

N^o d'ordre : 05/2007-M/IN

REPUBLIQUE ALGERIENNE DEMOCRATIQUE ET POPULAIRE
MINISTERE DE L'ENSEIGNEMENT SUPERIEUR ET DE LA RECHERCHE
SCIENTIFIQUES
UNIVERSITE DES SCIENCES ET DE LA TECHNOLOGIE HOUARI
BOUMEDIENNE
Faculté d'Electronique et Informatique



MEMOIRE

Présenté pour l'obtention du diplôme de **MAGISTER**

En : Informatique

Spécialité : Intelligence Artificielle et Bases de Données Avancées.

Par : KHORSI Hassiba

Thème

LA NEGOCIATION DE SERVICES DANS LE WEB

Soutenu le 02/05/2007, devant le jury composé de :

Mr N.Badache, Professeur, USTHB

Président

Mr A. Belkhir, Maître de Conférences, USTHB

Directeur de Thèse

Mr M.Ahmed Nacer, Professeur, USTHB

Examineur

Mr H.Azzoune, Maître de Conférences, USTHB

Examineur

Remerciements

Je remercie le bon dieu d'avoir exaucer mon vœux est de terminer ce travail.

Je remercie Mr BELKHIR Abdelkader, Maître de Conférences à l'USTHB d'avoir diriger et suivi ce travail, et sa patience continue durant le développement de ce mémoire.

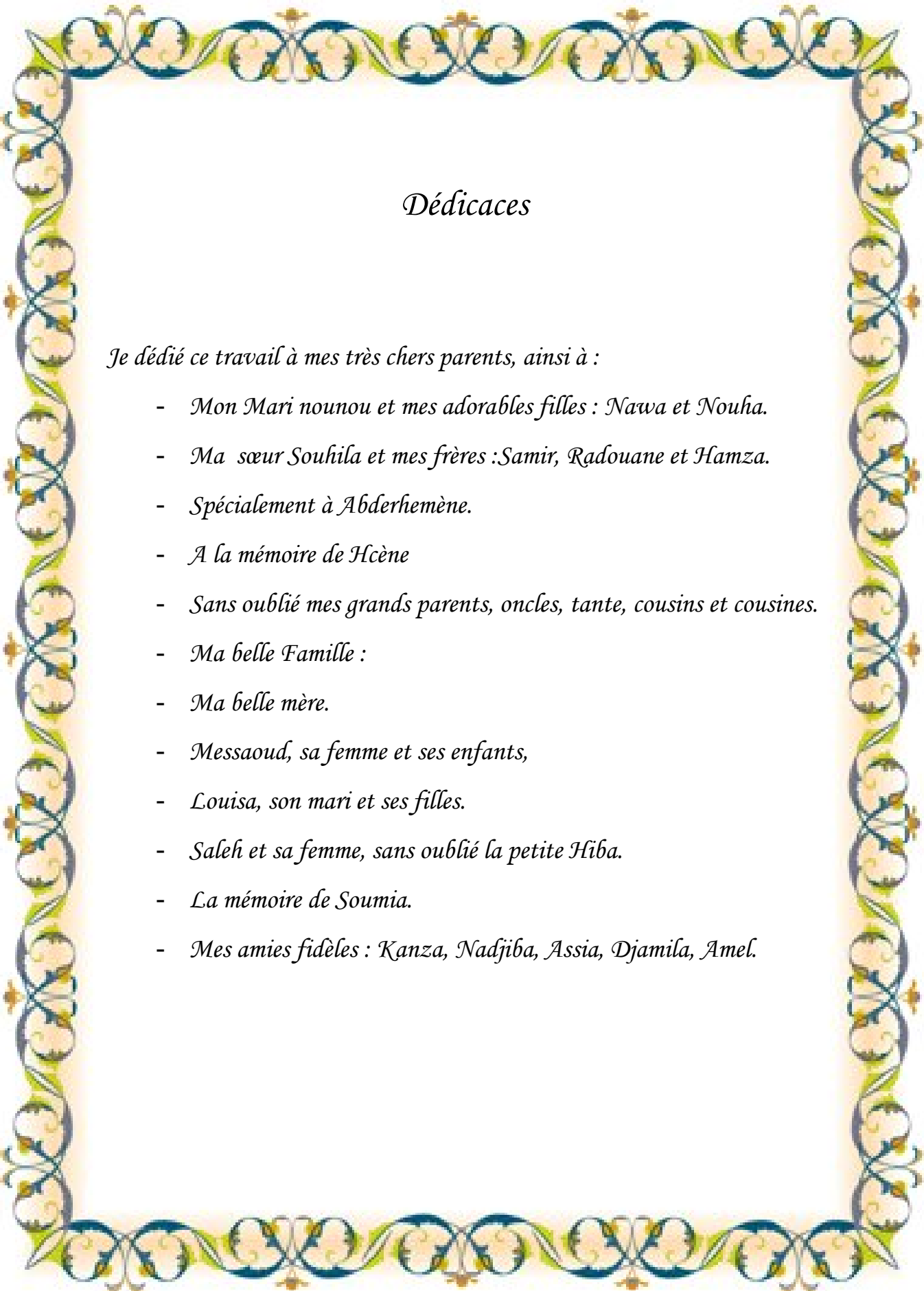
Je remercie également les membres du jury :

Mr BADACHE Nadjib Professeur à l'USTHB de m'avoir fait l'honneur de présider mon jury.

Mr AHMED NACER Mohamed Professeur à l'USTHB et Mr AZZOUNE Hamid Maître de Conférences à l'USTHB d'avoir acceptés de juger ce travail.

Je remercie toute ma famille, mes parents, et mon mari qui était toujours à mes cotés qui sans eux je ne serais pas là. Je remercie également ma sœur et mes frères.

Enfin, je tiens à remercie Mr Tebbani, Hakima, Fatiha, Nocéra, Faiza et rachida pour leurs aides, sans oublié tous le personnel de la DFCRI.

A decorative border with a repeating floral and scrollwork pattern in blue, green, and gold, framing the central text.

Dédicaces

Je dédie ce travail à mes très chers parents, ainsi à :

- *Mon Mari nounou et mes adorables filles : Nawa et Nouha.*
- *Ma sœur Souhila et mes frères : Samir, Radouane et Hamza.*
- *Spécialement à Abderhemène.*
- *A la mémoire de Hcène*
- *Sans oublié mes grands parents, oncles, tante, cousins et cousines.*
- *Ma belle Famille :*
- *Ma belle mère.*
- *Messaoud, sa femme et ses enfants,*
- *Louisa, son mari et ses filles.*
- *Saleh et sa femme, sans oublié la petite Hiba.*
- *La mémoire de Soumia.*
- *Mes amies fidèles : Kanza, Nadjiba, Assia, Djamila, Amel.*

Table des matières

Introduction Générale.....	1
 Chapitre I : Etat de l'Art	
1- Introduction.....	3
2- Architecture NAC	4
3- Le protocole HTTP	5
4 -Le protocole SIAN.....	7
5- Conclusion	8
 Chapitre II : Les principaux protocoles et outils de communication multimédia	
1- Introduction.....	9
2- Le protocole d'initiation de session SIP	9
2.1- Origine	9
2.2- Fonctionnalités.....	10
3- Architecture de SIP	10
4- Les composants d'un système SIP	13
5- Ouverture d'une session.....	17
5.1- Localisation d'un utilisateur SIP	17
5.2- Localisation d'un serveur SIP	18
5.3- Exemples de transactions	19
6- URL (Uniform Ressource Locators) SIP.....	22
7- Format des messages SIP	24
7.1- Ligne de requêtes	25
7.2- Ligne de réponse (d'état)	26
7.3- Les méthodes.....	27
7.4- Champs d'en-tête	28
7.5- Le corps du message	30
8- Fiabilité et sécurité du protocole SIP	33
8.1- Fiabilité	33
8.2- Sécurité et authentification.....	34

9- Concurrents de SIP	35
9.1- Le protocole H323.....	35
9.2- MGCP/MEGACO	36
10- Conclusion	36
11. Le Protocole de Description de Session : SDP	37
11.1- Introduction	37
11.2- Transport.....	37
11.3- Syntaxe	38
11.4- Conclusion	44

Chapitre III : Le Protocole CC/PP/ Gestion du profil

1- Introduction.....	45
2- Le protocole CC/PP	46
2.1- Origine	46
2.2- Définition.....	46
2.3- La structure d'un profil CC/PP	47
2.4- L'utilisation et l'extensibilité du vocabulaires CC/PP	55
3- La conformité.....	55
3.1- La conformité du document CC/PP	55
3.2- La conformité du producteur CC/PP	56
3.3- La conformité du consommateur CC/PP	56
3.4- Les revendications de conformité.....	56
4- Conclusion	57

Chapitre IV : Connexion, Etat de profil, et Négociation de Services

1- Introduction.....	58
2- Problématique	58
3- Solution proposée.....	60
4- Schéma descriptif.....	63
5- Qu'est ce qu'une métaphore ?.....	64
6- Les éléments d'une bonne métaphore	65
7- Solution métaphorique	65

8- Diagramme de négociation	67
9- Application.....	70
10- Conclusion	79
Conclusion générale	80
Annexes	
Annexe A.....	81
Annexe B.....	99
Annexe C.....	107
Bibliographie.....	111

Liste des Figures et des Tableaux

➤ FIGURES

Figure 1 : Organisation générale de l'architecture NAC	4
Figure 2 : Le principe du type MIME	6
Figure 3 : Modèle de la négociation du protocole HTTP 1.0.....	6
Figure 4 : Le principe du protocole SIAN.....	8
Figure 5 : L'architecture en couche du protocole SIP (Pile du protocole SIP).....	11
Figure 6 : Les différents types de communication avec SIP.	13
Figure 7 : Exemple d'établissement d'une session SIP entre deux User Agents.	14
Figure 8 : Les différents composants d'un système SIP.	17
Figure 9 : Etablissement de session à travers un serveur Proxy.....	20
Figure 10 : Etablissement de session à travers un Redirect Server.....	21
Figure 11 : Exemple d'un message SIP. 5060 : port par défaut.....	31
Figure 12 : Exemple d'un message SIP, en utilisant les abréviations.....	32
Figure 13 : Exemple de description SDP.	41
Figure 14 : Un profil sous forme d'un arbre.	47
Figure 15 : Les composants d'un profil CC/PP.	49
Figure 16 : Les composants d'un profil CC/PP en XML.....	49
Figure 17 : Une déclaration RDF décrivant un attribut du client.....	50
Figure 18 : Exemple de profil CC/PP complet en utilisant les valeurs complexes.....	51
Figure 19 : Un profil CC/PP utilisant des valeurs par défaut.....	52
Figure 20 : Exemple d'utilisation des redéfinitions des valeurs par défaut.....	53
Figure 21 : Exemple d'utilisation des redéfinitions des valeurs par défaut.....	53
Figure 22 : Un profil CC/PP référençant des valeurs par défaut externes en XML.....	54
Figure 23 : Processus de fourniture de services dans le Web.	63
Figure 24 : Le processus de négociation.	67
Figure 25 : L'intervention du processus de Négociation dans un service Web.....	70
Figure 26 : Utilisation de la notion d'interface papier pour présenter les offres.....	74
Figure 27 : Le service offert	78

➤ TABLEAUX

Tableau 1 : Champs d'en-têtes.	29
Tableau 2 : Les champs des types autorisés par SDP.	40
Tableau 3 : Types de média.....	43

Résumé

Le Web évolue vers des contenus intégrant des médias de plus en plus riches et variées, accédés en tout lieu et à tout moment. En plus des contenus multimédia, des services sont fournis pour la satisfaction des clients. Mais cette satisfaction dépend des préférences, des capacités des clients ainsi que de leurs environnements.

Afin d'atteindre cet objectif, un accord doit être établi entre le client et le fournisseur de service; un ensemble de protocoles sont étudiés dans le but d'établir la communication, la description des préférences et les capacités du client. De plus, l'introduction d'un processus de négociation est très importante pour assurer un échange d'informations cohérent qui permet d'aboutir à arrangement (accord) entre le client et le fournisseur. Ce mémoire s'intéresse à la mise en œuvre d'un processus de négociation de services en tenant comptes des contraintes d'environnement et des préférences utilisateurs.

Mots clés :

Protocole de signalisation, Communication, Préférences, Profil, Offres, Négociation,...

Introduction Générale

Le volume des réseaux a rapidement augmenté et ce phénomène ne s'est pas arrêté à l'Internet mais il s'est étendu aux organisations et entreprises. De nos jours, les réseaux informatiques modernes constituent un média innovateur qui supporte les nouvelles formes de coopération et de communication entre les utilisateurs.

En quelques mois, les Web Services sont devenus le nouveau point de convergence technologique de l'ensemble des acteurs du marché de l'informatique.

L'objectif des Web Services est de faciliter l'accès aux applications entre entreprises et ainsi de simplifier les échanges de données. Ils poursuivent un vieux rêve de l'informatique distribuée où les applications pourraient interopérer à travers le réseau, indépendamment de leur plate-forme et de leur langage d'implémentation. Dans ce sens, ils s'inscrivent dans la continuité d'initiatives telles que CORBA (Common Object Request Broker Architecture, de l'OMG) en apportant toutefois une réponse plus simple, s'appuyant sur des technologies et standards reconnus et maintenant acceptés de tous [NIQ06].

L'échange d'informations ou l'offre des services dans le web n'est pas le seul objectif des nouvelles technologies, mais ces dernières cherchent à satisfaire l'utilisateur en lui proposant des services qui s'accordent à leurs préférences et leur environnement. Une approche qui permet d'atteindre cet objectif est l'introduction de la notion de " Négociation de services", afin de permettre au client de "créer" son propre service.

Face à l'explosion du multimédia, il est devenu nécessaire de développer de nouvelles méthodes d'accès à l'information qui ne reposent plus sur un accès purement linéaire. Elles permettent à partir de nouvelles structures organisationnelles et opérationnelles de documents, un accès rapide et naturel à l'information. La prise en compte de cet aspect passe par la réalisation d'une interface graphique facilitant le parcours, la recherche et la visualisation des médias. Ainsi l'utilisation des protocoles qui permettent le transfert et l'interprétation de ces médias en temps réel sur IP [AND03], s'avère nécessaire.

Afin d'illustrer les différents aspects cités avant, ce mémoire sera organisé comme suit:

Nous allons commencer par un état de l'art à travers lequel nous allons illustrer quelques travaux qui utilisent le « processus de négociation ».

Ensuite, nous allons présenter le protocole d'initiation de sessions SIP, qui permet l'établissement, la modification et la terminaison des sessions multimédia interactives sur IP entre des terminaux, en présentant son fonctionnement, son architecture, la structure de ses messages (requêtes et réponses) [MHE99]. Nous introduisons le protocole de description de sessions SDP comme un outil élémentaire pour la compréhension des messages SIP. Il est utilisé pour annoncer les conférences multimédias et spécifier les éléments nécessaires pour la participation [HAJ06].

Le troisième chapitre est consacré pour la description du protocole CC/PP (Capacités Combinées/Profils de Préférences), ce dernier permet de décrire les profils (préférence) des utilisateurs ainsi leur environnement [JJS04].

Enfin, nous allons présenter notre processus de **Négociation de services** par lequel nous allons essayer de satisfaire les besoins d'un client, ainsi qu'une application à ce processus. Dans cette application, nous allons utiliser la notion d'interface papier pour la présentation des offres. [CHS04]

Chapitre I : Etat De L'art

1- Introduction

Le web a profondément changé depuis ses origines. L'époque est loin où on partageait de simples pages de texte et quelques images à partir de stations de travail et de PC connectés à l'Internet. Des évolutions particulièrement marquantes ont eu lieu ces dernières années dans le domaine des contenus et dans celui des moyens d'accès réseau. Les contenus du web comprennent maintenant des médias continus, qu'il s'agisse de vidéo, de son ou d'animations. On trouve aussi de véritables documents multimédias, où plusieurs ressources de différents médias sont intégrées, se complètent mutuellement et sont mises en relation étroite les unes avec les autres. Les moyens d'accès ont également beaucoup changé depuis les débuts du web, qu'il s'agisse des terminaux ou des supports de communication.

Lorsqu'une application cliente accède à un document stocké sur un serveur, les différents profils peuvent être exploités afin de restituer un contenu adapté au client. Malheureusement cette opération suppose la présence sur le serveur d'un service couvrant un large spectre de profils. Dans le cas général, il est supposé aussi que le client supporte un certain nombre de fonctionnalités et le serveur se restreint à offrir un nombre limité de services d'accès. Dans ce cas il faut établir une procédure de négociation entre ces deux entités permettant d'identifier le service optimal qui permet de satisfaire au mieux les fonctionnalités du client et les préférences de l'utilisateur [TAN02]. Il existe deux stratégies de négociation, [TAL04]:

- 1- **La négociation à la demande** : Dans ce type de négociation, le processus d'adaptation est invoqué lors de la réception de la requête de l'application cliente. Dans ce cas, l'ensemble des profils et documents qui peuvent exister au niveau du serveur ne subissent pas de changement.
- 2- **La négociation fondée sur des caches** : au contraire du premier type de négociation, cette négociation se fonde sur l'utilisation des versions du contenu source préalablement sauvegardées. Dans ce cas, l'ensemble des profils s'enrichit par les nouveaux profils qui correspondent au contenu stocké.

La négociation dans le Web est un thème qui a été abordée par plusieurs chercheurs, tel que son introduction dans les versions du protocole HTTP, dans l'architecture NAC (Negociation Adaptation Core) [TAL04], dans la résolution de quelques problèmes des Systèmes Multi-Agent (SMA) [PIR03],.....

2- Architecture NAC

NAC (Negociation Adaptation Core) s'inscrit dans une ligne de travaux très actifs sur l'adaptation et la négociation des multimédias dans le web. [NTV05]

NAC vise à assurer l'accès adapté aux contenus multimédias, où qu'ils se trouvent sur le web et quel que soit le protocole utilisé pour les transférer. Mais le serveur détenant l'information peut ne disposer d'aucun moyen d'adaptation et utiliser un protocole insuffisant en matière de négociation de contenu. Le client peut également souffrir des mêmes limitations. La solution consistait à interposer un *proxy* entre le client et le serveur. Ce proxy prend en charge l'essentiel de la négociation et de l'adaptation. En contact à la fois avec le client et avec le serveur, il a une vision globale du contexte. [NTV05]

L'approche de NAC consiste à séparer la négociation et le transfert de contenu dans deux protocoles différents. Ainsi, même si un client ne supporte pas la négociation, le proxy peut s'engager pour lui, à partir de son profil, dans une négociation avec un serveur et lui retourner ensuite le contenu négocié. Pour cela, le client n'a besoin que du protocole de transfert habituel dans ses échanges avec le proxy. L'architecture NAC est organisée comme suit :

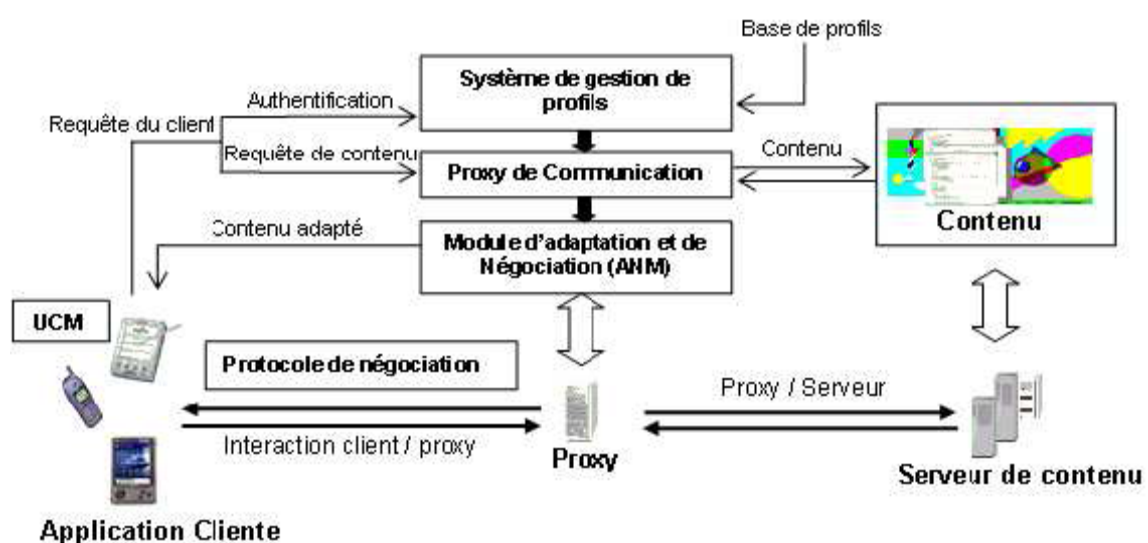


Figure 1 : Organisation générale de l'architecture NAC.

Cette architecture est composée de cinq entités, [NTV05] :

1- Le proxy de communication : assure l'accès au contenu et la communication qui se déroulent classiquement entre le client et le serveur. En réservant un port additionnel, le proxy assure aussi la communication orientée négociation avec le module UCM du client.

2- Le module d'adaptation et de négociation (ANM) : ce module assure l'adaptation de contenu et la négociation. Ce module coopère avec les autres entités de l'architecture afin de prendre la meilleure décision de négociation : choix de version de contenu, choix de méthode d'adaptation, choix de méthode de transmission du contenu final, etc.

3- Le module du contexte de l'utilisateur (UCM) : est utilisé côté client pour permettre une négociation avancée du contenu, d'une manière complètement indépendante de l'application cliente utilisée.

4- Le protocole de négociation permet de mettre en oeuvre une stratégie de négociation avancée. Il définit un mode d'interaction entre le module UCM côté client et le processus de négociation du module ANM. Cette interaction s'effectue par l'échange de requêtes et de réponses. Par exemple, le processus de négociation peut interroger le terminal sur ses capacités et ses préférences courantes ou il peut vérifier si le profil client a subi un changement depuis la dernière interrogation.

5- Le système de gestion de profils assure l'analyse et la gestion des descriptions de contexte (la description du client, du contenu, des méthodes d'adaptation, etc.) au profit de l'adaptation de contenu.

Le protocole de négociation est un aspect important de NAC. L'absence d'un tel protocole dans les systèmes d'adaptation disponibles sur le web réduit considérablement leurs possibilités, puisqu'il est alors impossible d'avoir une vue précise et actualisée de tous les aspects de contexte.

3- Le protocole HTTP

Dans le web, le protocole le plus utilisé est HTTP. Son rôle essentiel est de transmettre les contenus entre serveur et client. Dans sa première version, il offrait très peu de possibilités de négociation. Tout au plus permettait-il au client d'énoncer quelques préférences (par l'entête Accept) et les langues ou les médias acceptés (par le types MIME). (Voir la figure 2)



Figure 2 : Principe du type MIME.

Le protocole HTTP/1.0 suit le modèle de négociation de contenu présenté par la Figure 3. Lorsqu'une application cliente envoie une requête pour une ressource qui existe sur le serveur, les entêtes de type *Accept* (incluses dans la requête) expriment les capacités de l'application et les préférences de l'utilisateur. Le serveur utilise ces entêtes pour choisir la meilleure variante disponible du contenu source. La variante choisie est envoyée comme réponse à la requête. [TAL04]

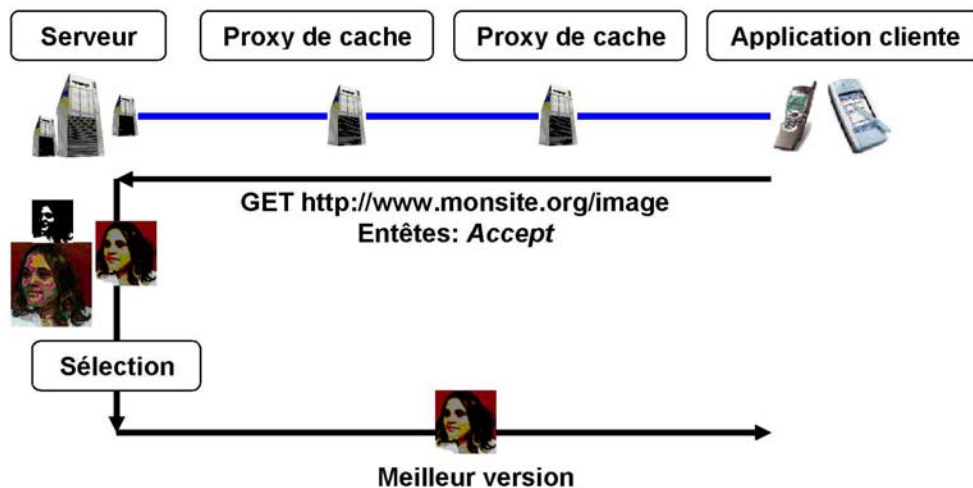


Figure 3 : Modèle de négociation du protocole HTTP/1.0.

Le problème majeur du modèle de négociation du protocole HTTP/1.0 réside lors de l'expression de toutes les capacités et préférences dans les entêtes *Accept* qui peut être très volumineuse.

Le protocole HTTP/1.1 a été proposé pour pallier les inconvénients du protocole HTTP/1.0 et faire face à ses limitations telles que l'organisation hiérarchique des proxy, la gestion du cache, les connexions persistantes, la négociation du contenu, etc. [TAL04]

Le protocole HTTP/1.1 permet la gestion des différentes versions (ou variantes) du même contenu sous une unique URI. Le processus de négociation du contenu de HTTP/1.1 consiste à effectuer une sélection de la meilleure variante en fonction des caractéristiques du client cible. Ce principe est le même que dans l'ancienne version du protocole; cependant HTTP/1.1 définit de nouvelles stratégies et considère la négociation du côté des entités de cache.

Le protocole HTTP/1.1 définit le processus de négociation de contenu, comme un mécanisme de sélection de la représentation la plus appropriée en réponse à une requête cliente. La représentation des ressources peut être négociée dans n'importe quelle réponse du serveur.

4 -Le protocole SIAN

Depuis quelques années, les systèmes multi-agent (SMA) ont été étudiés selon différentes approches. Les points principaux d'études ont été la communication et la coopération entre les agents. Dans les SMA, atteindre la coordination entre agents autonomes est un problème majeur. En générale, les agents coordonnent leurs actions en échangeant des messages. Un processus de coordination est une séquence de ces interactions entre agents.

Dans ce cas, la négociation s'effectue sur un contrat pour obtenir des ressources communes et sur demande d'un agent appelé **Initiateur**. Elle réunit l'initiateur et un ensemble d'agents appelés **participants**, auxquels le contrat est proposé et se déroule jusqu'à ce qu'un accord satisfaisant un pourcentage de participants soit trouvé. Les participants cherchent également à obtenir la meilleure solution possible pour eux tout en donnant le minimum d'information aux autres. [PMV05]

Les protocoles de négociation permettent de définir un cadre pour l'utilisation d'un langage. Ils peuvent être normalisés de plusieurs manières (graphe de raisonnement, réseau de Pétri, langage formel), cependant la plus classique est celle du graphe état-transition. Chaque noeud représente l'état d'un agent, relié par des arcs représentant l'émission-réception d'un acte de langage. Quand on passe directement d'un noeud à un autre, on passe en fait du noeud état de l'agent A au noeud état de l'agent B. Tous les agents partagent le même protocole.

Le protocole de SIAN [PIR03] est un protocole dédié à la négociation (à l'origine pour s'accorder sur la meilleure interprétation de l'environnement). Des propositions sont émises, discutées et mènent à un accord ou un désaccord.

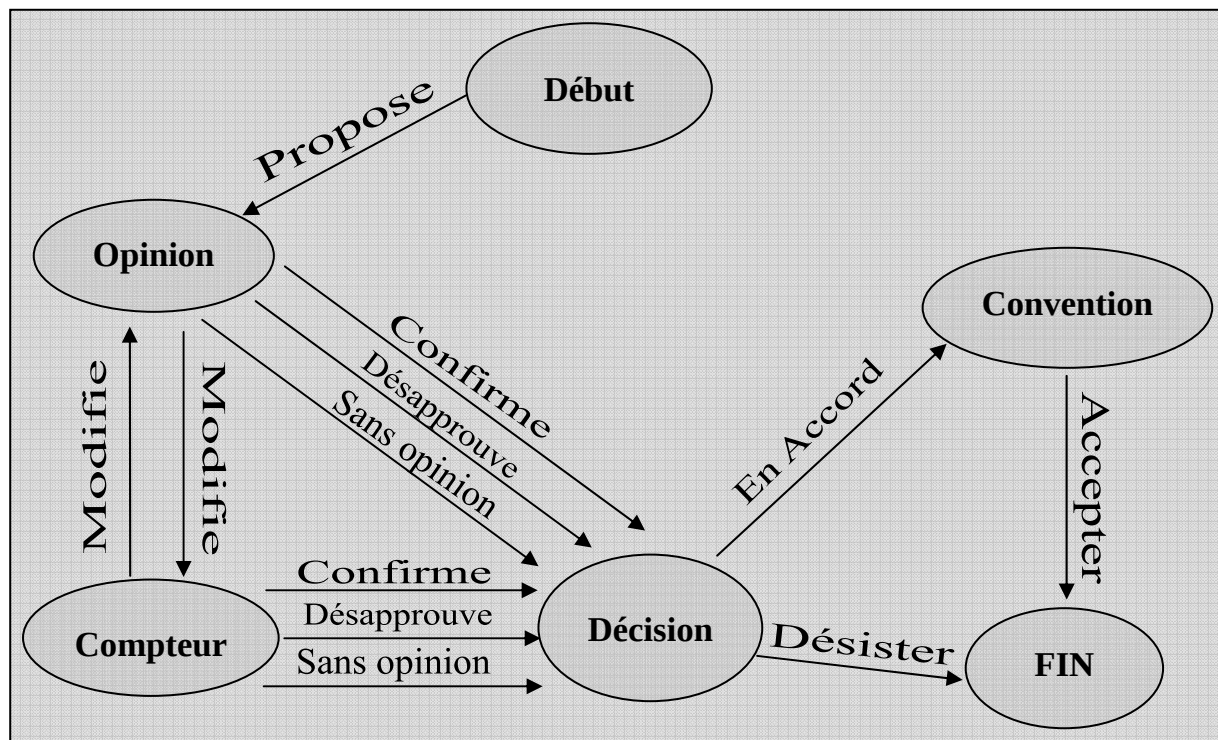


Figure 4 : Le principe du Protocole SIAN.

Les nœuds du graphe représentent les états du dialogue, les arcs représentent les transitions d'un état conversationnel à l'autre, et les branches (suite de transition) représentent des transitions possibles.

Les nœuds adjacents représentent alternativement l'agent qui a émis le premier message du protocole puis l'agent destinataire de ce message. De façon duale, pour un agent qui suit un protocole les nœuds représentent alternativement des états d'émission ou de réception de message.

5- Conclusion

D'après les travaux présentés, nous pouvons constater que le protocole de négociation peut être employé dans plusieurs domaines d'application très différents (tel que l'adaptation, la communication, les SMA). Dans chaque domaine le protocole de négociation joue un rôle important mais le but essentiel est d'arriver à la satisfaction des utilisateurs de ce protocole, en utilisant le maximum d'informations disponibles pour le bon déroulement du processus.

Chapitre II : Les Principaux Protocoles Et Outils De Communication Multimédia

1- Introduction

Les applications de vidéo ou audioconférence ou de téléphonie vont envahir le monde IP (Internet Protocol) qui jusqu'alors, ne pouvait raisonnablement pas supporter ce genre d'applications (délai de propagation de la voix important, images saccadées, ...).

Les services multimédias temps réel sur IP sont gérés par des protocoles de signalisation [NIB05], ces derniers concernent l'ensemble des informations échangées par les terminaux, les passerelles et tous les éléments assurant l'établissement, le contrôle et la rupture d'une connexion temps réel. Parmi les protocoles de signalisation, nous citons le protocole d'initiation de session **SIP** [MHE99] qui permet l'établissement, la modification et la terminaison des sessions multimédias interactives sur IP entre des terminaux, dont le rôle est de donner une description complète d'une connexion (média, application utilisé, ce qui peut être supporté, etc.). SIP est utilisé en conjonction avec un autre protocole important qui est le SDP (Session Description Protocol) [DAD04].

Dans ce qui suit, nous allons décrire les deux protocoles (SIP et SDP) qui représentent des principaux protocoles et outils de communication multimédia.

2- Le protocole d'initiation de session SIP

2.1- Origine

L'I.E.T.F. (Internet Engineering Task Force) est née en 1986 afin d'institutionnaliser les recherches internationales effectuées autour d'Internet. Elle a largement contribué à faire évoluer le réseau mondial Internet dans l'intérêt de tous. En effet, les groupes de travail qui le constituent produisent constamment des documents visant à améliorer le réseau Internet [MHE99].

Le protocole d'initiation de session (SIP) est un protocole normalisé par l'IETF (RFC 2543), appartenant à la couche application du modèle OSI [MHE99]. Le protocole SIP n'a pas percé dans la téléphonie IP face à H.323 et pourtant c'est un protocole porteur de nouveaux services réseaux, dans la téléphonie, en entreprise comme dans le monde des opérateurs [FBE03]. Il permet de simplifier la connexion à un réseau, représente une avancée significative, pour l'utilisateur distant qui cherche à intégrer un réseau virtuel d'entreprise. Ce protocole permet d'améliorer sensiblement les capacités de collaboration d'une communauté distante, telle que la sécurité des communications.

Son rôle est d'ouvrir, modifier et libérer les sessions ou appels ouverts entre un ou plusieurs utilisateurs. L'ouverture de ces sessions permet de réaliser de l'audio ou vidéoconférence, de l'enseignement à distance, de la voix (téléphonie) et de la diffusion multimédia sur IP essentiellement.

2.2- Fonctionnalités

SIP possède les fonctionnalités suivantes, [COL04] :

- Localisation du terminal appelé.
- Analyse du profil et des ressources du destinataire.
- Négociation du type de média (voix, vidéo, données,...), et des paramètres de communication.
- Disponibilité de l'appelé : 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.

3- Architecture de SIP

L'architecture en couches de SIP, telle que la présente le modèle OSI, fait apparaître une palette de nombreux protocoles, [MHE99]:

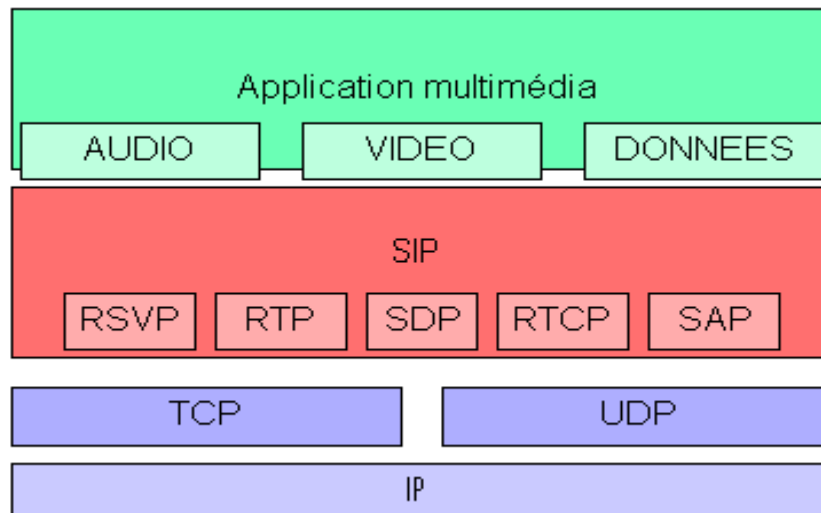


Figure 5 : L'architecture en couche du protocole SIP (Pile du protocole SIP).

A chacune des couches de l'architecture SIP sont associés des protocoles tels que :

- **RSVP (Ressource Reservation Protocol)** : est un protocole de signalisation pour allouer dynamiquement de la bande passante aux applications orientées réseaux. Il utilise la table de routage locale dans les routeurs pour déterminer les routes vers les destinations appropriées. RSVP travaille au-dessus de IP (IPv4 ou IPv6). RSVP n'est pas un protocole de routage mais travaille avec les protocoles de routage unicast et multicast. [ANA01]

RSVP peut être utilisé avec RTP/RTCP mais ce n'est pas obligatoire.

- **R.T.P. (Real-time Transport Protocol)** : fournit un moyen uniforme de transmettre sur IP des données soumises à des contraintes de temps réel (audio, vidéo,...). Son rôle principal consiste à mettre en œuvre des numéros de séquence de paquets IP pour reconstituer les informations de voix ou vidéo. Donc, RTP permet d'identifier le type de l'information transportée, d'ajouter des marqueurs temporels et des numéros de séquence à l'information transportée et de contrôler l'arrivée des paquets à destination. [NJJ06]

- **R.T.C.P. (Real-Time streaming Control Protocol)** : est basé sur des transmissions périodiques de paquets de contrôle par tous les participants dans la session. C'est un protocole de contrôle des flux RTP, permettant de véhiculer des informations basiques sur les participants d'une session, et sur la qualité de service. RTP et RTCP sont des protocoles qui se situent au niveau de la couche application et utilisent les protocoles sous-jacents de transport TCP ou UDP. [NJJ06]
- **S.A.P. (Session Announcement Protocol)** : permet aux ressources réseau, d'annoncer leurs adresses réseau et les services qu'elles fournissent. Chaque service est identifié par un numéro appelé identificateur SAP. Les mises à jour SAP sont envoyées toutes les 60 secondes. SAP est également utilisé pour écouter et émettre périodiquement des annonces SAP pour tenir à jour une table des services disponibles sur le réseau.[OLF04]
- **S.D.P. (Session Description Protocol)** : fournit un format universel pour décrire des sessions de multimédia, des annonces ou des invitations (SAP). Il est indépendant de la façon dont cette information est transportée. Il emploie un format de données entièrement textuel pour maximiser la portabilité dans les transports. Il est utilisé par différents protocoles de transport appropriés, y compris le protocole d'annonce de session SAP, le protocole de déclenchement de session SIP, le protocole des flux en temps réel RTP/RTSP, etc. Plus loin, on va voir en détails ce protocole, car c'est le protocole principal utilisé par SIP pour transmettre les types de média utilisé d'une session. [DAD04]

SIP permet l'ouverture de sessions en mode, [MHE99] :

- o Point-à-point (unicast) : communication entre 2 stations (voir figure 6.a) ;
- o Diffusif (multicast) : plusieurs utilisateurs via une unité de contrôle M.C.U. (Multipoint Control Unit) (voir figure 6.b) ;
- o Combinatoire : plusieurs utilisateurs pleinement interconnectés en multicast via un réseau à maillage complet de connexions (voir figure 6.c) ;

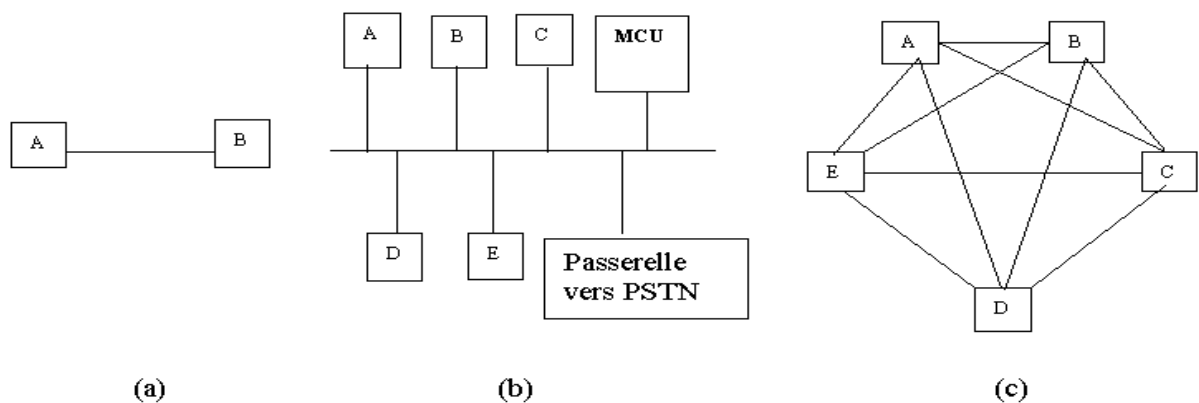


Figure 6 : Les différents types de communication avec SIP.

Remarque : Les utilisateurs reliés au Réseau Téléphonique Commuté Public (P.S.T.N. Public Switched Telephone Network) peuvent utiliser SIP car le PSTN est interconnecté au réseau des réseaux grâce à des passerelles (gateways).

Pour établir et terminer des communications multimédia, SIP utilise les cinq (5) fonctions suivantes :

- **User location** : permet de localiser le poste terminal utilisé pour communiquer ;
- **User capabilities** : détermine quels médias vont être échangés (voix, vidéo, données...) ainsi que les paramètres associés ;
- **User availability** : détermine si le poste appelé souhaite communiquer et autorise l'appelant à la contacter ;
- **Call setup** ou " **ringing** " : avertit les parties appelant et appelé de la demande d'ouverture de session (sonnerie ou message de réception d'appel) et mise en place des paramètres d'appel.
- **Call handling** : gère le transfert et la fermeture des appels.

4- Les composants d'un système SIP

Le principal objectif du protocole SIP est de mettre en communication plusieurs utilisateurs (qu'on appelle des agents), et ceci via un ensemble de serveurs dont chacun a son rôle, d'où SIP comprend deux types de composants : les «**User Agent** » (Agents utilisateurs) et les «**Serveurs** » [MHE99].

4.1- Les User Agent

Le premier composant SIP représente l'application de l'utilisateur final, par exemple :

- un terminal de téléphonie ou de visioconférence sur IP,
- un serveur audio ou vidéo ou encore une passerelle vers un autre protocole.

Ce type de composant, joue deux rôles :

- le premier est celui d'un client, appelé "User Agent Client (UAC)", lorsqu'il envoie des requêtes SIP,
- et le second est celui d'un serveur, appelé "User Agent Server (UAS)", lorsqu'il reçoit des requêtes SIP.

La figure suivante représente l'établissement de session entre deux agents (UA) :

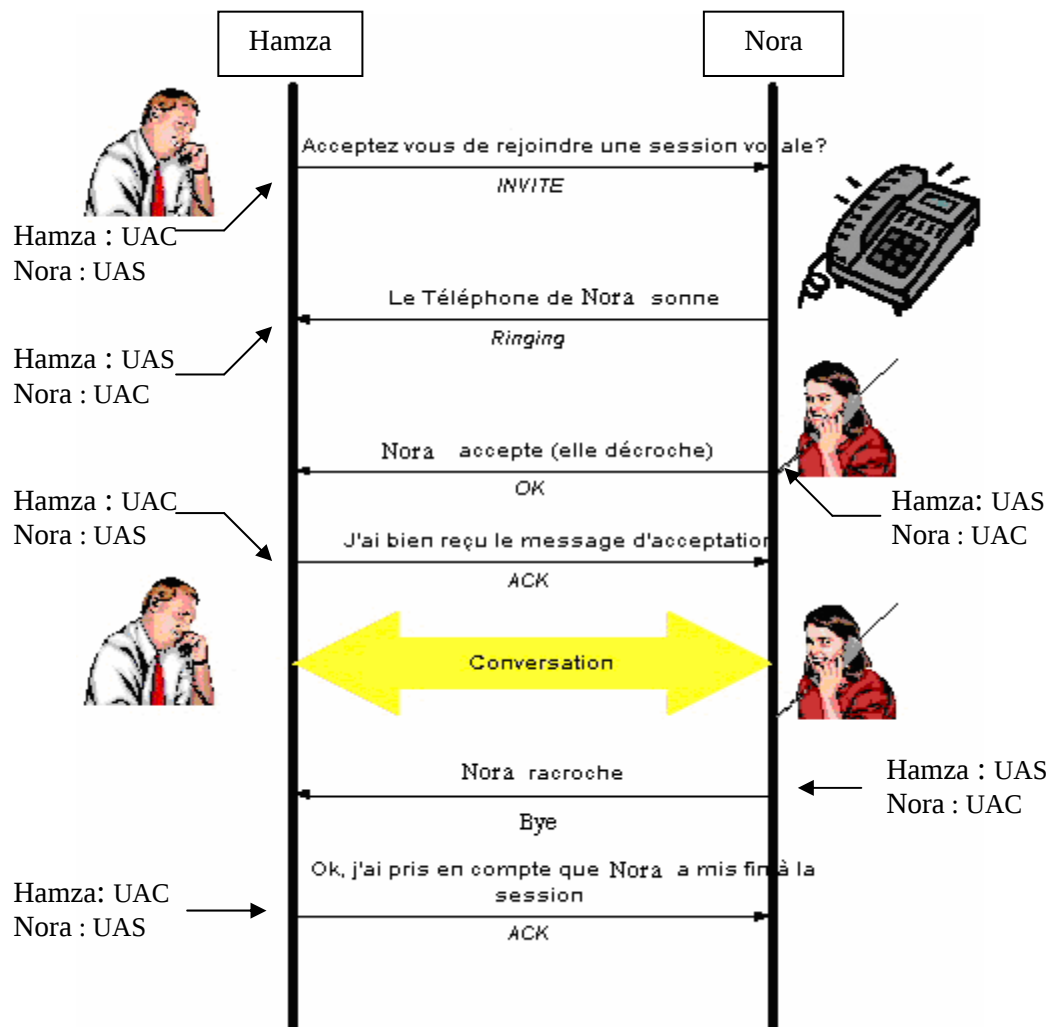


Figure 7 : Exemple d'établissement d'une session SIP entre deux User Agents.

Cette figure représente les échanges qui ont lieu entre deux UA pour l'établissement d'une connexion. Ce cas est le plus simple car les deux UA connaissent l'adresse IP de leur interlocuteur c'est pourquoi ils peuvent se joindre directement.

Dans le cas contraire, pour joindre un autre utilisateur, il est nécessaire de connaître son adresse SIP. Les utilisateurs, dans un environnement SIP sont identifiés par des URLs (Uniform Resource Locators). (Voir la section 6)

4.1.1- Scénario de comportement d'un UAC et un UAS :

L'établissement de session ou la communication entre deux utilisateurs se fait par échange de requête, par exemple : Lorsqu'un UAC appelant souhaite initier un appel, il émet une requête INVITE dont les champs d'en-tête To (ou la requête URI), From et Contact (éventuellement) contiennent respectivement l'adresse de l'appelé, l'adresse de l'appelant et l'adresse où l'UAC souhaite être contacté pour des transactions en provenance de l'appelé.

Lorsqu'un UAS appelé reçoit un appel, il peut l'accepter, le refuser ou le rediriger et émet une réponse vers l'appelant dans tous les cas. Cette réponse doit recopier les champs TO, From, CALL-ID, Cseq et Via de la requête.

4.2- Serveur Proxy

Le relais mandataire ou P.S. (Proxy Server) : auquel est relié un terminal fixe ou mobile (lors de son déplacement, le terminal est relié au PS le plus proche et change constamment de PS) agit à la fois comme client et serveur.

Un serveur proxy a la charge de router les messages SIP. Il a uniquement un rôle dans la signalisation et il ne gère pas de media. Un tel serveur peut interpréter et modifier les messages qu'il reçoit avant de les retransmettre.

Un PS peut prendre 2 états : **stateful** ou **stateless**. Un PS stateful mémorise l'association entre la requête entrante et les requêtes sortantes associées alors qu'un PS stateless est sans mémoire une fois que les requêtes sortantes sont générées.

Un type particulier de serveur proxy, appelé serveur «forking proxy », peut dupliquer une requête et l'envoyer vers plusieurs destinataires. Ces serveurs peuvent par exemple être utilisés pour contacter plusieurs terminaux appartenant à la même personne [MHE99].

4.3- Serveur Registrar

Afin de pouvoir joindre une personne à partir de son adresse SIP, une entité dans le réseau doit maintenir une correspondance (mapping) entre les adresses IP et les adresses SIP : c'est le rôle du Serveur Registrar [MHE99]. Un utilisateur peut s'inscrire auprès du Registrar Server via une requête Registrar en lui indiquant son adresse SIP et son adresse de machine sur le réseau. C'est un serveur qui accepte les **requêtes REGISTER** et offre également un service de localisation. Chaque serveur proxy ou serveur redirect est généralement relié à un Register.

4.4- Location Serveur

C'est un serveur qui permet de localiser la position courante d'un utilisateur, demandé par une communication, qui a traversé le proxy serveur et le serveur registrar auxquels il est rattaché [MHE99].

Donc lorsqu'un serveur proxy recevra un message INVITE destiné à un utilisateur X, le serveur de localisation lui indiquera l'adresse IP de X et le serveur proxy routera le message vers cette adresse IP.

Généralement dans les usages, le serveur Registrar est souvent associé au serveur Proxy et que le Location Server est le plus souvent utilisé comme une entité logique.

4.5- Serveur Redirect

Les Serveurs Redirect aident à localiser les User Agent SIP en fournissant une adresse alternative à laquelle l'utilisateur appelé peut être joint. Il réalise simplement une association (**Mapping**) d'adresse vers une ou plusieurs nouvelles adresses, en générant une réponse de redirection contenant la ou les prochaines adresses à contacter pour joindre le destinataire final [MHE99].

Les **Redirect Servers** (RS) ne prennent en charge aucune requête eux-même. Après avoir reçu une requête autre que CANCEL, le RS interroge une liste de positions possibles et retourne à l'émetteur de la requête une réponse 3xx (redirection) ou refuse la requête. Cette réponse met fin à la transaction SIP.

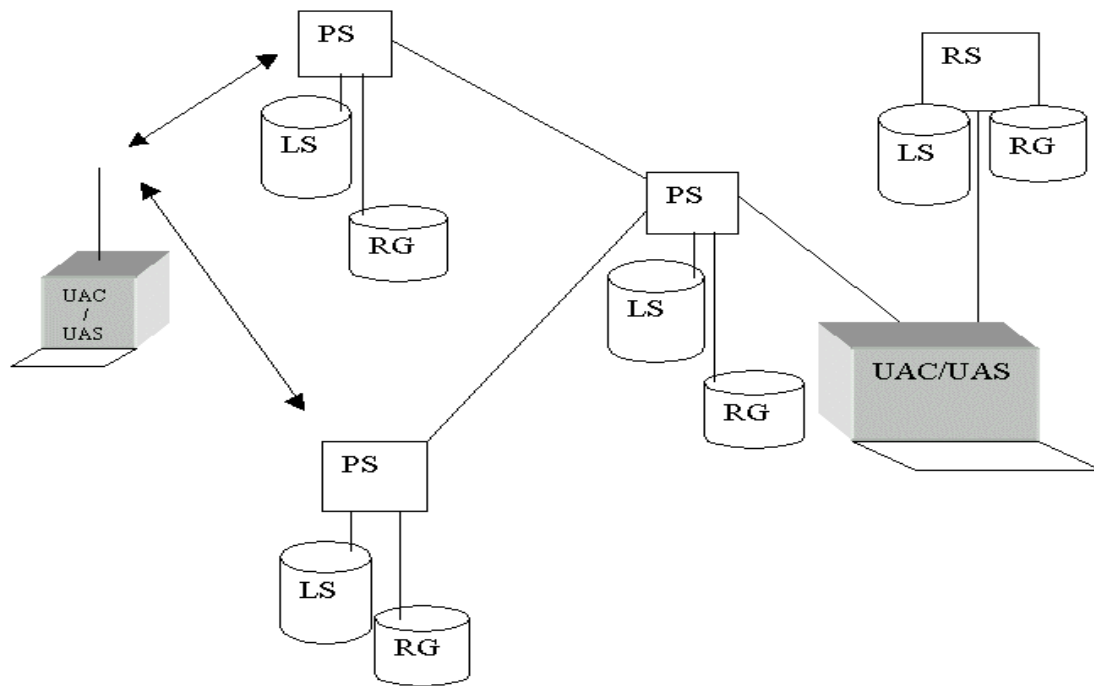


Figure 8 : Les différents composants d'un système SIP.

5- Ouverture d'une session

Lorsqu'un utilisateur utilise une application dont le but est de communiquer avec d'autres utilisateurs, ceci fait appel au protocole SIP en lui précisant la nature d'échange, ce qui permet à SIP de choisir parmi les cinq protocoles de son architecture, les mieux adaptés à ces échanges et détermine le nombre de sessions à ouvrir [MHE99].

Pour ouvrir une session, l'utilisateur émet une invitation transportant un descripteur de session permettant aux utilisateurs souhaitant communiquer de s'accorder sur la compatibilité de leur média. Ainsi SIP peut relier des stations mobiles en transmettant ou redirigeant les requêtes vers la position courante de la station appelée.

Par exemple ; pour véhiculer de la vidéo, deux (2) sessions doivent être ouvertes : l'une pour l'image et l'autre pour le son. Et pour un échange en unicast, quatre (4) sessions seront ouvertes : deux (2) pour la communication vidéo montante et deux (2) pour la descendante.

5.1- Localisation d'un utilisateur SIP :

Un terminal utilisateur peut constamment se déplacer, de ce fait sa position doit être enregistrée dynamiquement par un serveur de localisation. Ce serveur enregistre plusieurs positions pour un même terminal, qui est relié à plusieurs PS à la fois lorsqu'il se déplace (les PS les plus proches).

Lorsqu'un serveur SIP interroge son serveur de localisation, il établit une liste des positions possibles de l'utilisateur à partir des résultats reçus. Cette liste contient 0 position ou plus. Pour que le serveur de localisation communique la nouvelle position d'un utilisateur, au serveur SIP, le terminal client de l'utilisateur lui envoie une **requête REGISTER**.

Pour établir la communication entre l'UAC du terminal appelant et l'UAS du terminal appelé, ceux-ci doivent être identifiés. Chaque utilisateur et sa machine sont identifiés par une adresse appelée URL SIP qui se présente comme une URL Mailto ou Telnet : utilisateur@machine.

Où :

- La partie utilisateur : est composée d'un nom ou d'un numéro de téléphone;
 - La partie machine : est un nom de domaine ou une adresse numérique de réseau.
- (voir URL SIP en détaille dans la section 6).

Dans la plupart des cas, une adresse SIP est obtenue à partir de l'adresse e-mail.

5.2- Localisation d'un serveur SIP :

Pour localiser un serveur SIP, le client envoie une requête URI au serveur proxy le plus proche. Le client envoie une requête URI directement au destinataire lorsqu'il connaît son adresse, en précisant le protocole utilisé (TCP ou UDP), l'adresse IP et le port d'entrée du serveur du destinataire. Dans le cas contraire, si le port d'entrée du serveur n'est pas précisé, c'est le numéro 5060 utilisé par défaut.

Lorsque le serveur destinataire reçoit une requête, celui-ci la redirige vers la couche transport TCP ou UDP de son architecture. Si la requête URI ne précise aucun protocole, UDP est utilisé par défaut, puis TCP en cas d'échec. Si le serveur n'est pas joignable, un message ICMP est envoyé au client pour l'en informer (Dans le cas d'applications nécessitant un transport en temps-réel, UDP est indispensable, par contre, pour un transport fiable de données, TCP sera utilisé).

En cas d'échec de localisation du serveur SIP distant, le client utilise D.N.S (Domain Name Server) pour obtenir l'URL SIP d'un autre serveur SIP.

Une fois le client connecté à un serveur SIP distant, il peut lui adresser une ou plusieurs requêtes SIP et recevoir une ou plusieurs réponses de ce serveur. Les réponses contiennent certains champs identiquement remplis à ceux des requêtes. Dans le cas d'un transport TCP et d'une transaction simple (requête + réponse), requête et réponse SIP sont transportées par la même connexion TCP.

Si plusieurs requêtes émanent d'un même client à destination d'un même serveur (unicast), soit une seule connexion est ouverte, soit une connexion pour chaque requête est ouverte. Dans le cas d'un transport UDP unicast, la réponse à la requête d'un client est retournée à l'adresse contenue dans le champ d'en-tête via la réponse. Si la requête est transportée par UDP vers plusieurs serveurs (multicast), la réponse est dirigée vers la même adresse de multicast et le même port de destination.

En résumé, voici les différents éléments intervenant dans l'ouverture d'une session :

- Le choix des protocoles les mieux adaptés (RSVP, RTP, SDP, ...) suivant la nature des échanges.
- Détermination du nombre de sessions.
- Chaque utilisateur et sa machine sont identifiés par une URL SIP.
- Requête URI : permet de localiser le Proxy Server auquel est rattachée la machine de l'appelé.
- Requête SIP : une fois le client (machine appelante) connecté à un serveur SIP distant, il peut lui adresser une ou plusieurs requêtes SIP et recevoir une ou plusieurs réponses de ce serveur.

5.3- Exemples de transactions :

Pour l'ouverture d'une session, on doit faire appel à SIP, l'application de l'UAC appelant envoie une requête INVITE au Proxy Server (PS) auquel il est relié. Ce serveur, via d'autres PS, transmet cette requête à l'UAS auquel est relié l'appelé. Cette requête demande à l'appelé s'il veut rejoindre un forum de discussion, assister à une visioconférence ou établir simplement une communication privée avec l'appelant. Si l'appelé est d'accord, il renvoie une réponse OK (code 200) à l'appelant qui confirme alors qu'il a bien reçu la réponse de l'appelant. Pour cela, il envoie une requête ACK, acquittement (acknowledgement) à l'appelé. De la même manière, si l'utilisateur souhaite se déconnecter, l'application de l'utilisateur émet une requête BYE au lieu de ACK;

Nous distinguons deux façons différentes pour l'établissement de cette session entre les participants et ceci selon le type du serveur, [COL04] :

a) Par l'intermédiaire d'un Proxy Server :

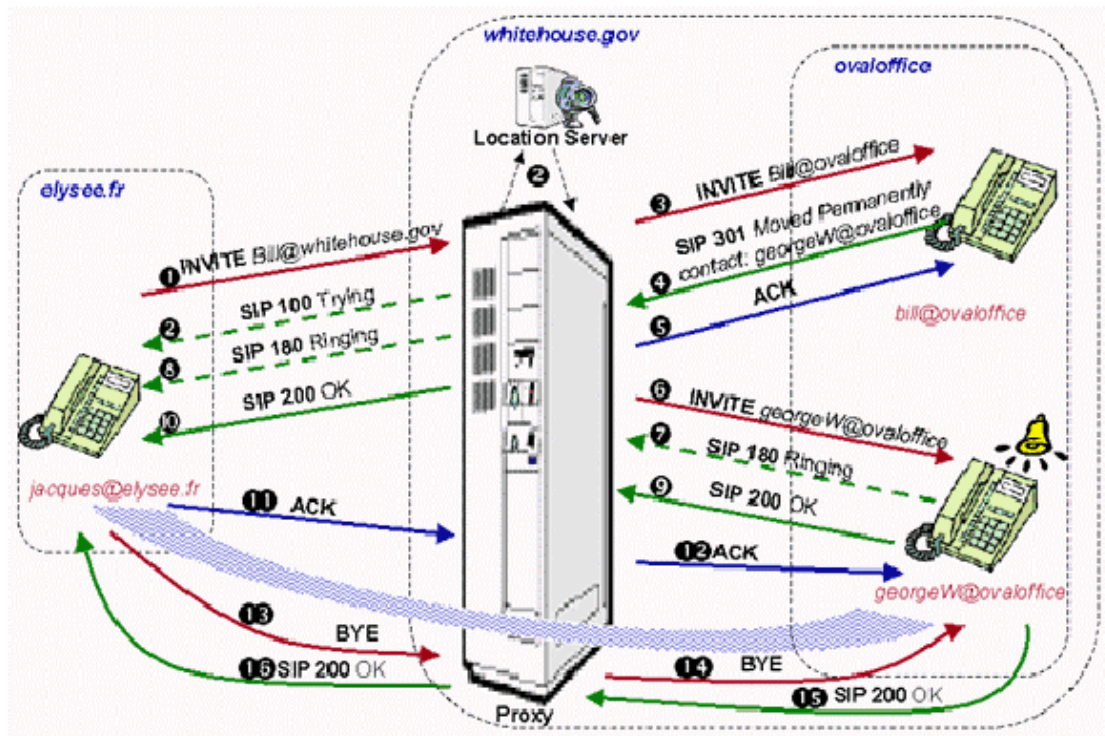


Figure 9 : Etablissement de session à travers un serveur Proxy.

1. L'application cliente du terminal de Jacques va émettre une requête SIP INVITE au serveur Proxy dont l'URL SIP spécifiée est : « **Bill@whitehouse.gov** ». Le serveur Proxy l'accepte.
2. Jacques reçoit une réponse (code 100) trying de la part du proxy distant, ce qui signifie que celui-ci cherche à localiser l'appelé ; le serveur SIP, selon le champ destination consulte sa propre base de données pour vérifier si l'enregistrement de l'utilisateur existe déjà. Il peut aussi pendant ce temps, chercher la location de Bill dans un **Location server**.
3. Le serveur Proxy obtient une localisation précise de Bill et transmet le requête INVITE de Jacques au terminal de BILL.
4. Le terminal de Bill va répondre au Proxy un message (code 301 : Moved Permanently) signifiant un départ permanent de l'appelé, et donne l'adresse d'un appelé alternatif, dans notre cas, georgeW@ovaloffice.gov.
5. Le Proxy acquitte l'information.
6. Le Proxy réémet la requête de Jacques (qui attend toujours) vers le terminal de GeorgeW.

- 7.8** Le terminal de GeorgeW sonne, et en notifie le Proxy, qui transmet l'information à Jaques.
- 9.10** Lorsque George décroche, son terminal informe le Proxy du succès de la requête INVITE, et le Proxy transmet l'information au terminal de Jaques.
- 11.12** Le terminal appelant (Jaques) acquitte l'information au terminal de l'appelé (George) par le biais du Proxy. La communication s'établit alors directement.
- 13 à 16** La terminaison de la session se déroule par la transmission d'une requête invoquant la méthode BYE.

b) Par l'intermédiaire d'un serveur de redirection :

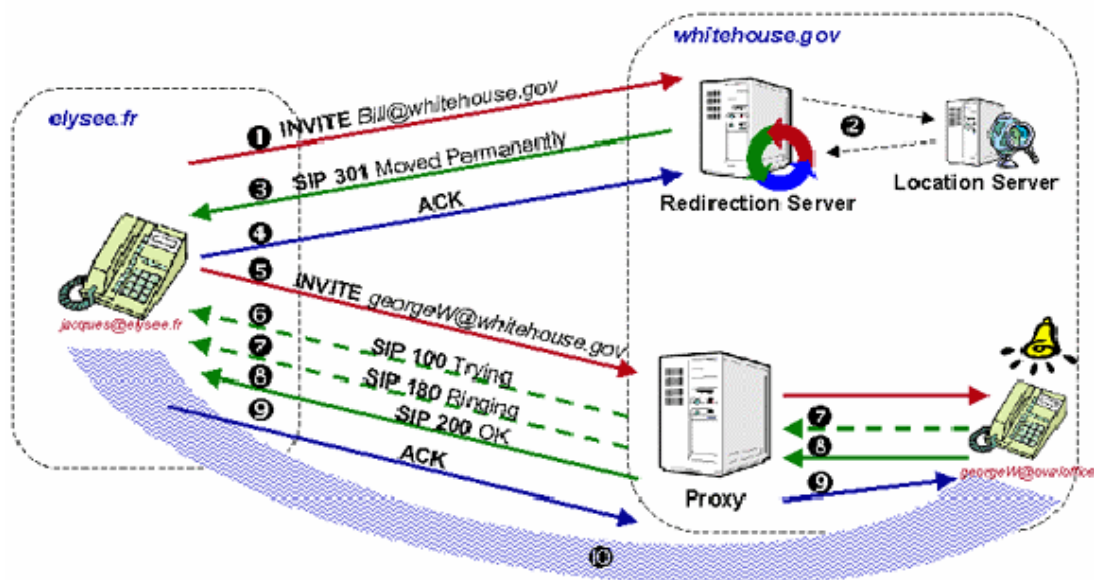


Figure 10 : Etablissement de session à travers un Redirect Server.

1. L'application cliente du terminal de Jacques va émettre une requête SIP **INVITE** au Redirect serveur dont l'URL SIP spécifiée est : « Bill@whitehouse.gov ».
- 2.3 Après une éventuelle consultation du Location serveur, le Redirect serveur retourne à Jacques un message (code 301) signifiant un départ permanent de l'appelé, et donne l'adresse d'un appelé alternatif : « GeorgeW@ovaloffice ».
4. Le terminal appelant (UAC) acquitte l'information.

- 5.6 Cette fois-ci, c'est le terminal qui va réémettre la requête de Jacques (qui attend toujours) vers le terminal de GeorgeW. La requête est relayée par un Proxy distant, qui notifie l'appelant à l'aide d'un message (code 100) de la nature de cette tentative.
7. Le terminal de GeorgeW sonne, et en notifie le Proxy, qui transmet l'information à Jacques.
8. Lorsque George décroche, son terminal informe le Proxy du succès de la requête INVITE, et le Proxy transmet l'information au terminal de Jacques.
9. Le terminal appelant (Jacques) acquitte l'information au terminal de l'appelé (George) par le biais du Proxy.
10. La communication s'établit alors directement entre les deux protagonistes.

La terminaison de la session se déroule de la même manière que dans l'exemple précédent.

A travers ces deux (02) exemples, nous remarquons que si certains paramètres de la session doivent être changés, un nouveau INVITE est émis tout en conservant le Call-ID mais un Cseq plus grand doit être utilisé.

6- URL (Uniform Resource Locators) SIP

Les messages SIP sont basés texte (caractères standards de l'ISO 10646 et codage UTF-8 du RFC 2279) c'est-à-dire que les émetteurs doivent terminer une ligne de message par un C.R.L.F -Carriage-Return Line-Feed (retour chariot)- et les récepteurs les interprètent comme tels, ce qui permet de les écrire en langage Java, Tcl et Perl [MHE99].

Les messages SIP doivent être accompagnés de :

- URL SIP de l'expéditeur représenté par l'URL **FROM**,
- URL de la destination courante, car elle peut changer si l'appelé est mobile, représenté par la **Requête URI**,
- URL du destinataire : représenté par l'URL **TO**,
- URL de redirection exprimé par l'URL **CONTACT**, et l'URL **EXTERNAL**.

URL FROM	URL COURANTE (REQUETE URI)	URL TO	URL CONTACT	URL EXTERNAL
----------	----------------------------	--------	-------------	--------------

Donc, il existe cinq types d'URL SIP (FROM, COURANTE, TO, CONTACT, EXTERNAL) dont la structure est la suivante :

SIP : <userinfo>@ <host> : <port> ; <URL-parameters>; Où :

- **<userinfo>** : Désigne un nom d'utilisateur et éventuellement un mot de passe ce qui donne le format <user : pass word>, mais peut être aussi un numéro de téléphone si le paramètre user = phone.
- **<host>** : Est le nom de domaine d'un hôte du réseau ou une adresse IP.
- **<port>** : Le numéro de port utilisé pour recevoir les requêtes. S'il n'est pas présent, le client utilise le numéro " 5060 " par défaut.
- **<URL-parameters>** : définissent des paramètres spécifiques de la requête qui sont :
 - Transport : (UDP ou TCP) détermine le mécanisme de transport à utiliser ; si le paramètre transport n'est pas présent on utilise le protocole UDP.
 - User : (Phone ou IP) si (user = phone) il indique que l'hôte est une passerelle qui relie un réseau téléphonique et que le nom d'utilisateur est un numéro de téléphone.
 - Maddr : (adresse IP multicast) ce paramètre fournit l'adresse multicast du serveur à contacter pour cet utilisateur.
 - Ttl (Time to live) : de 0 à 255, désigne le temps de vie d'un paquet UDP en multicast, l'unité étant le nombre de serveurs traversés.
 - Tag : compteur en Hexadécimale.
 - Méthode : la méthode utilisée. (voir la section 7.3)

Seules les URL CONTACT et EXTERNAL contiennent les paramètres transport, method, ttl, maddr et les en-têtes. Si un autre type d'URL contient ces paramètres, ceux-ci sont ignorés.

Le paramètre **tag** est utile dans l'URL TO et permet de désigner de nouvelles options dans SIP. Une nouvelle option doit être enregistrée à l'**I.A.N.A.**(Internet Assigned Numbers Authority).

Afin de mieux visualiser les URL SIP, voici quelques exemples :

- Pour contacter Nawa@USTHB.dz en multicast avec l'adresse IP 239.255.255.1 et un temps de vie de 15, l'URL SIP suivante sera utilisée :
sip : Nawa@USTHB.dz; maddr=239.255.255.1 ; ttl=15
- sip : Nawa : mot_de_passe@USTHB.dz
- sip : +1-212-555-1212 :1234@passerelle.fr ;user=phone
- sip : Nawa@registrar.fr ; method=REGISTER.

Dans ce qui suit, nous citons quelques caractéristiques de chaque URL, [MHE99] :

- **URL TO** : contient l'adresse dont l'enregistrement est en cours ou prédaté.
- **URL FROM** : contient l'adresse d'enregistrement de l'autorité qui en est responsable. Lors du premier enregistrement, elle est identique à celle de l'URL TO.
- **URL COURANTE** : la **requête URI** précise la destination de la requête REGISTER c'est-à-dire son domaine d'enregistrement. La requête REGISTER est transmise de serveur en serveur jusqu'à ce qu'elle ait atteint le serveur dont le domaine correspond à celui listé dans la requête URI.
- **URL CONTACT** : les requêtes autres que REGISTER à destination de l'URL TO sont redirigées aux adresses données dans l'URL CONTACT. Si la requête ne possède pas de champ d'URL CONTACT, alors l'enregistrement reste inchangé. Par contre, lorsque le champ d'URL CONTACT existe, il est utile pour obtenir la liste courante des enregistrements dans la réponse. Un client peut annuler un enregistrement en envoyant une requête REGISTER avec un "ttl" nul.
- **URL EXTERNAL** : réservée à des applications futures.

7- Format des messages SIP

Pour transporter plusieurs transactions à la fois, SIP peut utiliser une simple connexion TCP (mode flux) ou des datagrammes UDP (mode bloc). Seulement, les datagrammes UDP, y compris les en-têtes, ne doivent pas excéder une certaine longueur (M.T.U. Maximum Transmission Unit). Si la MTU est inconnue, elle est de 1500 octets par défaut. Cette taille permet l'encapsulation des datagrammes UDP ou segments TCP dans des paquets IP sans fragmentation [MHE99].

Un message SIP peut être à la fois **une requête** d'un client vers un serveur ou **une réponse** d'un serveur vers un client [COL04]. Ces deux types de messages SIP utilisent le format suivant :

Ligne de requête (Méthode, Requête URI, version SIP)
En-tête général ou de requête ou d'entité
CRLF *(permet de spécifier la fin du champ d'en-têtes, et le début du corps du message)
Corps du message

Requête d'un client vers un serveur

Ligne d'état (version SIP, code d'état, Reason Phrases)
En-tête général ou de réponse ou d'entité
CRLF (permet de spécifier la fin du champ d'en-têtes, et le début du corps du message)
Corps du message

Requête d'un serveur vers un client.

7.1- Ligne de requêtes

Les échanges entre un terminal appelant et un terminal appelé se font par l'intermédiaire de requêtes [MHE99]. La ligne de requête distingue un message de requête et suit le format suivant :

Méthode [SP] Requete-URI [SP] Version-SIP [CRLF]

- **Méthode** : Il existe six méthodes qui peuvent être utilisées dans une requête SIP, qui sont : INVITE, ACK, CANCEL, OPTION, BYE, REGISTER
- **Requête-URI** : La requête URI est une requête URL SIP ou une URI général indiquant l'utilisateur ou le service à qui cette requête est adressée.
- **Version-SIP** : La version actuelle du protocole SIP est la SIP/2.0

* **SP** : est un caractère qui sert de séparateur.

* **CRLF** : Il correspond à un retour chariot suivi d'un saut de ligne qui est une balise pour indiquer la fin des en-têtes et le début du corps du message.

7.2- Ligne de réponse (d'état)

Un message de réponse est caractérisé par la ligne de réponse qui a le format suivant :

Version-SIP [SP] Code d'état [SP] Reason Phrases[CRLF]

- **Code d'état** (Statut du code) : est un entier sur trois bits indiquant un résultat à l'issue de la réception d'une requête. Le code d'état est destiné à l'automate gérant l'établissement des sessions SIP.
- **Reason Phrases** (Phrases de reason) : introduite pour donner un petit texte explicatif du code d'état qui correspond au motif du refus ou de l'acceptation d'une requête destiné au programmeurs.

Le premier bit indique la classe de la réponse, il existe six classes de réponses et donc de codes d'état, représentés ainsi :

- **1xx : Informational** (informationnelle) : la requête est reçue et va être analysée.
Par exemple : le code d'état 180 représente le « ringing ».
- **2xx : Success** (Succès) : la requête a été reçue, comprise et acceptée, exprimé par le OK qui a le code 200.
- **3xx : Redirection** : d'autres actions doivent être menées pour valider la requête.
Par exemple, le code 300 indique qu'il y a plusieurs choix, et le code 380 exprime qu'il existe des services alternatifs.
- **4xx : Client Error** (erreur client) : la requête contient une erreur syntaxique ou ne peut pas être traitée par ce serveur. Le code 406, par exemple, signifie que l'erreur n'est pas acceptée.
- **5xx : Server Error** (erreur serveur) : le serveur ne peut traiter une requête apparemment correcte. Le code 505 indique que la version SIP citée n'est pas supportée.
- **6xx : Global failure** (échec global) : la requête ne peut être traitée par aucun serveur. Par exemple : le code d'état 606 : indique que la requête n'est pas acceptée.

Les codes de réponses sont basés sur les six types de réponses et peuvent être facilement étendus car peu de codes possibles sont définis (voir Annexe A). Il est nécessaire que les applications déterminent la classe (x) d'un code de réponse qui est indiqué par le premier chiffre, et traitent toute réponse non reconnue comme équivalente à la première réponse de la classe (le code x00).

La version SIP.02 possède de nombreux codes en commun avec la version HTTP/1.1. Les nouveaux codes ajoutés par SIP commencent à x80 pour éviter tout conflit avec HTTP. De plus SIP apporte une nouvelle classe 6xx de codes d'état. Les codes d'état des réponses SIP sont donc extensibles.

7.3- Les méthodes

Les échanges client-serveur se font à l'aide de requêtes. Plus les requêtes **INVITE**, **ACK**, **BYE**, **REGISTER**, **OPTIONS** et **CANCEL**, la requête URI constitue un type à part de requête que nous détaillerons par la suite [MHE99].

Les méthodes non supportées par un PS ou un RS sont traitées en tant que méthodes **OPTIONS**. Les méthodes non supportées par un UAS ou RG sont ignorées.

Dans ce qui suit, nous étudions en détails ces six (6) méthodes :

- **INVITE** : indique que l'application ou l'utilisateur est invité à participer à une session, dont le **Corps du message** décrit cette session (média supportés par l'appelant, par exemple). En cas de réponse favorable à l'invitation, l'invité doit spécifier également les médias qu'il supporte dans son Corps du message.

Le serveur chargé de répondre à une telle invitation, est identifié par son CALL-ID. Si un UAS reçoit une requête **INVITE** avec un numéro de séquence Cseq supérieur à celui de la dernière requête **INVITE** de même CALL-ID reçue, celui-ci doit mettre à jour cette dernière. Cette méthode est supportée par les PS, RS, UAS et UAC.

- **ACK** : confirme que le client a reçu une réponse définitive à une requête **INVITE**. La requête **ACK** possède, dans son Corps de message, une description définitive de la session que doit utiliser l'appelé. Si le Corps du message est vide (car il est optionnel), l'appelé utilise le descripteur de session de la requête **INVITE**. Un PS doit toujours déterminer si la requête **ACK** lui est destinée ou si elle est destinée à un autre PS. Cette détermination se fait en examinant le paramètre tag de l'URL TO. Le **ACK** lui est destiné si :

- 1) Le tag de l'URL TO de la requête **ACK** est égal au tag de l'URL TO de la réponse ;
- 2) L'URL FROM, le CALL-ID et le Cseq de la requête **ACK** sont égaux à ceux de la réponse.

Cette méthode est supportée par les PS, RS, UAS et UAC.

- **OPTIONS** : un PS en mesure de contacter l’UAS appelé, doit répondre à une requête OPTIONS en précisant ses capacités à contacter l’UAS. Cette méthode est supportée par les PS, RS, UAS, RG et UAC.
- **BYE** : est utilisée par l’UAS de l’appelé pour signaler au PS local qu’il ne souhaite plus participer à la session. Si la requête INVITE contient une URL CONTACT, l’appelé envoie la requête BYE directement à cette adresse plutôt qu’à l’URL FROM. Cette méthode est supportée par les PS et devrait être supportée par les RS et UAS.
- **CANCEL** : la requête CANCEL est envoyée par un UAC ou un PS pour annuler une requête non validée par une réponse finale d’état. Par exemple, un PS peut envoyer une requête CANCEL aux destinataires n’ayant pas retourné une réponse finale après avoir reçu une réponse du code 2xx ou 6xx du PS. Pour distinguer une réponse à une requête CANCEL et une réponse à la requête originale, l’en-tête général est de type Cseq dans le format du message de la requête CANCEL. Cette méthode est supportée par le PS et par tous les autres types de serveurs SIP.
- **REGISTER** : cette méthode est utilisée par le client pour enregistrer son adresse SIP listée dans l’URL TO par le serveur auquel il est relié. L’**adresse d’enregistrement** du client est définie sous le format suivant :

[Utilisateur@Domaine](#)

Cette méthode assure un service de localisation.

7.4- Champs d’en-tête

Il existe quatre (4) types d'en-tête utilisé dans les usages SIP, qui sont :

- **En-têtes général** : (General header) il est appliqué au message de requête et de réponse.
- **En-têtes d’entité** : (Entity headers) définit le type d’informations contenu dans le corps du message ou des ressources identifiées par la requête si le corps du message est absent.
- **En-têtes de requête** : (Request headers) autorise le client à ajouter des informations additionnelles concernant sa requête et même la destination du serveur.
- **En-têtes de réponse** : (Response headers) permettent aux serveurs d’ajouter des informations supplémentaires sur les réponses, qui ne peuvent pas être placées dans la ligne d’état de la réponse, ces informations concernent la réponse et l’accès aux ressources identifiées dans la requête URI.

Les différents champs d'en-tête qu'utilise SIP [MHE99] ne nécessitent pas d'ordre particulier sauf dans le cas de l'en-tête général **Via** où l'ordre des champs est important. Nous distinguons deux types de champs d'en-tête :

- Les champs d'en-tête transmis **saut par saut** : ce sont les champs qui peuvent être traités ou modifiés par tous les serveurs qu'ils traversent.
- Les champs d'en-tête transmis de **bout en bout** : c'est des champs qui sont interprétés par les émetteurs et les destinataires uniquement et ne sont pas modifiés par les serveurs qu'ils traversent.

Les champs d'en-tête transmis saut par saut doivent apparaître avant les champs d'en-tête transmis de bout en bout. Les serveurs Proxy ne doivent pas réordonner les champs d'en-tête mais peuvent ajouter éventuellement des champs **Via** ou autres champs de type saut par saut. Pour voir en détail ces champs (voir l'Annexe A), dans ce qui suit nous citons quelques champs d'en-têtes [LAK01] :

En-têtes	Champ	Signification
En-tête général	- Call-Id - Date - From - To - Via	- Identifie une invitation précise. - Date d'émission de la requête ou de la réponse. - Indique la personne à l'origine du message - indique l'adresse du récepteur - indique le chemin traversé par le message
En-tête d'entité	- Content-Length - Content-Type	- Taille du corps du message envoyé. - Indique les types de média utilisés dans le corps du message.
En-tête de requête	- Priority - Max-Forwards	- Indique le niveau de priorité de la requête. - Sert à limiter le nombre de PS traversé par la requête jusqu'à l'UAS de l'UAC.
En-têtes de réponse	- Allow - Retry-After	- Indique les méthodes valides supportées par les entités identifiées par la requête URI. - Indique à l'émetteur de la requête quand est – ce un service peut répondre.

Tableau 1 : Champs d'en-têtes.

Remarque : Les champs non identifiés sont traités comme des champs d'entité.

L'emploi de ces champs dépend de la méthode utilisée. Par exemple, les champs d'en-tête **Call-Id**, **CSeq**, **From**, **To** et **Via** sont requis par toutes les méthodes, sauf dans le cas de la méthode **OPTIONS**, où il faut ajouter en plus le **champ d'en-tête Allow**). Ces champs d'en-tête sont de type "de bout en bout". Alors que le champ d'en-tête **Expires** n'est utilisé que par la méthode **REGISTER**.

7.5- Le corps du message

Dans le cas des requêtes, un Corps de message est ajouté ou non selon la méthode utilisée, représentées ainsi :

Méthode	Contenu du Corps du message
ACK	Description de session
INVITE	Description de session
OPTIONS	Description de session
BYE	Pas de Corps de message
REGISTER	Réservé pour le futur
CANCEL	Non précisé

Dans le **cas des réponses**, le **Corps du message est obligatoire** et son type et l'interprétation dépendent de la méthode et des codes d'état utilisés.

Le **Type** du Corps du message doit être donné par le champ d'en-tête **Content-Type**. Si le Corps du message a subi un codage tel que la **compression** (JPEG, GIF, MPEG...), cela doit être indiqué par le champ d'en-tête **Content-Encoding** (sinon ce champ doit être omis). La **Longueur du message** (en octets) est donnée par le champ d'en-tête **Content-Length**.

La figure suivante décrit la structure d'un message INVITE :

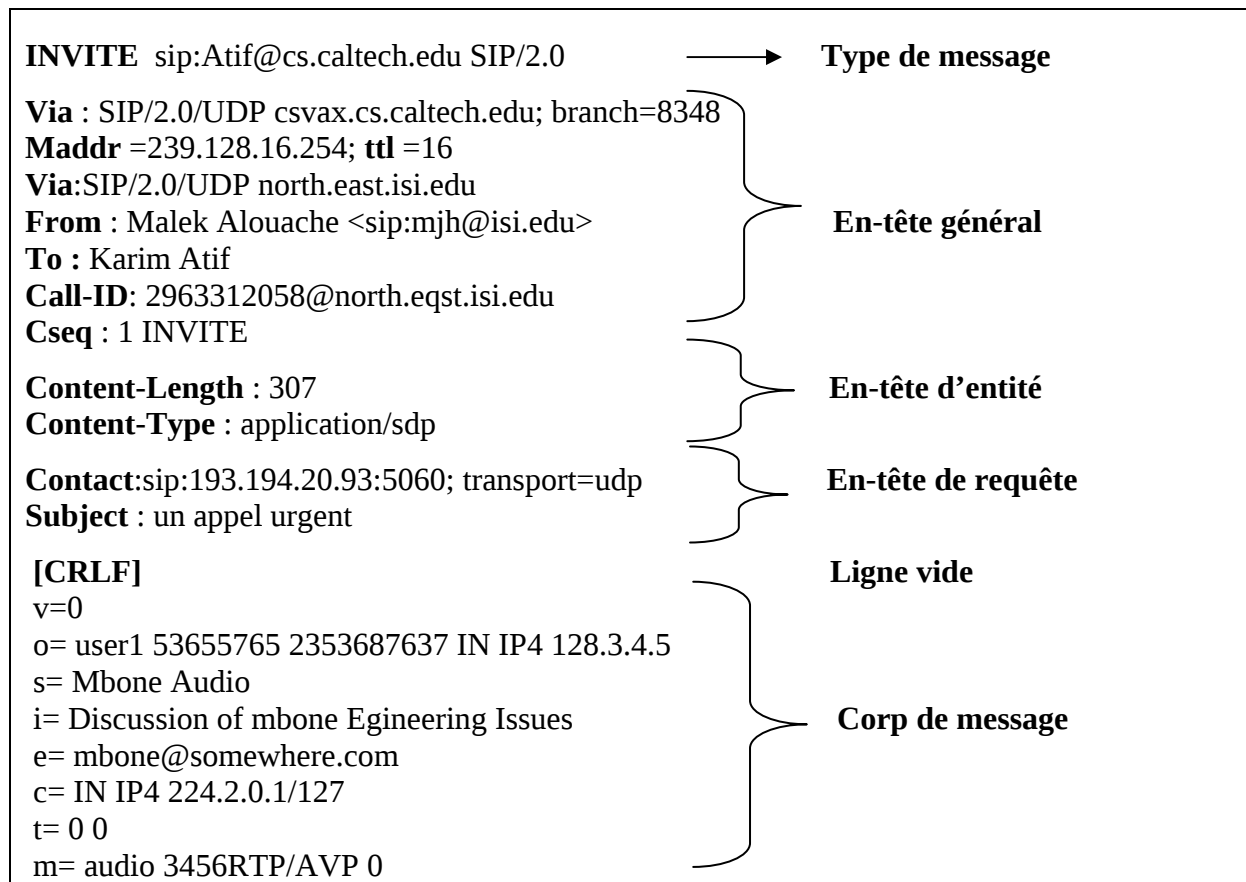


Figure 11 : Exemple d'un message SIP. 5060 : port par défaut.

Dans l'**en-tête général** nous retrouvons les champs : **From**, **To** et **Via** qui précisent les intervenant dans la session SIP et le trajet du message ; ainsi, le champ **Call-ID** qui fixe un identifiant unique de la session, il est de la forme <numéro>@<URL SIP de l'émetteur>, tous les messages échangés lors de cette session doivent avoir le même identifiant. Enfin, le champ **Cseq** portant un numéro de séquence qui identifie le message, ce numéro est incrémenté à chaque fois qu'une nouvelle requête est émis. Un serveur se contente de répondre avec le même numéro **Cseq** de la requête.

Dans l'**en-tête d'entité** nous retrouvons des informations sur le corps du message. Des informations supplémentaires sont dans l'**en-tête de requête**.

Contrairement aux protocoles standard tels que IP ou TCP, où le format des paquets ou segments est bien déterminé, **le format des messages SIP n'est pas standard**. Lorsque les messages SIP sont transportés par UDP, avec authentification et une description de session complexe, il arrive que la taille du message SIP de requête ou réponse dépasse la MTU. Pour

résoudre ce problème, un **format compact** a été défini utilisant des abréviations pour les champs d'en-tête suivants, [MHE99] :

Champ d'en-tête	Abréviation associée
Content-Type	c
Content-Encoding	e
From	f
CALL-ID	i
Contact	m (pour " moved ")
Content-Length	l
Subject	s
To	t
Via	v

Ainsi, le message d'invitation donné dans la figure 11 sera compacté comme suit :

```

INVITE sip:Atif@vlsi.caltech.edu SIP/2.0
v:SIP/2.0/UDP 131.215.131.131; maddr=239.128.16.254;ttl=16
v:SIP/2.0/UDP 128.16.64.19
f:sip:mjh@isi.edu
t:sip:Atif@cs.caltech.edu
i:62729-27@128.16.64.19
c:application/sdp
CSeq: 4711 INVITE
l:187
v:0
o:=user1 53655765 2353687637 IN IP4 128.3.4.5
s:Mbone Audio
i:Discussion of Mbone Engineering Issues
e:mbone@somewhere.com
c:IN IP4 224.2.0.1/127
t:0 0
m:=audio 3456 RTP/AVP 0

```

Figure 12 : Exemple d'un message SIP, en utilisant les abréviations.

Les serveurs doivent accepter les deux formats de message (forme normale, forme compact).

8- Fiabilité et sécurité du protocole SIP

8.1- Fiabilité

Le protocole SIP a été défini pour utiliser à la fois les deux protocoles de transport (TCP et UDP). De plus il fournit son propre mécanisme pour fiabiliser les sessions. [MHE99]

- **Fiabilisation des requêtes :**

Les serveurs en recevant une ou plusieurs copies des requêtes, retransmettent la réponse appropriée.

Lorsqu'un PS reçoit une requête CANCEL d'un client, il renvoie une requête CANCEL sur toutes les branches n'ayant pas encore reçu de réponse finale.

Quand un UAS reçoit une requête, il compare le CALL-ID de cette requête à ceux des appels en cours, comme suit :

- Dans le cas où le CALL-ID correspond à l'un d'eux, l'UAS compare alors le tag de l'URL TO de la requête avec celui de l'appel en cours, la requête sera rejetée si les « tag » ne sont pas correspondents. Dans le cas contraire, l'UAS compare également le tag de l'URL FROM de la requête avec ceux des appels existants. S'il correspond à l'un d'eux, ce sont les Cseq qui seront comparés. Si le Cseq de la requête est inférieur ou égal à celui de l'appel courant, la requête est une retransmission, sinon, c'est une nouvelle requête. Dans ce cas, si le champ d'en-tête de la nouvelle requête ne correspond à aucun de ceux des appels existants, un nouvel appel est créé.
- Si le CALL-ID ne correspond à aucun d'eux, un nouvel appel est créé.

- **Fiabilisation des réponses :**

Généralement, si un PS, RS, UAS ou RG ne peut pas répondre à une requête dans les 200 ms, il doit renvoyer une réponse d'information provisoire (du code 1xx) dès que possible.

La correspondance entre les réponses et les requêtes se fait grâce aux en-têtes TO, FROM, CALL-ID, Cseq et le premier Via d'en-tête.

Il arrive qu'un PS reçoit une réponse qui ne le concerne pas, dans ce cas, si le champ d'en-tête Via indique que le prochain serveur auquel sera transmis la réponse utilise TCP, le PS contacté ouvre une connexion TCP à cette adresse. Ainsi, les PS doivent se préparer à recevoir des réponses aux entrées où sont ouvertes des connexions TCP.

Il existe deux types de connexion TCP, qui sont :

- Connexion TCP **passive** : c'est une connexion acceptée par un PS mais ouverte par une autre entité.
- Connexion TCP **active** : c'est une connexion initialisée par un PS.

Un PS doit conserver l'état d'une transaction pendant au moins 32 secondes après réception de la première réponse définitive autre que la réponse OK (code 200) afin de pouvoir retransmettre cette réponse.

8.2- Sécurité et authentification

Comme les messages SIP peuvent contenir des données confidentielles, le protocole SIP possède 3 mécanismes de cryptage, [COL04] :

- Cryptage de bout en bout du Corps du message SIP et de certains champs d'en-tête sensibles aux attaques.
- Cryptage au saut par saut (hop by hop) à fin d'empêcher des pirates de savoir qui appelle qui.
- Cryptage au saut par saut du champ d'en-tête Via pour dissimuler la route qu'a emprunté la requête.

De plus, afin d'empêcher à tout intrus de modifier et retransmettre des requêtes ou réponses SIP, des mécanismes d'intégrité et d'authentification des messages sont mis en place. Pour des messages SIP transmis de bout en bout, des clés publiques et signatures sont utilisées par SIP et stockées dans les champs d'en-tête Authorization.

Une autre attaque connue avec TCP ou UDP est le « déni de service », lorsqu'un Proxy Server intrus renvoie une réponse de code 6xx au client (signifiant un échec général, la requête ne peut être traitée). Le client peut ignorer cette réponse. S'il ne l'ignore pas et émet une requête vers le serveur "régulier" auquel il était relié avant la réponse du serveur "intrus", la requête aura de fortes chances d'atteindre le serveur intrus et non son vrai destinataire.

9- Concurrents de SIP

9.1- Le protocole H323

La norme H323 regroupe un ensemble de protocoles de communication de la voix, de l'image et de données sur les réseaux IP. Il représente le principal concurrent de SIP pour la transmission de sons, données et vidéo en temps réel. Normalisé depuis 1996, elle fait partie de la série H.32x qui traite de la vidéoconférence au travers des différents réseaux et concerne les réseaux LAN et Internet. Elle a été développée par l'**IUT** pour des réseaux ne garantissant pas une qualité de service. **[MAJ06]**

L'avantage incontesté d'H.323 réside dans sa **maturité**, mais il est relativement **complexe** contrairement à SIP qui tente de simplifier les échanges en utilisant une sémantique proche de HTTP.

Il existe en effet de nombreux produits (plus de 30) utilisant ce standard approuvé maintenant internationalement, adopté par de grandes entreprises telles **Cisco, IBM, Intel, Microsoft, Netscape**, etc.

Les cinq principaux logiciels suivant, de visioconférence utilise la norme H323 sous IP : Picturel 550 (adaptation de Netmeeting de Microsoft), Proshare 500 (adaptation de Netmeeting de Microsoft), Trinicon 500 plus (PC Conférence), Smartstation (adaptation de Netmeeting de Microsoft), Cruiser 150 (Meetingpoint 4.0).

La conséquence directe de la compatibilité de tous les logiciels de communication avec la norme H.323 est un niveau d'**interopérabilité** très élevé, c'est un grand point fort d'H.323. En effet, cela permet à plusieurs utilisateurs d'échanger des données audio et vidéo sans faire attention aux types de média qu'ils utilisent.

Cette norme détermine quatre entités fonctionnant ensemble, **[MHE99]** :

- ❖ Le **terminal** est la machine qui permet la communication : il peut être un PC multimédia ou bien un téléphone adapté à IP.
- ❖ La **passerelle** relie le réseau de voix sur IP et le réseau téléphonique commuté (RTC).
- ❖ La **MCU** (Multipoint Control Unit) gère les conférences en multicast (plus de 3 utilisateurs).
- ❖ Et enfin le **gatekeeper (garde barrière)** combine des fonctions d'administration de la bande passante avec celles de PBX virtuels. Il établit la relation entre les alias des terminaux (adresse H.323 de l'utilisateur) et leur adresses IP. Comme pour SIP les adresses de type email sont possibles, ainsi que les numéros de téléphone.

9.2- MGCP/MEGACO

La convergence de la voix et des données conduit à l'évolution d'une nouvelle génération de réseaux et crée l'opportunité d'un nouveau marché. HSS (Hughes Software Systems), par exemple, a développé une série de protocoles pour suivre la convergence des différentes technologies, MEGACO en est issu.

Megaco peut être combiné avec Multiquick Voice (marque déposée par HSS) afin d'apporter une solution rapide de voix sur réseaux de paquets (IP) ou de mettre en place une passerelle avec les réseaux téléphoniques existant (RTC). A la différence de H.323 et de SIP qui reposent sur une architecture distribuée, MEGACO et MGCP sont orientés client/serveur.

Ce protocole gère l'interface entre le **Media Gateway Controller** (MGC, traite la signalisation) et les **Media Gateway** (MG, convertit les signaux audio en paquets) définis par l'architecture H.323 adopté par l'IETF et l'ITU. [MHE99]

10- Conclusion

SIP peut être assimilé à une boîte à outils, composée de modules autorisant le développement de nouvelles applications de communication d'entreprise, qui dépassent la téléphonie pour atteindre le multimédia. La simplicité, la rapidité et la légèreté d'utilisation du protocole SIP ainsi que ses avancées en matière de sécurités de messages sont autant d'arguments qui pourraient permettre à SIP d'être un concurrent redoutable dans le monde IP. Notons aussi que SIP est un protocole extensible (grâce au code d'état) et standard (grâce au format de ses messages), ce qui lui permettra d'assurer l'interopérabilité multifournisseur au niveau des fonctionnalités.

Après la présentation du protocole SIP, son architecture, ses composants ainsi leurs fonctions, ceci nous permet de dire que SIP pourrait être à l'origine d'une profonde évolution des services téléphoniques, mettant à profit tout le potentiel des technologies d'Internet. La téléconférence téléphonique se marierait, dans ce cas, avec des outils de gestion web, de présence des participants, et de partage d'applications.

Le protocole SIP utilise le protocole SDP comme un outil important pour l'ouverture des sessions, car ce dernier permet de décrire la session et le multimédia utilisé, ce qui aide l'utilisateur à prendre la décision d'acceptation ou de refus de participation à la session. Ce protocole SDP sera abordé en détail dans la section suivante.

11. Le Protocole de Description de Session : SDP

11.1- Introduction

Depuis le lancement des conférences multimédias (flux de vidéos en ligne, ou d'autres sessions en temps réel) un protocole s'est imposé pour transporter des détails de médias (des adresses de transport,...) et tout autre donnée de description de session aux participants. C'est pour cela qu'un protocole de description de session a été créé. Le protocole SDP (RFC-2327) est un produit de MMUSIC (Multiparty Multimedia Session Control), un groupe actif de L'IETF (Internet Engineering Task Force).[DAD04]

En général, SDP doit transporter une information suffisante pour permettre l'accès à une session et annoncer les ressources à utiliser, par exemple :

- Nom et but de la session.
- Horaires où la session est active.
- Le contenu médiatique de la session.
- L'information sur comment recevoir ces médias tels que les adresses, les ports et les formats.
- Des informations supplémentaires, par exemple : la bande passante qui est utilisée dans la conférence et les informations nécessaires pour contacter la personne responsable de la session établie.

Dans ce qui suit, nous aborderons la manière dont l'information est transportée et son utilité en unicast ou multicast. Ensuite nous étudierons la syntaxe de SDP en donnant un exemple détaillé.

11.2- Transport

A) Les informations transmises

En général, il y a beaucoup d'informations incluses dans une description de session, et les informations concernant le média utilisé sont essentielles pour l'implémentation [HAJ06].

Les informations transmises sont donc :

- Le type de média (vidéo, sonore, etc.)
- Le protocole du transport (RTP/ UDP/ IP, H.320, etc.)
- Le format du média (vidéo H.261, vidéo MPEG, etc.)

Pour une session IP multicast, les informations suivantes sont aussi transmises :

- Adresse multicast pour le média
- Port du transport pour le média

Pour une session IP d'unicast, l'information suivante est transmise :

- Adresse distante pour le média
- Port du transport pour l'adresse de contact.

B) Transport

Dans une diffusion en mode multicast, SDP a pour objectifs de communiquer l'existence d'une session, et de donner l'information suffisante sur la session pour permettre à des utilisateurs de s'associer et de participer à la session. Dans un mode unicast, seulement le dernier but est réalisable.

SDP ne possède pas de protocole de transport, il est situé au-dessus dans cette couche (voir figure 1 : Pile du protocole SIP). Il est encapsulé dans des formats soit TCP ou UDP. Cependant ce protocole doit rester fiable afin d'éviter les pertes de paquets de description. Pour minimiser cet effet, SDP est utilisé dans différents protocoles tels que celui d'annonces ou d'initiations de sessions, SAP ou SIP. Ces données sont alors de taille très réduite (de l'ordre du kilo byte au maximum pour SAP). La perte de paquet est donc minimisée. **[DAD04]**

11.3- Syntaxe

SDP définit une syntaxe pour décrire une session de multimédia avec l'information suffisante pour découvrir et participer à cette session.

Les descriptions de session de SDP sont entièrement textuelles en utilisant le jeu de caractères ISO 10646. La forme textuelle a été choisie afin d'augmenter la portabilité, pour permettre à une variété de transports d'être employée (par exemple, description de session dans un message d'email) et pour permettre la flexibilité. **[DAD04]**

Cependant, puisque SDP peut être employé dans les environnements où la taille maximum d'une description de session est limitée (par exemple des annonces de SAP, SIP transporté par UDP), le codage est délibérément compact.

Une description de session de SDP se compose d'un certain nombre de lignes du texte de la forme :

< type>= <value>

Où le < type > doit être exactement un caractère significatif et la < valeur > est un texte structuré dont le format dépend du < type >.

En général la < valeur > est un certain nombre de champs délimités par un caractère d'espace simple. Il ne faut pas avoir d'espace blanc de chaque côté du symbole "=".

Une description de session se compose d'une section de **niveau-session** suivie par des **niveaux-media**.

- La section de **niveau-session** commence par "v =" et continue jusqu'à la première section de **niveau-media**.

- La description de médias commence par "m =" et continue jusqu'aux prochains médias décrits ou la fin de la description entière de session.

Généralement, il y a beaucoup d'informations à inclure dans une description de session et les informations concernant le média utilisé sont essentielles pour l'implémentation, quelques lignes dans chaque description sont exigées et certaines sont facultatives (marqués avec un «*»).

Dans une description SDP, toutes les lignes apparaissent exactement dans l'ordre donné ci-après [HAJ06].

La Description De La Session	
TYPE	DESCRIPTION
v	Version du protocole
o	Le propriétaire/créateur et identifiant de la session
S	Nom de la session
i*	Information sur la session
u*	URI de description
e*	Adresse email
p*	Numéro de téléphone
c*	Information sur la connexion **
b*	Information sur la bande passante
z*	Ajustement de la zone du temps
k*	Clef de codage
a*	Zéro ou plusieurs lignes d'attributs de session
** n'est pas exigé s'il est inclus dans tous les médias	

La Description Du Temps	
t	Le temps où la session est active
r*	Indique un nombre de répétition

La description Du média	
m	Media et adresse du transport
i*	Titre du média
c*	Information sur la connexion **
b*	Information sur la bande passante
k*	Clef de codage
a*	Zéro ou plusieurs lignes d'attributs de session
** facultatif s'il est inclus au niveau de session	

Tableau 2 : Les champs des types autorisés par SDP.

Remarques :

- 1- L'ordre fixe des lignes augmente considérablement la détection des erreurs et tient compte d'un analyseur simple.
- 2- Le nombre de lettre est inextensible. L'analyseur ignore les autres lettres rajoutées, ou les messages incompréhensibles.

- **Exemple de description SDP :**

Pour clarifier les concepts présentés précédemment, voici un exemple de session SDP (voir figure 13). Quelques champs sont obligatoires et d'autres sont facultatifs selon l'application. Les plus significatifs tels que : l'origine (o), le nom de la session (s), le temps où la session est active (t), l'information sur la connexion (c) et l'annonce du média (m) sont décrits en détail.

```
v=0
o=Samir 2890844526 2890842807 IN IP4 130.233.154.68
s=ICW SIP
i=une session pour le test de SDP avec SIP
u=http://www.kaced.net/staff/samir/sdp.03.ps
e=samir@kaced.net (samir)
c=IN IP4 130.233.154.68
t=2873397496 2873404696
m=audio 49170 RTP/AVP 01
m=video 51372 RTP/AVP 31
```

Figure 13 : Exemple de description SDP.

La description détaillée [HAJ06] de chaque champ :

- **Origine :**

Le champ 'o=' donne l'initiateur de la session (son nom et l'adresse de sa hôte) plus l'identifiant de la session et le numéro de la version de la session. La structure générale du champ 'o' est :

o=<username><id de la session><version><type du réseau><type de l'adresse><adresse>

- **<username>** est le nom de l'utilisateur sur l'hôte origine ou bien c'est '-' si l'hôte origine ne supporte pas le concept d'ID de l'utilisateur. **<username>** ne doit pas contenir de caractère d'espaces (blanc)
- **<id de la session>** est une chaîne de caractères numériques, tel que le tuple de **<username>**, **<id de la session>**, **<version>**, **<type du réseau>**, **<type de l'adresse>** et **<adresse>** forme un identifiant unique pour la session.
- **<version>** est le numéro de la version de cette annonce. Il est nécessaire pour les annonces du Proxy de détecter laquelle de plusieurs annonces pour la même session est la plus récente. Ainsi, le paramètre **<version>** est augmenté quand une modification est faite aux informations de la session. Pour cette raison, il est recommandé qu'une étiquette temporelle soit utilisée.

- **<type du réseau>** est une chaîne de caractère qui donne le type du réseau. Initialement 'IN' est défini pour avoir la signification 'Internet'.
- **<type de l'adresse>** est une chaîne de caractère qui donne le type de l'adresse. Initialement IP4.
- **<adresse>** est l'adresse unique de la machine à partir de laquelle la session a été créée. En général, c'est le nom complet du domaine qualifié de la machine.

Dans l'exemple précédent : o=Samir 2890844526 2890842807 IN IP4 130.233.154.68

- **Le nom de la session :**

Le champ 's=' est le nom de la session. Il doit y avoir un et un seul champ 's=' par SDP, et il doit contenir des caractères ISO 10646. Par exemple : s=ICW SIP.

- **Information sur la connexion :**

Le champ 'c=' contient les informations sur la liaison (connexion). La structure générale du champ est :

c=<type du réseau><type de l'adresse><adresse de la connexion>

Une annonce de la session doit contenir un champ 'c=' dans chaque description de média ou un seul dans le niveau de la session.

- **<type du réseau>** est une chaîne de caractères qui donne le type du réseau. Initialement 'IN' signifie Internet.
- **<type de l'adresse>** permet à SDP d'être utilisé pour les sessions qui ne sont pas basées IP. Actuellement seulement IP4 est défini.
- **<adresse de la connexion>** est le troisième sous champ. Pour les adresses IP4, la connexion est définie comme suit : Si la session n'est pas multicat, alors l'adresse de la connexion contient le nom complet du domaine qualifié ou bien l'adresse IP de l'unicast de la source des données attendues.

- **Temps d'activité de la session :**

Dans le champ 't=', le premier et le deuxième sous champs donnent respectivement le temps de début et le temps d'arrêt pour la conférence. Ces valeurs sont la représentation décimale des valeurs du temps du protocole de temps du réseau NTP (Network Time Protocol time values) en secondes. La structure générale du champ 't=' est :

t=<temps de début de la session><temps de fin de la session>

Par exemple : $t=2873397496\ 2873404696$

Si le temps de fin de session est mis à zéro, alors la session n'est pas limitée.

Si le temps de début de session est mis à zéro, la session est considérée comme permanente.

- **Annonces des médias**

Une description de sessions peut contenir plusieurs descriptions du média. Chacune d'elle commence avec un champ 'm=', et se termine soit par le prochain champ 'm=' soit par la fin de la description.

Par exemple :

$m=AUDIO\ 49170\ RTP/AVP\ 0$

$m=VIDEO\ 51372\ RTP/AVP\ 31$

$m=APPLICATION\ 32416\ UDP\ WP$

La structure générale d'un champ média est :

$m=<média>\ <port>\ <transport>\ <fmt\ list>$

1. **<média>** : est le type du média. Les médias définis actuellement sont inclus dans le tableau suivant :

TYPES DE MEDIA
Application
Audio
Contrôle
Information
Vidéo
Nouvelle sortie (new emerging)
Modalités de transmission

Tableau 3 : Types de média.

La différence entre l'application et l'information est que la première est un flux médiatique tel que les informations dans un tableau, et la dernière est le transfert des données en vrac tel que la multi-diffusion de programmes exécutables qui ne seront pas affichés pour l'utilisateur. Le contrôle est utilisé pour spécifier un canal supplémentaire du contrôle de la conférence pour la session.

2. Le deuxième sous champ est le **port** du transport auquel le flux du média sera envoyé.

3. Le troisième sous champ est le protocole de **transport**. Les protocoles de transport suivants sont définis préliminairement :

RTP/AVP – L'IETF Real-time Transport Protocol qui utilise le profil Audio/Vidéo transporté sur UDP (User Datagram Protocol).

4. Le quatrième sous champ spécifie la **liste de formats** de type utile, c-à-d il spécifie les formats qui peuvent être utilisés dans la session, le premier de ces formats est le format par défaut pour la session.

11.4- Conclusion

SDP est un protocole de description qui contient un certain nombre d'informations sur une session.

SDP est un protocole de description de session fiable mais on ne peut lui faire confiance que si le protocole de transport utilisé est lui-même fiable. Un protocole fréquemment utilisé associé à SDP est SAP (Session Announce Protocol). SAP fournit les mécanismes de cryptage et authentification.

Le format de description de session SDP, présente un outil très efficace pour les besoins d'annonce de média, d'initiation de session et d'autres protocoles tel que SIP.

On constate aussi, que c'est un protocole qui fournit un format universel pour décrire des sessions de multimédia, les annonces (SAP) ou les invitations (SIP). Ce protocole est indépendant de la façon dont cette information est transportée. SDP définit une syntaxe pour décrire une session de multimédia avec l'information suffisante pour découvrir et participer à cette session.

SDP est prévu pour être employé dans un éventail d'environnements et d'applications de réseau. **[DAD04]**

Chapitre III

Le Protocole CC/PP/ Gestion du Profil

1- Introduction

Il y a maintenant une variation énorme dans les capacités entre les équipements (les plus petits capables de se connecter au Web, les stations de travail ou ordinateurs individuels classiques). Entre ces deux extrêmes, il y a de nombreux autres terminaux qui peuvent accéder au Web, comme les systèmes de télévision interactive, les assistants personnels, les téléphones intelligents et les appareils domestiques.

Cette variation énorme des appareils connectés à l'Internet impose la disponibilité de beaucoup d'informations concernant aussi bien le client ou son environnement (poste de travail). Il existe des techniques qui fournissent des informations limitées, telles que les entêtes Accept de http [TRH96] et les attributs alt de HTML [VAL06].

Dans le cadre de la satisfaction des préférences d'un client, un format de profil universel est nécessaire pour la contextualisation d'un contenu qui décrit les capacités d'un agent utilisateur et les préférences de son utilisateur. Pour cela le format CC/PP (Capacités Combinées/Profils de Préférences) est utilisé car il répond à cette attente.

Un profil CC/PP est une description des *capacités d'un appareil* et des *préférences* d'un utilisateur. Cette description désignée souvent par *contexte de remise* d'un appareil, peut être utilisée pour guider l'adaptation du contenu présenté à l'appareil en question [JJS04].

Le but de CC/PP est de fournir le fondement de l'indépendance vis à vis des équipements capables de se connecter au Web.

Dans ce chapitre nous allons présenter ce protocole, ainsi que son apport à la négociation de services.

2- Le protocole CC/PP

2.1- Origine

L'un des objectifs primordiaux du W3C est l'accès universel au Web à partir de terminaux variés. De ce fait, plusieurs aspects doivent être pris en considération tel que les capacités matériel, logiciel, langue d'origine, culture, lieu géographique ou encore aptitude physique.

Le Consortium World Wide Web (W3C) a publié la recommandation CC/PP 1.0 (Capacités Combinées/Profils de Préférences (CC/PP)) [JYM04]. Il le considère comme un environnement qui permet de commenter les capacités d'un terminal, les préférences utilisateurs et le contexte d'utilisation. Cet environnement est décrit à l'aide des profils CC/PP, ces derniers constituent un ensemble d'informations basé sur le concept des RDF (Resource Description Framework) [ORL99]. Il est fourni par le client pour piloter l'adaptation du contenu Web demandé [JYM04]. Tout terminal peut communiquer à un serveur ses capacités détaillées à l'aide de ce profil, du fait qu'il permet un format de description des données standard.

La situation suivante illustre le besoin de description du profil : « Je suis un téléphone mobile et la taille d'affichage de mon écran ne me permet pas de visualiser un site Web avec cadres. Merci de me fournir un contenu Web sous forme de listes ». Ceci constitue un exemple de contexte de distribution de contenu, où les caractéristiques du terminal, les préférences utilisateur et ses contraintes, créent des conditions sur la manière dont le contenu Web sera délivré.

2.2- Définition

Le protocole CC/PP est vu sous forme de structure et vocabulaires permettant la définition d'un **format des données d'un profil client**. CC/PP représente un cadre pour l'incorporation des caractéristiques particulières d'une application ou d'un **environnement d'exploitation**. Ce protocole ne définit pas comment le profil est généré (c-à-d quels attributs seront reconnus) ou comment seront transmis. Le but de la conception de ce protocole est son utilisation au sein d'un **cadre applicatif global**, pour cela peu de protocoles entrent dans sa conception [JJS04].

Bien que ce protocole a été destiné à être largement indépendant d'un autre protocole, il a fait l'objet d'une attention particulière en ce qui concerne son utilisation avec le protocole HTTP pour la recherche de ressources Web.

CC/PP s'appuie sur les RDF (Ressource Description Framework) [ORL99]. Ces derniers sont considérés comme un langage universel pour la description des métadonnées ou des propriétés lisibles par les machines sur le Web. Les RDF fournissent les moyens et les outils d'extensibilité du vocabulaire, par le biais des espaces de nommage XML.

2.3- La structure d'un profil CC/PP

Un profil CC/PP est construit sous forme d'un arbre ou une hiérarchie à deux niveaux :

Un profil est constitué d'un ensemble de *composant* (au moins un), et chacun d'eux contient au moins un *attribut* [JJS04].

A)- Les composants

Les composants d'un profil correspondent aux branches initiales de l'arbre qui décrit ce profil, par exemple :

- Nous avons le profil (P1) qui contient trois Composants (C₁, C₂, C₃), dont chacun a ses propres attributs (a₁, a₂,a₆), qui peuvent être décrit à l'aide de l'arbre suivant :

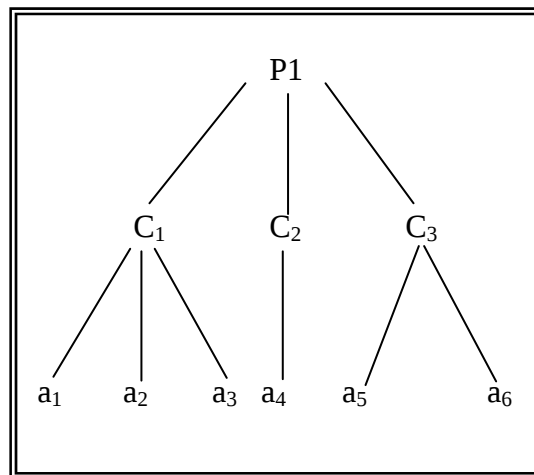


Figure 14 : Un profil sous forme d'un arbre.

P1 peut être le profil qui décrit les trois principaux composants suivants :

- C₁ : la plate-forme matérielle sur laquelle le logiciel s'exécute,
- C₂ : la plate-forme logicielle sur laquelle toutes les applications sont hébergées,
- C₃ : une application individuelle, comme un navigateur.

Concernant le code XML qui décrit ce profil **[JJS04]**; nous trouvons que chaque composant est décrit par une ressource de type *ccpp:Component* (ou une sous classe RDF) et relié à la ressource du profil client par une propriété *ccpp:component*.

A l'aide de la propriété *rdf:type* (ou une structure RDF équivalente), contenu dans l'objet d'une ressource *ccpp:Component*, nous pouvons déterminer quelle sorte de composant est décrit. La présence du type de composant permettra au processeur de disposer de suffisamment d'informations pour les interpréter correctement. Le processeur CC/PP doit être capable de prendre en charge les composants qui n'indiquent pas leurs type.

Dans ce qui suit, nous allons présenté un exemple qui illustre les composants d'un profil :

```
[exemple:LeProfil]
|
+--ccpp:component-->[exemple:MaterielFinal]
+--ccpp:component-->[exemple:LogicielFinal]
+--ccpp:component-->[exemple:NavigateurFinal]
```

Figure 15: Les composants d'un profil CC/PP.

Le code XML correspondant :

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ccpp="http://www.w3.org/2002/11/08-ccpp-schema#"
  xmlns:exemple="http://www.example.com/schema#">

  <rdf:Description rdf:about="http://www.example.com/profil#LeProfil">

    <ccpp:component>
      <rdf:Description
        rdf:about="http://www.example.com/profil#MaterielFinal">
        <!-- Les propriétés de MaterielFinal ici -->
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description
        rdf:about="http://www.example.com/profil#LogicielFinal">
        <!-- Les propriétés de LogicielFinal ici -->
      </rdf:Description>
    </ccpp:component>

    <ccpp:component>
      <rdf:Description
        rdf:about="http://www.example.com/profil#NavigateurFinal">
        <!-- Les propriétés de NavigateurFinal ici -->
      </rdf:Description>
    </ccpp:component>

  </rdf:Description>
</rdf:RDF>
```

Figure 16: Les composants d'un profil CC/PP en XML.

B)- Les attributs des composants

Chaque composant appartenant à un profil CC/PP décrit les capacités et préférences d'un client en utilisant un certain nombre d'attributs, d'où chaque composant et ses attributs représentent un sous arbre de l'arbre du profil CC/PP.

Une capacité est décrite souvent au moyen d'un nombre minimum d'attributs CC/PP, ayant chacun une valeur atomique simple.

Pour décrire les capacités et les préférences d'un client, ce dernier est considéré comme une ressource depuis laquelle des arêtes étiquetées, se dirigent vers des valeurs objets correspondantes **[JJS04]**.

[Ressource composant du client] --nomAttribut--> (valeur-attribut)
--

Figure 17 : Une déclaration RDF décrivant un attribut du client.
--

Dans le code XML correspondant, les noms d'attributs sont décrits à l'aide des adresses URI.

Quand des valeurs plus complexes sont nécessaires, on peut les construire comme des sous-graphes RDF [voir l'Annexe B]. L'emploi de valeurs d'attribut complexes peut se révéler utile pour représenter des valeurs de remplacement, par exemple, un navigateur pouvant gérer plusieurs versions de HTML.

Un profil hypothétique pourrait ressembler à ce qui suit :

```

[ex:LeProfil]
|
+--ccpp:component-->[ex:MaterielFinal]
|
|   +--rdf:type----> [ex:PlateformeMateriel]
|   +--ex:displayWidth--> "320"
|   +--ex:displayHeight--> "200"
|
+--ccpp:component-->[ex:LogicielFinal]
|
|   +--rdf:type----> [ex:PlateformeLogiciel]
|   +--ex:name-----> "EPOC"
|   +--ex:version--> "2.0"
|   +--ex:vendor---> "Symbian"
|
+--ccpp:component-->[ex:NavigateurFinal]
|
|   +--rdf:type----> [ex:AgentUtilNavigateur]
|   +--ex:name-----> "Mozilla"
|   +--ex:version--> "5.0"
|   +--ex:vendor---> "Symbian"
|   +--ex:htmlVersionsSupported--> [ ]
|
|   -----
|   |
|   +--rdf:type---> [rdf:Bag]
|   +--rdf:_1-----> "3.2"
|   +--rdf:_2-----> "4.0"

```

Figure 18 : Exemple de profil CC/PP complet en utilisant les valeurs complexes.

C)- Les valeurs par défaut

Les attributs décrivant un profil ne se limitent pas à leur incorporation au sein du profil seulement, mais ces attributs peuvent exister au sein d'un autre profil, désigné par *Profil par défaut*, qui sont stockés séparément au profil courant. Ces attributs vont être référencés par une adresse URI, car chaque ressource par défaut est un document RDF référencé par une adresse URI [JJS04]. [Voir l'Annexe C]

Le stockage des profils par défaut à un emplacement différent des profils qui le référencent, est avantageux et appréciable particulièrement dans les environnements en liaison sans fil, tels que les réseaux cellulaires. Dans le cas où le profil est volumineux ou la liaison du client est coûteuse, l'utilisation des valeurs par défaut est intéressante car une seule partie du profil sera transmise par le réseau sans fil puisque le reste est référencé.

L'utilisation d'un profil par défaut est en quelque sorte similaire au concept d'héritage dynamique. Elle rend possible certaines optimisations importantes.

Les valeurs par défaut d'un composant de profil CC/PP sont signalés par un arc *ccpp:defaults* depuis le composant concerné vers un composant qui définit les valeurs par défaut. Voici un exemple d'utilisation des valeurs par défaut:

```
[ex:LeProfil]
|
|--ccpp:component--> [ex:MaterielFinal]
|
|       |
|       |--rdf:type-----> [ex:PlateformeMateriel]
|       |--ccpp:defaults--> [ex:MaterielDefault]
|
|--ccpp:component--> [ex:LogicielFinal]
|
|       |
|       |--rdf:type-----> [ex:PlateformeLogiciel]
|       |--ccpp:defaults--> [ex:LogicielDefault]
|
|--ccpp:component--> [ex:NavigateurFinal]
|
|       |
|       |--rdf:type-----> [ex:AgentUtilNavigateur]
|       |--ccpp:defaults--> [ex:AgentUtilDefault]

[ex:MaterielDefault]
|
|--rdf:type----> [ex:PlateformeMateriel]
|--ex:displayWidth--> "320"
|--ex:displayHeight--> "200"

[ex:LogicielDefault]
|
|--rdf:type----> [ex:PlateformeLogiciel]
|--ex:name-----> "EPOC"
|--ex:version--> "2.0"
|--ex:vendor---> "Symbian"

[ex:AgentUtilDefault]
|
|--rdf:type----> [ex:AgentUtilNavigateur]
|--ex:name-----> "Mozilla"
|--ex:version--> "5.0"
|--ex:vendor---> "Symbian"
|--ex:htmlVersionsSupported--> [ ]
|
|       |
|       |--rdf:type---> [rdf:Bag]
|       |--rdf:_1-----> "3.2"
|       |--rdf:_2-----> "4.0"
```

Figure 19 : Un profil CC/PP utilisant des valeurs par défaut.

Remarques

1- Si une valeur d'attribut donnée est appliquée directement à une ressource composant et si elle apparaît aussi dans une ressource appelée par la propriété *ccpp:defaults*, alors c'est la valeur appliquée directement qui a le privilège :

```
[ex:LeProfil]
|
+--ccpp:component--> [ex:MaterielFinal]
      |
      +--rdf:type-----> [ex:PlateformeMateriel]
      +--ccpp:defaults---> [ex:MaterielDefault]
      +--ex:memoryMb-----> "32"

[ex:MaterielDefault]
|
+--rdf:type----> [ex:PlateformeMateriel]
+--ex:displayWidth--> "320"
+--ex:displayHeight--> "200"
+--ex:memoryMb---> "16"
```

Figure 20: exemple d'utilisation des redéfinitions des valeurs par défaut.

Dans cet exemple, le composant par défaut indique une capacité mémoire de 16Mb, mais cette valeur est écrasée par la propriété *memoryMb* appliquée directement au composant profil. De ce fait, dans ce profil, l'attribut *memoryMb* a une valeur de 32.

2- Si un composant appelé par *ccpp:defaults* contient un attribut absent du composant de profil appelant, alors l'effet est comme si la valeur d'attribut dans le composant par défaut était appliquée directement au composant de profil. Par exemple, le profil de la figure 20 devrait s'interpréter comme le profil décrit dans la figure suivante:

```
[ex:LeProfil]
|
+--ccpp:component--> [ex:MaterielFinal]
      |
      +--rdf:type-----> [ex:PlateformeMateriel]
      +--ex:displayWidth--> "320"
      +--ex:displayHeight--> "200"
      +--ex:memoryMb-----> "32"
```

Figure 21 : Exemple d'utilisation des redéfinitions des valeurs par défaut.

L'exemple suivant est la sérialisation XML du profil avec des références vers des valeurs par défaut externes (exemple de la figure 20) :

Le profil de l'appareil appelant les valeurs par défaut :

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ccpp="http://www.w3.org/2002/11/08-ccpp-schema#"
  xmlns:ex="http://www.example.com/schema#">

  <rdf:Description
    rdf:about="http://www.example.com/profil#LeProfil">
    <ccpp:component>
      <rdf:Description
        rdf:about="http://www.example.com/profil#MaterielFinal">
        <rdf:type
          rdf:resource="http://www.example.com/schema#PlateformeMateriel" />
        <ccpp:defaults
          rdf:resource="http://www.example.com/profilMateriel#MaterielDefault" />
        <ex:memoryMb>32</ex:memoryMb>
        </rdf:Description>
      </ccpp:component>

    </rdf:Description>
  </rdf:RDF>
```

Les valeurs par défaut de PlateformeMateriel:

```
<?xml version="1.0"?>
<rdf:RDF xmlns:rdf="http://www.w3.org/1999/02/22-rdf-syntax-ns#"
  xmlns:ex="http://www.example.com/schema#">
  <rdf:Description
    rdf:about="http://www.example.com/profilMateriel#MaterielDefault">
    <rdf:type
      rdf:resource="http://www.example.com/schema#PlateformeMateriel" />
    <ex:displayWidth>320</ex:displayWidth>
    <ex:displayHeight>200</ex:displayHeight>
    <ex:memoryMb>16</ex:memoryMb>
    </rdf:Description>
  </rdf:RDF>
```

Figure 22 : Un profil CC/PP référençant des valeurs par défaut externes en XML.

2.4- L'utilisation et l'extensibilité du vocabulaires CC/PP

Généralement des applications différentes utilisent des vocabulaires différents. Afin que ces applications puissent communiquer ensemble dans le cadre de CC/PP, elles doivent définir un vocabulaire commun et d'autres spécifiques à chaque application ou une méthode de conversion entre les vocabulaires différents. Les espaces de nommage XML [STE03] peuvent garantir l'absence de conflit entre les noms d'applications différentes, mais ils ne fournissent pas de base commune pour l'échange d'informations entre des applications différentes. [JJS04].

Tout vocabulaire existant peut être utilisé et réutilisé dans n'importe quelle expression CC/PP. Ce vocabulaire est extensible aussi, car CC/PP permet l'introduction de nouveaux vocabulaires (attributs).

3- La conformité

Chaque producteur peut faire une revendication (par exemple, des vendeurs pour leurs propres produits, des tierces parties pour ces mêmes produits, des journalistes, etc.). Ces revendications peuvent être publiées partout (par exemple, sur le Web ou dans la documentation d'un produit). Les prétendants sont seuls responsables de leurs revendications. Si le sujet de la revendication (par exemple, un logiciel) change après la date où celle-ci est intervenue, le prétendant est responsable de la mise à jour de la revendication. Ils sont censés modifier ou rétracter une revendication si cette dernière n'est pas valide [JJS04].

Il existe trois classes de produits CC/PP :

- ❖ Les documents (par exemple, une ressource Web).
- ❖ Les producteurs (par exemple, un client Web).
- ❖ Les consommateurs (par exemple, un serveur Web).

3.1- La conformité du document CC/PP

Les documents peuvent exister comme ressources accessibles via un URL ou peuvent être transmis comme données dans un message. Un document est conforme à CC/PP quand il satisfait aux 23 critères cités en Annexe B, parmi lesquels nous citons les suivants :

- 1- Le document DOIT être valide en RDF et sérialisé en XML, et se baser sur un ou plusieurs vocabulaires dérivés du schéma RDF.
- 2- Le document DOIT employer une syntaxe valide pour les déclarations d'espace de nommage.

- 3- L'élément profil DOIT contenir un ou plusieurs composants.
- 4- Chaque composant dans le profil DOIT contenir un ou plusieurs attributs.
- 5- Les références des composants par défaut DOIVENT être des URL valides.
- 6- Les attributs d'un composant PEUVENT contenir à la fois une valeur par défaut et une valeur appliquée, cette dernière valeur ayant le privilège.

3.2- La conformité du producteur CC/PP

Un producteur est conforme à CC/PP quand tout document de profil CC/PP qu'il génère, est un document conforme à CC/PP.

3.3- La conformité du consommateur CC/PP

Un consommateur est conforme à CC/PP quand il accepte tout document conforme à CC/PP et en extrait les informations CC/PP.

Les consommateurs CC/PP se rangent dans deux catégories de conformité :

1. **Conforme** : un consommateur CC/PP peut revendiquer être un *consommateur conforme à CC/PP 1.0* s'il accepte n'importe quel profil CC/PP valide et en extrait les informations.
2. **Validant** : un consommateur CC/PP peut revendiquer être un *consommateur validant conforme à CC/PP 1.0* s'il est conforme et s'il rejette tous les profils CC/PP invalides.

3.4- Les revendications de conformité

3.4.1- La validité

Une revendication de conformité est valide si elle est **bien formée** et satisfait aux critères de conformité nécessaires pour la classe de produit concernée comme indiquée ci-dessus.

3.4.2- Bien formé

Une revendication de conformité est bien formée si elle comprend les informations suivantes :

- La date de la revendication.
- La classe de produit (document, producteur ou consommateur).
- La catégorie de consommateur (conforme ou conforme validant), le cas échéant
- Le titre et l'URI daté de ce document.
- Le nom du produit (identité), y compris une version, une date ou un autre identificateur qui identifie le produit de manière unique.

4- Conclusion

Généralement, les différents protocoles de description se sont focalisés sur la description des capacités des appareils, par contre, le protocole CC/PP peut véhiculer des informations concernant les capacités d'un appareil et les préférences de son utilisateur. Ces informations devraient permettre aux serveurs Web d'améliorer l'accessibilité des sites Web.

Faisant partie d'une **application globale**, CC/PP ne peut couvrir entièrement de telles questions mais son utilisation conjointe avec des mécanismes appropriés est prévue, tel que les protocoles de signalisation et ainsi les protocoles de transport.

Les futures versions de CC/PP pourront introduire des structures supplémentaires apportant une correspondance normalisée des profils clients avec d'autres ressources de métadonnées.

Notre travail, est consacré à l'étude du mécanisme de négociation de services et la satisfaction de l'utilisateur, en respectant ses préférences et les contraintes de son environnement. Une gamme de services est offerte au client, ce qui lui permet de rentrer en négociation avec le fournisseur de ces services pour aboutir à un accord.

L'emploi et la complémentarité entre les différents protocoles étudiés, à travers les chapitres précédents, pour satisfaire les besoins d'un utilisateur, représentent le noyau de notre mémoire et ce qui sera décrit dans le chapitre suivant.

Chapitre IV : Connexion, Etat de Profil Et Négociation de Services

1- Introduction

Les progrès technologiques récents ont permis l'apparition d'une grande variété de nouveaux moyens pour accéder et utiliser des services à travers le Web en tout lieu et à tout moment. Aujourd'hui, on trouve une multitude de formats de services complexes avec de nouvelles fonctionnalités.

Face à cette évolution, il est nécessaire de concevoir des systèmes qui permettent aux utilisateurs l'accès et de bénéficier des services qui existent dans le Web sous une forme qui correspond à leurs préférences et les contraintes imposées par leur environnement.

La négociation de contenu existe depuis longtemps sur le Web, comme partie du protocole HTTP. Ses applications pratiques ont souvent été limitées car HTTP a été conçu pour la description de navigateurs, et non pas celle des utilisateurs, des contextes et des terminaux [JYM04]. En autorisant des descriptions complexes et complètes des contextes de distribution, CC/PP fournit des informations nécessaires pour adapter les contenus Web aux besoins des utilisateurs.

2- Problématique

Notre étude au protocole CC/PP, nous a permis de considérer ce dernier comme un élément important dans la négociation de service, car il permet de décrire les préférences des utilisateurs, en utilisant au parfait usage ses avantages et de compléter ses manques.

CC/PP est un cadre de format conçu pour une utilisation dans le contexte d'un environnement applicatif ou opérationnel global. Cette spécification ne définit pas comment utiliser CC/PP avec un protocole ou une application particulières. Plusieurs questions peuvent être posées pour la spécification d'un protocole applicable, qui sera utilisé pour acheminer les profils CC/PP, pour une utilisation efficace du cadre CC/PP, les règles d'exploitation de l'environnement global doivent spécifier [JJS04] :

- Le protocole d'échange de capacités;
- le modèle de fiabilité;
- les mécanismes de sécurité;

- les règles pour l'identification des requêtes de profil;
- le vocabulaire ;
- les contraintes sur les types de valeur d'attribut admis;
- les règles de traitement et/ou de correspondance des valeurs d'attribut;
- le vocabulaire d'un serveur mandataire et son traitement;
- les informations supplémentaires à inclure dans toutes les données de ressource transmises;
- les formes d'URI admises pour l'identification des documents de profil référencés (par exemple, les profils par défaut);
- les mécanismes de localisation et de rapport des documents de profil référencés;
- les interactions avec les mécanismes de négociation existants dans le protocole hôte;

Avant de décrire l'environnement applicatif du protocole CC/PP, nous allons citer quelques manques de ce protocole, que nous devons par la suite les compléter pour l'obtention d'un processus cohérent. Nous citons ce qui suit :

1- *La sécurité* : nous pouvons remarquer qu'aucun mécanisme de sécurité n'est utilisé, bien que certaines informations contenues dans un profil peuvent être sensibles. Il faut donc déployer des mécanismes de confiance et de sécurité adéquats afin de protéger la vie privée des utilisateurs.

2- *Corps du profil* : il n'existe pas une structure de contexte imposé pour l'utilisateur, ce dernier peut envoyer un profil incomplet (informations importantes et nécessaires) qui peut influencer sur le temps d'exécution.

Par exemple : dans le cas où l'utilisateur oublie de donner le type du composant, ceci oblige le processeur CC/PP à prendre en charge le profil qui induit à :

- une perte de temps;
- parfois à un conflit entre les composants.

3- *Les valeurs par défaut* : le moment du chargement des valeurs, et la vérification de la cohérence de l'adresse URI.

4- *Redéfinition des attributs* : dans ce cas l'utilisateur n'a aucune information sur le type de l'attribut pour donner une valeur cohérente.

5- *Vocabulaire* : le vocabulaire existant est limité.

3- Solution proposée

Notre vision est que la solution du problème de négociation de contenu, doit définir un ensemble de composants qui collaborent afin de trouver un consensus sur la meilleure stratégie à suivre pour répondre aux requêtes du client. Ces composants doivent inclure une prise en charge complète des différentes dimensions du contexte et aux objets media du contenu. Les services de la négociation doivent comprendre :

- La notion du profiling,
- un mécanisme de recherche pour choisir les meilleures offres adéquates aux spécifications du client,
- l'utilisation des technologies Web telle que XML et RDF.

Pour cela, nous avons proposé d'utiliser ce qui suit :

- Les deux protocoles SIP et SDP : pour l'ouverture et la description de la session de communication ainsi la description du matériel utilisé. Ces deux protocoles permettent aussi de véhiculer les différentes données échangées.
- La notion d'interface papier : cette dernière est utilisée pour la présentation des informations médiatiques, car elle permet la présentation du média ou du service fourni ainsi qu'un texte descriptif. **[CHS04]**
- Le protocole CC/PP : on va l'utiliser afin de décrire les préférences de l'utilisateur.

Pour la réalisation de cette vision, on doit avoir au moins :

- 1- L'application cliente, c'est-à-dire l'application ou le protocole utilisé par le client pour décrire, émettre le contenu de sa requête, ainsi que le traitement des réponses reçus. Dans notre cas nous avons choisi d'utiliser le protocole CC/PP, auquel nous proposons ces ajouts :
 - Concernant la sécurité : le protocole SIP qui s'encharge de véhiculer le profil, assure une sécurité d'informations. **[NIF04]**
 - Les attributs : on doit définir dans le profil des champs obligatoires, ce qui assure que l'information soit complète, et dans le cas où l'utilisateur ne fournit pas l'information le rejet se fera à son niveau, avant même que le profil soit envoyé, ce qui augmente du rendement du processeur CC/PP et évidemment du processus de négociation.
 - Concernant les valeurs par défaut : le chargement des composants référencés doit se faire lors d'analyse de la requête, c à d au niveau du serveur de contexte, surtout dans le cas des réseaux cellulaires.

- Il faut ajouter des fonctionnalités, qui indiquent lors de redéfinition des attributs, le type et le composant auxquels fait référence l'attribut redéfini ; Par exemple : juste à la fin de la saisie d'un attribut, si ce dernier existe déjà dans un autre composant, un sous-texte apparaît pour indiquer le type et le composant auxquels appartient cet attribut.
- Cependant, pour l'extension du vocabulaire : on propose de définir une bibliothèque qui sera organisée par domaine, et divisée comme suit :
 - ❖ Des classes de vocabulaire commune à des différents domaines d'application.
 - ❖ Et d'autres spécifiques, dans lesquels nous définissons le vocabulaire le plus utilisé dans chaque domaine.

Pour chaque utilisateur, cette bibliothèque sera offerte, après avoir choisi un domaine d'application. Il aura le vocabulaire correspondant pour en choisir un, dans le cas contraire, il définit son propre terme, et ce dernier peut être rajouté à la bibliothèque. Mais on doit définir des méthodes de vérification qui permettent d'effectuer un filtrage à cette bibliothèque, pour enlever les mots répétitifs, et synonymes pour ne pas saturer l'espace mémoire.

- 2- Un système de communication, qui assure le moyen de dialogue et d'échange de messages entre les entités du réseau et en particulier entre le client et le serveur. Un système de communication inclut un protocole prédéfini qui organise l'échange des messages et leur donne une sémantique. Dans notre étude nous avons présenté le protocole SIP qui permet d'initier et d'ouvrir des sessions de communication et le protocole SDP qui permet d'aider le système à prendre des décisions grâce à sa description des média qui sont utilisés en cours de la communication. Le protocole SIP doit exister au niveau du client, le serveur SIP, et le serveur de contexte. Dans ce cas, une extension à SIP est introduite, car les messages échangés sont interprétés à l'aide des réponses SIP, ayant chacune un nouvel code d'état appartenant à la classe Informationnelle (1xx).

- 3- Un serveur de contexte, qui est l'entité du réseau dont la tâche principale est de satisfaire les requêtes clientes reçues. Généralement, satisfaire une requête revient à envoyer le contenu demandée dans la requête. Ce serveur doit être doté du protocole CC/PP pour analyser et répondre à la requête reçue, ainsi du protocole SIP pour rester en communication avec l'utilisateur. Ce serveur utilise aussi la notion des interfaces papier pour présenter ses offres; du fait que cette représentation permet l'usage du multimédia. Ce serveur consulte constamment un serveur de contenu, afin de connaître les différentes offres et les arrangements possibles qui peuvent être offerts.
- 4- Un processus de négociation, qui est le noyau du système, responsable d'appliquer une stratégie permettant de prendre la meilleure décision vis-à-vis des contraintes du contexte courant. Le processus de négociation décide si le contenu source, tout en respectant les contraintes courantes, peut être satisfait directement ou non. Dans le cas négatif, le processus de négociation échange des messages d'arrangements (négociation) avec le client, et éventuellement avec d'autres composants système, afin d'effectuer le meilleur effort pour satisfaire le contexte courant. Une fois la décision prise, le processus de négociation s'arrête et le système de communication prend en charge de délivrer la (ou les) réponse (es) au client.

4- Schéma descriptif

Dans ce qui suit nous allons présenter un schéma qui nous permet de mieux voir ce processus de négociation et ses différents composants, afin de satisfaire la requête de l'utilisateur.

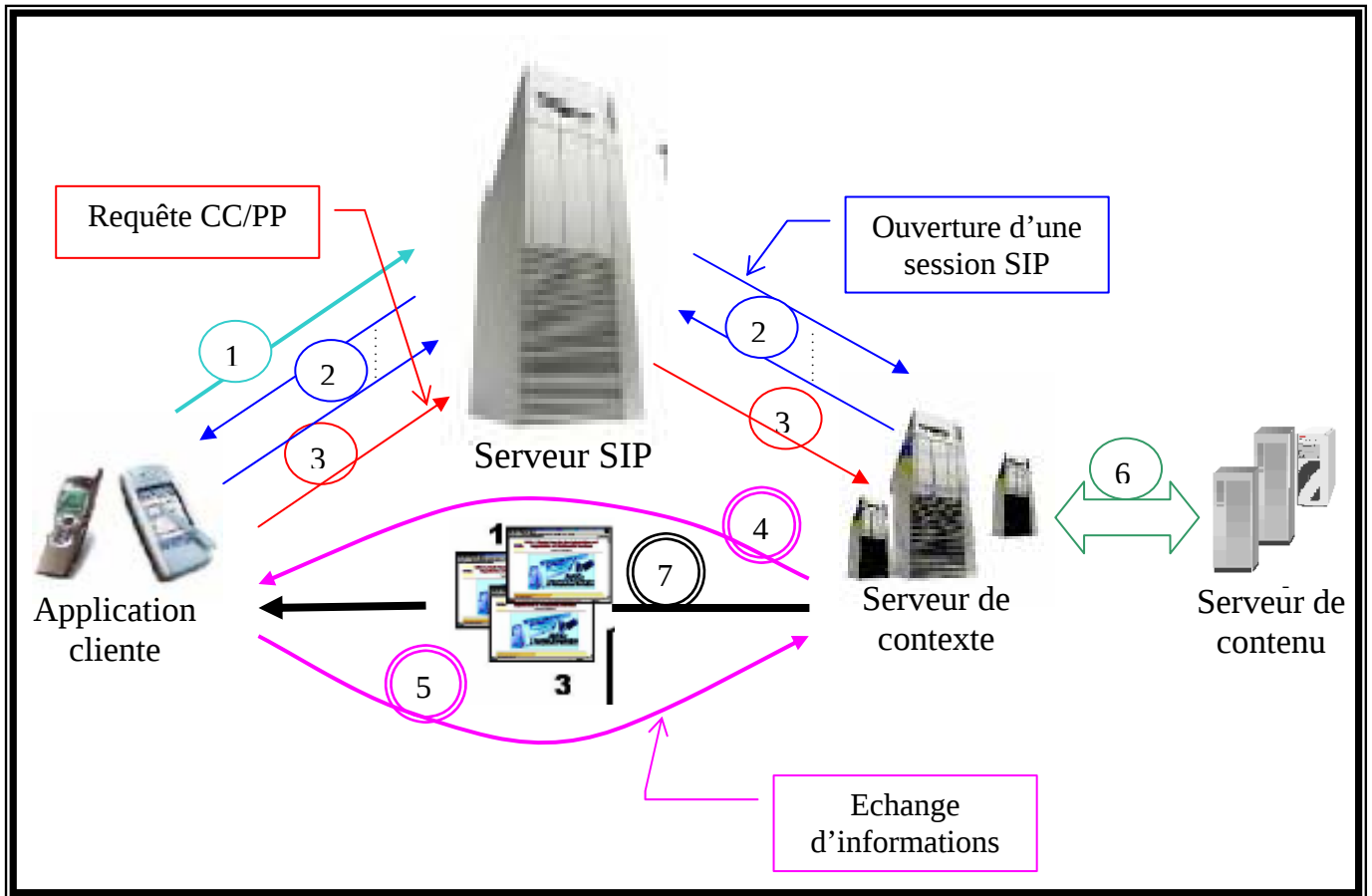


Figure 23 : Processus de fourniture de services dans le Web.

- 1- Le client émet la requête " INVITE " au serveur SIP pour inviter le Serveur de contexte à ouvrir une session de communication.
- 2- L'échange des différents messages pour l'établissement de la session en utilisant le protocole SIP (Voir le premier chapitre pour l'ouverture d'une session de communication), en utilisant aussi le protocole SDP.
- 3- Une fois la connexion est établie, le client génère sa requête CC/PP en utilisant le protocole CC/PP, qui sera vérifier à son niveau; et bien sûr en utilisant le protocole SIP et un des protocole de transmission UDP ou TCP, la requête sera transmise au serveur de contexte à travers le serveur SIP.

- 4-5 Représente le processus de négociation qui s'effectue par échange d'informations, entre le client et le serveur de contexte par l'intermédiaire du serveur SIP, afin de satisfaire le client, tout en consultant le serveur de contenu, car ce dernier décrit les différentes possibilités d'arrangements et de modification qui peuvent être introduits sur le service d'origine.
- 6- La consultation du serveur de contenu.
- 7- Le service fourni par le serveur de contenu selon le contexte final de la négociation fourni par serveur de contexte.

Le processus de négociation peut être décrit à l'aide d'une vision, qui nous permet de sortir du cadre fixé habituellement. Les vraies solutions sont à chercher en dehors de ce cadre, sans que les personnes (entité) impliqués rentrent dans le cadre donné, c'est ici que la métaphore peut nous être utile. [STP02]

5- Qu'est ce qu'une métaphore ?

Sheldon Kopp la définit comme "une façon de parler dans laquelle une chose est exprimée en termes d'une autre, jetant par ce rapprochement un nouvel éclairage sur le caractère de ce qui est décrit" [STP02].

Une métaphore est un procédé d'expression où l'on substitue au sens propre d'un mot un sens emprunté à une autre réalité qui offre avec elle des ressemblances ou des analogies [ROD99]. Mais le mot substitué n'a pas totalement disparu, il reste présent dans la contamination du sens.

La métaphore transporte la réalité dans un autre espace. Ce sera la figure idéale pour le rêveur qui ne se contente pas du réel que lui impose la perception [JSC06].

En d'autres termes, la métaphore consiste en un rapprochement de deux domaines différents d'expériences, le rapprochement prenant des formes différentes selon que les deux domaines sont syntaxiquement confrontés l'un à l'autre ou substitués l'un à l'autre. Elle consiste à utiliser un concept d'un domaine ontologique autre que celui qui est attendu.

Dans toute métaphore, un domaine d'expérience déborde sur un autre, le premier (le domaine *débordant*) sera dénommé le métaphorisant, le second (le domaine *débordé*) le métaphorisé. [CED05].

Quelques exemples :

- 1- Qualifier mon voisin de poète ne veut pas dire qu'il est écrit de la poésie, mais qu'il a le caractère du poète.
- 2- Pour évoquer la force de quelqu'un on dit : « Ce char (la personne musclée) m'a fait reculer de deux pas en arrière».
- 3- Les prix grimpent : les prix sont vus comme une entité animée et l'augmentation des prix comme un déplacement vers le haut.

6- Les éléments d'une bonne métaphore

La métaphore peut être utilisée comme un vecteur du changement. Afin qu'elle ait son plein effet, la métaphore doit alors comprendre les éléments suivants :

- 1- Une description fidèle de la situation de départ ;
- 2- une mise en scène de la "crise" qui fait désirer un changement ;
- 3- le changement souhaité, l'objectif désiré ;
- 4- la solution proposée pour atteindre cet objectif.

7- Solution métaphorique

La métaphore consiste à transposer une relation existante dans un domaine conceptuel dans un autre domaine en appliquant certaines qualités choisies de l'un sur l'autre. Cette structuration ne s'effectue pas au hasard : elle est généralement due non seulement à l'expérience physique mais aussi aux expériences culturelles et sociales.

Comme nous nous intéressons à la création d'une métaphore qui reflète le processus de négociation, on doit développer les quatre éléments cités précédemment.

- 1- Description de la situation de départ :

L'attente de la réponse (décision) du client envers les offres proposées par le serveur de contenu.

- 2- La crise :

Non satisfaction complète du client des services proposés, qui exprime l'hésitation du client entre le rejet et l'acceptation.

- 3- Le changement souhaité (objectif désiré) :

Donner une représentation ou interprétation concrète à l'état déséquilibré du client qui varie entre le rejet et l'acceptation et aboutir à une décision finale.

4- Solution proposée :

On va assimiler notre problème à une palette de couleur qui se dégrade du **vert** vers le **rouge** ; où :

Le vert : reflète l'acceptation ;

Le rouge : exprime le rejet ;

Les couleurs qui se dégradent vers le vert : indiquent le degré et le taux de satisfaction du client et qui tend vers l'acceptation ;

Cependant, les couleurs qui tendent vers le rouge : ces dernières reflètent le degré d'hésitation du client et qu'il tend vers le rejet.

La métaphore est la suivante :

« Tant qu'on est dans la zone qui est proche du vert le service est négociable, par conséquent on va atteindre le point vert; par contre, si on tend vers le rouge ou le rouge lui-même, le service est entrain d'être rejeté, sauf seulement si (les arguments, c-à-d la négociation) on fait un saut vers la zone qui se dégrade vers le vert ».

A partir de cette métaphore, on peut concrétiser ce processus sous une forme formel et applicatif, pour cela on l'a modélisé à l'aide d'un automate.

8- Diagramme de négociation

Le diagramme de négociation peut être vu comme un automate, constitué d'un ensemble d'états qui reflètent les activités du client et du serveur, et un alphabet qui représente l'ensemble d'informations échangées.

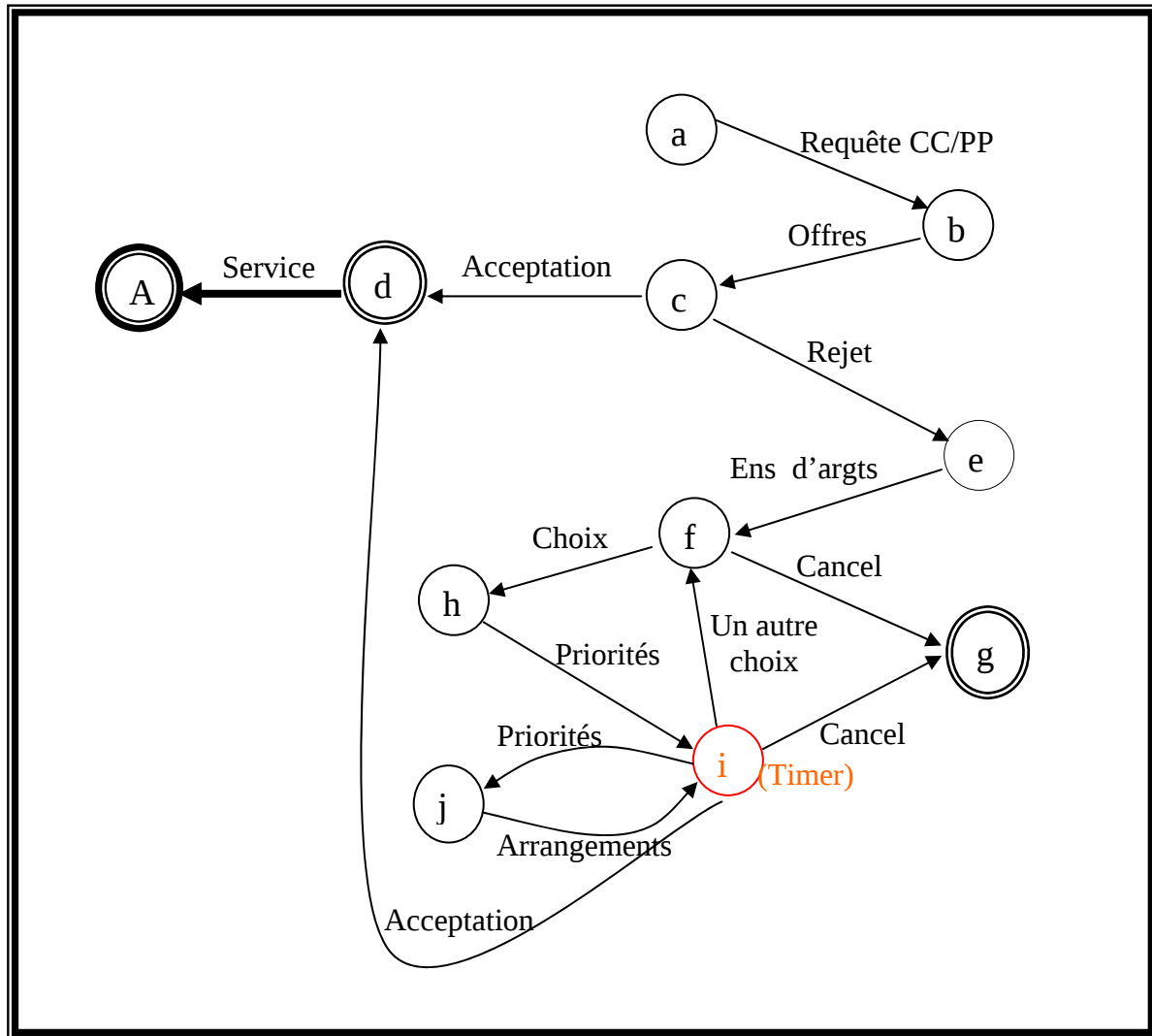


Figure 24 : Le processus de négociation.

Acceptation : est exprimé par la requête **OK** du protocole SIP.

Rejet : est exprimé lui aussi par une nouvelle réponse "**Rejet**" qui sera incorporé aux réponses SIP.

Description de l'automate :

Etat (a) : dans cet état le client génère sa requête en utilisant le protocole CC/PP et qui sera transmise au serveur du contexte en utilisant le protocole SIP.

Etat (b) : à cet état le serveur reçoit la requête du client, et en fonction de ses préférences, il lui propose une gamme de services (offres).

Etat (c) : à ce niveau , Si le client est satisfait par les offres du serveur, il émet la méthode **OK** du protocole SIP pour exprimer son accord. Sinon il émet la réponse **Rejet** (voir codes d'états de SIP et leurs extensions, Annexe A), pour exprimer la non satisfaction du client et que le service est refusé.

Etat (d) : le serveur reçoit le **OK** du client pour le service proposé. Cet état indique la fin de la négociation et qui a abouti à un choix, pour cela on le représente par un état final.

Etat (e) : lorsque le serveur du contexte reçoit le "Rejet", il ne met pas fin à la communication, mais ce dernier émet un ensemble d'arguments au client afin de le convaincre de l'intérêt des services offerts.

Etat (f) : lorsque le client reçoit les arguments envoyés par le serveur, à ce niveau ;

- Soit le client rejette toujours ces services, il envoie la requête CANCEL pour quitter la session,
- Soit le client exprime son choix parmi les offres proposées mais en précisant les difficultés (la raison pour laquelle il n'a pas choisi le service à partir du premier coup), qu'il rencontre avec son choix, par exemple les services de voyage :
 - Il peut être trop cher pour le client;
 - le client n'est pas d'accord sur le grade d'hôtel proposé ;
 - le client non disponible durant la période proposée ;
 - le véhicule du client non adéquat pour le voyage ;

Par exemple aussi dans le cas des études :

- suivre des études à distances pour les personnes qui travaillent.
-

Etat (g) : le serveur reçoit le **CANCEL** du client qui reflète le rejet total malgré les arguments proposés, ou le refus des arrangements proposés; et à ce niveau on met fin à la négociation; pour cela l'état (g) est représenté par un état final.

Etat (h) : le serveur reçoit le choix du client ainsi ses contrariétés, ce qui reflète le rejet partiel du service. Il demande au client d'exprimer ses **priorités** dans le service qu'il a choisi.

Etat (i) : le client émet au serveur ses **priorités** (voir codes d'états de SIP et leurs extensions, Annexe A).

Etat (j) : le serveur à ce niveau en fonction des informations fournies par le client, propose les différents **arrangements** (voir codes d'état de SIP et leurs extensions, Annexe A) possibles qui peuvent être introduit sur le service.

Remarque : nous remarquons bien dans l'automate qu'il y a un échange d'informations (une boucle) entre le client (état i) et le serveur (état j) ; cet échange abouti a :

- Soit à un accord, donc le client émet la requête **OK** pour exprimer sa satisfaction pour les arrangements proposés concernant le service choisi.
- Soit à un rejet total des services proposés, malgré les arrangements introduits, par conséquent le client émet la requête **CANCEL** pour mettre fin à la communication.
- Soit à un rejet total du service courant et aux arrangements proposés ; et exprime son désir pour un autre service ce qui lui permet de retourner à l'état F ;
- Soit à une boucle infinie ; pour palier à ce problème nous proposons de disposer d'un Timer (ou un compteur) au niveau du client, qui se déclenche dès la première fois qu'on sollicite l'état i, dont ce Timer aura le même rôle sur l'autre boucle qui relie l'état i, l'état f et l'état h.

Grâce à ce Timer Si au bout d'un certain temps, on ne quitte pas les deux boucles, une fermeture automatique de la session s'effectuera. On propose que le timer déclenche la fermeture de la session lorsqu'il dépasse le temps moyens de la négociation de deux choix.

Le Timer sera mis à zéro dès qu'on passe à l'état (g) ou (d).

Etat (A) : représente la partie de fourniture du service auquel a abouti la négociation et son adaptation à l'environnement de l'utilisateur.

9- Application

Notre processus de négociation peut être calqué sur des services concrets et quotidiens; par exemple les services offerts par les agences touristiques et qui les offrent via le Web, comme le montre la figure suivante :

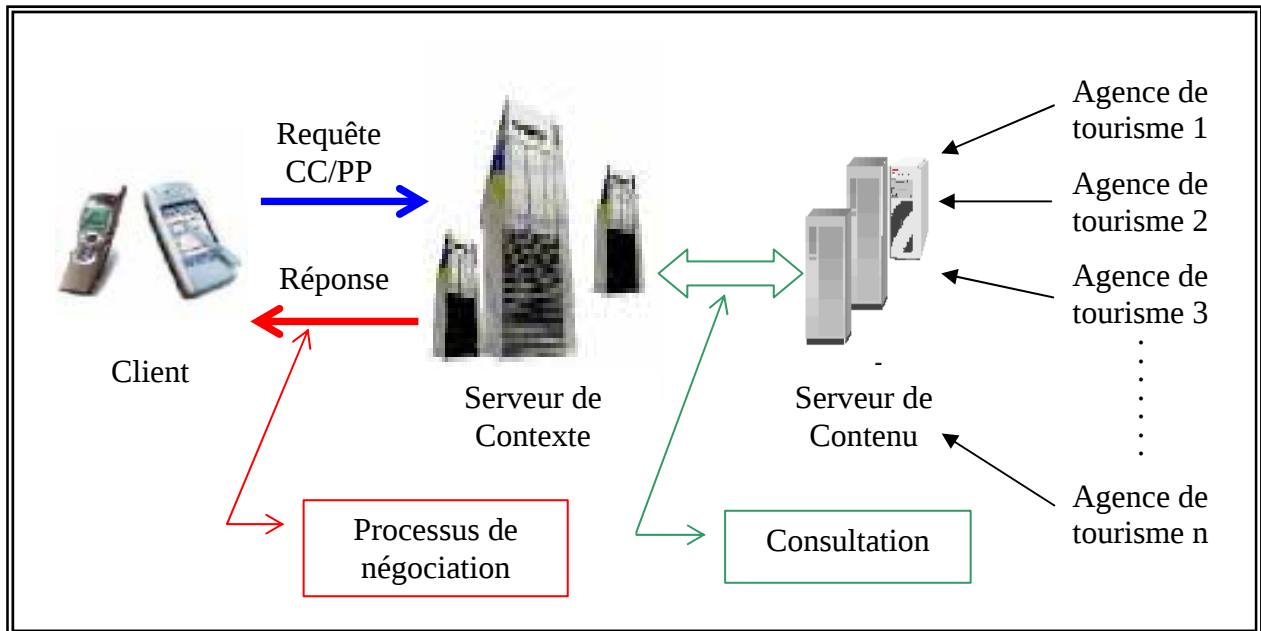


Figure 25 : L'intervention du processus de Négociation dans un service Web.

Nous allons dans ce qui suit présenter l'exécution de notre processus en utilisant le protocole SIP, afin de satisfaire un client qui veut passer des vacances à un endroit précis selon des critères souhaités.

Nous proposons dans cette solution, une extension au protocole SIP par l'introduction de nouvelles réponses (code d'état), ainsi de nouveaux attributs dans le corps du message pour décrire la requête CC/PP, pour l'exécution du processus de Négociation au sein du protocole SIP. Toute réponse utilisée dans le processus de négociation, sera ajoutée aux réponses prédéfinis de SIP.

Etape 1:

Le client émet une requête « Invite » pour l'établissement d'une session de communication SIP; comme il peut au sein du message de la requête « Invite » de préciser le sujet de la connexion est qu'il s'agit de « demande de service ».

Le message est le suivant :

```
C-> S : INVITE sip:Tourism-agence@ Holidays.agence.com SIP/2.0
Via: SIP/2.0/UDP kton.agence.com
From: A. Nourredine <sip:noualou@yahoo.fr>
To: Tourism-agence <sip:Tourism@agence.com>
Call-ID: 3298420296@north.east.isi.edu
CSeq: 1 INVITE
Subject: Demande des offres de voyages
Content-Type: application/sdp
Content-Length: 4765

v=0
o=nourredine 53655765 2353687637 IN IP4 128.3.4.5
c=IN IP4 holidays.agence.com
m=audio 3456 RTP/AVP 0 3 4 5
m=video 4003 RTP/AVP 03
```

Etape 2 :

Une fois la session est établie (voir le premier chapitre), le client émet une requête CC/PP qui décrit les caractéristiques et les critères des endroits où il veut aller.

Par exemple, le client exprime à l'aide d'une requête CC/PP le désir suivant :

« Je veux visiter une ville historique ou une cote extrêmement belle, résidé à un hôtel confortable au milieu de la ville, possédant une piscine si c'est possible, n'oubliant pas la qualité de restauration. Si c'est possible que vous me donner le planning des soirée programmées».

Le code du message ressemble à ce qui suit, notons que dans ce message nous avons introduit un nouveau code d'état **107** qui indique que c'est un profil qui est transmis :

C-> S : SIP/2.0 107 Profil CC/PP

Via: SIP/2.0/UDP kton.agence.com

From: A. Nourredine <sip:noualou@yahoo.fr>

To: Tourism-agence <sip:Tourism@agence.com>

Call-ID: 3298420296@north.east.isi.edu

CSeq: 2 INVITE

v=0

o= nourredine 53655765 2353687637 IN IP4 128.3.4.5

s= Offre de services de voyage

****u= http://example.profil.edu

a=* N = Alouache

P = Nouredine

Domaine préféré = Sport

Service = un voyage à une ville, de prix moyen

Ville = historique ou une merveille du monde

Hôtel = Bonne qualité, confortable

Nbre de voyageur = un couple et 2 enfants

c= IN IP4 holidays.agence.com

m= **audio 3456** RTP/AVP **0345**

m= **video 4003** RTP/AVP **03**

***Le profil sera stocké à l'adresse URI cité ci-dessus ou décrit par les attributs définis par l'attribut a*.

Etape 3 :

Une fois la requête reçue par le serveur de contexte, ce dernier se met en contact avec le serveur de contenu afin de proposer au client les services les plus proches de sa requête en utilisant les présentations d'interface papier qui contiennent de l'audio, d'images, et du texte descriptif. Les offres sont données dans le corps du message sous forme d'une adresse URI défini par l'attribut "u" du protocole SDP.

La réponse utilisée dans ce cas est du code d'état 300 (Multiple choices). Le code est le suivant :

S->C : SIP/2.0 300 Multiple Choices

Via: SIP/2.0/UDP kton.holidays.agence.com

From: A. Nourredine <sip:noualou@yahoo.fr>

To: Tourism-agence <sip:Tourism@agence.com>

Call-ID: 3298420296@north.east.isi.edu

Cseq: 3 INVITE

V=0

o=A.nourredine 4858949 4858949 IN IP4 192.1.2.3

s= Offre des services

i= Faites un choix parmi les offres proposées

u= <http://www.pachatour/ville/vacance.html>

c= IN IP4 Holidays.agence.com

m= audio **5004** RTP/AVP **03**

m= video **4003** RTP/AVP **03**

A l'adresse URI indiqué ci-dessus, on trouve les représentatives suivantes :

Séjours <u>Tunisie</u> Entre le 07-09-2006 et le 29-10-2006 à pd. € 255 Très bel hôtel situé en plein centre de la zone touristique Yasmine Hammamet, à environ 180 mètres de sa belle plage de sable fin et à proximité de la Marina.		Séjours <u>Istanbul</u> Entre le 11-09-2006 et le 31-10-2006 à pd. € 485 Résidence à l'hôtel Marmara, situé en plein centre de la ville. Où vous allez visiter des Mosquées historiques. Pendant ce séjour un festival est organisé.	
Île Maurice Grande Baie Hôtel Le Veranda *** 7 Jours A partir de € 868 Départs de Paris. Entre le 05-09-2006 et le 31-10-2006 Baigné par les eaux turquoise de Grand Baie dans le nord ouest de l'île Maurice.		Egypte Sharm el Sheikh Mercure Luna Sharm el Sheikh 3* 8 Jours À partir de € 598 Entre le 09-09-2006 et le 21-10-2006 Découvrez la magnifique station balnéaire de Sharm El Sheikh avec un rapport qualité prix exceptionnel!	

Figure 26 : Utilisation de la notion d'interface papier pour présenter les offres.

Selon l'automate de la négociation de services, on se trouve à l'état C; Si le client est d'accord pour une offre donnée (par exemple l'offre N°2), il émet la requête suivante:

C->S : SIP/2.0 200 OK

Via: SIP/2.0/UDP kton.holidays.agence.com
From: A. Nourredine <sip:noualou@yahoo.fr>
To: Tourism-agence <sip:Tourism@agence.com>
Call-ID: 3298420296@north.east.isi.edu
CSeq : 4 INVITE

V=0

o=A.nourredine 4858949 4858949 IN IP4 192.1.2.3

s= Offre des services de voyage

i= l'offre N°2

u= <http://www.pachatour/ville/vacance.html>

c=IN IP4 Holidays.agence.com

m=**audio 5004** RTP/AVP 03

m=**video 4003** RTP/AVP 03

Sinon, il envoie la réponse 108 pour exprimer le rejet :

C->S: Sip 2.0 108 Rejet

Via: SIP/2.0/UDP kton.holidays.agence.com
From: A. Nourredine <noualou@agence.com>
To: Tourism-agence <tourism@agence.com>
Call-ID: 33298420296@north.east.isi.edu

Lorsque le serveur reçoit la réponse "Rejet" (à l'état **e**), il envoie une réponse du code d'état 380 qui désigne d'autres services (qui correspond aux arguments de notre automate); le message sera le suivant :

S->C: SIP/2.0 380 Alternative Service

Via: SIP/2.0/UDP kton.agence.com

From: A. Nourredine <noualou@agence.com>

To: Tourism-agence <tourism@agence.com>

Call-ID: 3298420296@north.east.isi.edu

Content-Length: 725

v=0

o= nourredine 53655765 2353687637 IN IP4 128.3.4.5

s= Offre de services de voyage

***u= http://example.profil.edu

a* : s1= vous bénéficier des journées de soins

s2= réduction pour les enfants de moins de 10 ans

.....

c=IN IP4 holidays.agence.com

m=**audio 3456** RTP/AVP **0 3 4 5**

m=**video 4003** RTP/AVP **03**

Suite à la réponse 380 (à l'état **F**), le client peut toujours refusé et émettre la requête **Cancel** suivante, pour mettre fin à la communication (aller à l'etat **g**) :

C->S: CANCEL sip: kton.holidays.agence.com

Via: SIP/2.0/UDP kton.holidays.agence.com

From: A. Nourredine <noualou@agence.com>

To: Tourism-agence <tourism@agence.com>

Call-ID: 33298420296@north.east.isi.edu

CSeq: 1 CANCEL

Content-Length: 0

Comme il peut émettre la requête **Choix** pour désigner un choix et citer les points non convaincants de l'offre, comme suit :

C->S : SIP/2.0 109 Choix

Via: SIP/2.0/UDP kton.agence.com

From: A. Nourredine <sip:noualou@yahoo.fr>

To: Tourism-agence <sip:Tourism@agence.com>

Call-ID: 3298420296@north.east.isi.edu

V=0

o=A.nourredine 4858949 4858949 IN IP4 192.1.2.3

s= Offre des services

i= l'offre N° 3, mais trop cher, il n'y a pas de piscine

u= <http://www.pachatour/ville/vacance.html>

c=IN IP4 Holidays.agence.com

m=**audio 5004 RTP/AVP 0 3**

m=**video 4003 RTP/AVP 03**

Le serveur de contexte en recevant cette réponse, il envoie une réponse du code d'état **110** demandant au client de préciser ses capacités financière, et les critères prioritaires...

C->S : SIP/2.0 110 Priorités

Via: SIP/2.0/UDP kton.agence.com

From: A. Nourredine <sip:noualou@yahoo.fr>

To: Tourism-agence <sip:Tourism@agence.com>

Call-ID: 3298420296@north.east.isi.edu

V=0

o=A.nourredine 4858949 4858949 IN IP4 192.1.2.3

s= Offre des services

i= Citez vos priorités et capacités financière.

u= <http://www.pachatour/ville/vacance.html>

c=IN IP4 Holidays.agence.com

m=audio 5004 RTP/AVP 0 3

m=video 4003 RTP/AVP 03

Une fois ce message reçu, le client rentre en échange des messages de code d'état **110 (priorités)** et d'autres de code **111 (arrangements)** avec le serveur, dont les informations sont exprimées par l'attribut "i" du corps de message. Au bout d'un certain temps (timer), la fin de communication aboutit :

- Soit à un accord pour les arrangements proposés et le client émet la requête **OK**.
- Soit le client n'est pas d'accord pour les arrangements proposés, et il opte pour un autre voyage en envoyant une réponse de type **Autre Choix**.
- Soit le client n'est pas d'accord pour les arrangements, et il quitte la session en envoyant le message **CANCEL**.

Par exemple, dans le cas où le client a accepté les arrangements, on aura le service suivant offert :

Île Maurice Grande Baie Le coin de Mire 2* 7 Jours À partir de € 799 Départs de Paris. Entre le 05-09-2006 et le 31-10-2006 L' Hôtel Coin de Mire Ile Maurice : un charme fou ! Hôtel de charme dominant la lagon spectaculaire de Bain bœuf, niché entre Grand Baie et sa multitude de restaurants et de boutiques et le village de pêcheur de Cap Malheureux, havre de tranquillité!!	
--	---

Figure 27 : L'offre N° 3 après Négociation.

10- Conclusion

La négociation de services est un processus très important, qui permet de servir d'autres processus tel que l'adaptation; pour cela nous avons vu qu'on peut le modéliser à l'aide d'un automate déterministe à états finis.

Cependant nous avons aussi pu l'interpréter à l'aide d'une métaphore. Les métaphores ne sont pas seulement utiles pour la gestion de conflit ou la recherche de solutions nouvelles, mais elles rendent aussi de précieux services lors de la communication d'un changement important, lorsqu'il faut expliquer des processus complexes ou très techniques en des termes simples et compréhensibles pour chacun, ou quand il s'agit de partager sa vision avec ses collaborateurs et ses clients.

A travers l'application présentée dans ce chapitre, nous avons essayé d'éclaircir au mieux le rôle et le but de notre processus au lecteur, en l'appliquant sur un exemple concret celui d'aller en vacance.

La négociation de services dans le Web est un axe très important, qui a des effets bénéfiques sur la vie quotidienne et d'autres domaines (le côté commercial, diversités des offres, gains en temps,...). Ce processus de négociation peut être vu à travers plusieurs modèles mais le point commun est la satisfaction du client.

Une notion importante peut être incorporée à ce processus afin de lui donner une rigueur plus forte et proche de l'utilisateur est celle des web sémantiques car ces dernières permettent la génération des données sémantiques à partir de la saisie d'information par les utilisateurs. [CHN06]

Conclusion générale

L'étude du processus de négociation de service dans le web, nous a permis de découvrir d'importants protocoles de communication (SIP) et de description (SDP, CC/PP) qui peuvent être employés et étendus. Cette étude nous a permis aussi de connaître quelques applications qui ont utilisé le processus de négociation, et ses différents emplois et interprétations. Dans l'architecture NAC, la négociation consistait à trouver la meilleure présentation d'après le contexte courant et le choix de la méthode d'adaptation. Cependant, le protocole http consistait à choisir la meilleure variante d'après les entêtes « Accept », ce qui laisse toujours http limité. Par ailleurs, le protocole SIAN utilise un graphe état-transition ; il permet de présenter mieux la négociation grâce aux différents messages qui peuvent être échangés.

A travers le processus de négociation dans le Web que nous avons étudié dans ce mémoire, nous avons essayé de poser un problème important dans le Web. Il s'agit des offres de services aux clients selon leurs besoins.

Cette négociation permettra d'aboutir à un arrangement qui sera établi entre le client et le fournisseur.

Certes, ce travail nécessite une étude plus approfondie, afin d'établir un processus plus consistant ayant un enrichissement sémantique. Ceci peut être atteint en fournissant une balise de données plus riche en informations et cohérente afin de faciliter les tâches au processus de négociation. C'est une continuité des travaux déjà entrepris dans ce sens [RCO06].

Nous considérons que l'introduction du concept des **Web Sémantiques** est très important à l'interprétation des profils et les besoins du client. Cette notion permet la génération des données sémantiques à partir de la saisie d'information par les utilisateurs. [CHN06]

Ce processus joue un rôle important pour le protocole d'adaptation, car une fois le choix est fixé par le client et le fournisseur, il ne reste au protocole d'adaptation qu'à fournir le service et prendre en considération que les critères de l'environnement.

A.1- Les codes d'états et leurs extensions

Les codes de réponses sont basés sur les six types de réponses et peuvent être facilement étendus car peu de codes possibles sont définis. Il est nécessaire que les applications déterminent la classe (x) d'un code de réponse qui est indiqué par le premier chiffre, et traitent toute réponse non reconnue comme équivalente à la première réponse de la classe (le code x00) [MHE99]. Voici les codes de réponses listés dans le tableau suivant ; dont les réponses de code d'état de 107 à 112 sont une extension qui est introduite à SIP dans notre application :

Information	100 107 108 109 110 111 112 180 181 182	Trying Profil CC/PP Rejet Choix Priorités Arrangements Autre choix Ringing Call Is Being Forwarded Queued
Succès	200	OK(1)
Redirection (2)	300 301 302 303 305 380	Multiple Choices Moved Permanently Moved Temporarily See other Use Proxy Server Alternative Service

Annexe A

Erreur du client	400	Bad Request
	401	Unauthorized
	402	Payment Required(3)
	403	Forbidden
	404	Not Found
	405	Method Not Allowed
	406	Not Acceptable
	407	Proxy Server Authentication Required
	408	Request Timeout
	409	Conflict(4)
	410	Gone(5)
	411	Length Required
	413	Request Entity Too Large
	414	Request URI Too Large
	415	Unsupported Media Type
	420	Bad Extension
	480	Temporarily not available
	481	Call Leg/Transaction Does Not Exist(6)
	482	Loop Detected(7)
	483	Too Many Hops
	484	Address Incomplete
	485	Ambiguous(8)
	486	Busy Here
Erreur du serveur	500	Internal Server Error
	501	Not Implemented
	502	Bad Gateway
	503	Service Unavailable
	504	Gateway Timeout
	505	SIP Version not supported

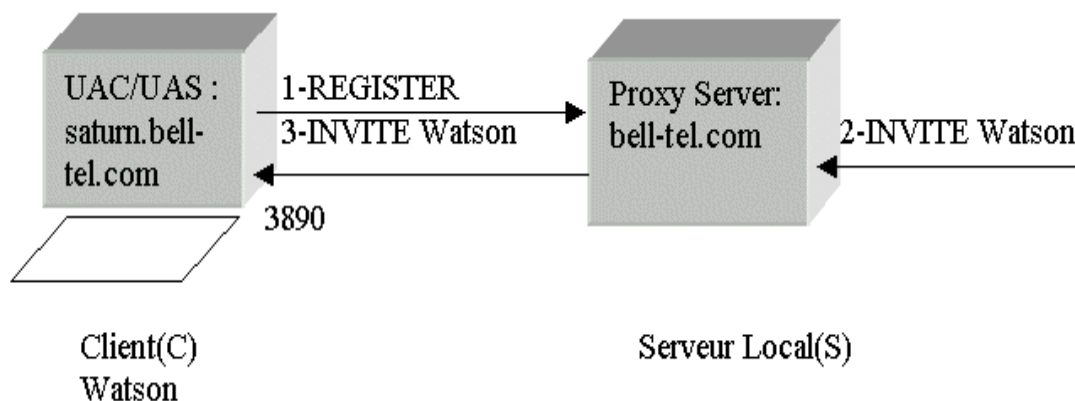
Echec général	600	Busy Everywhere
	603	Decline(9)
	604	Does not exist anywhere
	606	Not Acceptable

Tableau1 : Récapitulatif des codes d'état et des motifs pour la version SIP/2.0

A.2- Exemples de transactions SIP

1) Enregistrement :

Prenons le cas d'un utilisateur dont la machine (UAC/UAS), «saturn », est reliée au serveur local «bell-tel.com » et possède l'adresse «saturn.bell-tel.com ». Supposons que cet utilisateur souhaite s'enregistrer, pour participer à une session SIP en multicast, auprès de son serveur SIP et recevoir les requêtes/réponses SIP sur le port UDP 3890. [MHE99]



Alors, le client devra envoyer le message SIP suivant au serveur local :

```

C->S: REGISTER sip:bell-tel.com SIP/2.0
Via: SIP/2.0/UDP saturn.bell-tel.com
From: sip:watson@bell-tel.com
To: sip:watson@bell-tel.com
Call-ID: 70710@saturn.bell-tel.com
CSeq: 1 REGISTER
Contact: <sip:watson@saturn.bell-tel.com:3890;transport=udp>
Expires: 7200

```

Remarquons que cet enregistrement expirera au bout de 7200 secondes, soit 2 heures. Si une invitation à une session SIP, à destination de watson@bell-tel.com, parvient au serveur local bell-tel.com, alors elle sera redirigée vers la station «saturn» à l'adresse watson@saturn.bell-tel.com et sur le port UDP 3890. Si l'utilisateur souhaite être joignable n'importe où il doit le préciser dans les 2 dernières lignes de son message (notons que pour cette nouvelle requête REGISTER, le numéro de séquence a été incrémenté d'une unité) :

C->S: **REGISTER** sip:bell-tel.com SIP/2.0

Via: SIP/2.0/UDP saturn.bell-tel.com

From: sip:watson@bell-tel.com

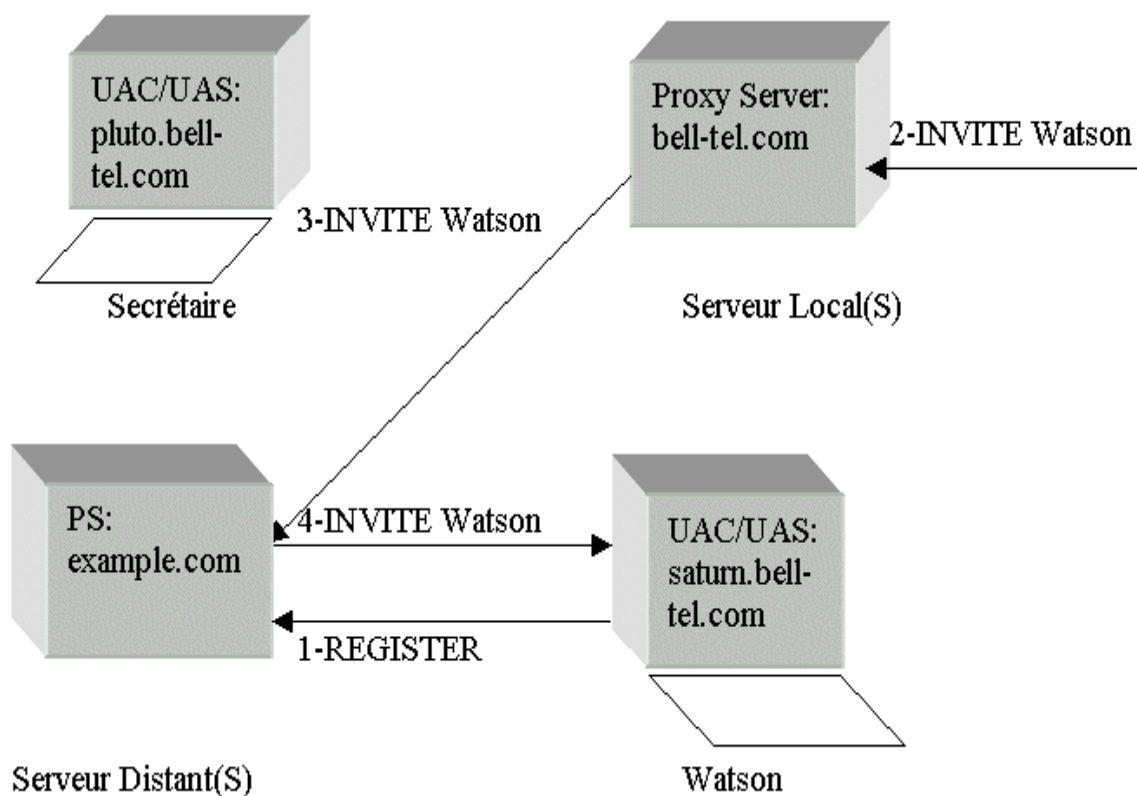
To: sip:watson@bell-tel.com

Call-ID: 70710@saturn.bell-tel.com

CSeq: 2 REGISTER

Contact: *

Expires: 0



De plus l'utilisateur doit préciser la nouvelle adresse où il souhaite être contacté dans la dernière ligne de son message :

C->S: REGISTER sip:bell-tel.com SIP/2.0

Via: SIP/2.0/UDP saturn.bell-tel.com

From: sip:watson@bell-tel.com

To: sip:watson@bell-tel.com

Call-ID: 70710@saturn.bell-tel.com

CSeq: 3 REGISTER

Contact: sip:tawatson@example.com

Maintenant, le serveur local bell-tel.com retransmettra toutes les requêtes pour Watson au serveur distant example.com en utilisant la requête URI tawatson@example.com pour le localiser. Par contre, l'utilisateur nouvellement raccordé au serveur distant devra, au préalable, envoyer une requête REGISTER au nouveau serveur local afin d'être joignable. Mais si Watson a oublié d'enregistrer sa nouvelle adresse, son secrétaire, jon.diligent peut le faire à sa place (ce cas ne figure pas sur le schéma) :

C->S: REGISTER sip:bell-tel.com SIP/2.0

Via: SIP/2.0/UDP pluto.bell-tel.com

From: sip:jon.diligent@bell-tel.com

To: sip:watson@bell-tel.com

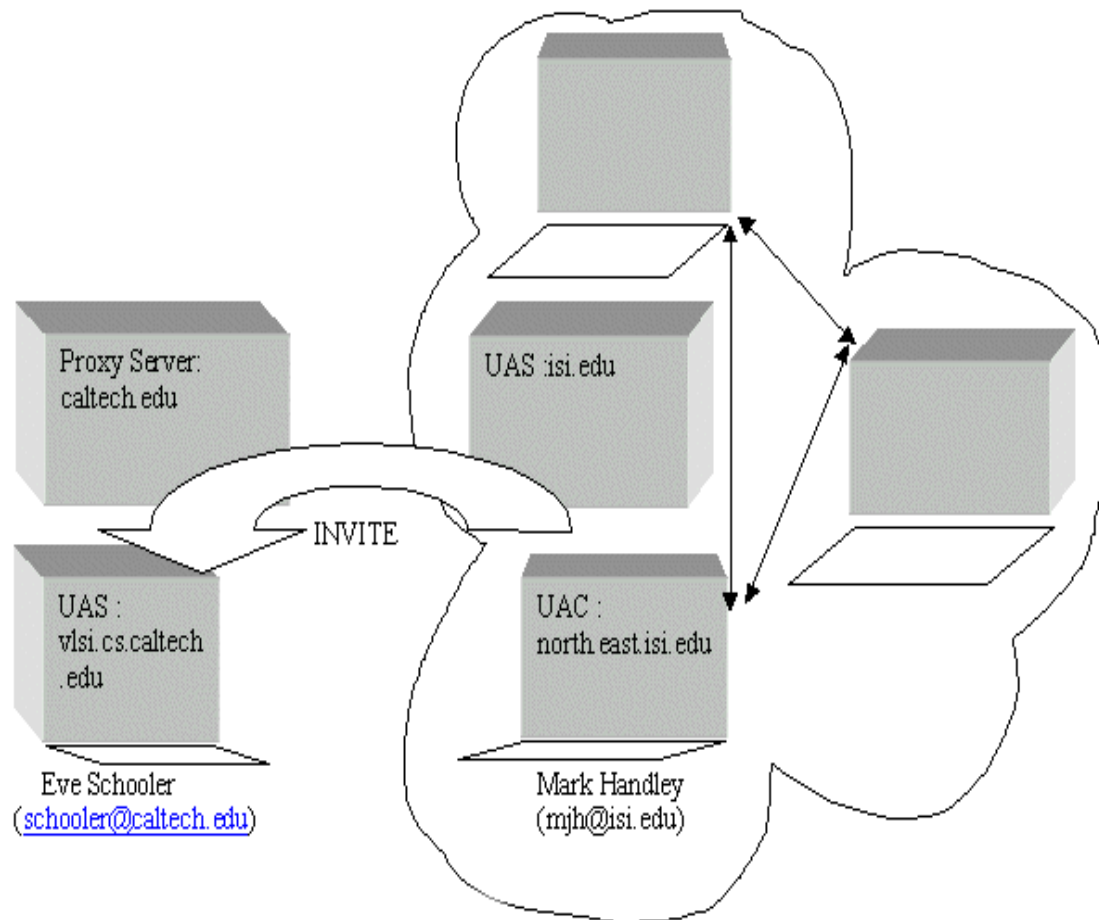
Call-ID: 17320@pluto.bell-tel.com

CSeq: 1 REGISTER

Contact: sip:tawatson@example.com

Ainsi toute requête pour Watson sera envoyée soit vers le RG (Registrar) du serveur local bell-tel.com ou vers le serveur distant example.com. Ainsi, le serveur distant (Proxy Server) transmettra la requête à l'adresse indiquée par la requête URI. Ensuite, le champ d'en-tête **Max-Forwards** pourra être utilisé pour restreindre l'enregistrement au serveur distant uniquement.

2) Invitation à une conférence en multicast:



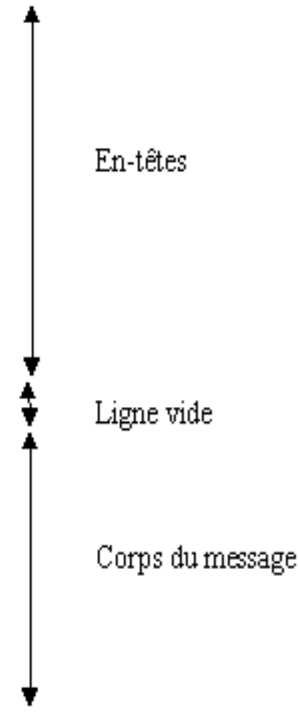
Dans cet exemple, 3 stations communiquent à travers la même session SIP multicast (SIP utilise le protocole SDP-Session Description Protocol- comme format de description de la session SIP ouverte). Supposons que les 3 participants souhaitent en inviter un 4^{ème}, schooler@caltech.edu, relié au serveur local cs.caltech.edu. L'un des participants déjà en communication, par exemple mjh@isi.edu, va adresser une requête INVITE au 4^{ème} participant, dont la trace est la suivante :

```

C->S: INVITE sip:schooler@cs.caltech.edu SIP/2.0
  Via: SIP/2.0/UDP csvax.cs.caltech.edu;branch=8348
    ;maddr=239.128.16.254;ttl=16
  Via: SIP/2.0/UDP north.east.isi.edu
  From: Mark Handley <sip:mjh@isi.edu>
  To: Eve Schooler <sip:schooler@caltech.edu>
  Call-ID: 2963313058@north.east.isi.edu
  CSeq: 1 INVITE
  Subject: SIP will be discussed, too
  Content-Type: application/sdp
  Content-Length: 187

v=0
o=user1 53655765 2353687637 INIP4 128.3.4.5
s=Mbone Audio
i=Discussion of Mbone Engineering Issues
e=mbone@somewhere.com
c=IN IP4 224.2.0.1/127
t=0 0
m=audio 3456 RTP/AVP 0

```



Les champs d'en-tête Via listent les D.N.S.(Domain Name Server) traversés le long du chemin reliant l'initiateur de la requête au destinataire de celle-ci. La requête URI indique que l'invitation est adressée à schooler@cs.caltech.edu, l'adresse courante proposée par le serveur csvax.cs.caltech.edu. L'en-tête ci-dessus est suivi d'une ligne vide et du Corps du message qui contient la description de la session.

Une fois l'UAS de l'appelé contacté, il renvoie une réponse à l'UAC de l'appelant, lui signalant qu'il sonne bien ("ringing") :

```

S->C: SIP/2.0 180 Ringing
Via: SIP/2.0/UDP csvax.cs.caltech.edu;branch=8348
;maddr=239.128.16.254;ttl=16
Via: SIP/2.0/UDP north.east.isi.edu
From: Mark Handley <sip:mjh@isi.edu>
To: Eve Schooler <sip:schooler@caltech.edu> ;tag=9883472
Call-ID: 2963313058@north.east.isi.edu
CSeq: 1 INVITE

```

Annexe A

Dès que l'invité accepte de participer à la session SIP, il retourne une réponse OK précisant dans le champ d'en-tête Contact à quel PS il est relié (es@jove.cs.caltech.edu) :

S->C: SIP/2.0 200 OK

Via: SIP/2.0/UDP csvax.cs.caltech.edu;branch=8348

;maddr=239.128.16.254;tll=16

Via: SIP/2.0/UDP north.east.isi.edu

From: Mark Handley <sip:mjh@isi.edu>

To: Eve Schooler <sip:schooler@caltech.edu> ;tag=9883472

Call-ID: 2963313058@north.east.isi.edu

CSeq: 1 INVITE

Contact: sip:**es@jove.cs.caltech.edu**

Enfin, l'appelant confirme l'invitation par une requête ACK , adressée au PS précisé dans le champ d'en-tête Contact:

C->S: ACK sip:es@jove.cs.caltech.edu SIP/2.0

Via: SIP/2.0/UDP north.east.isi.edu

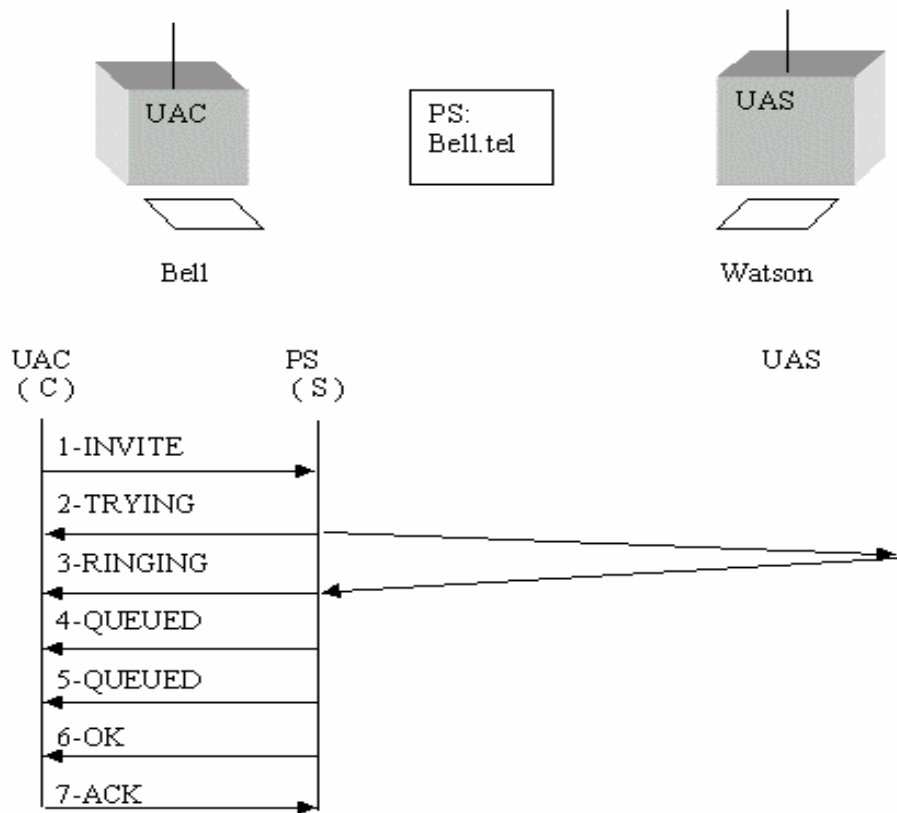
From: Mark Handley <sip:mjh@isi.edu>

To: Eve Schooler <sip:schooler@caltech.edu> ;tag=9883472

Call-ID: 2963313058@north.east.isi.edu

CSeq: 1 ACK

3) Appel entre 2 utilisateurs:



Dans le cas d'un appel téléphonique sur IP entre 2 utilisateurs (Bell et Watson), l'appelant (Bell) doit préciser dans son invitation quels types de média il supporte (dans notre exemple, il s'agit des types 0 (PCMU), 3 (GSM), 4 (G.723) et 5 (DVI4)) :

C->S: INVITE sip:watson@boston.bell-tel.com SIP/2.0

Via: SIP/2.0/UDP kton.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:watson@bell-tel.com>

Call-ID: 3298420296@kton.bell-tel.com

CSeq: 1 INVITE

Subject: Mr. Watson, come here.

Content-Type: application/sdp

Content-Length: ...

v=0

o=bell 53655765 2353687637 IN IP4 128.3.4.5

s=Mr. Watson, come here.

c=IN IP4 kton.bell-tel.com

m=**audio 3456** RTP/AVP **0 3 4 5**

S->C: SIP/2.0 100 Trying

Via: SIP/2.0/UDP kton.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311

Call-ID: 3298420296@kton.bell-tel.com

CSeq: 1 INVITE

Content-Length: 0

S->C: SIP/2.0 180 Ringing

Via: SIP/2.0/UDP kton.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311

Call-ID: 3298420296@kton.bell-tel.com

CSeq: 1 INVITE

Content-Length: 0

S->C: SIP/2.0 182 Queued, 2 callers ahead

Via: SIP/2.0/UDP kton.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311

Call-ID: 3298420296@kton.bell-tel.com

CSeq: 1 INVITE

Content-Length: 0

S->C: SIP/2.0 182 Queued, 1 caller ahead

Via: SIP/2.0/UDP kton.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311

Call-ID: 3298420296@kton.bell-tel.com

CSeq: 1 INVITE

Content-Length: 0

S->C: SIP/2.0 200 OK

Via: SIP/2.0/UDP kton.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: <sip:watson@bell-tel.com> ;tag=37462311

Call-ID: 3298420296@kton.bell-tel.com

CSeq: 1 INVITE

Contact: sip:watson@boston.bell-tel.com

Content-Type: application/sdp

Content-Length: ...

v=0

o=watson 4858949 4858949 IN IP4 192.1.2.3

s=I'm on my way

c=IN IP4 boston.bell-tel.com

m=**audio 5004** RTP/AVP **0 3**

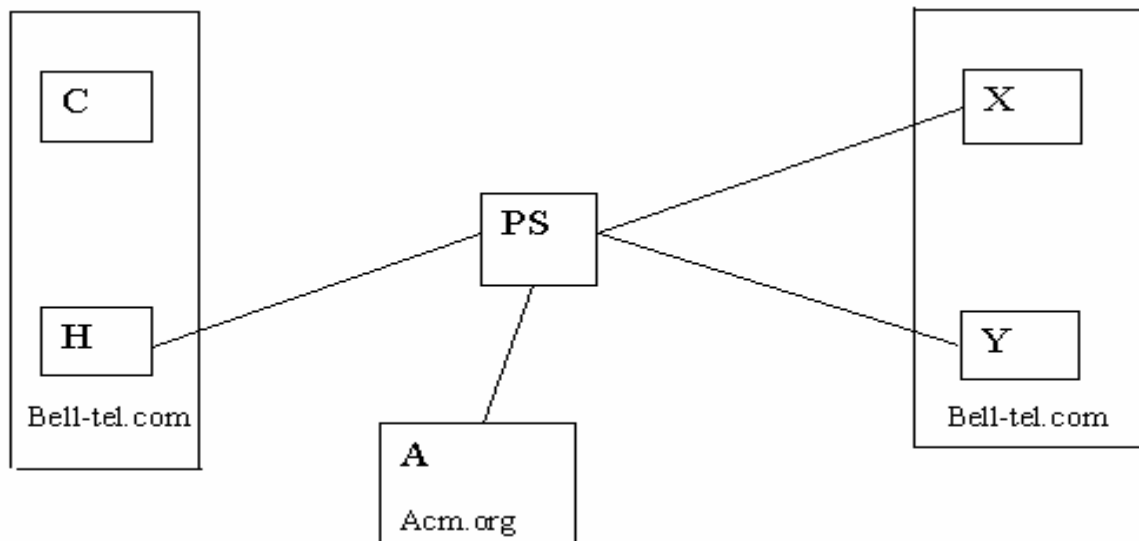
Cet exemple illustre bien l'utilisation des codes d'état dans les réponses que l'UAS retourne à l'UAC. La réception de l'appel par le PS auquel est relié l'appelant est confirmée par une réponse TRYING (code 100) et la sonnerie d'appel chez l'appelé par une réponse RINGING (code 180), puis 2 réponses signalent que l'appel a été mis en attente en précisant l'état de la file d'attente : "Queued, 2 callers ahead" et "Queued, 1 caller ahead" (code 182). Enfin, quand l'appelé décroche, une réponse OK le signale à l'appelant et précise quels types de média il supporte : 0 (PCMU) et 3 (GSM). Notons que Bell et Watson s'échangeront des données audio (voix sur IP par exemple). Watson les enverra vers le port d'entrée 3456 de la machine c.bell-tel.com de Bell, et Bell les enverra vers le port d'entrée 5004 de la machine boston.bell-tel.com de Watson. Si nos deux interlocuteurs n'avaient pas précisé les ports d'entrée et le protocole de transport des données, RTP et RTCP auraient été choisis par défaut car ils permettent de véhiculer et d'assurer le contrôle de flux des données en temps-réel. Les paquets RTCP auraient alors été reçus par les machines de Bell et Watson sur les ports d'entrée 3457 et 5005 respectivement. Une fois que les 2 locuteurs se sont accordés sur les types de média utilisés, l'appelant confirme l'appel par une requête ACK :

```
C->S: ACK sip:watson@boston.bell-tel.com SIP/2.0
Via: SIP/2.0/UDP kton.bell-tel.com
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: T. Watson <sip:watson@bell-tel.com> ;tag=37462311
Call-ID: 3298420296@kton.bell-tel.com
CSeq: 1 ACK
```

4) Terminer un appel:

Pour mettre fin à un appel, l'appelant ou l'appelé peut émettre un requête BYE :

```
C->S: BYE sip:watson@boston.bell-tel.com SIP/2.0
Via: SIP/2.0/UDP kton.bell-tel.com
From: A. Bell <sip:a.g.bell@bell-tel.com>
To: T. A. Watson <sip:watson@bell-tel.com> ;tag=37462311
Call-ID: 3298420296@kton.bell-tel.com
CSeq: 2 BYE
```

5) Duplication des requêtes par les PS:

Dans le cas où Bell souhaite appeler Watson qui est "logé" sur 2 machines simultanément - X et Y - au moment de l'appel. Seulement Watson possède 4 adresses SIP enregistrées par le Registrar du même du PS (ieee.org): A (watson@acm.org), H (watson@h.bell-tel.com), X (t.watson@x.bell-tel.com) et Y (watson@y.bell-tel.com). Bell ne travaille que sur une seule station (c.bell-tel.com) à l'adresse C (a.g.bell@bell-tel.com). L'UAC auquel Bell est relié envoie une requête INVITE à Watson à l'adresse t.watson@ieee.org reconnue par le PS de nom de domaine ieee.org :

C->PS: INVITE sip:t.watson@ieee.org SIP/2.0

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ieee.org>

Call-ID: 31415@c.bell-tel.com

CSeq: 1 INVITE

Le PS SIP essaie les 4 adresses de Watson en parallèle. Il envoie par exemple le message suivant à la machine H :

PS->H: INVITE sip:watson@h.bell-tel.com SIP/2.0

Via: SIP/2.0/UDP sip.ieee.org ;branch=1

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ieee.org>

Call-ID: 31415@c.bell-tel.com

CSeq: 1 INVITE

Mais comme Watson n'est pas logé sur cette machine actuellement, une réponse Not Found (code 404) est retournée au PS par la station appelée :

H->PS: SIP/2.0 404 Not Found

Via: SIP/2.0/UDP sip.ieee.org ;branch=1

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ieee.org>;tag=87454273

Call-ID: 31415@c.bell-tel.com

CSeq: 1 INVITE

Afin que la station H ne retransmette pas indéfiniment sa réponse, le PS lui envoie une requête ACK :

PS->H: ACK sip:watson@h.bell-tel.com SIP/2.0

Via: SIP/2.0/UDP sip.ieee.org ;branch=1

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ieee.org>;tag=87454273

Call-ID: 31415@c.bell-tel.com

CSeq: 1 ACK

Parallèlement, le PS tente de contacter Watson à l'UAS des stations A, X et Y :

PS->A: INVITE sip:watson@acm.org SIP/2.0

Via: SIP/2.0/UDP sip.ietf.org ;branch=2

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ietf.org>

Call-ID: 31415@c.bell-tel.com

CSeq: 1 INVITE

PS->X: INVITE sip:t.watson@x.bell-tel.com SIP/2.0

Via: SIP/2.0/UDP sip.ietf.org ;branch=3

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ietf.org>

Call-ID: 31415@c.bell-tel.com

CSeq: 1 INVITE

PS->Y: INVITE sip:watson@y.bell-tel.com SIP/2.0

Via: SIP/2.0/UDP sip.ietf.org ;branch=4

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ietf.org>

Call-ID: 31415@c.bell-tel.com

CSeq: 1 INVITE

Comme Watson sur la station X et l'un de ses collègues sur la station Y entendent simultanément la sonnerie d'invitation et décrochent : X et Y renvoient simultanément une réponse OK à Bell, via le PS :

X->PS: SIP/2.0 200 OK

Via: SIP/2.0/UDP sip.ietf.org ;branch=3

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ietf.org> ;tag=192137601

Call-ID: 31415@c.bell-tel.com

CSeq: 1 INVITE

Contact: sip:t.watson@x.bell-tel.com

Y->PS: SIP/2.0 200 OK

Via: SIP/2.0/UDP sip.ietf.org ;branch=4

Via: SIP/2.0/UDP c.bell-tel.com

Contact: sip:t.watson@y.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ietf.org> ;tag=35253448

Call-ID: 31415@c.bell-tel.com

CSeq: 1 INVITE

Les 2 réponses sont simultanément retransmises à Bell par le PS alors que l'UAS de la station A n'a toujours pas répondu et cherche toujours Watson dans sa base de données. Pour lui éviter cette recherche infructueuse, le PS annule sa requête INVITE vers A, ayant obtenu des réponses positives aux requêtes envoyées aux stations X et Y. Pour cela, le PS envoie une requête CANCEL à l'UAS associée à la station A :

PS->A: CANCEL sip:watson@acm.org SIP/2.0

Via: SIP/2.0/UDP sip.ietf.org ;branch=2

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ietf.org>

Call-ID: 31415@c.bell-tel.com

CSeq: 1 CANCEL

L'UAS de la station A stoppe alors sa recherche et envoie une réponse OK au PS :

A->PS: SIP/2.0 200 OK

Via: SIP/2.0/UDP sip.ietf.org ;branch=2

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ietf.org>

Call-ID: 31415@c.bell-tel.com

CSeq: 1 CANCEL

Lorsque Bell reçoit les réponses à son invitation des stations X et Y, il lui faut choisir son interlocuteur parmi Watson et son collègue. Pour cela, il acquitte les 2 appels séparément afin de localiser Watson et ne conserver que l'appel lui étant destiné :

C->X: ACK sip:t.watson@x.bell-tel.com SIP/2.0

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ietf.org>;tag=192137601

Call-ID: 31415@c.bell-tel.com

CSeq: 1 ACK

C->Y: ACK sip:watson@y.bell-tel.com SIP/2.0

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ietf.org>;tag=35253448

Call-ID: 31415@c.bell-tel.com

CSeq: 1 ACK

Après avoir discuté avec Bell et son collègue, Bell sait que Watson se trouve sur la station X et envoie une requête BYE vers la station Y afin d'en terminer l'appel. La station Y confirme la fin de l'appel en envoyant à Bell (station C) une réponse OK :

C->Y: BYE sip:watson@y.bell-tel.com SIP/2.0

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ieee.org>;tag=35253448

Call-ID: 31415@c.bell-tel.com

CSeq: 2 BYE

Y->C: SIP/2.0 200 OK

Via: SIP/2.0/UDP c.bell-tel.com

From: A. Bell <sip:a.g.bell@bell-tel.com>

To: T. Watson <sip:t.watson@ieee.org>;tag=35253448

Call-ID: 31415@c.bell-tel.com

CSeq: 2 BYE

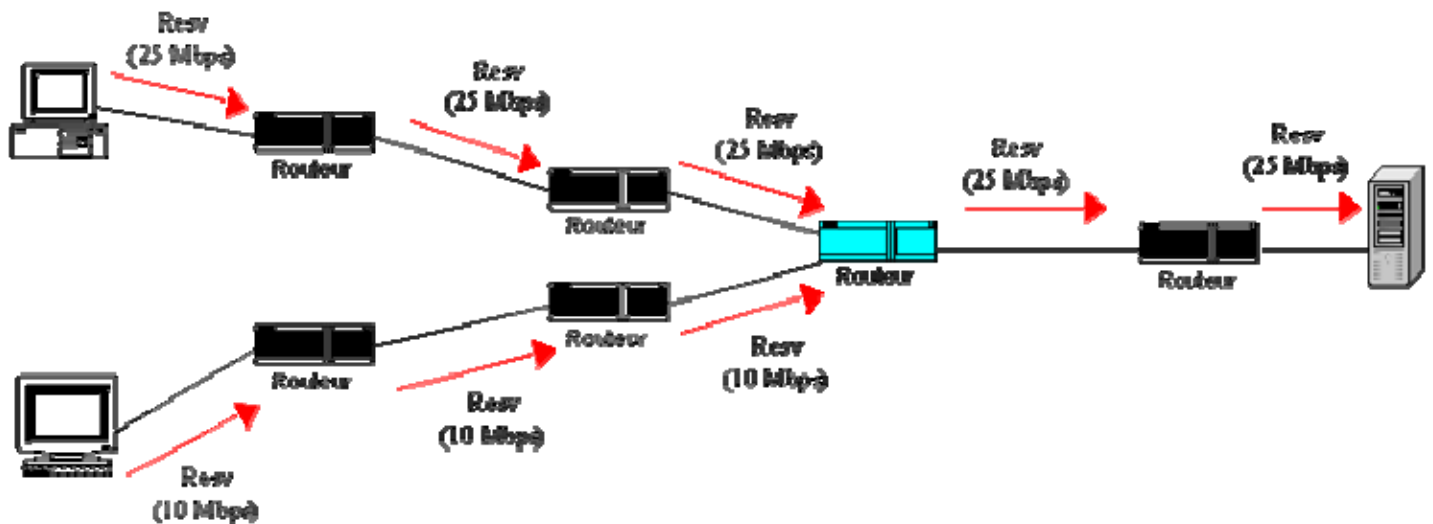
Les protocoles complémentaires du protocole SIP

B.1- Le protocole de réservation de ressources RSVP :

En 1995, Le protocole RSVP (Ressource Reservation Protocol a été développé et mis au point par Xerox PARC, le MIT ainsi que l'Information Sciences Institute et le Computer Science Department, deux entités de l'université de Californie. RSVP est un protocole de signalisation pour allouer dynamiquement de la bande passante aux applications orientées réseau dans des environnements traditionnellement datagramme. Il est particulièrement utile pour les applications multimédias de type CBR. RSVP est utilisé dans le modèle IntServ (Integrated Service) mais il peut être utilisé hors de ce contexte. RSVP est un service de type demande-acceptation : les réservations demandées par le client sont acceptées ou refusées en fonction de stratégies ou de la disponibilité des ressources réseau. [ANA01]

RSVP utilise la table de routage locale dans les routeurs pour déterminer les routes vers les destinations appropriées. La QoS étant nécessaire dans le routage, il est important de fournir une sorte d'interface entre ces deux mécanismes distincts (réservation et routage).

RSVP rend obligatoire la demande de QoS par le récepteur (l'application participante) plutôt que par l'émetteur (l'application source). Le récepteur apprend les spécifications du flux multimédia par un mécanisme hors-bande. Le récepteur peut ainsi faire les réservations qui lui sont appropriées. Cela est très utile dans le cas d'une transmission multicast. En effet, dans le cas où on aurait prévu que la demande de ressources soit faite par l'émetteur, une QoS identique à tous les émetteurs aurait été mise en œuvre et n'aurait pas été adaptée aux besoins du récepteur. D'autre part, certains émetteurs auraient eu tendance à toujours demander la réservation la plus importante qui aurait nui au système dans sa globalité. Le fait que le récepteur décide des ressources dont il a besoin permet une facturation différenciée par récepteur.



Réservation de ressources dans un flux multicast
Fusion des messages Resv

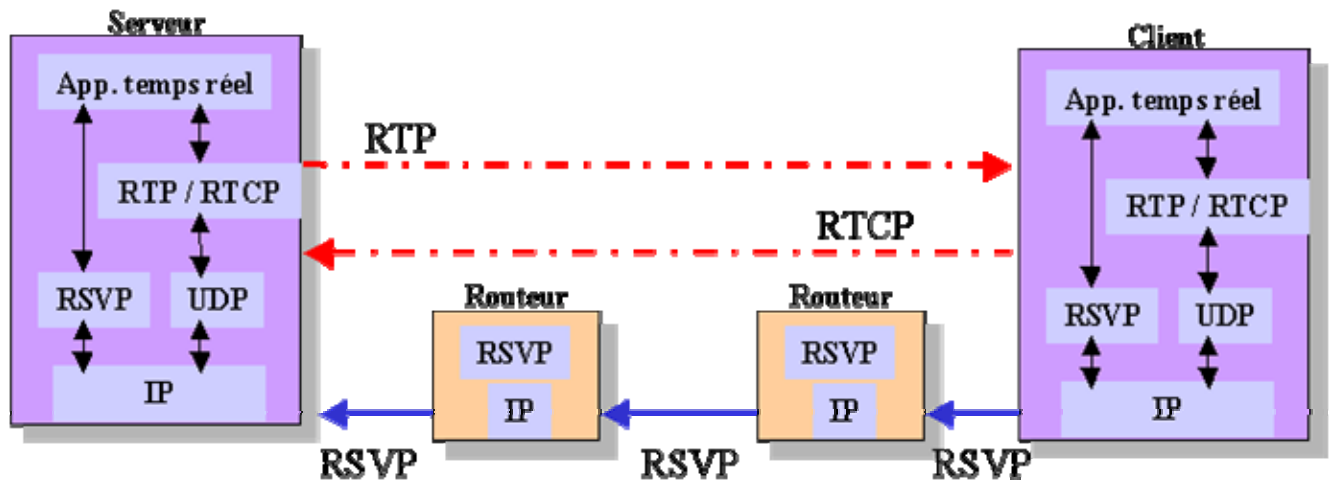
Les équipements d'interconnexions (routeurs), sur le chemin du flot des données, répondent aux requêtes RSVP, établissent et maintiennent les connexions. Les routeurs communiquent via RSVP pour initialiser et gérer la QoS réservée aux sessions.

RSVP travaille au-dessus de IP (IPv4 ou IPv6) et occupe la place d'un protocole de transport dans la pile des protocoles mais ne transporte pas de données utilisateurs. Dans les cas où le système ne permet pas l'utilisation de services réseau directement ("raw"), RSVP est encapsulé dans des paquets UDP. RSVP passe de façon transparente les routeurs non RSVP.

RSVP n'est pas un protocole de routage. Il est censé travailler avec les protocoles de routage unicast et multicast

RSVP fait des réservations de ressources pour les applications unicast et multicast et s'adapte dynamiquement aux évolutions (participants, changements de routes). Il demande des ressources dans une seule direction et traite l'émetteur et le récepteur de manière différente. Il est utilisé par un "host" pour le compte d'une application, pour demander une QoS au réseau (bande passante garanti, débit crête,..). Il est utilisé par les routeurs pour le contrôle de la QoS et l'établissement et maintien du service demandé.

RSVP peut être utilisée avec RTP/RTCP mais ce n'est pas obligatoire.



Une proposition du groupe de travail RSVP de l'IETF définit une interface entre RSVP et routage appelée RSRR : Routing Support for Routing Reservation. La communication entre ces deux protocoles est réalisée via l'échange de questions et réponses asynchrones. Cette interaction permet à RSVP de s'adapter aux modifications de routage.

- **Types de messages RSVP :**

Sept types de messages RSVP ont été prévus :

- Path : envoyé par la source pour indiquer la liste des routeurs du chemin suivi par les données ;
- Resv : demande de réservation ;
- PathErr : message d'erreur concernant le chemin ;
- ResvErr : message d'erreur de demande de réservation ;
- PathTear : indique aux routeurs d'annuler les états concernant la route ;
- ResvTear : indique aux routeurs d'annuler les états de réservation (fin de session);
- ResvConf (optionnel) : message de confirmation envoyé par le routeur au demandeur de la réservation ;

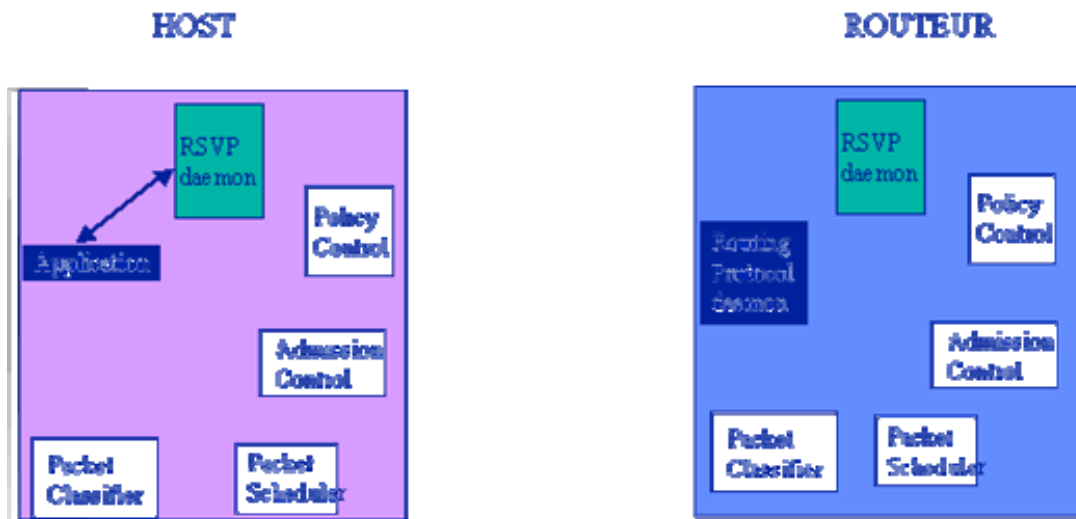
Un message RSVP est constitué d'un en-tête et d'un nombre variable d'objets qui dépend du type du message.

L'entête est constitué de 64 bits :

Vers	Flags	Type du Msg	Checksum
Send_TTL		Réservé	Longueur Msg.

- Vers (4 bits) : version du protocole RSVP (=1) ;
- Flags (4) : non utilisé à ce jour;
- Type de Msg (8) : 1 à 7 selon le type ci-dessus;
- Checksum (16) : Contrôle d'erreurs ;
- Send_TTL (8) : valeur du TTL IP à comparer avec le TTL du paquet IP pour savoir s'il y a des routeurs non-RSVP ;
- Longueur (16) : longueur du message en octets (en-tête et objets).

1) Le récepteur envoie la requête de QoS au démon RSVP local



2) La requête Qos est passée à 2 modules de décision :

- Admission Control : qui détermine si les ressources sont suffisantes en fonction de la requête (débit, taux de pertes...)
- Policy Control : qui vérifie que l'utilisateur a l'autorisation administrative de faire cette réservation. Il authentifie le demandeur et permet la facturation.

Si un des modules a une réponse négative alors envoie d'une notification d'erreur au demandeur.

3) Envoie des paramètres de QoS aux deux modules de gestion de flots : Packet Classifier et Packet Scheduler

- Le packet Classifier trie les paquets. Il détermine la route et la classe de QoS pour chaque paquet entrant.

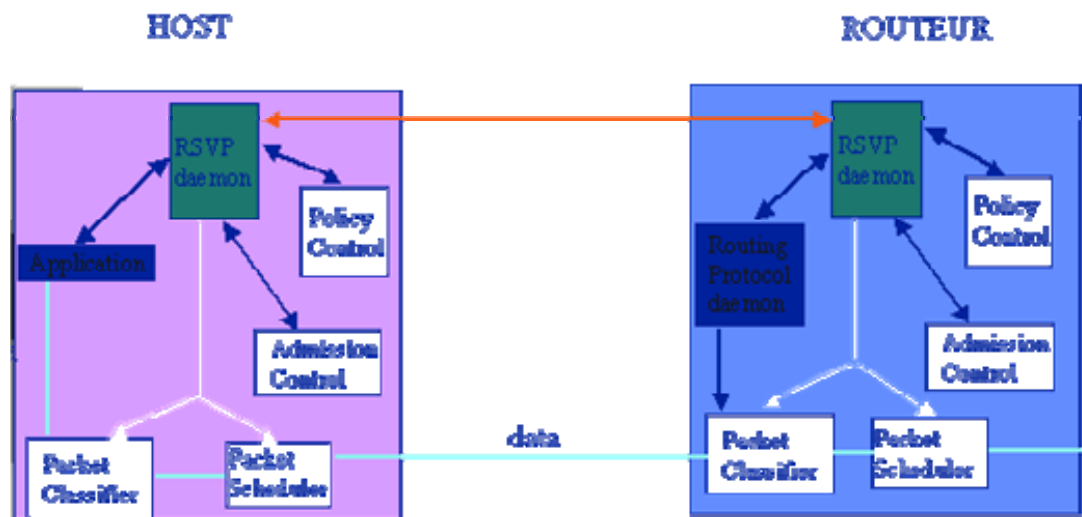
- Le Packet Scheduler prend la décision de transmettre chaque paquet en choisissant sa place dans la file d'attente correspondant au service demandé. Il peut demander une allocation supplémentaire du temps CPU ou une augmentation de la taille de mémoire tampon dans le routeur...

4) Le scénario se reproduit pour le "hop" suivant :

Un cache pour le contrôle du trafic est construit dans chaque routeur. Pour répondre aux modifications des sessions multicast par exemple, RSVP envoie des messages de rafraîchissement le long du chemin.

En l'absence de messages de rafraîchissement, le cache relatif à une réservation est détruit.

[RSV]



B.2- Les protocoles de transfert RTP et RTCP :

RTP (Realtime Transport Protocol) et son compagnon RTCP (Realtime Transport Control Protocol) permettent respectivement de transporter et de contrôler des flots de données qui ont des propriétés temps-réel.

Le but de RTP est de fournir un moyen uniforme de transmettre sur IP des données soumises à des contraintes de temps réel (audio, vidéo, ...). Le rôle principal de RTP consiste à mettre en œuvre des numéros de séquence de paquets IP pour reconstituer les informations de voix ou vidéo même si le réseau sous-jacent change l'ordre des paquets.

Plus généralement, RTP permet :

- d'identifier le type de l'information transportée,
- d'ajouter des marqueurs temporels et des numéros de séquence l'information transportée,
- de contrôler l'arrivée à destination des paquets.

De plus, RTP peut être véhiculé par des paquets multicast afin d'acheminer des conversations vers des destinataires multiples.

Le protocole RTCP est basé sur des transmissions périodiques de paquets de contrôle par tous les participants dans la session.

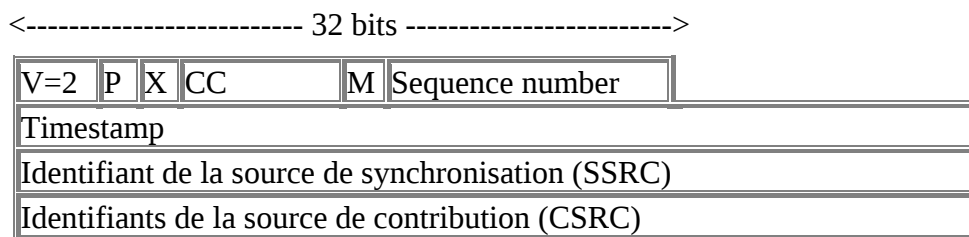
C'est un protocole de contrôle des flux RTP, permettant de véhiculer des informations basiques sur les participants d'une session, et sur la qualité de service.

RTP et RTCP sont des protocoles qui se situent au niveau de l'application et utilisent les protocoles sous-jacents de transport TCP ou UDP. Mais l'utilisation de RTP/RTCP se fait généralement au-dessus de UDP.

RTP et RTCP peuvent utiliser aussi bien le mode Unicast que le mode Multicast.

• Format des en-têtes et leurs contenus :

L'en-tête RTP comporte les informations suivantes :



- **Le champ Version V** : de longueur 2 bits, indique la version du protocole (V=2),
- **Le champ padding P** : de longueur 1 bit, si P est égal à 1 alors le paquet contient des octets additionnels de bourrage (padding) pour finir le dernier paquet,
- **Le champ extension X** : de longueur 1 bit, si X=1 l'en-tête est suivi d'un paquet d'extension,
- **Le champ CSRC count CC** : de longueur 4 bits, contient le nombre de CSRC qui suivent l'entête
- **Le champ marker M** : de longueur 1 bit, son interprétation est définie par un profil d'application (profile),
- **Le champ payload type PT** : de longueur 7 bits, ce champ identifie le type du payload (audio, vidéo, image, texte, html, etc.),
- **Le champ séquence number** : de longueur 16 bits, sa valeur initiale est aléatoire et il s'incrémente de 1 à chaque paquet envoyé, il peut servir à détecter des paquets perdus,
- **Le champ timestamp** : de longueur 32 bits, reflète l'instant où le premier octet du paquet RTP a été échantillonné. Cet instant doit être dérivé d'une horloge qui augmente de façon monotone et linéaire dans le temps pour permettre la synchronisation et le calcul de la gigue à la destination,
- **Le champ SSRC** : de longueur 32 bits, identifie de manière unique la source, sa valeur est choisie de manière aléatoire par l'application. Le champ SSRC identifie la source de synchronisation (ou dit simplement "la source"). Cet identificateur est choisi de manière aléatoire avec l'intérêt qu'il soit unique parmi toutes les sources d'une même session La liste des CSRC identifie les sources (SSRC) qui ont contribué à l'obtention des données contenues dans le paquet qui contient ces identificateurs. Le nombre d'identificateurs est donné dans le champ CC,
- **Le champ CSRC** : de longueur 32 bits, identifie les sources contributantes.

L'en-tête RTCP comporte les informations suivantes :

- **Le champ version** (2 bits),
- **Le champ padding** (1 bits),
- **Le champ reception report count** (5 bits): nombre de comptes-rendus dans le paquet,
- **Le champ packet type** (8 bits) 200 pour SR,
- **Le champ length** (16 bits) longueur du paquet en mots de 32 bits,

- **Le champ SSRC** (32 bits): identification de la source spécifique à l'émetteur,
- **Le champ NTP timestamp** (64 bits),
- **Le champ RTP timestamp** (32 bits),
- **Le champ sender's packet count** (32 bits),
- **Le champ sender's octet count** (32 bits) statistiques,
- **Le champ SSRC-n** (32 bits) numéro de la source dont le flux est analysé,
- **Le champ fraction lost** (8 bits),
- **Le champ cumulative number of packets lost** (24 bits),
- **Le champ extended highest sequence number received** (32 bits),
- **Le champ interarrival jitter** (32 bits). C'est une estimation de l'intervalle de temps d'un packet de données RTP qui est mesuré avec le timestamp et qui est sous forme d'un entier. C'est en fait le temps relatif de transit entre deux paquets de données. La formule pour le calculer est : $J = J + (|D(i-1,i)| - J) / 16$. L'interarrival jitter est calculé à chaque packet de donnée reçu par la source SSRC_n .
i --> Premier paquet
i-1 --> paquet précédent
D --> différence
J --> Second paquet,
- **Le champ last SR timestamp** (32 bits),
- **Le champ delay since last SR** (32 bits).

RTCP est un protocole de contrôle associé à RTP, il mesure les performances, par contre il n'offre pas de garantie. Pour cela il faut, employer un protocole de réservation du type RSVP ou bien s'assurer que les liens de communications utilisés sont correctement dimensionnés par rapport à l'utilisation qui en est faite.

RTP/RTCP est au-dessus du transport UDP/TCP, mais pratiquement au-dessus de UDP. RTP est un protocole de session, mais il est placé dans l'application. C'est au développeur de l'intégrer.

RTP n'a rien à voir avec le type de flux, il est au-dessus de UDP lui-même au-dessus de IP. Le type de flux est théoriquement utilisé dans IP. RTP apporte un numéro de séquence, un timestamp et un identificateur unique de la source (SSRC).

C.1- Sommaire de la hiérarchie de classes CC/PP

rdfs:Resource	
ccpp:Profile	{Profil livrable au serveur origine}
ccpp:Component	
rdfs:Literal	
ccpp:string	{Une valeur textuelle d'un attribut CC/PP}
ccpp:integer	{Une valeur entière d'un attribut CC/PP}
ccpp:rational	{Une valeur rationnelle d'un attribut CC/PP}
rdf:Bag	{Un ensemble de valeurs pour un attribut CC/PP}
rdf:Seq	{Une séquence de valeurs pour un attribut CC/PP}
rdf:Property	
ccpp:Property	{Une propriété appliquée à une instance ccpp:Resource}
ccpp:Structure	{Une propriété structurelle dans un profil CC/PP}
ccpp:Attribute	{Une propriété dénotant un attribut CC/PP}

Figure C-1 : La hiérarchie de classes CC/PP

C.2- La notation de graphe RDF

La structure sous-jacente de RDF est un graphe étiqueté orienté. RDF emploie une sérialisation XML pour représenter ces graphes dans les échanges entre systèmes informatiques. Cette notation XML est plutôt volumineuse et difficile pour le discours humain, de sorte qu'ici on emploie une notation plus visuelle pour décrire les structures des graphes RDF, [JJS04] :

[Ressource-sujet] --nomDePropriété--> [Ressource-objet]

Désigne une arête de graphe étiquetée par 'nomDePropriété' depuis une ressource RDF nommée 'Ressource-sujet' vers une autre ressource RDF nommée 'Ressource-objet'.

[Ressource-sujet] --nomDePropriété--> "Valeur de propriété"

Désigne une arête de graphe étiquetée par 'nomDePropriété' depuis une ressource RDF nommée 'Ressource-sujet' vers une chaîne littérale contenant la valeur indiquée.

[Ressource-sujet] --nomDePropriété--> { "Valeur1", "Valeur2", ... }

Ceci est un raccourci pour une propriété dont la valeur est une ressource *rdf:Bag* contenant les valeurs indiquées.

[<Type-du-sujet>] --nomDePropriété--> [<Type-de-l-objet>]

Les noms entre crochets représentent une ressource RDF du type indiqué (c'est-à-dire ayant la valeur de propriété indiquée pour *rdf:Type*, sans indication d'un nom particulier pour la ressource. C'est utile pour montrer les classes RDF qui peuvent être reliées par une propriété.

[Ressource-sujet] --nomDePropriété--> [Ressource-objet]

```

      |
      |
      |-----
      |
      | +--propriété1--> (valeur1)
      | +--propriété2--> (valeur2)
      | :
      | (etc.)

```

Les arcs de propriété peuvent s'enchaîner et plusieurs arcs peuvent partir d'une ressource sujet.

Figure C-2 : Notation d'un graphe RDF

C.3- La conformité du document CC/PP

Les documents peuvent exister comme ressources accessibles via un URL ou peuvent être transmis comme données dans un message. Un document est conforme à CC/PP quand il satisfait aux critères suivants, [JJS04] :

1. Le document DOIT être valide en RDF et sérialisé en XML, et se baser sur un ou plusieurs vocabulaires dérivés du schéma RDF.
2. Le document DOIT employer une syntaxe valide pour les déclarations d'espace de nommage.
3. L'élément profil DOIT contenir un ou plusieurs composants.
4. Chaque composant dans le profil DOIT contenir un ou plusieurs attributs.

5. Les noms de composant PEUVENT se trouver dans les attributs *rdf:about* ou *rdf:ID*.
6. Les composants DOIVENT être indiqués à l'aide d'une propriété *ccpp:component*, dans lequel l'espace de nommage utilisé pour qualifier le composant se trouve dans l'espace de nommage CC/PP ou UAProf.
7. Les noms de composant, les types de composant et les noms d'attribut doivent tous se rapporter aux différents URI au sein d'un profil.
8. Si on donne un type de composant comme nom d'un élément et comme élément *rdf:type*, ceux-ci doivent se rapporter au même URI.
9. Les références des composants par défaut DOIVENT être des URL valides.
10. Les composants par défaut PEUVENT s'écrire *ccpp:defaults* ou *ccpp:Defaults*.
11. Les composants par défaut DOIVENT être indiqués à l'aide des propriétés *ccpp:defaults* ou *ccpp:Defaults*, pour lesquelles l'espace de nommage utilisé pour qualifier defaults ou Defaults est l'espace de nommage CC/PP ou UAProf.
12. Les attributs d'un composant PEUVENT contenir à la fois une valeur par défaut et une valeur appliquée, cette dernière valeur ayant préséance.
13. Les composants PEUVENT contenir des valeurs par défaut incorporées.
14. Les composants NE DOIVENT PAS contenir à la fois des valeurs par défaut incorporées et des valeurs par défaut référencées.
15. Les composants PEUVENT référencer un document par défaut qui n'a pas d'attribut *rdf:type*.
16. Les attributs PEUVENT avoir des ensembles de valeurs (Bag).
17. Les attributs PEUVENT avoir des séquences de valeurs (Seq).
18. Les attributs PEUVENT avoir des valeurs chaîne de caractères.
19. Les attributs PEUVENT avoir des valeurs entières.
20. Les attributs PEUVENT avoir des valeurs rationnelles.
21. Un composant NE DOIT PAS contenir plusieurs attributs avec le même nom.
22. Des attributs ayant le même nom PEUVENT se trouver dans des composants différents.
23. Les profils PEUVENT utiliser plusieurs espaces de nommage pour les attributs.

C.4- Les valeurs par défaut

Un document par défaut utilise un élément *rdf:Description* pour nœud racine. L'instance *rdf:Description* est nommée à l'aide d'un attribut *rdf:about* dont la valeur est un URI (Uniform Ressource Identifier). Celui-ci DOIT correspondre à la valeur de l'attribut XML *rdf:resource* de l'élément *ccpp:defaults* dans le document appelant. Les URL utilisés dans les exemples de documents par défaut cités dans ce mémoire se sont des URL des documents de valeurs par défaut externes. Cependant, l'URI de la ressource par défaut n'est pas nécessairement l'URL (Uniform Ressource Locator) du document, aussi longtemps que l'URI est identifié de manière unique, que le même URI est utilisé dans le document source et le document de valeurs par défaut externe, et que le logiciel de traitement peut localiser et ramener le document contenant la ressource par défaut [JJS04].

Une ressource indiquée par une propriété par défaut peut apparaître dans un document séparé, auquel cas l'utilisateur devrait donner une référence d'URI **absolue** pour la ressource par défaut. Dans ce cas, la partie URI de l'identificateur de ressource par défaut (c'est-à-dire, en excluant la partie identificateur de fragment) est utilisée pour rechercher un document RDF contenant la description de la ressource par défaut. Par exemple, si la ressource par défaut se nomme <http://example.com/profilMateriel#PlateformeMateriel>, alors l'adresse URI <http://example.com/profilMateriel> sera utilisée pour rechercher un document RDF et, au sein de ce document, la ressource ayant pour identificateur local [#PlateformeMateriel](#) sera retenue comme ressource par défaut. (Une telle ressource au sein du document cible pourrait être définie par `about='http://example.com/profilMateriel#PlateformeMateriel'` ou `ID='PlateformeMateriel'`).

Bibliographie

- ANA01** : André Aoun, "IntServ et RSVP", Université Paul Sabatier (Toulouse) 2001, édition du 08/12/2001, <http://www.httr.ups-tlse.fr/pedagogie/cours/tcp-ip/rsvp/>
- AND03** : Antoine Delley, <<Réseaux IP- Voix et Multimédia IP>>, 2003
[http:// ditwww.epfl.ch/SIC/SA/publications/FI03/fi-1-3/1-3-page3.html](http://ditwww.epfl.ch/SIC/SA/publications/FI03/fi-1-3/1-3-page3.html)
- CED05** : Cédric Detienne, « La métaphore : produit, processus et producteur », édition Février 2005; http://www.fabula.org/atelier.php?La_m%26eacute%3Btaphore_%3A_produit%2C_processus_et_producteur
- CHN06** : [CharlesNepote](#), « Web Sémantique », <http://www.soapuser.com/fr/basics2.html>, Edition de 28-11-2006.
- CHS04** : M. Christian Tchepnda, M. Hervé Stéphane Mbida, " Interface papier pour des présentations multimédias", Mémoire de fin d'études, USTHB 2004.
- COL04** : Corre Olivier, « Le protocole d'Initiation de Session », édition 2004, <http://www-rp.lip6.fr/~corre/SIP-EXPOSE.pdf>
- DAD04** : Da Silva Alexandre et Decaillon Tony, «SDP: Session Description Protocol», <http://www-rp.lip6.fr/~corre/sdp.doc>
- FBE03** : Frédéric Bergé, «Le protocole SIP brigue la téléphonie Internet», 01 Réseaux, édition du 28/05/2003, <http://www.01net.com/article/208827.html>
- HAJ06** : M.Handley, V.Jacobson, "SDP: le Protocole de Description de Session » ; Traduit par Claude Brière de l'Isle le 04/2006; <http://abccdrfc.free.fr/rfc-vf/rfc2327.htm>
- JJS04** : Grahm Klyn, Franklin Reynolds, Chris Woodow, Hidetaka Ohto, Johan jelem, Mark H.Butker, Luu Tran; "Les profiles composites de capacités /préférences (CC/PP) structure et vocabulaire 1.0"; édition Janvier 2004; <http://www.w3c.com>
- JSC06** : JSCPA lyon « Ecole de communication et de journaliste », « Les métaphores », N°307139 ; <http://jaseur.free.fr/jaseur/metaphore.htm>
- JYM04** : Janet Daly, Yasuyuki Hirakawa, Marie-Claire Forgue, "Le W3C publie la recommandation CC/PP 1.0", édition Janvier 2004; www.w.org/2004/01/ccpp-release.HTML.fr

- LAK01** : M. Layaida, M. Kaced, «PFE : Protocole de signalisation SIP et services multimédias pour la troisième génération de téléphone mobiles », mémoire de fin d'études, USTHB 2001.
- MAJ06** : Malcolm Johnson , « ITU-T Recommendation H323 », <http://www.itu.int/ITU-T>
- MHE99**: M.Handley, H.Schulzrinne E. Schooler et J.Rosenberg, «SIP: Session Initiation protocol», Request for comment 2543, Internet Engineering Task Force, Mars 1999, <http://www.IETF.com>
- NIF04** : Nicolas FISCHBACH, « Sécurité de la voix sur IP (VOIP) », Acte du symposium SSTIC04, édition 2004, <http://actes.sstic.org/SSTIC04/VOIP/SSTIC04-article-Fischbach-VOIP.pdf#search=%22type%20de%20securit%C3%A9%20SIP%22>
- NIB05** : Nicolas Bareil, " projet Ilty : I'm listing to you (via VoIp), Acte du symposium SSTIC05, édition 2005, http://actes.sstic.org/SSTIC05/ILTY_Im_Listening_to_you_via_VOIP/SSTIC05-article-Bareil-VOIP_Projet_ILTY.pdf#search=%22les%20protocole%20de%20signalisation%22
- NIQ06** : Nicholas Quaine, « Web Services (les services web) et le Service Web (le Web des Services) », <http://www.soapuser.com/fr/basics2.html>.
- NJJ06** : Nico VanHaute, Julien Barascud et Jean-Roland Conca, "Internet-Les protocoles RTP/RTCP", Issu de l'encyclopédie Informatique : Comment ça marche 2006; <http://www.commentcamarche.net/internet/rtcp.php3>
- NTV05** : Nabil Layaïda, Tayeb Lemlouma, Vincent Quint ; « NAC, une architecture pour l'adaptation multimédia sur le web », édition 2005, <http://wam.inrialpes.fr/publications/2005/TSI-NAC.pdf>
- OLF04** : Olivier Fonty, "Voix sur Ip, Protocole et Standart", paru le 27/04/2004, http://www.supinfo-projects.com/fr/2004/voix_sur_ip/4/
- ORL99** : Ora Lassila, Ralph R.Swick, "Spécification du modèle et la syntaxe du cadre de description des ressource", www.w3.org/TR/1999/REC-rdf-syntax-19990222
- PIR03** : Pierre Ruffez, « Etude de la négociation dans le cadre d'un jeu de rôle sur le Sylvopastoralisme », Rapport de stage de DEA IARFA, Soutenu le 17 Septembre 2003, <http://cormas.cirad.fr/pdf/deaRuffez.pdf>

- PMV06** : Phillippe Mathieu, Marie-Hélène Verrons, « Resolution par l'AGL de négociation ANTS », http://www2.lifl.fr/~mathieu/rapport_edt.pdf
- RCO06** : R.C De Olivera, « Adaptation de contenu par transformation et gestion de la qualité de service pour des machines hétérogènes », édition 2006, <http://194.199.20.6/prog-recherche.htm>
- ROD99** : Robert Dubuc, « Au plaisir des mots ; Feu vert aux métaphores », édition Janvier 1999; http://home.ican.net/~lingua/fr/chroniques/chron_13.htm
- STE03** : Stefan Munz/ Traduit par Serge François, " Espace de nommage XML", http://Fr.selfhtml.org/xml/regles/espaces_nom.htm
- STP02** : Stefan Merckelbach, Pascaline Caliginri, «Des métaphores en entreprise », édition juin 2002 ; <http://www.ordinata.ch/fr/minute/minute6.htm>
- TAL04** : Tayeb LEMLOUMA, « Architecture de Négociation et d'Adaptation de Services Multimédia dans des Environnements Hétérogènes », Thèse de Doctorat le 9 Juin 2004, <http://wam.inrialpes.fr/publications/2004/TheseLemlouma.pdf>
- TAN02** : Tayeb Lemlouma, Nabil Layaïda, « Projet : OPERA ; Négociation de contenu pour des machines hétérogènes » ; <http://www.inria.fr/rapportsactivite/RA2002/opera/module18.html>
- TRH96** : T.Berners-Lee, R.Fielding, H.Frystyk ; "Hyper Text Transfer Protocol http 1.0, RFC1945", <http://72.14.203.104/search?q=cache:Qtafz7QxRQUJ:abcdrfc.free.fr/rfc-vf/rtf/rfc1945.rtf+T.berners-Lee+et+R.fielding&hl=fr&ct=clnk&cd=10>
- VAL06** : Van Lanker Luc, 'Apprendre le langage HTML', www.ccim.be/ccim328/HTML
- VEM03** : Véronique MORICEAU, « Un modèle de représentation sémantique de la métaphore : le cas des métaphores d'orientation », Université Paul Sabatier ; Toulouse III, Année 2002-2003 ; <http://www.irit.fr/recherches/ILPL/essais/site-vero/dea.pdf>