploiement dans nos universités et nos laboratoires. Voici plus en détail les différents avantages:

### 6.1 Support de l'hétérogénéité du parc de machines

Comme mentionné précédemment, des installateurs natifs de SIP Communicator sont disponibles pour plusieurs systèmes d'exploitation. Actuellement nous maintenons des versions pour:

- Debian ;
- Fedora ;
- Installateur générique linux ;
- Mac OS X ;
- Windows ;
- Installateur java pour tout système d'exploitation sur lequel il existe une machine virtuelle java.

### 6.2 Gratuité d'utilisation

SIP Communicator est développé et distribué sous la licence open source GNU Lesser General Public Licence (LGPL) [9]. Ceci implique que toute utilisation de cette application dans un contexte commercial, de recherche ou autre n'est soumise à aucun frais d'utilisation.

### 6.3 Développement ouvert et transparent

Aujourd'hui tout le monde connaît, ne serait-ce que de nom, le logiciel Skype [6]. Ce dernier dispose effectivement de nombreux atouts: la quasi gratuité des appels téléphoniques, une très bonne qualité audio lors des conversations et surtout une forte adaptabilité aux infrastructures réseaux, ce qui lui permet de fonctionner dans la plupart des configurations réseaux existantes notamment lors de l'utilisation d'un mécanisme de translation d'adresses (NAT). Cependant les concepteurs de Skype n'ont pas consenti à lever le secret sur le mode de fonctionnement de leur logiciel, rendant son utilisation incompatible avec la plupart des politiques de sécurité officielles. Son utilisation a de ce fait été interdit au sein des laboratoires du CNRS, les établissements de recherche et d'enseignement supérieur[7].

SIP Communicator est entièrement développé sous la licence LGPL. Son code source étant disponible, son utilisation ne souffre pas du même effet boîte noire. Son fonctionnement est transparent et sécurisé. Il répond ainsi aux impératifs de sécurité qui sont fixés par le gouvernement et son utilisation est entièrement compatible avec la mise en œuvre d'une politique de sécurité des systèmes d'informations.

## 6.4 Facilité de déploiement et d'administration

Le déploiement d'une solution de communication basée sur SIP Communicator ne nécessite que peu d'effort d'installation ou maintenance. Il suffit d'installer un serveur SIP afin de gérer l'établissement des communications.

De plus, avec le support avancé des entrepôts OSGi qui est actuellement en cours de développement, une équipe d'administration pourra, avec l'installation locale d'un tel entrepôt, rendre les mises à jour automatiques. Ceci permet un délai d'intervention très court en cas d'urgence (e.g. installation de mise à jour de sécurité).

## 6.5 Conformité aux normes

Au sein de la communauté SIP Communicator, nous nous efforçons de garder au maximum le côté standard de l'application. Le support et le développement donne la priorité à des protocoles tels que Jabber et SIP, définis de manière stricte par l'*IETF* ou *The XMPP Foundation*. Pour les universités ou les laboratoires, ceci garantit l'indépendance par rapport à un protocole particulier et donne la liberté aux administrateurs de décider à tout moment la migration vers une nouvelle application.

## 6.6 Support d'un grand nombre de protocoles

Aujourd'hui SIP Communicator supporte la majorité des protocoles de communication populaires (la liste complète est décrite dans la section 3.2). Ainsi une politique prévoyant son utilisation dans un établissement particulier ne risque pas d'engendrer des conflits d'utilisation. Autrement dit, les employés de cet établissement pourront profiter des services proposés par l'infrastructure locale et continuer d'utiliser leur compte privé sans avoir recours à deux logiciels différents.

## 6.7 Facilement extensible et customisable

Comme nous l'avons décrit précédemment, il est particulièrement facile de modifier SIP Communicator pour l'adapter à un usage précis ou de rajouter de nouvelles fonctionnalités.

Un établissement qui souhaite offrir une intégration plus étroite de cette application avec ses services existants peut rajouter les modules manquant pour un coût minime de développement.

Par exemple, si une université souhaite que ses employés puissent profiter des services de voix sur IP tout en ayant accès à un annuaire LDAP (SIP Communicator ne supporte pas actuellement l'accès au serveur LDAP), elle

pourrait développer un plug-in léger qui effectue des recherches LDAP et donnerait ainsi la possibilité de passer des appels en utilisant ces résultats. Ce plug-in ne nécessiterait que très peu d'efforts et serait facilement réalisable dans le cadre d'un projet étudiant.

## 6.8 Compatibilité avec IPv6

SIP Communicator peut-être mis en œuvre aussi bien dans des réseaux IPv4 que dans des réseaux IPv6. Il est donc compatible avec le déploiement en cours dans le réseau RENATER du protocole Internet de nouvelle génération.

De plus, les efforts qui sont actuellement mis en œuvre pour l'implémentation du protocole ICE et son intégration dans SIP Communicator, garantissent une migration transparente pour l'utilisateur entre ces deux protocoles.

## 7 Conclusion

Dans un contexte d'austérité et de réduction des coûts, la téléphonie sur IP retient de plus en plus l'intérêt des établissements publics. La migration vers ce type de solution est à envisager à court terme.

Étant donné l'importance des communications, le choix des outils mis en œuvre devra être mûrement réfléchi. Il devra se faire au regard de certaines contraintes et être réalisé dans le respect des politiques de sécurité des systèmes d'informations existantes.

Son déploiement et son administration devra garantir la même qualité et facilité que la téléphonie conventionnelle. De plus le passage à cette nouvelle technologie devrait permettre d'offrir de nouveaux services de communications et collaboration en ligne.

En tant que projet open source SIP Communicator garantit une parfaite transparence de son fonctionnement et permet ainsi de contrôler la sécurité des échanges.

La simplicité de son architecture facilite le développement d'extensions et donne la possibilité aux établissements de l'adapter aisément à leurs besoins.

La communauté open source, soutenant le projet a prouvé sa vitalité et sa pérennité garantissant ainsi un suivi et une évolution constante et aisée du produit.

L'ergonomie et les fonctionnalités du logiciel le rendent également très attrayant pour l'utilisateur final.

SIP Communicator constitue donc à notre sens un candidat sérieux comme client de communication dans le cadre d'un déploiement d'une architecture de voix sur IP.

## 8 Remerciements

Nous aimerions remercier tout les membres de la communauté SIP Communicator qui lui ont permis de devenir l'application mature qu'il est aujourd'hui.

Depuis sa création, beaucoup de gens ont aidé de diverses façons à sa croissance. Nous tenons donc exprimer

notre gratitude à Thomas Noël de l'équipe Réseaux et Protocoles du LSIIT, les membres du Centre Réseaux Communication de Strasbourg et à l'Université Louis Pasteur en général pour leur soutien constant.

## Bibliographie

- [1] J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler, *SIP: Session Initiation Protocol*, IETF RFC 3261, Juin 2002.
- [2] Scott Ludwig, Joe Bedam, Peter Saint-Andre, Robert McQueen, Sean Egan et Joe Hildebrand, *XEP-0166: Jingle*, XMPP Standards Foundation, Juin 2007.
- [3] Jonathan Rosenberg, *Interactive Connectivity Establishment (ICE): A Protocol for Network Address Translator (NAT) Traversal for Offer/Answer Protocols*. IETF Work in progress, draft-ietf-mmusic-ice-17, Juillet 2007
- [4] M. Spencer, B. Capouch, E. Guy, F. Miller et K. Shumard, *IAX2: Inter-Asterisk eXchange Version 2*. IETF Work in progress, draft-guy-iax-03, Avril 2007
- [5] Joe Hildebrand et Peter Saint-Andre, *XEP-0080: User Location*, XMPP Standards Foundation, Juin 2007
- [6] Skype. <http://www.skype.com>
- [7] Message d'alerte du Fonctionnaire de Sécurité Défense du CNRS, *Recommandations pour limiter les risques de l'utilisation de Skype dans les laboratoires CNRS*, 11 Octobre 2005
- [8] Google Summer of Code  
<http://code.google.com/soc/2007/>
- [9] GNU Lesser General Public licence (LGPL),  
<http://www.gnu.org/licenses/lgpl.html> Juin 2007
- [10] OSGi™ - The Dynamic Module System for Java™;  
<http://www.osgi.org>
- [11] Apache Felix, <http://felix.apache.org>
- [12] Huib-Jan Imbens, *XEP-0113: Simple Whiteboarding*, XMPP Standards Foundation, Juillet 2003