Université Abou Bekr Belkaid
Tlemcen Algérie
جامعة أبي بكر بلقايد
تلمسان الجزائر

République Algérienne Démocratique et Populaire
Université Abou Bakr Belkaid– Tlemcen
Faculté des Sciences
Département d'Informatique

Mémoire de fin d'études

Pour l'obtention du diplôme de Master en Informatique

Option: *Réseau et Système Distribuer (R.S.D)*

Thème

Réalisation d'un serveur SIP pour réseaux multimédia

Réalisé par :

- **DIB Mohamed Ramzi**
- **YOUBI Ali**

Présenté le ***23 Juin 2014*** *devant le jury composé de MM.*

- BENAMAR Abdelkrim *(Président)*
- *ABDELLAOUI Ghouti* *(Encadreur)*
- *SMAHI Ismail* *(Co-Encadreur)*
- *BENMAMMAR Badr* *(Examinateur)*
- BENDEMRAD Tarik *(Examinateur)*

Année universitaire : 2013-2014

Annexe A : Signalisation SS7 [10] [27]

SS7 ou « Common Channel Signaling System N°4 » est un standard global de télécommunication défini par l'ITU. Ce standard définit les procédures et les protocoles par lesquels les éléments du réseau, dans un réseau à commutations de circuits, s'échangent des informations de contrôle et de routage sur un réseau digital de signalisation.

SS7 est utilisé pour :

- L'établissement et la gestion des appels ;
- Les services mobiles comme le roaming ;
- La portabilité des numéros ;
- Les services intelligents (0800, 0900…) ;
- Les services de transfert (call forwaeding).

La signalisation SS7 s'effectue par transmission de messages, entre les éléments du réseau, à une vitesse de 56 à 64 Kbps sur un canal bidirectionnel appelé « signaling link ». La signalisation se transmet en dehors de la bande de transmission réservée au transport proprement dit des données. C'est de la signalisation « Out Of Band ».

Cette signalisation offre, en comparaison avec la signalisation « In Band », de meilleurs temps de réponse pour l'établissement des appels (cfr : Multi Frequency Signaling Tones), une meilleurs utilisation des canaux voix et un support pour les réseaux intelligents.

Annexe B : MGCP (Media Gateway Control Protocol) :

Le protocole MGCP (Media Gateway Control Protocol) **[11]** résulte de la fusion des deux protocoles SGCP 1.1 (Simple Gate Control Protocol) et du protocole IPDC 1.0 (Internet Protocol Device Control). Le protocole SGCP est utilisé pour contrôler l'activité d'une « Telephony Gateway » à travers un élément de contrôle d'appel externe nommé « Call

Agent ». Le protocole IPDC est utilisé pour réaliser le control de connexions du « Media Gateway » et le transport de la signalisation. MGCP définit donc le protocole entre les passerelles et un équipement permettant de contrôler ces dernières dans le contexte d'un réseau public.

Ce protocole suppose une architecture de contrôle d'appelle dans laquelle l'élément intelligent de contrôle se trouve en dehors de Gateway, et c'est lui qui assure le contrôle de l'activité de la passerelle multimédia à l'aide d'un protocole de contrôle. Cet élément de contrôle s'appelle « Media Gateway Contoller » ou « Call Agent ».

Dans l'architecture MGCP il existe plusieurs « Call Agent ». Le protocole MGCP suppose que ces différents « Calls Agents » doivent se synchroniser l'un avec les autres pour envoyer des messages cohérents aux passerelles sous leur contrôle.

Finalement MGCP est un protocole maître /esclave ou on s'attend à ce que les Gateways exécutent des commandes envoyées par les Call Agents.

Etant développé par la communauté des opérateurs des télécommunications, MGCP s'adresse à l'interconnexion de la signalisation SS7 et VoIP. L'architecture H.323 n'étant pas compatible avec le monde des réseaux publics de téléphonie intégrant de multiples passerelles et la signalisation SS7, il a fallu trouver un autre protocole: MGCP. On constate donc que ce protocole est plutôt adressé aux grands réseaux et aux téléopérateurs.

Cette nouvelle architecture a donc abandonné le model « Gatekeeper » et a supprimé la signalisation du Gateway en la mettant dans les « Media Gateway Controllers ». Ce contrôleur appelé aussi « Call Agent », s'occupe du contrôle de multiples « Media Gateways »

MGCP utilise plusieurs commandes pour signaler les différents événements :

- **Notification Request**
- **Notification Command**
- **Create Connection**
- **Modify Connection**
- **Delete Connection**
- **Audit Endpoint**
- **Audit Connection**
- **Restart In Progress**

Annexe C: [1] [24]

SIPv1 est un protocole qui a été défini par la RFC 2543 en mars 1999. Il est basé sur UDP ou TCP et est utilisé pour la création de sessions à participants multiples, comme les applications de vidéo/audio conférence. En juin 2002, le standard SIPv2, défini par la RFC 3261, remplace complètement SIPv1. Actuellement, SIPv1 n'est plus utilisé puisque cette version n'inclut aucun mécanisme de défense efficace.

1. Détails des en-têtes des messages réponses

General-Header	En-TÊTE GÈNÈRAL : identique à celui de la Requête (voir plus haut)
Entity header	En-TÊTE D'ENTITÈ : identique à celui de la Requête (voir plus haut)
Response Header	En-TÊTE DE RÈPONSE .

Allow	Autorise les <<méthodes>> de l'appelé.
Also	Indique au destinataire que la réponse concerne tous les correspondants qui sont listés.
Location	L'en-tête de localisation peut être utilisé avec les réponses de code 2xx ou 3xx, pour indiquer d'autres adresses possibles permettant de joindre le correspondant, ou pour fournir des précisions concernant le transport (TCP au lieu d'UDP, mlticast, etc.). Dans le cas de codes réponses 301 ou 302 (changements d'adresses), le champ <<Location >> doit contenir les nouvelles adresses.
Proxy-Authenticate	Authentification par un proxy.
Retry-after	Ce champ peut être utilisé avec les réponses de code 503 (Service Indisponible-Unavailable), 404(correspondant introuvable-Not Found), 600(Occupation-Busy), 603(Décline l'Invitation-Decline) pour indiquer quand le correspondant sera disponible. La valeur de ce champ peut être optionnel peut être rajouté. Exemple : Retry-After ; Mon, 8 Feb 1999 20 :30 :00 GMT(je suit en réunion).
Server	Logiciel serveur, syntaxe HTTP
Unsupported	Liste l'ensemble des fonctionnalités qui ne sont pas reconnues par le serveur appelé.
Warning	Peut contenir des informations additionnelles concernant les réponses. Par exemple, << multicast impossible>>
WWW-Authenticate	Indique la manière de s'identifier. Ce champ d'en-tête doit toujours être inclus dans les réponses de code 404 ; <<non autorisé>>.

Tableau C.1 : Détails des en-têtes des messages réponses

2- Détail des codes réponses :

INFORMATIONAL-INFORMATION		
100	Trying	Réponse de temporisation, envoyée par le serveur qui ne peut pas satisfaire la requête dans un délai de 200 ms. La procédure est en cours, mais le correspondant n'a pas encore été localisé. Patienter.
180	Ringing	Le correspondant a été trouvé, et son <<poste>> est en train d'être <<sonné>>.
181	Queued	L'appel est mis dans une file d'attente. La réponse revoie le nombre d'appels déjà en attente pour ce même poste.
SUCCESS-SUCCÈS		
200	OK	Le correspondant a fait part de son accord pour répondre à l'invitation.
REDIRECTION-REROUTAGE		
300	Multiple choices	La recherche a trouvé plusieurs adresses de localisation possibles. C'est au client appelant de choisir celle vers laquelle l'appel sera émis.
301	Moved permanently	Changement d'adresse définitif. L'appel doit être renouvelé à la nouvelle adresse indiquée dans ce champ.
302	Moved Temporarily	Changement d'adresse temporaire. L'appel doit être renouvelé à la nouvelle adresse indiquée dans ce champ. La validité de cette adresse peut être indiquée dans l'en-tête <<expires>> des messages SIP.
303	See Other	Essayer une autre adresse suggérée dans ce champ.
305	User Proxy	Utiliser un proxy à l'URL indiquée.
380	Alternative Service	La requête, telle qu'elle à été proposée, n'est pas acceptable, mais d'autre services décrits dans le corps du message de la réponse sont disponibles, par exemple une messagerie

		vocale, page, etc.
381	Ambiguous	L'adresse du correspondant sollicité était ambiguë. Elle ne permet pas de mener l'appel à son terme. Une liste d'adresses alternatives, offrant des similitudes avec celle demandée, pourra être proposée à l'appelant.
CLIENT ERROR-ERROR CLIENT (appelant)		
400	Bad Request	La requête n'a pu être interprétée à cause d'erreurs de syntaxe.
401	Unauthirized	L'authentification de l'appelant est exigée. La requête n'est pas autorisée.
402	Payment required	Paiement requis. Utilise pour des services de facturation.
403	Forbidden	La requête a été correctement interprétée, mais elle est refusée par le correspondant, sans autre explication.
404	Not Found	Il n'y a pas d'abonné au nom de domaine indiqué.
405	Method Not Allowed	Ce type de requête n'est pas accepté par ce serveur. Une liste de requêtes valides sera transmise.
407	Proxy Authentication Required	Semblable à 401, demande d'authentification par un proxy préalable à la communication.
408	Request Timeout	La temporisation indiquée pour la validité de la requête a été dépassée. << Le poste ne répond pas >>. La réponse peut contenir une adresse vers laquelle faire suivre l'appel (suivi sur non-réponse).
409	Conflict	Un conflit a été détecté.
410	Gone	Parti.
411	Length Required	L'entête « length » est requise.
412	Precondition Failed	La preconditions a échoué.
413	Request Message Body Too Large	Le corps du message de la requête est trop long.

414	Request-URI Too Large	L'URI est trop longue.
415	Unsuppored Media Type	Ce type de média n'est pas reconnu.
420	Bad Extension	Le serveur n'a pas reconnu l'extension de protocole spécifié.
480	Temporarily not available	Le correspondant est indisponible à l'heure où la requête lui parvient. Il proposera éventuellement une heure plus appropriée, dans le champ d'en-tête <<Retry-After>>.
481	Invalid Call-Id	L'appelant a renvoyé un message ACK ou BYE avec un Call-ID inconnu.
482	Loop Detected	Le serveur de destination a reçu une requête, dont les champs VIA contenaient sa propre adresse. Il a donc détecté une boucle.
483	Too Many Hops	Le serveur a reçu une requête dont le champ VIA contenait un nombre de noeud intermédiaires (hops) traversés dépassant le nombre fixé dans le champ d'en-tête de requête << maxforward>>
SERVER ERROR-ERROR SERVEUR(Appelé)		
500	Internal server error	Une erreur interne au serveur l'empêche de répondre à la requête.
501	Not Implemented	Le serveur ne reconnaît pas la fonctionnalité demandée.
502	Bad gateway	Le serveur, faisant fonction de proxy ou de gateway, a reçue un message d'erreur de serveurs situés en amont.
503	Service Unavailable	Le serveur ne peut pas prendre en compte la requête pour des raisons de maintenance ou de surcharge. Si l'indisponibilité est temporaire, le serveur peut indiquer une heure plus favorable dans un champ d'en-tête << Retry After >>.
504	Gateway Timeout	Le serveur, agissant en tant que gateway, a subit un dépassement de délai (Timeout) de la part d'un serveur en amont, par exemple un serveur de localisation.
505	SIP Version not supported	Cette version de protocole SIP n'est pas reconnue.
GLOBAL FAILURE-ÈCHEC GLOBAL		
600	Busy	Le correspondant a reçu la requête, mais il ne peut pas prendre l'appel car il est occupé à une autre tâche. Le champ

		d'en-tête << Retry-after >> peut indiquer une heure plus favorable pour renouveler l'appel. La réponse peut contenir une adresse vers laquelle faire suivre l'appel.
603	Decline	Le correspondant décline l'invitation. Le champ d'en-tête << Retry-after >> peut indiquer une heure plus favorable pour renouveler l'appel.
604	Does not exist anywhere	Le serveur sait avec certitude que ce correspondant n'existe plus.
606	Not Acceptable	Certains paramètres de la communication contenus dans la requête ne peuvent pas être acceptés pas le serveur : bande passante, protocole, format, multicast, etc.

Tableau C.2 : Liste des codes réponses.

Les codes réponses sont extensibles, et de nouvelles références peuvent apparaître au fil des besoins. Les applications SIP peuvent ne pas savoir interpréter tous les codes réponses ; cette ignorance ne remet pas en cause la compréhension de la réponse, et c'est le résultat x00 qui sera communiqué. En revanche, les applications doivent être capables d'identifier la classe à laquelle elles appartiennent, sur la base du premier chiffre.

Les en-têtes **SIP** sont similaires à ceux de **HTTP**, par leur vocabulaire et par leur syntaxe. L'ordre des en-têtes est sans importance ; pourtant, les proxies doivent s'abstenir de modifier cet ordre, ainsi que le contenu des champs, hormis **VIA** ou un nouveau champ (nœud à nœud). Les proxies et les locations serveurs n'émettent pas tous les mêmes types de réponses.

Propriétés	**Serveur de Reroutage (Redirect Server)**	**Serveur proxy**	**Agent Client (User Agent Server)**
Peut également être client	NON	OUI	OUI
Émet des réponses de code 1xx	OUI	OUI	OUI
Émet des réponses de code 2xx	NON	OUI	OUI

Émet des réponses de code 3xx	OUI	OUI	OUI
Émet des réponses de code 4xx	OUI	OUI	OUI
Émet des réponses de code 5xx	OUI	OUI	OUI
Émet des réponses de code 6xx	NON	OUI	OUI
Insère des en-tête VIA	NON	OUI	NON
Accepte des ACK	NON	OUI	OUI

Tableau C.3 : Émetteurs de messages réponses

2. SDP (Session Description Protocol) : [28]

Les requêtes INVITE peuvent contenir, dans le corps du message et après la pile d'en-tête, la description de la conférence à laquelle est invité le correspondant.

Proposé par le groupe de travail MMUSIC (Multiparty Multimedia Session Control) de l'IETF, SDP offre un format de description de conférences multimédias, qui peut être employé dans les requêtes SIP.

Les principaux champs sont décrits ci-après :

v	Protocol version	Version du protocole
o	Owner/creator	Créateur ou propriétaire de la présentation

s	Session name	Nom de la session
i	Session information	Information sur la session
u	URI of description	URI (URL absolu) de la description
e	Email address	Adresse e-mail
p	Phone number	Numéro de téléphone
c	Connection information	Information sur la connexion (adresse sue Internet ou PSTN)
b	Bandwidth available/needed	Bande passante disponible/nécessaire
z	Time zone adjustement	Ajustement de la zone horaire
k	Encryption key	Clé de cryptage
a	Session attributes	Attributs de session, description de médias
m	Media name and transport address	Nom des médias et adresses de transport
t	Media title	Titre de media
c	Connection information	Information sur la connexion
b	Bandwidth (kbps)	Bande passante
k	Encryption key	Clé de cryptage
a	Media attributes	Attributs de médias

Tableau C.4 : Codage des noms de champs

Mais SDP reste encore limité, et ne permet par exemple pas d'exprimer des alternatives, des préférences, ou d'affiner la description de présentations.

CHAPITRE I :

L'objectif de ce chapitre est de présenter les différentes technologies utilisant le réseau pour transmettre des objets multimédia, voix, image, et texte et qui utilisent des mécanismes de signalisation sera présentée dans le premier paragraphe. La signalisation, avec ses deux protocoles, les plus connus, H323 et SIP, sera détaillée par la suite. La présentation théorique de ces technologies nous permettra de situer l'application que nous avons développée et définir son domaine application.

1- La Téléphonie sur IP (ToIP)

Au début des années 80, des expérimentations ont été faites pour la transmission de la voix par Internet VoIP. Cette dernière consiste à transformer la voix en paquets de données (échantillonnage numérisation compression) et transmettre les conversations via le réseau IP (Internet Protocol). [8]

Ces paquets de données transportant la voix ont des contraintes différentes de ceux transportant du texte. En effet, les techniques de compression utilisées d'une part, la vitesse et la qualité des lignes de transmission, le degré de congestion du réseau par lequel transitent ces paquets et les mécanismes de routage d'autres parts, influent sur la qualité de la voix à la réception. Les pertes de paquets, avec une trop grande latence dans la transmission, ainsi qu'une variation des délais (gigue) peuvent rendre le signal reçu inaudible (Figure I.1)

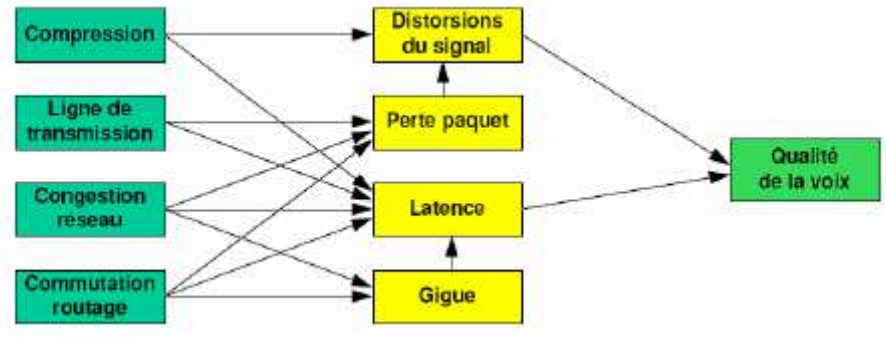


Figure I.1 : Les contraintes de la VoIP [8]

1.1-Différents types de téléphonie sur IP :

On peut, en première approche, distinguer trois types de téléphonie sur IP selon le terminal utilisé par chacun des deux correspondants. [5] [13]

1.1.1- Téléphonie entre micro-ordinateurs ("PC to PC") :

Les deux correspondants utilisent leurs micro-ordinateurs, avec les haut-parleurs généralement livrés en série et en y adjoignant des microphones (**voir Figure I.2**). On l'a vu, ce mode de fonctionnement nécessite actuellement que les correspondants se fixent un rendez-vous préalable sur Internet ou soient connectés en permanence et, bien sûr, qu'ils utilisent des logiciels de voix sur IP compatibles. De plus, les adresses IP changeant à chaque connexion, les correspondants doivent se mettre d'accord sur la consultation d'un annuaire ("dynamique", car mis à jour à chaque connexion par chaque correspondant potentiel qui doit s'y enregistrer) pour permettre à l'appelant de connaître l'adresse de l'appelé (cette procédure est grandement facilitée pour des utilisateurs connectés en permanence à Internet).

RTC : Réseau téléphonique commuté
IAP : Fournisseur d'accès à Internet

- En grand public -

Nota : le site 2 et le site 1 peuvent être confondus

- En entreprise –

Figure I.2 : Communication de micro-ordinateur à micro-ordinateur (PC to PC) [13]

1.1.2- Téléphonie entre micro-ordinateur et poste téléphonique ("PC to phone", voire "phone to PC") :

L'un des correspondants est sur son micro-ordinateur ; s'il désire appeler un correspondant sur le poste téléphonique de celui-ci, il doit se connecter sur un service spécial sur Internet, offert par un fournisseur de service (un "ISP") ou par son fournisseur d'accès à

Internet (son "IAP"), mais qui doit mettre en œuvre une "passerelle" ("gateway") avec le réseau téléphonique. C'est cette passerelle qui se chargera de l'appel du correspondant et de l'ensemble de la "signalisation" relative à la communication téléphonique, du côté du correspondant demandé

(Voir **Figure I.3**).

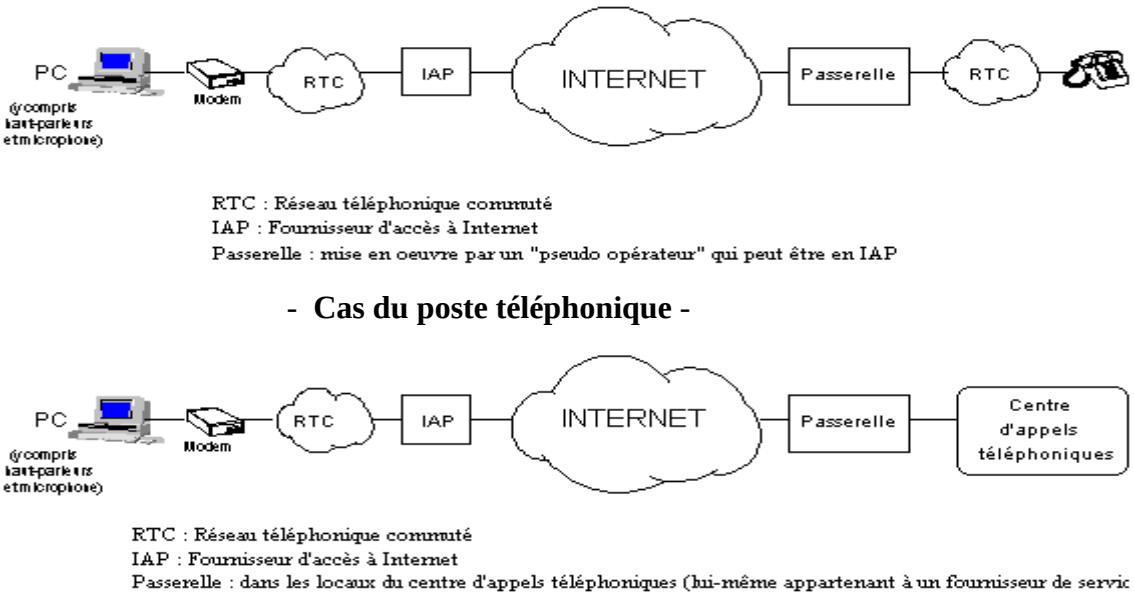


Figure I.3: Communication de micro-ordinateur à poste téléphonique (PC to phone) [13]

Si le correspondant qui appelle est sur son poste téléphonique et qu'il veut joindre un correspondant sur internet, il devra appeler le numéro spécial d'une passerelle qui gérera l'établissement de la communication avec le réseau internet et le correspondant sur ce réseau pourvu, là aussi, qu'il soit au rendez-vous (à moins qu'il ne soit connecté en permanence).

Un cas particulier important de la communication d'un micro-ordinateur vers un poste téléphonique est celui où le correspondant appelé est un centre d'appels téléphoniques intégré à une application internet (voir **Figure I.3** ci-dessus).

1.1.3- Téléphonie entre postes téléphoniques ("phone to phone") :

Un scénario plus évolué, proche de la téléphonie classique est possible pour faire dialoguer deux postes téléphoniques via un réseau IP. Deux méthodes sont possibles :

- En utilisant des téléphones ordinaires (**Figure I.4**)

➢ En utilisant des téléphones dédiés IP (**Figure I.5**)

RTC : Réseau téléphonique commuté

Passerelle : mise en oeuvre par un "pseudo opérateur" (par exemple un fournisseur d'accès à Internet) dans ses locaux

Figure I.4 Téléphones ordinaire.

PBX : Autocommutateur téléphonique interne au site

Passerelle : dans les locaux du site de l'entreprise

Figure I.5 Téléphones dédiés IP. [13]

L'avantage de téléphonie sur IP par rapport à la téléphonie classique est incontestable le coût qui est presque nul même sur les longues distances. Quand la qualité et la sécurité des réseaux IP seront garanties il est fort possible que la téléphonie classique disparaître.

1.1.4- Conclusion

La téléphonie et l'internet sont deux technologies qui ont contribué à l'avènement de la technologie **ToIP.**

La Téléphonie IP est le moyen de communication actuel et d'avenir par excellence.

La multiplication des offres d'accès à l'internet haut débit par satellite ou via l'ADSL est en train d'opérer une véritable révolution dans le monde de la Téléphonie IP. Mais il ne suffit pas d'avoir de très hauts débits pour résoudre les problèmes de transport des applications multimédia. Il faut prendre en considération les protocoles de signalisation [1]. Cette dernière concerne l'ensemble des informations échangées par terminaux ainsi que les entités de gestion et les passerelles participant à toutes les phases d'établissement, de contrôle, de rupture et de fermeture de connexion.

2 Signalisation

2.1 Définition :

La signalisation concerne l'échange d'informations entre les nœuds d'un réseau. Ces informations servent à l'établissement et au contrôle des connexions à travers le réseau (établir et terminer des appels). La signalisation permet également le transfert d'informations concernant la gestion du réseau et de ses ressources. [1]

Les protocoles de signalisation font partie de la couche 7 (Application) du modèle OSI. Le respect de procédures de signalisation communes est l'une des conditions nécessaires à l'interopérabilité des produits de différents constructeurs. Deux catégories d'acteurs ont activement contribué à l'établissement des procédures de signalisation :

- Les organismes de standardisation, ITU, IETF ou ETSI ;
- Les consortiums d'industriels et d'opérateurs de télécommunications.

2.2 - Historique et généralité sur les protocoles de signalisation :

La signalisation est une des plus importantes fonctions dans l'infrastructure des télécommunications puisqu'elle permet aux composants du réseau de communiquer entre eux pour établir et terminer des appels.

Voice Over IP, dont le but est d'établir des canaux de communication vocaux entre utilisateurs, requiert l'utilisation de protocoles de signalisation pour établir et terminer les appels.

Au début des années 90, il n'existait pas, à proprement parler, de protocoles standardisés permettant la signalisation entre logiciels intégrant Voice Over IP. Dés lors, il était impossible de communiquer entre participants si ceux-ci ne possédaient pas rigoureusement le même logiciel. En effet, chaque développeur construisait son logiciel d'après ses propres protocoles de signalisation. Pour pallier à ce problème, l'ITU(Union Internationale des Télécommunications) entreprit l'idée de standardiser le signalisation sur Voice standard IP. C'est en 1996 que le groupe de travail numéro 16 de l'ITU proposa le standard H.323 version 1. Il s'agissait du premier standard de signalisation concernant la transmission d'informations multimédia en temps réel sur des réseaux ne possédant pas de qualité de service.

Ensuite vint l'engouement pour l'interconnexion du monde IP et des réseaux à commutation de circuits (SCN). Il fallait développer des outils matériels qui puissent effectuer des conversions de médias et dont la tâche la plus importante consistait à comprendre les messages de signalisation envoyés de part et d'autre des réseaux hétérogènes. C'est la tâche des « gateways ». Ces nouveaux développements rendirent le standard H.323 limité car celui-ci n'intégrait pas la signalisation SS7 (voire les détailles sur annexe A), propre aux réseaux SCN. [15]

A nouveau, il fallut attendre 1998 pour que le standard MGCP (Media Gateway Control Protocol) ou « Megaco » soit établi par l'ITU groupe 16(voire les détailles sur annexe B). Ce standard régit l'utilisation du Media Gateway et du Media Gateway Contrôler dont les buts sont de, respectivement, convertir les médias et contrôler les appels.

D ans d'autres perspectives d'évolution est également né SIP dans la fin des années 90. Plus simple qu'H.323, il attire aujourd'hui les regards de développeurs.

Dans les deux sessions suivantes en va détailler les deux protocoles qui sont en concurrences, et qui sont les plus connues dans le domaine de la signalisation.

2.3- Le standard de signalisation H.323

2.3.1- Introduction

Le standard de signalisation H.323, développé par l'ITU-T, est défini comme étant le standard spécifiant les éléments, les protocoles et les procédures d'établissement et de gestion d'appel pour réaliser des communications audio, vidéo et autres données en temps réel sur les réseau par communication de paquets. [9]

2.3.2- Architecture H.323

La recommandation H.323 définit un modèle architectural pour assurer le transport de la voix sur un réseau en mode paquets de type IP, c'est-à-dire sans qualité de service.

L'architecture matérielle H.323 comprend plusieurs éléments de réseaux :

Les terminaux, les portiers (*gatekeepers*), les passerelles (*Gateways* H.323 vers H.320/H.324/ téléphones classiques) et les contrôleurs multipoints (MCUs – MC, Multipoint Contrôler, MP – Multipoint Processor).

Les terminaux de type H.323 peuvent être intégrés dans des ordinateurs personnels ou implantés dans des équipements autonomes tels que des vidéophones. La prise en charge de la parole est obligatoire, tandis que celle des données et vidéo est facultative.

Ci-dessous les piles des sous protocoles et l'architecture de H.323 :

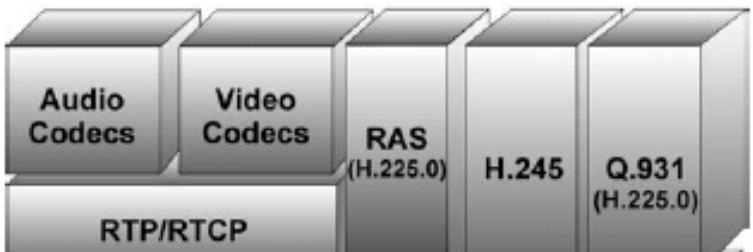


Figure I.6 : Pile protocolaire H 323 [9]

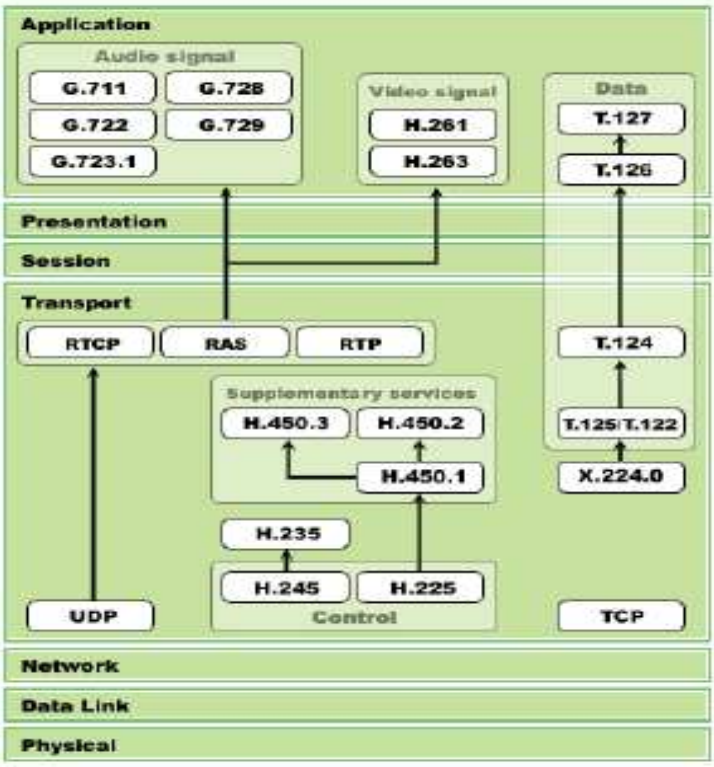


Figure I.7 : Mise en évidence de la pile protocolaire H.323 par rapport au modèle OSI [9]

2.3.3- Etude des protocoles utilisés en H 323

2.3.3.1- Le Terminal H.323

Un terminal peut être un PC, un PDA ou tout autre périphérique autonome supportant H.323 et les communications multimédia bidirectionnelles en temps réel. Il supporte des communications audio et optionnellement des applications vidéo et/ou de données. Il peut être utilisé dans des conférences multipoints (par utilisation de *MCUs*). Son rôle est de communiquer avec d'autres terminaux multimédias (H.324 du RTC, ou H.310 (sans fil) et H.321 du B-RNIS, ou encore avec les terminaux H.320 sur RNIS ou H.322 sur LAN avec qualité de service).

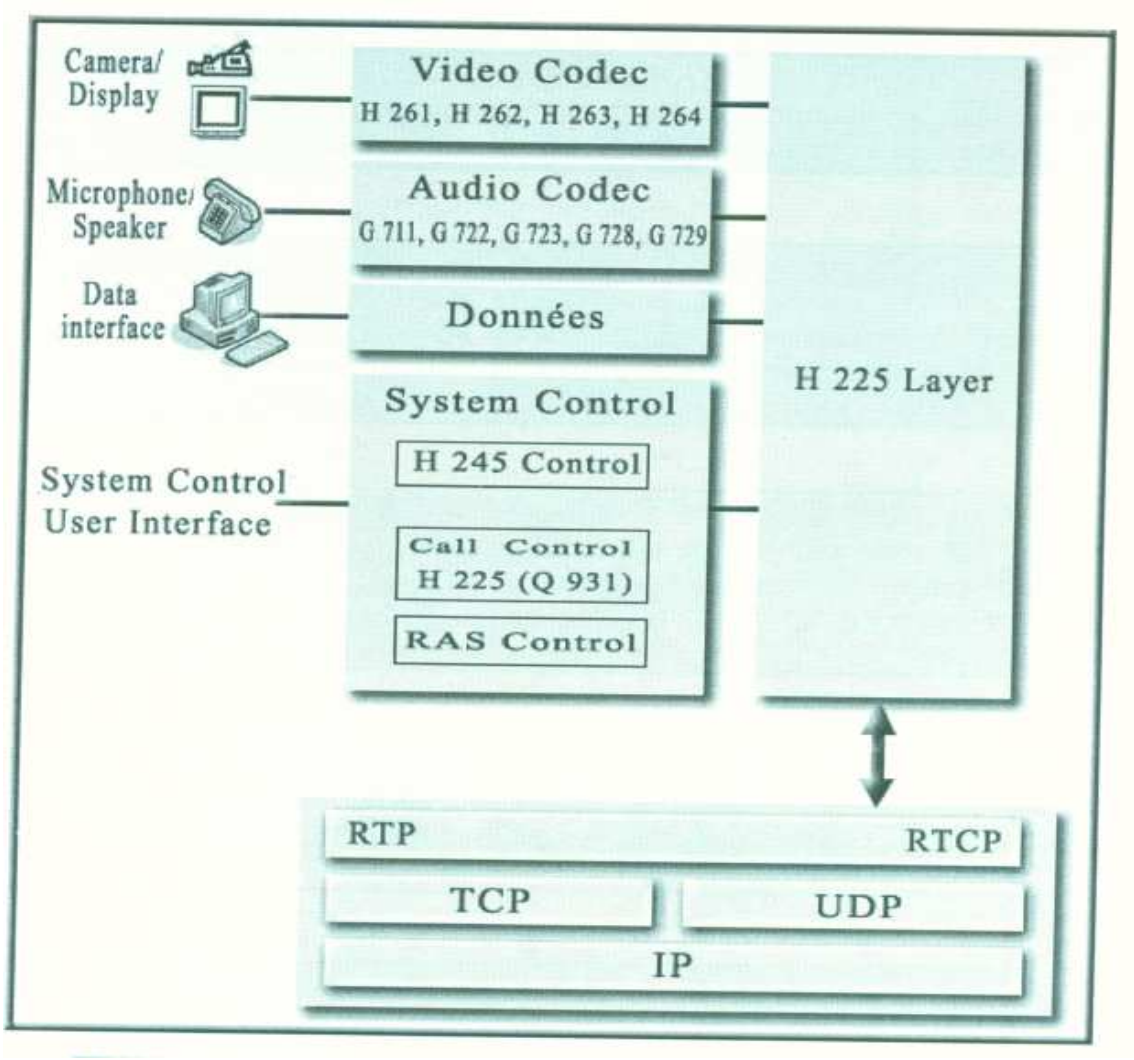


Figure I.8 : Décomposition fonctionnelle d'un terminal H.323

2.3.3.2- Les différents protocoles H.323

Signalisation d'enregistrement/contrôle d'admission RAS

H.225-RAS : Les messages RAS (Registration, Admission et Status) définissent la manière dont un terminal H.323 s'enregistre auprès d'un Gatekeeper, il permet également de contrôle d'admission.

Signalisation appel

H.225-Q.931 : Des messages H.225 sont intégrés dans les messages Q.931. Ils définissent les procédures de signalisation pour activer ou fermer les appels : initiation, contrôle d'admission, mise en forme des paquets, synchronisation, fermeture de la communication. H.225-Q.931 formate et reçoit les messages de contrôle des flux vidéo, audio qui utilisent RTP/RTCP.

Contrôle d'appel

H.245 : Lorsque la communication a été établie par H.225 et Q.931, H.245 active les canaux logiques (vidéo, audio ou données). Les messages transmis permettent de déterminer les formats des médias (codecs, débit, taille), d'ouvrir les canaux logiques à partie des adresses IP et des numéros de port UDP. H.245 permet de contrôler les médias.

Codecs vidéo

H.261 et H.263 : Le Codec vidéo (H.261, H.263 optionnel) encode la source vidéo à transmettre (caméra sur port USB ou carte d'acquisition vidéo), et décode les flux vidéo reçus pour un affichage sur écran.

Codecs audio

G 711, G 722, G 723, G 728 et G 729 : Les Codecs audio. Ce sont des normes d'encodage audio, la différence entre eux est le débit qui en découle (ex : G 711 donne un débit de 64 Kbits/s, G 728 donne un débit de 16 Kbits/s).

Données

T 120 : définit les échanges de données (transfert de fichiers, applications partagées, échange de texte, tableau blanc).

2.3.3.3- La zone et les éléments du H.323

Les composants d'un système H.323 sont :

- les *terminaux*, point de départ et d'arrivée d'une communication, assurent l'audio et optionnellement la vidéo et les données dans des conférences en point à- point ou en multipoint ;
- les *gateways* assurent l'interaction avec le réseau RTC
- les *gatekeepers* qui fournissent les services de contrôle d'admission et de translation d'adresses
- les *Multipoint Controllers* (*MC*), les *Multipoint Processors* (*MP*) et les *Multipoint Control Units* (*MCU*) fournissent un support pour les conférences multipoints.

Les *gatekeepers*, *gateways* et *MCU* sont logiquement séparés mais peuvent être implémentés en un seul dispositif physique.

Une collection de *terminaux*, *gateways* et *MCU* gérés par un seul *gatekeeper* est appelée « *zone H.323* ». Une zone contient au moins un terminal, peut inclure des *gateways* et des *MCU* et n'a qu'un seul *gatekeeper*. H.323 se situant au niveau applicatif de l'OSI, une zone peut être distribuée sur plusieurs réseaux IP connectés par des routeurs.

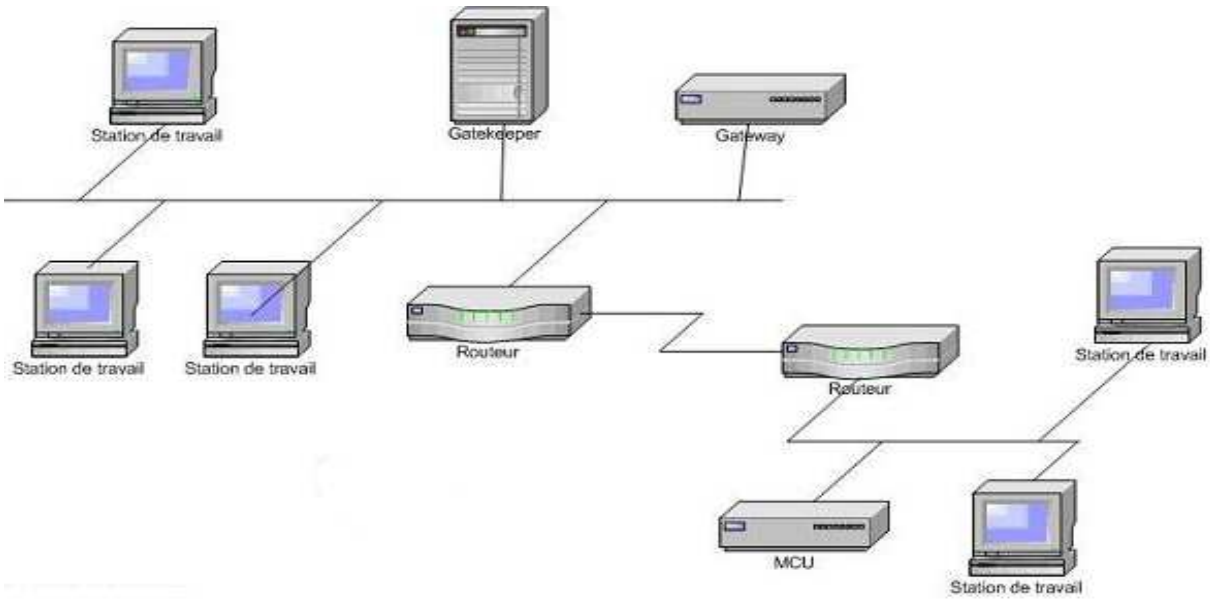


Figure I.9 : Zone H 323. [9]

La signalisation entre chaque terminal et le *gatekeeper* est transmise au-dessus d'une connexion TCP, selon les spécifications de *RAS*. Un *gatekeeper* peut participer à une

variété de modèles de signalisation dictés par le *gateway*. Des modèles de signalisation déterminent quels messages de signalisation passent à travers le *gatekeeper* et lesquels peuvent directement transiter entre les entités, comme le terminal et le *gateway*.

La figure I.9 illustre un modèle de *signalisation directe* (*direct signaling*) où l'échange des messages de signalisation n'implique pas le *gatekeeper*, alors que dans un modèle de *signalisation routée* (*gatekeeperrouted signaling*) (La figure I.10) seuls les flots de média transitent directement entre les terminaux.

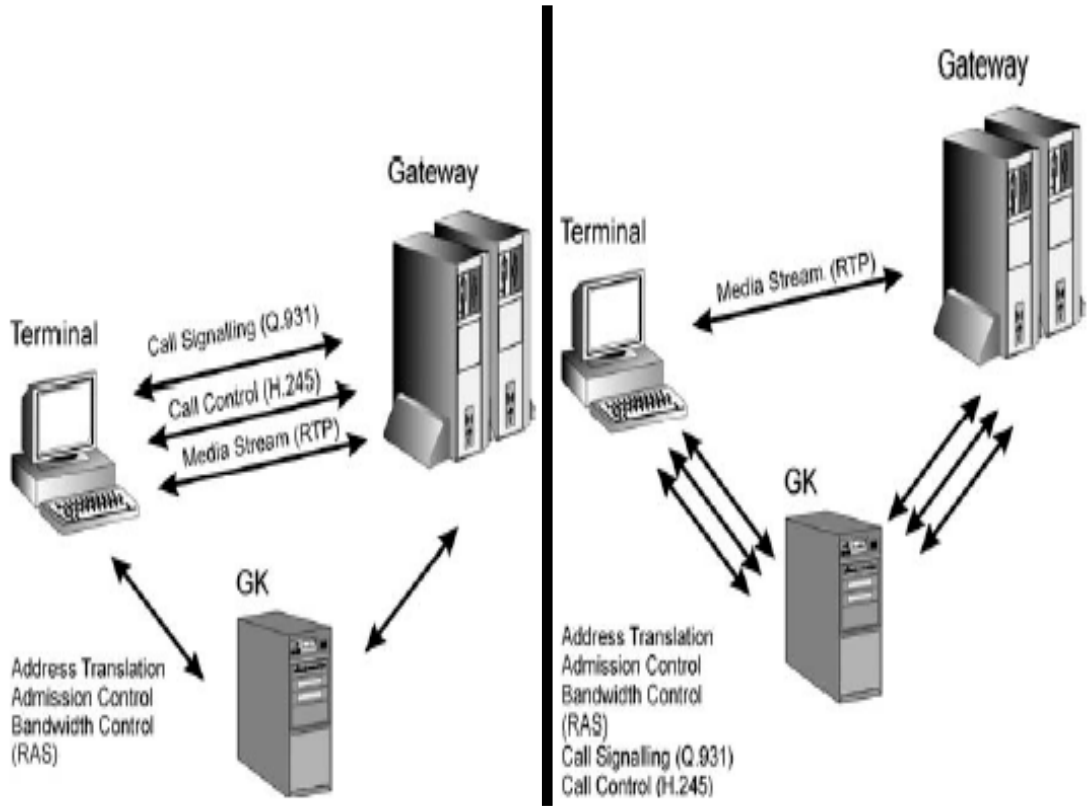
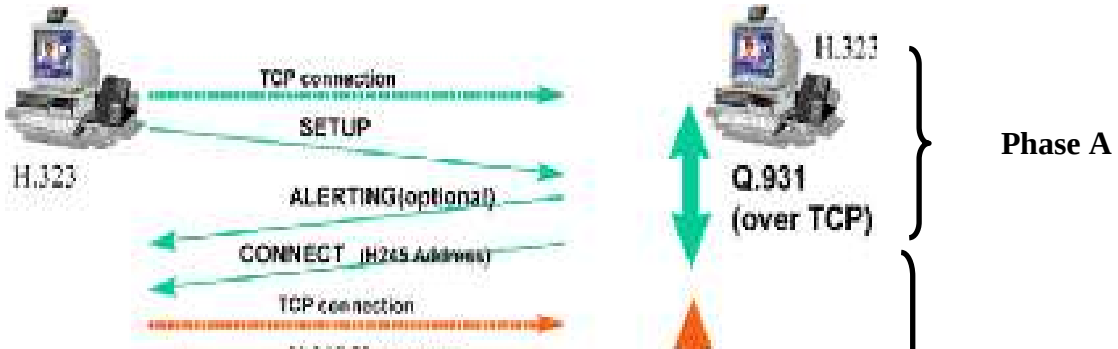


Figure 1.10 : Signalisation directe. **Figure 1.11 : Signalisation routée**.

2.3.3.4- Etablissement d'une connexion avec H 323



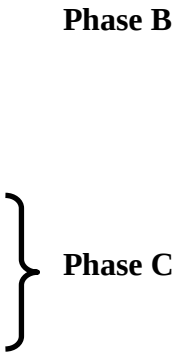


Figure I.12 : Etablissement d'une connexion avec H.323 [9]

Compte tenu de la complexité des échanges, l'établissement d'une communication H.323 peut prendre plusieurs secondes. Ce qui correspond à plusieurs phases. Lesquelles sont :

Phase A : phase d'établissement d'appel, le récepteur est averti qu'un appel va débuter. Cette phase utilise le protocole TCP.

Phase B : phase d'établissement des numéros de canaux logiques utilisables et échange des caractéristiques afin de déterminer les codecs qui pourront être utilisés. Dans cette phase, il y a une multitude d'aller-retour pour établir la connexion H.245.

Phase C : phase de communication (le transport se fait avec le protocole UDP encapsulé dans du RTP).

2.4- Le protocole de signalisation SIP (Session Initialisation Protocol)

Le protocole SIP (Session Initialisation Protocol) a été initié par le groupe MMUSIC (Multiparty Multimedia Session Control) [RFC 2543] et désormais repris et maintenu par le Groupe SIP de l'IETF [RFC 3261]. SIP est un protocole de signalisation appartenant à la couche application du modèle OSI. Il a été conçu pour l'ouverture, le maintient et la terminaison de sessions de communications interactives entre des utilisateurs. De telles sessions permettent de réaliser de l'audio, de l'enseignement à distance et de la voix (téléphonie) sur IP essentiellement. Pour l'ouverture d'une session, un utilisateur émet une

invitation transportant un descripteur de session permettant aux utilisateurs souhaitant communiquer de négocier sur les algorithmes et codecs à utiliser. SIP permet aussi de relier des stations mobiles en transmettant ou redirigeant les requêtes vers la position courante de la station appelée. Enfin, SIP est indépendant du médium utilisé et aussi du protocole de transport des couches basses. [16]

2.4.1- Architecture protocolaire

SIP est un protocole indépendant des couches de transport, il appartient aux couches applications du modèle OSI. Le SIP gère la signalisation et l'établissement des sessions interactives de communication multimédias et multipartites. Il est aussi basé sur le concept Client/Serveur pour le contrôle d'appels et des services multimédias. Conçu selon un modèle de type IP, il est hautement extensible et assez simple en conception architecturale, de sorte qu'il peut servir de base à la création d'applications et de services. Il est basé sur le protocole HTTP et peut utiliser UDP ou TCP. [14]

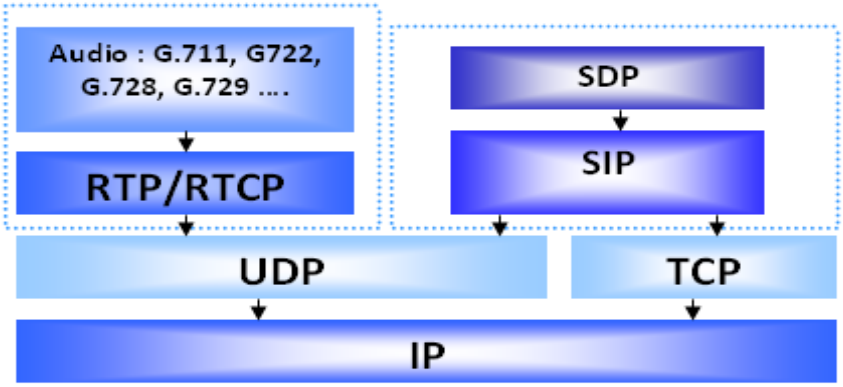


Figure I.13 : pile SIP [14]

2.4.2- Architecture d'une plate forme SIP

SIP est un protocole simple et flexible orienté messages. Les principaux composants d'un système basé sur SIP sont :

- **Terminal SIP (User Agent Client ou UAC) :** Peut être aussi bien un SoftPhone (logiciel) qu'un HardPhone (téléphone IP). Les UAC sont capables d'émettre et de recevoir de la signalisation SIP.
- **Proxy Server :** encore appelé serveur mandataire auquel est relié un terminal fixe ou mobile, agit comme serveur envers le client et comme client envers les autres UAS.
- **Redirect Server :** Ce serveur permet de rediriger les appels vers la position courante d'un utilisateur. Il réalise simplement une association d'adresses vers une ou plusieurs nouvelles adresses.

- **Location Server :** Il fournit la position courante des utilisateurs dont la communication traverse les serveurs mandataire et de redirection auxquels il est rattaché.
- **Registrar Server :** Ce serveur reçoit et accepte les inscriptions des utilisateurs (adresse IP, port, login).

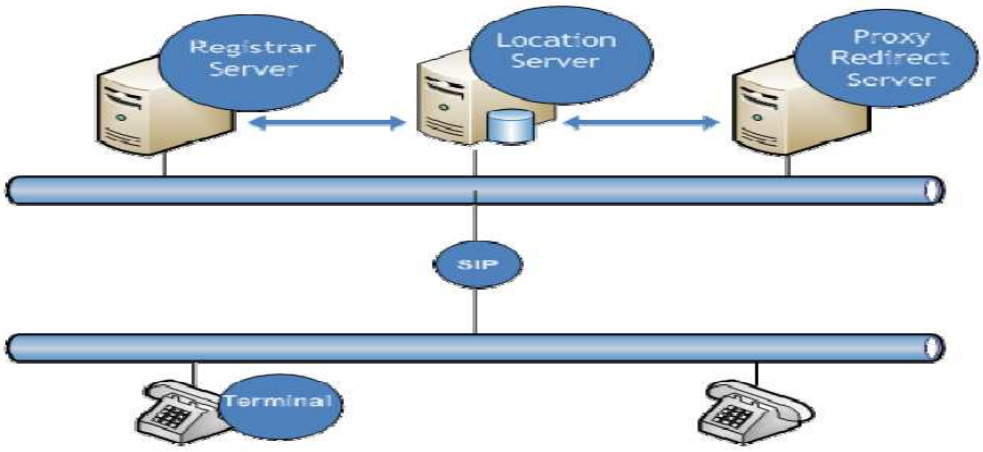


Figure I.14 : architecture SIP [14]

2.4.3- L'adressage SIP : [15] [16]

SIP exploite des formats d'adresses de type e-mail et ses extensions prévus dans les architectures DNS MX et SMTP EXPN. Les adresses SIP (des URL) peuvent être intégrées dans des pages HTML. SIP est indépendant du format d'adressage, et accepte des adresses de terminaux H.323 ou au format E.164 du RNIS. SIP peut être exploité pour le transport temps réel (exemple fax).

2.4.3.1 - Les adresses SIP :

Elles sont de type : user_ID@host_name.

Elles peuvent prendre des formes variées, telles que :

- user@domain ;
- user@host ;
- user@IP_adress ;

- phone-number@gateway, si le numéro appelé est sur le RTC.

La partie user_ID est un nom dépendant du système d'exploitation. Le host peut être un nom de domaine ou une adresse IP numérique

2.4.3.2 - Résolution d'adresse :

Le poste client SIP résout l'adresse du destinataire en suivant une démarche systématique.

- Si la partie « host » contient une adresse IP numérique, un serveur SIP y sera contacté.
- Sinon, le client exploitera les ressources du DNS en s'appuyant sur les protocoles UDP ou TCP, selon la disponibilité, pour résoudre l'adresse.
- En cas d'échec de la méthode précédente, l'appelant contactera le serveur SMTP du « host » ; Grâce à une commande SMTP EXPN, il tentera d'obtenir des adresses alternatives. Les nouvelles adresses ainsi obtenues seront résolues par le DNS, et un serveur SIP sera recherché.

2.4.3.3 - Les URIs SIP

Exemples d'URIs SIP :

Sip :m.barce@home.com
Sip :m.barce :secret@home.com ;transport=tcp
Sip : m.barce@home.com?subject=project
Sip :+65-4558989@gateway.com
Sip :jean@[110.51.82.3]
Sip : jean@110.51.82.3

La structure des URIs SIP est conforme à la recommandation RFC1630.

2.4.4- Les messages SIP :

SIP est un protocole de type client serveur. A cet effet, les échanges entre un terminal appelant et un terminal appelé se font par l'intermédiaire de requêtes et réponse SIP:

- Les requêtes (Request) émises par un client ;
- Les réponses (Response) envoyé par le poste sollicité, ou par une entité le représentant (proxy).

Tous les messages adoptent le format générique décrit dans le RFC822. Ces messages disposent de piles d'en-têtes très riches, mais ils peuvent également être échangés sous une forme simple, de quelques lignes, qui suffisent à établir un appel. Les types de requêtes, les codes de classes de réponses et les en-têtes de messages sont évolutifs. De nouveaux types, codes ou champs d'en-têtes, peuvent être facilement rajoutés pour satisfaire les besoins.

2.4.4.1 - Les requêtes :

Il y a plusieurs types de requêtes (par ailleurs appelées « méthodes »), mais trois suffisent à établir un appel : INVITE, ACK et BYE. Deux autres requêtes, OPTIONS et REGISTER permettent de réaliser toutes les applications de téléphonie. Les autres requêtes offrent des services complémentaires. De nouvelles requêtes SIP sont régulièrement ajoutées, et permettent d'enrichir le modèle.

Les types de requêtes :

Requête	Description
INVITE	La requête INVITE permet à un correspondant à participer à une conférence.
ACK	Les messages ACK est envoyé par l'appelant qui accuse réception de la réponse finale de l'appelé.
OPTIONS	Demande de précision concernant les options et les capacités d'une entité SIP.
BYE	L'appelant indique au serveur correspondant qu'il souhaite mettre un terme à une tentative d'appel, ou à une communication en cours.

CANCEL	Cette requête sert à interrompre une procédure ou une recherche de correspondant en cours. Mais elle ne peut être exploitée pour mettre fin à un appel déjà établi. Un proxy, qui a effectué une recherche parallèle de localisation, peut avoir reçu des réponses de code 200(OK) ou 600 (Echec global) de certains serveurs. Il pourra alors envoyer de requêtes CANCEL aux destinations qui n'auraient pas encore répondu.
REGISTER	Un usager emploiera cette requête pour enregistrer son adresse auprès d'un proxy, ou d'un serveur de redirection.

Tableau I.1 : Requêtes SIP

- **Structure des en-têtes des messages « Requêtes »**

Les requêtes décrites ci-dessus, et les réponses présentées plus loin, disposent d'une pile d'en-têtes conforme aux spécifications du RFC822, valable pour http 1.1.

La structure générale des en-têtes de requêtes est la suivante :

Début de ligne
En-tête de messages
CRLF : séparateur
Corps de message

Tableau I.2 : En-tête de requêtes.

Avec :

- Début de ligne : ligne de requête ou ligne d'état.
- En-tête : en-tête général ou en-tête de requête ou en-tête de réponse ou en-tête d'entité.
- CRLF : balise pour indiquer la fin du champ d'en-tête et début du corps du message (qui nécessite un en-tête d'entité).

Exemple :

INVITE sip : Ramzi@domain.com
Via : **From :** **To :** **Call-ID :**

GENERAL-HEADER	**EN-TÊTE GENERAL**

Cseq :
Subject :
Content-Type : **Content-Length :**
Ligne Vide
Donnée SDP

Figure I.15 : Allure d'une requête SIP.

Call-ID	Numéro d'identification d'une invitation ou d'un appel. Un « appel » est constitué de l'ensemble des participants à une conférence, invités par une même source. Un appel SIP est identifié par un « Call-ID » unique. C'est l'action d'inviter à une session qui définit l'appel. Si, par exemple un usager est invité par plusieurs participants différents à une même session multipoint, chacune de ces invitations constituera un appel. Le Call-ID n'est utilisé que par SIP. Il peut donc prendre le format le plus simple, du type local-id@host.
Cseq	*Command Sequence* : Numéro d'ordre de sortie des messages. Plusieurs messages peuvent avoir le même Call-ID, mais le numéro de séquence s'incrémentera d'une unité pour chaque nouveau message. Les messages de réponse aux requêtes utiliseront « CSeq » comme référence au message d'origine.
From	SIP URI indiquant l'initiateur de l'invitation.
Via	Indique la route empruntée par la requête jusqu'à ce nœud. Permet de prévenir les boucles, et garantie que la réponse prendra la même route que la requête. Le client appelant est le premier à insérer un champ VIA contenant son adresse. Par la suite, tous les proxies qui relaieront la requête devront ajouter leur champ VIA en tête de la liste.
ENTITY HEADER	**EN-TÊTE D'ENTITE (terme http 1.1)**
Content-Length	Taille en octets du corps de message.
Content-Type	Type de média utilisé pour composer le corps du message.
REQUEST-HEADER	**EN-TÊTE DE REQUÊTE**
To	SIP URI du correspondant appelé. Exemple : SIProtocoles : support@viop.org.

Tableau I.3: Détails de quelques champs d'en-tête.

2.4.4.2 - Les réponses :

➢ Structure des en-têtes des messages << Réponse >> :

Les réponses ont la même structure d'en-tête que les requêtes décrites précédemment.

Ligne d'état
En-tête général ou réponse ou d'entité
CRLF
Corps de message

Tableau I.4 : En-tête de réponses

La ligne d'état, est composée de trois parties ; c'est le caractère SP (single space) qui sert de séparateur.

- **SIP-version** : la version de SIP utilisée ;
- **Status-Code** : le code réponse est un code numérique composé de trois chiffres, qui peut être interpréter automatiquement par une machine, et qui fournit le résultat de la requête (voir ci-après la liste des codes réponses) ;
- **Reason-Phrase** : la raison écrite explique en quelques mots ce résultat à l'intention d'un opérateur humain.
- **Exemple :**

SIP/20. 302 Moved temporarity
From : **To :** **Call-ID :** **Localization :** **Exprires :** **Cseq :**
Subject :

Content-Type : **Content-Length :**
Ligne Vide
Donnée de réponse (SDP)

Figure I.16 : Allure d'une réponse SIP.

Les classes de code réponse :

A ces requêtes sont associées des réponses qui sont dans le même format que celles du protocole HTTP. Voici les plus importantes d'entre elles :

- **1XX :** messages d'informations (100 – essai, 180 – sonnerie, 183 – en cours)
- **2XX :** succès de la requête (200 –OK)
- **3XX :** Redirection de l'appel, la demande doit être dirigée ailleurs
- **4XX :** Erreur du client (La requête contient une syntaxe erronée)
- **5XX :** Erreur du serveur (le serveur n'a pas réussi à traiter une requête correcte)
- **6XX :** Echec général (606 – requête non acceptable par aucun serveur)

Exemple :

Exemple Requête SIP	**Exemple Réponse SIP**
INVITE sip:ramzi@tlemcen-univ.dz SIP/2.0	**SIP**/2.0 200 **OK**
Via: SIP/2.0/UDP 100.101.102.103 / 102.157.155.198	**Via :** SIP/2.0/UDP 100.101.102.103 102.157.155.198
From : sip:Ali@tlemcen-univ.dz	**From :** sip:Ali@tlemcen-univ.dz
To : sip:ramzi@tlemcen-univ.dz	**To :** sip:ramzi@tlemcen-univ.dz
Call-ID : 12554469@100.101.102.103	**Call-ID :** 12554469@102.157.155.198
CSeq : 1 **INVITE**	**CSeq :** 1 **INVITE**
Subject : où êtes-vous?	**Subject :** me voila
Content-Type : application/sdp	**Content-Type :** application/sdp
Content-Length : 102	**Content-Length :** 150

Tableau I.5 : Exemple de requête et réponse SIP.

2.4.5- Fonctionnement

SIP intervient aux différentes phases de l'appel :

- Localisation du terminal de l'interlocuteur.
- Analyse du profil et des ressources du destinataire.
- Négociation du type de média (voix, audio, vidéo…) et des paramètres de communication.
- Disponibilité du correspondant, détermine si le poste appelé souhaite communiquer, et autorise l'appelant à le contacter.
- Etablissement et suivi de l'appel, avertit les parties appelant et appelé de la demande d'ouverture de session, gestion du transfert et de la fermeture des appels.
- Gestion de fonctions évoluées : retour d'erreurs, …

Le schéma suivant illustre le scénario d'une communication SIP.

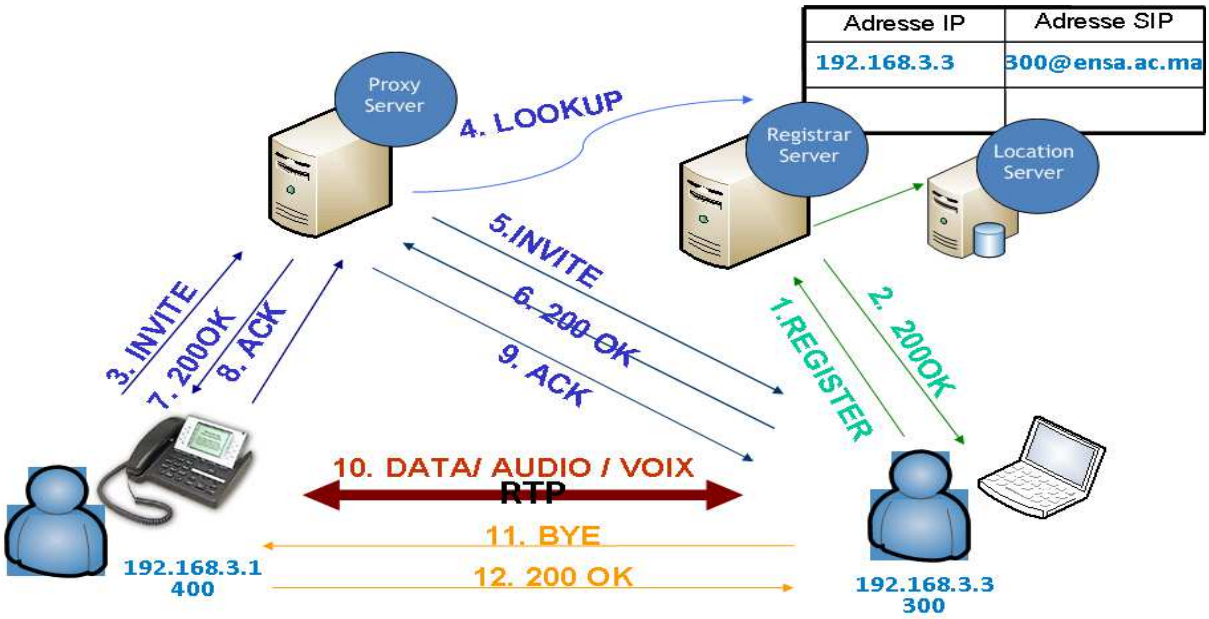


Figure I.17 : Exemple d'une communication SIP [14]

2.5 Comparaison H.323 et SIP [12]

SIP et H.323 sont tous deux conçus pour répondre aux besoins de contrôle de session et de fonctions de signalisation dans une architecture de contrôle d'appel distribuée. Bien que SIP et H 323 soient également prévus pour permettre à des stations sans intelligence embarquée de communication entre elles.

Néanmoins, puisqu'il a essentiellement été pensé dans ce but, le protocole SIP reste beaucoup plus simple d'implémentation que H.323. Un premier élément de cette comparaison peut être le nombre de pages de définition de chacun de ces protocoles.

Bien que les messages SIP ne soient pas directement compatibles avec H.323, les deux protocoles peuvent coexister dans le même réseau de téléphonie par paquets si une station qui gère l'interopérabilité entre les deux protocoles est disponible.

Les principales différences entre les deux protocoles se trouvent regrouper dans le tableau suivant :

Fonction	**H 323**	**SIP**
Normalisation	UIT H 323 V4	IETF RFC3261
Transport de la signalisation	TCP	TCP/UDP
Transport de flux multimédia	UDP	TCP/UDP
Etablissement de canaux logiques	Oui, un par sens	Non
Signalisation multicast	Non	Oui

Codage des primitives	Binaire	Texte
Evolutivité	Faible, beaucoup d'extensions propriétaires	Protocole ouvert
Gestion de conférences	Centralisée (MCU)	Distribuée
Transport audio/vidéo	RTP ou autre	RTP ou autre

Tableau I.6 : Comparaison entre H 323 et SIP [12]

Notre choix s'est porté sur le protocole SIP, du fait de sa simplicité et de son évolutivité.

3- Conclusion :

Nous avons vu dans ce chapitre la **ToIP**, les protocoles de signalisation les plus connus ainsi qu'une comparaison entre le H.323 et le protocole SIP pour justifier notre choix concernant le protocole SIP.

II Conception et architecture

1. Introduction

Dans les chapitres précédents nous avons décrit brièvement les différents protocoles, mis en jeu dans la réalisation d'une application multimédia, et relatifs à la signalisation des appels et au transport des données (voix et données), et nous avons terminé par une comparaison entre les deux protocoles de signalisation les plus connus et qui sont en concurrence aujourd'hui, à savoir H.323 et SIP. Pour les raisons évoquées nous avons choisi pour notre application SIP qui est simple d'implémentation et qui est un protocole d'avenir.

2. Spécification

La conception de notre application repose sur la réalisation des points suivants [16] :

- L'implémentation d'un protocole de signalisation se basant sur :
  - Le choix d'un protocole ouvert.
  - L'adaptabilité à tous les médias communicants.
  - L'utilisation du modèle client/serveur.
  - L'utilisation d'une architecture proxy-SIP.
  - L'enregistrement des profiles utilisateurs sur le serveur.
- L'application doit être multi-plateforme (interopérabilité).

3. Modèle de dialogue

SIP permet l'utilisation de deux architectures : une architecture impliquant un proxy utilisé dans le cas de postes de travail fixes et une deuxième impliquant en plus du proxy, un serveur de redirection utilisé dans le cas de terminaux mobiles.

Dans ce qui suit nous procédons à la modélisation générale d'un dialogue dans le cadre d'un SIP avec proxy qui s'adapte bien à notre application.

Notre application est de type client serveur. Elle permet de réaliser la signalisation d'appels SIP et le transport multimédia des données (voix et messages instantanés).

La **Figure II.1** représente le modèle de dialogue propre au SIP et à l'échange des paquets voix et données de notre application entre deux agents utilisateurs client serveur (*UAC/UAS*) A et B via un proxy :

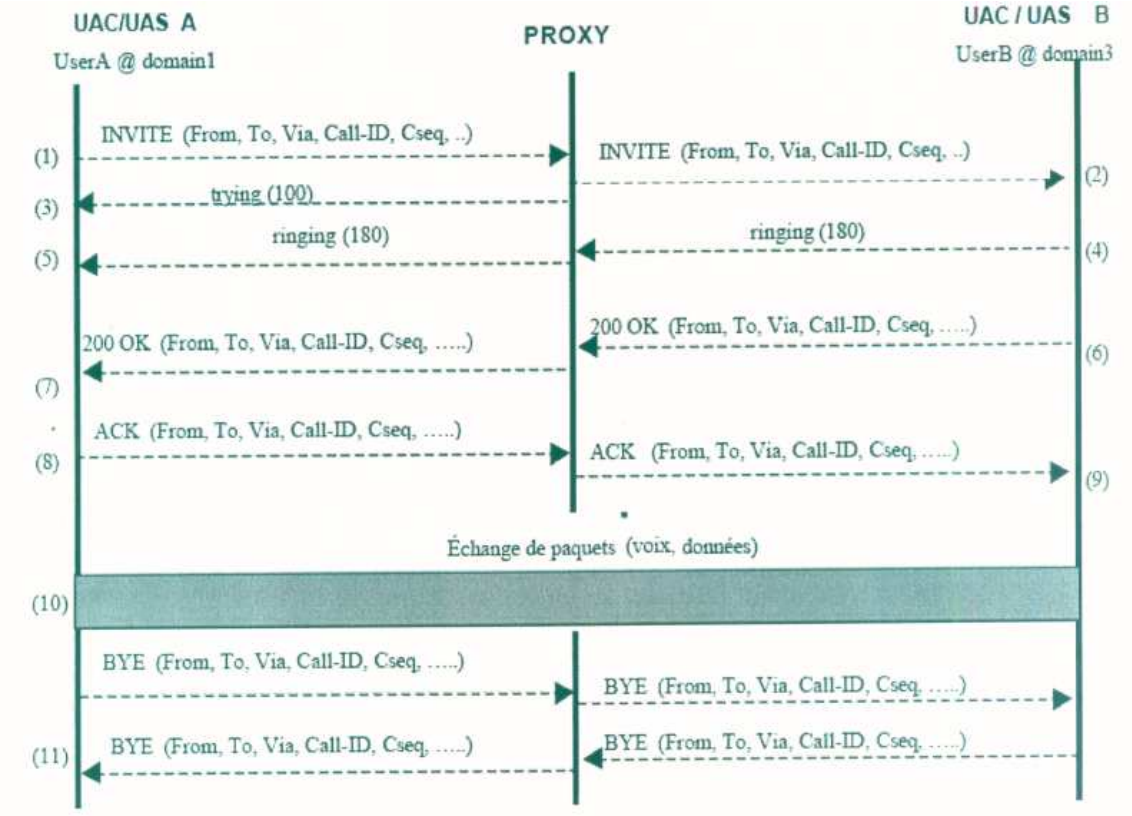


Figure II.1 : Modèle de dialogue [16]

L'établissement d'une communication entre clients se fait en unicast (point à point). Les étapes successives sont les suivantes [16] :

1. Le client A qui désire initier une communication émet une requête INVITE contenant les URL *From, To et Via*. Celle-ci est envoyée au serveur proxy.

2. Le proxy reçoit la requête INVITE du client, il en extrait l'URL To, consulte sa base de données pour vérifier si le destinataire existe et en fonction de cette vérification il renvoie un message d'état au client source le notifiant que destinataire n'est pas trouvé dans le cas où celui-ci n'existe pas.

   Dans le cas contraire il achemine la requête au destinataire à l'adresse IP contenue dans l'URL *To* pour signaler au destinataire une invitation à une communication.

3. En même temps que le proxy achemine la requête au destinataire, il envoie aussi un message d'état au client pour le notifier qu'il essaie d'établir un contact avec le destinataire.

4. Dés la réception de la requête INVITE par le client destinataire, un message d'invitation s'affiche à son écran lui indiquant le nom du client qui désire établir une communication. Un message d'état est envoyé de manière automatique au proxy qui l'acheminera vers le client source pour l'informer que le destinataire a reçus son invitation avec succès.

5. Le proxy reçoit la réponse du destinataire, il extrait l'URL *From* du message et de la même manière qu'à l'étape 2, il achemine celle-ci vers le client source. Un message *ringing* est affiché à l'écran de ce dernier.

6. Le client destinataire répond par message *Ok* au client source et accepte ainsi son invitation à communiquer. Cette réponse est envoyée au proxy.

7. Le proxy reçoit la réponse *Ok*. Il l'achemine au client source de la même manière qu'à l'étape 5.

8. A la réception de la réponse *Ok*. Un message d'acquittement est envoyé au proxy automatiquement contenant les mêmes URL From et *To* que les messages échangés précédemment.

9. De la même manière qu'à l'étape 2, proxy achemine ce message d'acquittement vers le destinataire.

10. La communication est établie dés que la réception des messages d'acquittement *Ack*.

    A cette étape les paquets voix et messages sont échangés entre les clients en multicast

Chaque fois qu'une communication est établie, le serveur proxy attribue une liaison (canal) directe entre les deux clients pour l'échange des paquets (vois et données).

11. Si un client désire mettre fin à sa participation à la conférence multimédia, il envoie une requête *bye* au proxy qui achemine à son tour aux autres clients pour les informer du départ de celui-ci.

4. Architecture des hiérarchies de classes

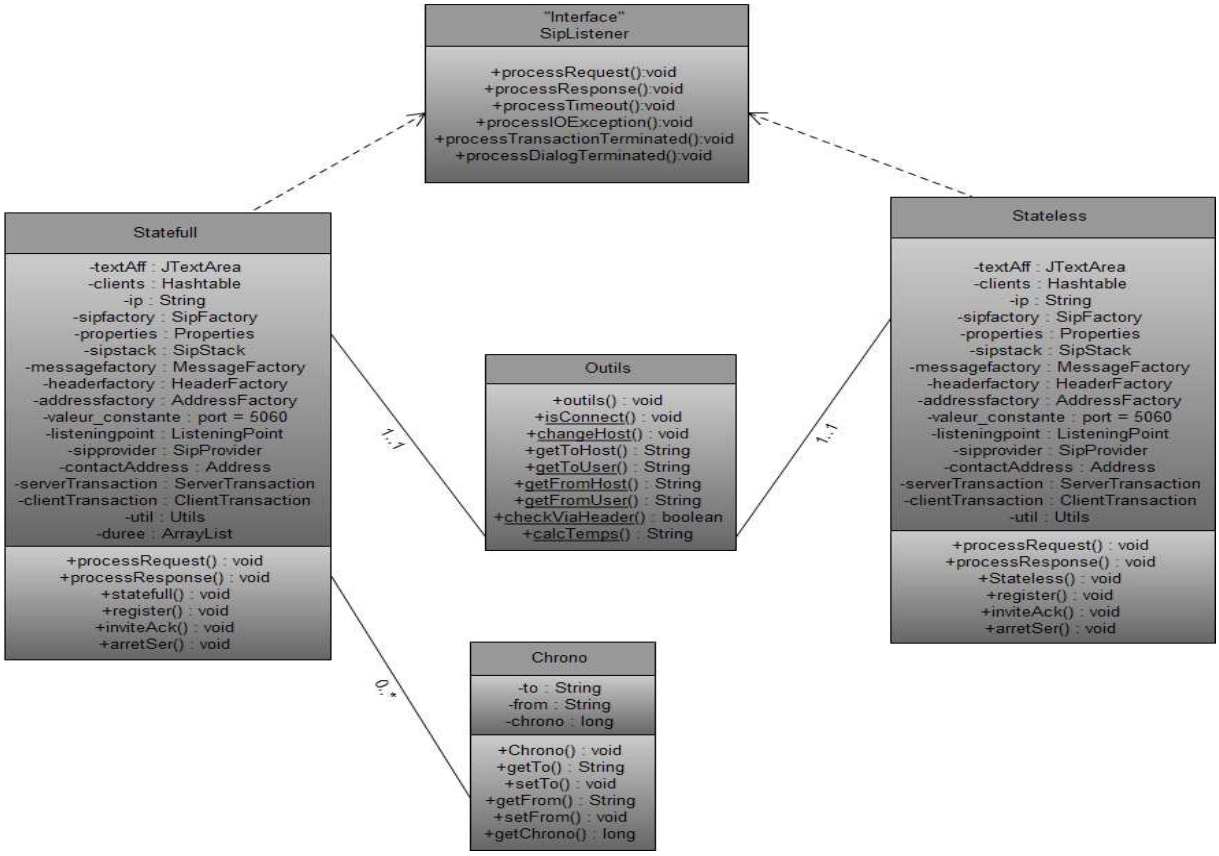


Figure II.2 : Diagramme des principales classes

4.1- Identification des classes :

Ci-dessus, nous avons représenté les principales classes de notre application. Notre programme assure la communication et la signalisation.

- **Statefull :** Définit le comportement d'un serveur registrar et proxy avec état (statefull) selon le **RFC 3261** [16]
- **Stateless :** Définit le comportement d'un serveur registrar et proxy sans état (stateless) selon le **RFC 3261**
- **Outils :** Présente les diffèrent méthode qui nous aide dans notre travail
- **Chrono :** C'est pour le calcul de la durée d'appel dans le cas d'un proxy Statefull

5. Conclusion :

Dans ce chapitre, nous avons décrit les spécifications et le modèle de dialogue propre à SIP, les différents composants qui constituent l'architecture système de notre application ainsi que leur fonctionnement.

II Conception et architecture

1. Introduction

Dans les chapitres précédents nous avons décrit brièvement les différents protocoles, mis en jeu dans la réalisation d'une application multimédia, et relatifs à la signalisation des appels et au transport des données (voix et données), et nous avons terminé par une comparaison entre les deux protocoles de signalisation les plus connus et qui sont en concurrence aujourd'hui, à savoir H.323 et SIP. Pour les raisons évoquées nous avons choisi pour notre application SIP qui est simple d'implémentation et qui est un protocole d'avenir.

2. Spécification

La conception de notre application repose sur la réalisation des points suivants [16] :

- L'implémentation d'un protocole de signalisation se basant sur :
  - Le choix d'un protocole ouvert.
  - L'adaptabilité à tous les médias communicants.
  - L'utilisation du modèle client/serveur.
  - L'utilisation d'une architecture proxy-SIP.
  - L'enregistrement des profiles utilisateurs sur le serveur.
- L'application doit être multi-plateforme (interopérabilité).

3. Modèle de dialogue

SIP permet l'utilisation de deux architectures : une architecture impliquant un proxy utilisé dans le cas de postes de travail fixes et une deuxième impliquant en plus du proxy, un serveur de redirection utilisé dans le cas de terminaux mobiles.

Dans ce qui suit nous procédons à la modélisation générale d'un dialogue dans le cadre d'un SIP avec proxy qui s'adapte bien à notre application.

Notre application est de type client serveur. Elle permet de réaliser la signalisation d'appels SIP et le transport multimédia des données (voix et messages instantanés).

La **Figure II.1** représente le modèle de dialogue propre au SIP et à l'échange des paquets voix et données de notre application entre deux agents utilisateurs client serveur (*UAC/UAS*) A et B via un proxy :

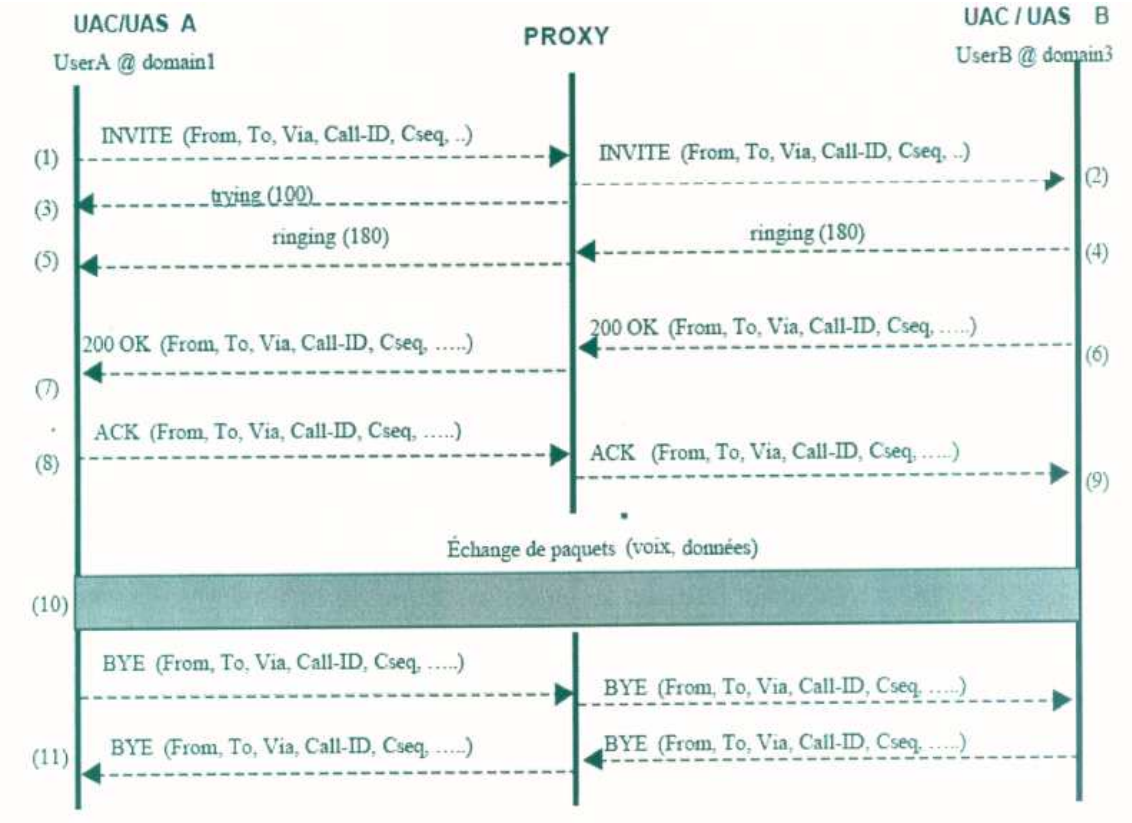


Figure II.1 : Modèle de dialogue [16]

L'établissement d'une communication entre clients se fait en unicast (point à point). Les étapes successives sont les suivantes [16] :

1. Le client A qui désire initier une communication émet une requête INVITE contenant les URL *From, To et Via*. Celle-ci est envoyée au serveur proxy.

2. Le proxy reçoit la requête INVITE du client, il en extrait l'URL To, consulte sa base de données pour vérifier si le destinataire existe et en fonction de cette vérification il renvoie un message d'état au client source le notifiant que destinataire n'est pas trouvé dans le cas où celui-ci n'existe pas.

   Dans le cas contraire il achemine la requête au destinataire à l'adresse IP contenue dans l'URL *To* pour signaler au destinataire une invitation à une communication.

3. En même temps que le proxy achemine la requête au destinataire, il envoie aussi un message d'état au client pour le notifier qu'il essaie d'établir un contact avec le destinataire.

4. Dés la réception de la requête INVITE par le client destinataire, un message d'invitation s'affiche à son écran lui indiquant le nom du client qui désire établir une communication. Un message d'état est envoyé de manière automatique au proxy qui l'acheminera vers le client source pour l'informer que le destinataire a reçus son invitation avec succès.

5. Le proxy reçoit la réponse du destinataire, il extrait l'URL *From* du message et de la même manière qu'à l'étape 2, il achemine celle-ci vers le client source. Un message *ringing* est affiché à l'écran de ce dernier.

6. Le client destinataire répond par message *Ok* au client source et accepte ainsi son invitation à communiquer. Cette réponse est envoyée au proxy.

7. Le proxy reçoit la réponse *Ok*. Il l'achemine au client source de la même manière qu'à l'étape 5.

8. A la réception de la réponse *Ok*. Un message d'acquittement est envoyé au proxy automatiquement contenant les mêmes URL From et *To* que les messages échangés précédemment.

9. De la même manière qu'à l'étape 2, proxy achemine ce message d'acquittement vers le destinataire.

10. La communication est établie dés que la réception des messages d'acquittement *Ack.*

    A cette étape les paquets voix et messages sont échangés entre les clients en multicast

    Chaque fois qu'une communication est établie, le serveur proxy attribue une liaison (canal) directe entre les deux clients pour l'échange des paquets (vois et données).

11. Si un client désire mettre fin à sa participation à la conférence multimédia, il envoie une requête *bye* au proxy qui achemine à son tour aux autres clients pour les informer du départ de celui-ci.

4. Architecture des hiérarchies de classes

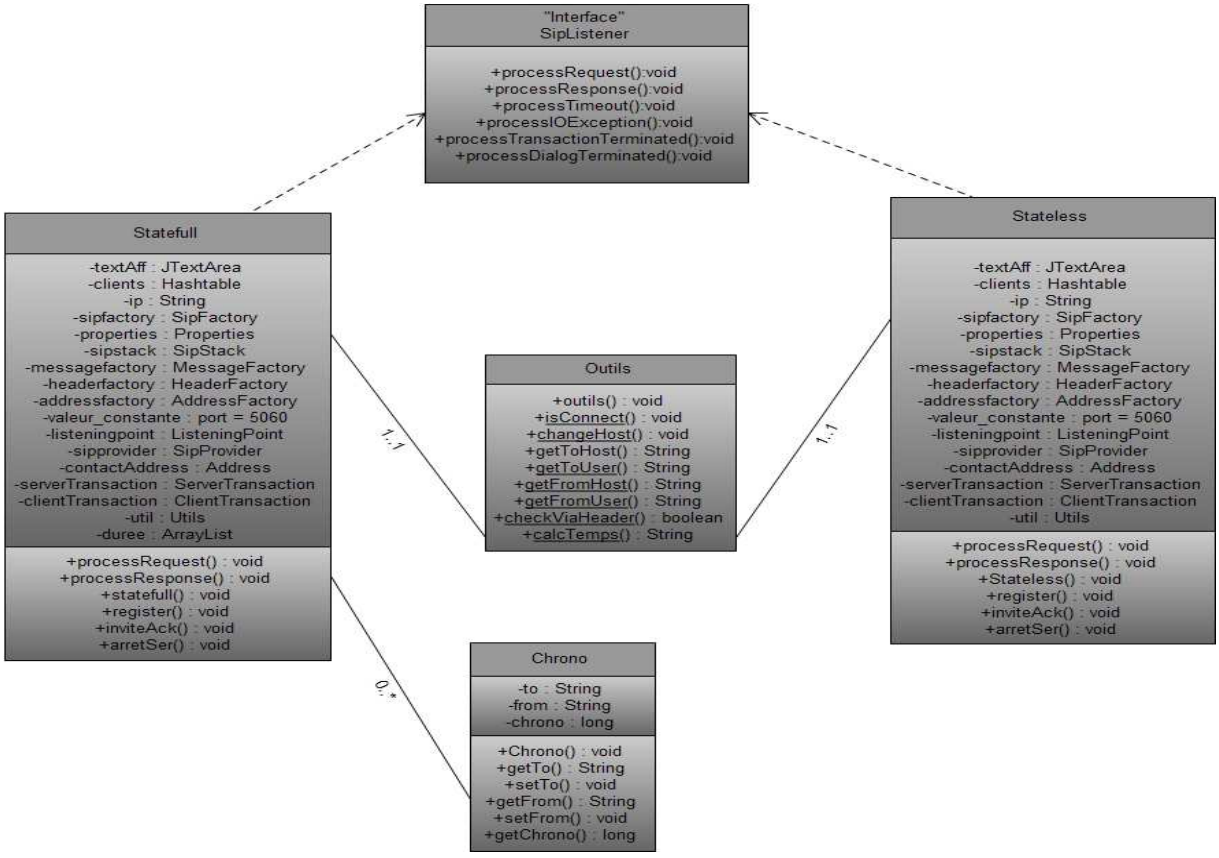


Figure II.2 : Diagramme des principales classes

4.1- Identification des classes :

Ci-dessus, nous avons représenté les principales classes de notre application. Notre programme assure la communication et la signalisation.

- **Statefull :** Définit le comportement d'un serveur registrar et proxy avec état (statefull) selon le **RFC 3261** [16]
- **Stateless :** Définit le comportement d'un serveur registrar et proxy sans état (stateless) selon le **RFC 3261**
- **Outils :** Présente les diffèrent méthode qui nous aide dans notre travail
- **Chrono :** C'est pour le calcul de la durée d'appel dans le cas d'un proxy Statefull

5. Conclusion :

Dans ce chapitre, nous avons décrit les spécifications et le modèle de dialogue propre à SIP, les différents composants qui constituent l'architecture système de notre application ainsi que leur fonctionnement.

CONCLUSION GENERALE :

Dans notre travail nous nous sommes intéressés aux réseaux multimédia tels que la téléphonie et surtout à l'aspect signalisation sous-jacente, phase nécessaire à l'établissement et au contrôle de la communication.

Nous avons étudié les protocoles de signalisation les plus connus et nous avons opté pour implémenter le serveur SIP. L'application que nous avons développée est permet d'établir toutes les phases de signalisation coté serveur avant l'établissement d'une connexion pour une session de messagerie instantanée ou de téléphonie. Le test de programme a donné des résultats concluant vis-à-vis des spécifications de notre projet. Cette application telle quelle tourne dans un environnement multi plateforme puisque programmée en Java. Elle est basée sur protocole ouvert.

Ce projet nous a permis de nous initier aux protocoles utilisés dans les réseaux multimédias. La programmation de cette application nous a permis de découvrir et de maîtriser.

Nous avons utilisé une API disponible (JAIN SIP) qui contient toutes les classes nécessaire pour le développement de notre application.

Notre application est extensible. Elle peut d'être enrichie par d'autre modules permettant :

- La redirection des requêtes SIP dans le cadre de mobilités des utilisateurs.

- D’incorporer des mécanismes de qualité de service et de réservation de ressources.
- D’incorporer aussi des modules de cryptage et d’authentification pour une meilleure sécurité des échanges.
- D’assurer une passerelle avec d’autres protocoles de signalisation, pour une meilleure interopérabilité.

INTRODUCTION :

Depuis l’invention du téléphone, de nombreux progrès se sont opérés dans les domaines des télécommunications.

Avec le développement des nouvelles technologies informatiques ainsi que le développement et l’utilisation croissante de l’internet, le monde des télécommunications est entré dans une ère nouvelle.

Le besoin de communication à moindres coût entre personnes et clients a fait basculer une partie du trafic issu des lignes téléphoniques conventionnelles (utilisant la commutation de circuits) sur le réseau internet (à commutation de paquets). Cette technique basée sur le réseau internet est appelée, dans le jargon des télécommunications, Voice Over IP.

Par extension de cette technique, la transmission de la voix et de la vidéo en *unicast* et *multicast* sur IP représente une avancée technologique importante dans le domaine du multimédia.

Cette transmission repose essentiellement sur un protocole d’échanges (*handshaking*) qui prend en charge le contrôle des échanges effectués. Elle est assurée par des protocoles tels que H.323 et SIP. Cet échange est propre de la signalisation effectuée entre les nœuds d’un réseau. La signalisation sert à l’établissement et au contrôle des connexions à travers le réseau (établir et terminer les appels). Elle permet également l’échange d’informations concernant la gestion du réseau et de ses ressources. Ces fonctions deviennent de plus en plus importantes dans la téléphonie mobile. Où les terminaux et les liaisons ont des caractéristiques très variées et devant être négociées avant l’établissement de la connexion.

L’avènement des protocoles standardisés a libéré les développeurs du fardeau d’un développement de protocole de signalisation qui les obligeaient à construire autant de protocoles propriétaires que de clients.

L’objectif de notre travail, dans le cadre du projet de fin d’études, est la conception et la réalisation d’une application basée sur un protocole de signalisation dans le domaine de la communication multimédia en utilisant les protocoles d’internet et les standards connus, précisément SIP. En effet, les logiciels tels que NetMeeting de Microsoft, qui font de la communication multimédia, utilisent un modèle de signalisations qui sont propriétaires.

Notre objectif s’inscrit dans l’utilisation et la manipulation de standards développés par les organismes de télécommunication tels que l’ITU-T et l’IETF. Ceci a pour avantage principal de faciliter l’interopérabilité des équipements et des applications dans ce domaine. Cette application est implantée avec une approche orientée objet en utilisant le langage de programmation Java pour des raisons de portabilité et de réutilisabilité.

L’application que nous avons développée est un serveur SIP, qui se charge d’analyser et traiter les messages de contrôles SIP, échangés entre les clients SIP distants pour établir et terminer les appels (protocole de signalisation SIP).

Notre choix s’est porté sur le protocole SIP. Ce dernier utilisé dans le contexte de la téléphonie sur Internet se trouve être un protocole extensible destiné à trouver des applications en dehors du cadre de la téléphonie.

Ce mémoire se présente comme suit : Le chapitre 1, présente l’état de l’art sur la téléphonie IP et la signalisation. Les principaux protocoles de signalisation seront présentés et comparés. Le chapitre 2 est consacré au modèle de dialogue SIP qui sera implémenté dans notre application et à l’architecture des principales classes développées (UML). Dans le chapitre 3 nous présentons la mise en œuvre, et les tests de l’application ainsi que son utilisation pour l’établissement de la communication téléphonie.

Liste des figures:

Liste des tableaux:

Liste des abréviations :

ADSL: Asymmetrical Data Subscriber Loop.

API: Application Programming Interface.

DHCP: Dynamic Host Configuration Protocol**.**

DNS: Domain Name Server.

DSP: Digital signal processor.

HTTP: HyperText Transfer Protocol**.**

IETF: Internet Engineering Task Force**.**

ETSI: European Telecommunications Standards Institute.

IP: Internet Protocol.

ITU: International Telecommunication Union.

L.S: Location Server.

MG: Media Gateway.

MGC: Media Gateway Controller.

OSI: Open System International.

PBX: Private Branch eXchange.

PSTN: Public Switched Telephone Network.

PABX: Private Automatic Branch eXchange.

PS: Proxy Server.

PSTN: Public Switched Telephone Network**.**

R.G: ReGistrar.

RNIS: Réseau Numérique à Intégration de Service**.**

RS: Redirect Server**.**

RSVP: ReSerVation Protocol.

RTC: Réseau Téléphonique Commuté.

RTCP: Real time Transport Control Protocol.

RTP: Real time Transport Protocol.

SDP: Session Description Protocol**.**

SG: Signalling Gateway.

SIP: Session Initiation Protocol.

SMTP: Simple Mail Transfer Protocol**.**

SS7: Signalling System No.7.

TCP: Transmission Control Protocol.

ToIP: Telephony over IP.

UA: User Agent.

UAC: User Agent Client.

UAS: User Agent Server.

UDP: User Datagram Protocol.

VoIP: Voice over Internet Protocol.

MGCP: Media Gateway Control Protocol.

JVM: Java Virtual Machine.

UML: Unified Modeling Language

Table des matières :

Table des matières :

ملخص:

مع تطور المعلومات الجديدة وتطوير وزيادة استخدام تقنيات الإنترنت، دخلت عالم الاتصالات حقبة جديدة.

نقل الصوت والفيديو عبر IP أحادي الإرسال والبث المتعدد يمثل طفرة تكنولوجية كبيرة في مجال الوسائط المتعددة.

هذا الإرسال يستند على بروتوكول تبادل (المصافحة) الذي يتكلف بالمراقبة و التبادل المنفذ. المؤمن من قبل بروتوكولات مثل H.323 و SIP.

الهدف من عملنا في إطار مذكرة التخرج، هو تصميم و إنجاز تطبيق الإشارات القائمة على بروتوكول في مجال الاتصالات والوسائط المتعددة باستخدام بروتوكولات الإنترنت والتطبيقات المعايير المعروفة، SIP على وجه التحديد.

التطبيق الذي وضعناه هو خادم SIP ، والذي هو مسؤول عن تحليل ومعالجة رسائل التحكم SIP المتبادلة بين SIP للعملاء البعيدين لربط وإنهاء المكالمات (بروتوكول للإشارات SIP).

كلمات مفاتيح: مراقبة، IP، بروتوكول، H.323 و SIP، خادم.

Résumé:

Avec le développement des nouvelles technologies informatiques ainsi que le développement et l'utilisation croissante de l'internet, le monde des télécommunications est entré dans une ère nouvelle.

La transmission de la voix et de la vidéo en *unicast* et *multicast* sur IP représente une avancée technologique importante dans le domaine du multimédia.

Cette transmission repose essentiellement sur un protocole d'échanges (*handshaking*) qui prend en charge le contrôle des échanges effectués. Elle est assurée par des protocoles tels que H.323 et SIP.

L'objectif de notre travail, dans le cadre du projet de fin d'études, est la conception et la réalisation d'une application basée sur un protocole de signalisation dans le domaine de la communication multimédia en utilisant les protocoles d'Internet et les standards connus, précisément SIP.

L'application que nous avons développée est un serveur SIP, qui se charge d'analyser et traiter les messages de contrôles SIP, échangés entre les clients SIP distants pour établir et terminer les appels (protocole de signalisation SIP).

Abstract:

With the development of new information and increasing use of internet technologies, the telecommunications world has entered a new era.

Transmission of voice and video over IP unicast and multicast represents a major technological breakthrough in the field of multimedia.

This transmission is essentially an exchange protocol (handshaking) that supports the control of the exchanges. It is ensured by protocols such as H.323 and SIP.

The objective of our work in the project graduation is the design and implementation of a protocol-based signaling in the field of multimedia communication using the Internet application protocols and known standards, specifically SIP.

The application we have developed is a SIP server, which is responsible for analyzing and processing the SIP control messages exchanged between the SIP remote clients to establish and terminate calls (SIP signaling protocol).

Mots-clés : unicast, multicast, IP, protocoles, H.323, SIP, signalisation, serveur.