

# Le Protocole SIP Avancé et ses Extensions

EFORT

<http://www.efort.com>

Le premier tutoriel sur le protocole SIP proposé par EFORT est disponible à l'URL : [http://www.efort.com/r\\_tutoriels/SIP\\_EFORT.pdf](http://www.efort.com/r_tutoriels/SIP_EFORT.pdf). Il introduit les entités SIP, le protocole SIP à un haut niveau, l'interfonctionnement SIP/ISUP et l'architecture de services SIP. Ce nouveau tutoriel présente le protocole SIP de manière approfondie ainsi que ses extensions.

## 1 Protocole SIP

SIP (Session Initiation Protocol) est un protocole de signalisation défini par l'IETF (Internet Engineering Task Force) permettant l'établissement, la libération et la modification de sessions multimédias.

Il hérite de certaines fonctionnalités des protocoles HTTP (Hyper Text Transport Protocol) utilisé pour naviguer sur le WEB, et SMTP (Simple Mail Transport Protocol) utilisé pour transmettre des courriels (E-mail).

Le protocole SIP fournit la signalisation nécessaire à la création, sur réseaux IP, des fonctions de téléphonie similaires à celles du protocole ISUP dans le monde SS7 ou à celles du protocole H.323. SIP fait l'unanimité dans le monde de la téléphonie sur IP par sa simplicité et sa bonne intégration à l'architecture Internet et en font le candidat idéal pour les terminaux légers et mobiles.

Dans le premier tutoriel SIP, il a été vu que le protocole SIP est constitué initialement de six requêtes (Figures 1 et 2):

- REGISTER permet l'enregistrement, le ré-enregistrement et le désenregistrement
- INVITE a pour but l'initiation de session de toute nature : session audio, session vidéo, session tchat, session fax, session IPTV, etc.
- ACK sert à confirmer la réponse finale à une méthode INVITE
- OPTIONS permet d'obtenir les capacités de l'entité interrogée
- BYE termine une session établie
- CANCEL permet d'annuler une session qui n'a pas encore été établie.

Les requêtes SIP sont acquittées par des réponses. Les réponses SIP indiquent :

- Soit une information de progression d'appel
- Soit une information d'état final

Les réponses SIP contiennent :

- Un code d'état (Status Code) qui est un entier sur 3 chiffres
- Une raison (Reason-Phrase) qui décrit textuellement la raison.

Il existe six classes de réponses :

- Classe 1xx : Information, la requête a été reçue, et est en cours de traitement.
- Classe 2xx : Succès, la requête a été reçue, comprise et acceptée.
- Classe 3xx : Redirection, la session requiert d'autres traitements avant de pouvoir déterminer si elle peut être réalisée.
- Classe 4xx : Erreur requête client, la requête ne peut pas être interprétée par le serveur ou la destination, telle que soumise. La requête doit être modifiée avant d'être renvoyée.
- Classe 5xx : Erreur serveur, le serveur échoue dans le traitement d'une requête apparemment valide.

- Classe 6xx : Echec global, la requête ne peut être traitée par aucun serveur.

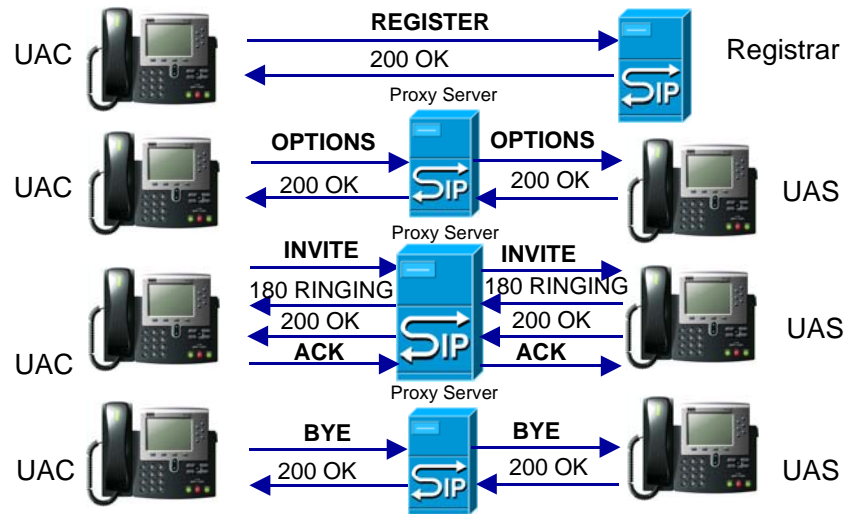


Figure 1 : Méthodes REGISTER, OPTIONS, INVITE, ACK et BYE

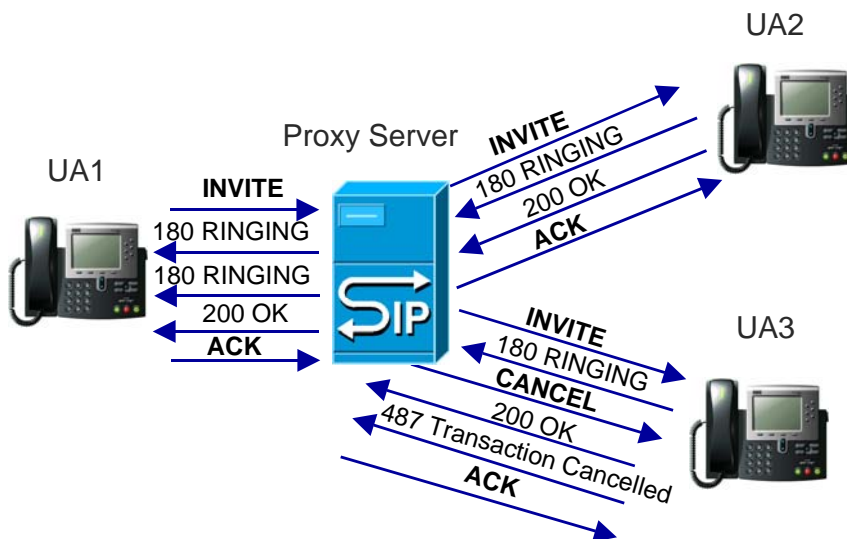


Figure 2 : Méthode CANCEL

## 2 Etablissement d'une session audio avec le protocole SIP

Dans l'exemple suivant représenté à la Figure 3, l'appelant a pour URL SIP sip:mary.taylor@tn.com, alors que celle de l'appelé est sip:mart.rich@tn.com.

Une méthode SIP INVITE est émise par le terminal SIP de l'appelant au Proxy Server. Ce dernier achemine la demande d'initiation de session à la destination. La requête INVITE contient un ensemble de headers obligatoires.

Par ailleurs, la requête SIP contient une syntaxe SDP (Session Description Protocol). Cette structure consiste en plusieurs lignes qui décrivent les caractéristiques du média que l'appelant " Mary " requiert pour la session.

Mary Taylor indique qu'il s'agit d'une session téléphonique (m=audio), que la voix paquetisée doit lui être délivrée à l'adresse de transport (port UDP = 45450, adresse IP = 192.23.34.45) avec le protocole RTP et en utilisant un format d'encodage défini dans le RFC 3551 AVP (Audio Video Profile) et pouvant être G.711  $\mu$ -law (0) ou G.729 (18).

La réponse 180 RINGING est retournée par le destinataire au terminal SIP de l'appelant.

Lorsque l'appelé accepte la session, la réponse 200 OK est émise par son terminal SIP et acheminée au terminal SIP de l'appelant.

Le terminal SIP de l'appelant retourne une méthode ACK au destinataire, relayée par l'entité SIP Proxy.

L'entité Proxy Server participe à l'acheminement de la signalisation entre UAs alors que les UAs établissent directement des canaux RTP pour le transport de la voix ou de la vidéo paquetisée sans implication du Proxy Server dans ce transport.

Lorsque Mary raccroche, son terminal SIP envoie une requête BYE pour terminer la session. Cette requête est délivrée au Proxy Server qui l'achemine au terminal SIP de Mark. Ce dernier retourne la réponse 200 OK.

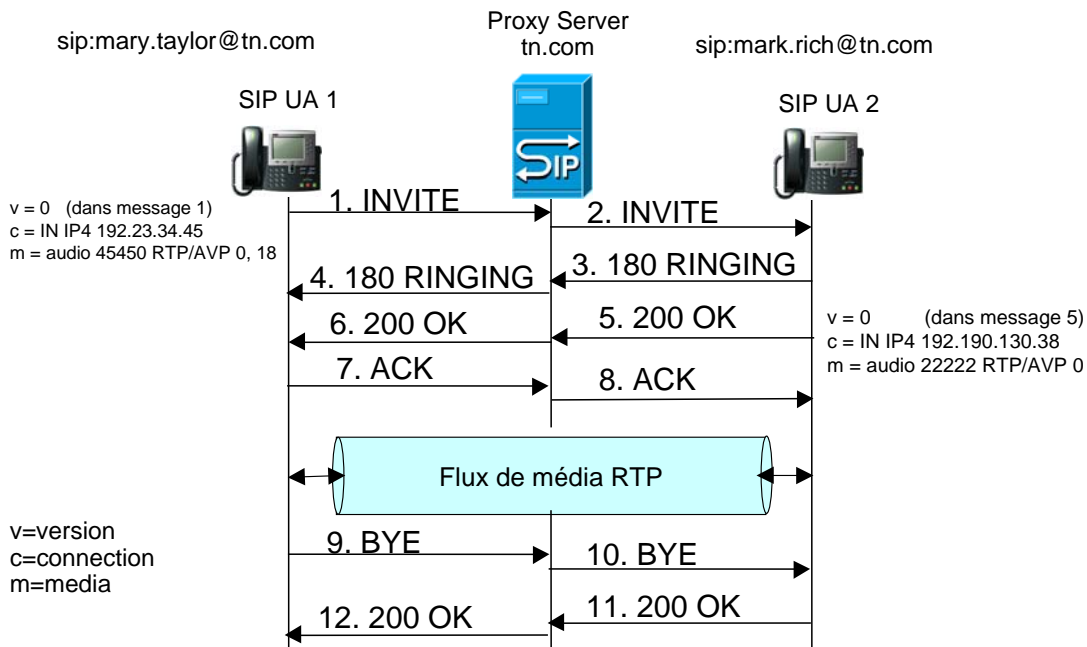


Figure 3 : Etablissement d'une session audio avec le protocole SIP

Une requête ou méthode SIP consiste en une request line, plusieurs headers, une empty line et un message body. Le « message body » est optionnel; certaines requêtes SIP n'en disposent pas.

Une Request line contient trois éléments : method, Request URI, et protocol version. La « method » indique le type de requête. Le request-URI (Uniform Resource Indicator) indique l'entité destinatrice de la requête. Finalement, la « protocol version » est SIP/2.0.

Une réponse SIP consiste en une status line, plusieurs headers, une empty line, et un message body. Le « message body » est optionnel; certaines requêtes SIP n'en disposent pas.

La première ligne de la méthode SIP, appelée request line liste la méthode (ici INVITE), le request URI et le numéro de version du protocole SIP utilisé (2.0). Alors qu'il est possible lors du routage d'une méthode SIP de modifier le request-URI, le header To doit rester inchangé.

Le premier Header suivant la ligne de début est le Header Via. Chaque entité SIP qui émet ou relaye une méthode SIP insère son adresse de host ou adresse IP dans ce Header Via. S'il s'agit d'une adresse de host, cette dernière peut être traduite en une adresse IP par le DNS. Le Header Via contient la version du protocole SIP, puis le type de transport pour le protocole SIP (ici un transport UDP), un espace, suivi du nom de host ou de l'adresse IP, de deux points, d'un numéro de port SIP et d'un paramètre branch. Ici, il s'agit du port SIP par défaut, à savoir 5060. Le paramètre branch identifie la transaction SIP.

Le Header Via permet de prévenir d'éventuelles boucles et assure que la réponse suivra le même chemin que la requête.

Les headers suivants sont les headers From et To qui indiquent les adresses SIP de l'émetteur et du récepteur de la requête. Toutefois le routage de la méthode se fait en utilisant le Request URI et non pas le header To. Lorsqu'un label de nom est utilisé (dans l'exemple, Mary Taylor et Marc Rich), alors l'URL SIP est délimitée par des crochets et permet de router le message. Ce label de nom peut être utilisé lors de l'alerte du terminal destinataire.

Le header From de la requête INVITE doit contenir un paramètre tag qui est identificateur aléatoire choisi par Mary Taylor. Lorsque l'appelé retourne des réponses, celles ci doivent inclure un paramètre Tag dans le header To. Ceci permet à l'appelant dans un scénario de forking de bien distinguer les différentes réponses reçues. Si par exemple, l'appelé est joignable sur deux terminaux, l'appelant recevra deux réponses 180 Ringing chacune contenant un Header To avec la même adresse SIP mais des tag différents. On peut donc conclure qu'une session est identifiée de manière unique grâce aux informations CallId, From-tag et To-tag.

Le Header Call-ID a une forme similaire à celle d'une adresse électronique mais ne fait que représenter un identificateur permettant de désigner une session SIP.

Le Header suivant est le Header Cseq. Il est constitué par un numéro suivi du nom de la méthode SIP. Dans l'exemple, il s'agit de la méthode INVITE. Le numéro est incrémenté pour chaque nouvelle méthode SIP émise. Le numéro de séquence initial choisi dans l'exemple est 1 mais il est possible de commencer par n'importe quelle valeur.

Le Header Contact est aussi requis dans la requête INVITE. Il contient l'URI SIP de terminal de Mary Taylor. Cet URI permet de router des requêtes SIP directement au terminal de Mary Taylor.

Tout message SIP doit contenir les paramètres obligatoires To, From, Max-Forwards, Via, Call-ID, C-Seq et Contact. Les autres headers sont optionnels.

Les headers content type et content length indiquent que le corps du message est une structure décrite avec SDP (Session Description Protocol) et contient 162 octets. Cette structure consiste en plusieurs lignes qui décrivent les attributs du média que l'appelant " Mary " requiert pour l'appel.

```
INVITE sip:mark.rich@tn.com SIP/2.0
Via : SIP/2.0/UDP station1.tn.com:5060; branch=z9hG4bK776asdrgrs
Max-Forwards : 20
To : Mark Rich <sip:mark.rich@tn.com>
From : Mary Taylor <sip:mary.taylor@tn.com>;tag=21272
Call-Id: j1sv235bh8965ws
CSeq: 1 INVITE
Contact: sip:mary.taylor@192.190.132.20
Content-Type: application/sdp
Content-Length:162
```

```
v = 0
c = IN IP4 192.190.132.20
m = audio 45450 RTP/AVP 0
```

Le message suivant est la réponse 180 Ringing envoyée en réponse à la requête INVITE. Cette réponse indique que l'appelé Mark Rich a reçu l'INVITE et que l'alerte a été initiée. L'alerte peut être faire sonner le téléphone, un message d'alerte sur l'écran du terminal ou toute autre méthode afin d'attirer l'attention de l'appelé Mark Rich.

La réponse 180 Ringing, parce qu'elle appartient à la catégorie "1XX" est une réponse "informationnelle". Une telle réponse est utilisée afin de transportée des information non critiques au sujet de la progression d'appel.

La phrase accompagnant le code de réponse (dans notre cas "Ringing") est suggérée dans le standard, mais toute texte peut être utilisé.

La réponse 180 Ringing est crée en copiant de nombreux headers de la requête INVITE, incluant Via, To, From, Call-ID, et CSeq, puis en ajoutant une ligne de réponse contenant la version du protocole SIP, le code de réponse, et la phrase associée.

Le header Via contient le paramètre « branch » mais dispose d'un paramètre supplémentaire "received". Ce paramètre contient l'adresse IP d'où la réponse a été reçue (192.190.132.23). Il s'agit de la même adresse que l'URI dans le Via résolue par interrogation DNS.

Notons que les headers To et From ne sont pas inversés dans la réponse. Bien que cette réponse soit envoyée par Mark Rich à Mary Taylor, les headers From et To indiquent le contraire. La raison est que les headers To et From dans SIP sont définis afin d'indiquer la direction de la requête et non pas la direction de la réponse. Puisque Mary Taylor initie la requête, toutes les réponses à l'INVITE pourront se lire : To: Mark Rich From: Mary Taylor.

Le header To contient maintenant un tag généré par Mark Rich. Toutes les réponses et requêtes futures dans cette session contiendront le tag généré par Mary Taylor et le tag généré par Mark Rich.

La réponse contient aussi un header Contact qui contient une adresse à laquelle Mark Rich peut être contacté directement lorsque la session est établie.

SIP/2.0 180 Ringing

Via : SIP/2.0/UDP proxy1.tn.com:5060; branch=z9hG4bK7745221  
received = 192.190.132.21

Via : SIP/2.0/UDP station1.tn.com:5060; branch=z9hG4bK776asdrgrs

To : Mark Rich <sip:mark.rich@tn.com>; tag 22454

From : Mary Taylor <sip:mary.taylor@tn.com>;tag=21272

Call-Id : j1sv235bh8965ws

Cseq : 1 INVITE

Contact : sip:mark.rich@192.190.23.16

Content-Length : 0

Lorsque l'appelé, Mark Rich décide d'accepter l'appel, une réponse 200 OK est retournée. Cette réponse indique aussiqu'un type de média proposé par l'appelant pour la session est acceptable. Le message body de la réponse 200 OK contient les information média de Mark Rich. L'appelé a l'obligation d'accepter le premier média de la liste proposée par l'appelant, supporté par l'appelé.

Cette réponse 200 OK est construite de la même façon que la réponse 180 Ringing et contient les mêmes To tag et Contact URI. Les capacités média du terminal de Mark Rich sont communiquées dans le body SDP ajouté à la réponse.

SIP/2.0 200 OK

Via : SIP/2.0/UDP proxy1.tn.com:5060; branch=z9hG4bK7745221  
received = 192.190.132.21

Via : SIP/2.0/UDP station1.tn.com:5060; branch=z9hG4bK776asdrgrs

To : Mark Rich <sip:mark.rich@tn.com>; tag 22454

From : Mary Taylor <sip:mary.taylor@tn.com>;tag=21272

Call-Id : j1sv235bh8965ws

Cseq : 1 INVITE

Contact : sip:mark.rich@192.190.23.16

Content-Type : application/sdp  
Content-Length :162

v = 0  
c = IN IP 192.190.23.16  
m = audio 52140 RTP/AVP 0

L'étape finale est de confirmer la session avec une requête ACK (Acknowledgment). La confirmation signifie que Mary Taylor a reçu avec succès la réponse de John Cook (200 OK de l'INVITE). Cet échange d'information média permet l'établissement de la session média via un autre protocole : Real Time Transport Protocol (RTP)

Le Cseq a le même numéro que celui de l'INVITE, mais la méthode est positionnée à ACK. A ce point la session média commence en utilisant les informations de média transportées dans les messages SIP (descriptions SDP).

Le paramètre branch dans le header Via contient un nouvel identificateur de transaction que l'INVITE puisque l'ACK émis pour acquitter la réponse 200 OK est considéré comme une nouvelle transaction.

L'échange de messages montre que SIP est un protocole de signalisation de bout en bout.

ACK sip:mark.rich@tn.com SIP/2.0  
Via : SIP/2.0/UDP station1.tn.com:5060; branch= z9hG4bK83bcetfa  
Max-Forwards : 20  
To : Mark Rich <sip:mark.rich@tn.com>; tag 22454  
From : Mary Taylor <sip:mary.taylor@tn.com>; tag=21272  
Call-Id : j1sv235bh8965ws  
Cseq : 1 ACK  
Content-Length : 0

La libération de la session requiert l'envoi d'une méthode BYE par le participant qui souhaite initier cette libération. Le header Via dans cet exemple est renseigné avec l'adresse du hostname de Mary Taylor et contient un nouvel identificateur de transaction puisque la méthode BYE est considérée comme une nouvelle transaction indépendamment des INVITE et ACK. Les headers To et From montrent que la requête est initiée par Mary Taylor. Si c'est Mark Rich qui décide de libérer la session, alors le header From indiquerait la SIP URI de Mark Rich.

BYE sip:mark.rich@tn.com SIP/2.0  
Via : SIP/2.0/UDP station1.tn.com:5060; branch= z9hG4bK883bcfrfg  
Max-Forwards : 20  
To : Mark Rich <sip:mark.rich@tn.com>; tag 22454  
From : Mary Taylor <sip:mary.taylor@tn.com>; tag=21272  
Call-Id : j1sv235bh8965ws  
Cseq : 2 BYE  
Content-Length : 0

La réponse à la méthode BYE est 200 OK. Cette réponse rappelle le Cseq de la requête BYE. Il n'y a pas d'ACK émis puisque l'ACK n'est envoyé que pour acquitter une réponse finale à l'INVITE.

SIP/2.0 200 OK  
Via : SIP/2.0/UDP proxy1.tn.com:5060; branch=z9hG4bK7745343  
received = 192.190.132.21  
Via : SIP/2.0/UDP station1.tn.com:5060; branch= z9hG4bK883bcfrfg  
To : Mark Rich <sip:mark.rich@tn.com>; tag 22454  
From : Mary Taylor <sip:mary.taylor@tn.com>;tag=21272

Call-Id : j1sv235bh8965ws  
Cseq : 2 BYE  
Content-Length : 0

### 3 Description SDP

La finalité de SDP (Session Description Protocol) est de transporter les informations se rapportant aux flux de données multimédia dans une session multimédia afin de permettre aux récepteurs de cette description de session de participer à cette session.

L'exemple suivant décrit les informations pouvant être transportées par le protocole SDP:

- v = 0. Version du protocole SDP (0)
- o = Mary 2890844526 2890844526 IN IP4 . Origine de la session décrite à travers le username (Mary), l'identificateur de la session (2890844526), la version de la session, le type de réseau (IN = Internet), le type d'adresse (IP4 = IPv4), et l'adresse réseau de la machine où la session a été créée (192.190.132.20).
- s = session name; Il s'agit d'une chaîne de caractères, nom de la session, qui peut être affichée aux participants de la session
- i= Information; contient des informations sur la session. Il peut contenir n'importe quel nombre de caractères.
- u= URI; contient une uniform resource indicator (URI) avec plus d'information sur la session.
- e= e-mail; contient une adresse E-mail du host de la session. Si un nom est aussi utilisé, l'adresse e-mail est mise entre <>.
- p= phone; contient un numéro de téléphone avec un format globalisé commençant par +, puis le code pays, puis un espace ou ' - ', puis le numéro local. Un commentaire peut être rajouté entre parenthèses.
- c = connection; IN IP4 192.190.132.20 . Données de connexion incluant le type de réseau (IN = Internet), le type d'adresse (IP4=IPv4) et l'adresse de connexion (192.190.132.20) à différencier du réseau et de l'adresse à laquelle la session a été créée.
- b= bandwidth; contient des informations sur la bande passante requise. L'information est de la forme b=modifier:bandwidth-value. Le modifier est soit CT pour Conference total ou AS pour Application Specific. CT est utilisé pour une session multicast pour spécifier le débit total qui peut être utilisé par tous les participants dans la session. AS est utilisé afin de spécifier le débit pour un unique site. Le débit est exprimé en kilobits par seconde.
- t = time; Instant de la session. Il indique les instants de début et de fin de la session Ces instants sont égaux à 0 pour une session établie dynamiquement comme un appel téléphonique. Ces instants sont positionnés à des valeurs particulières lorsqu'il s'agit d'une conférence pré-arrangée avec un début et une fin de conférence programmés
- m = media; audio 45450 RTP/AVP 0. Type de média (audio), port de transport où la voix ou la vidéo paquetisée doit être envoyée (45450), le protocole de transport (RTP) et le type de format (AVP 0 = G.711 m-law).
- a=attributes; contient des attributs de la session média. Il peut être utilisé pour étendre la description SDP avec plus d'information sur le média. Les informations fournies qui ne sont pas comprises par l'utilisateur SDP, peuvent être ignorées. Plusieurs lignes « a » peuvent être présentes pour chaque type de codec listé.

```
v=0
o=Mary 2890844526 2890844526 IN IP4 192.190.132.20
s=Résumé réunion
i=Traite des points importants de la dernière réunion
u=http://www.sipstation.com
e=Mary Taylor <mary.taylor@tn.com>
```

p=+33 672192222 (Appeler uniquement de 9h00 à 17h00)  
c=IN IP4 192.190.132.20  
b=AS:64  
t=2877631875 2879633673  
m=audio 45450 RTP/AVP 0 8 99  
a=rtpmap:0 PCMU/8000  
a=rtpmap:8 PCMA/8000  
a=rtpmap:99 iLBC/8000  
m=video 23422 RTP/AVP 31  
a=rtpmap:31 H261/90000

## 4 Enregistrement

La méthode REGISTER est utilisée par un User Agent pour informer le réseau SIP de son adresse IP et de son URL pour laquelle il souhaite recevoir des sessions.

L'enregistrement SIP est similaire à l'enregistrement du terminal mobile lors de son attachement au réseau GSM.

L'enregistrement est important afin d'être authentifié, afin d'être localisé et ainsi recevoir des appels entrants depuis des Proxy servers qui servent le domaine de l'utilisateur.

La méthode REGISTER est utilisée par un User agent afin d'indiquer au Registrar son adresse de localisation (e.g., adresse IP) ainsi que son adresse SIP (SIP URL).

Le user agent peut connaître par avance son Registrar par configuration (adresse IP du Registrar préconfigurée dans le User agent) auquel cas il émet une méthode REGISTER uniquement à ce Registrar. Le user agent peut aussi apprendre dynamiquement l'adresse du Registrar par exemple via un serveur DHCP.

Sinon, le User agent émet le message à tous les Registrars du domaine en utilisant l'adresse multicast (224.0.1.175).

La méthode SIP Register contient les headers suivants :

- Le header From indique l'adresse de l'entité ayant initié l'enregistrement.
- Le header To indique l'adresse de l'utilisateur enregistré.
- Les headers From et To ont la même valeur si l'utilisateur s'enregistre lui-même.
- L'adresse de la fonction REGISTRAR n'est indiquée que dans la première ligne de la commande REGISTER dans le champ Request-URI.
- Le header CallID. Un User agent utilisera un même Call-ID pour l'ensemble de ses enregistrements.
- Le header Cseq. A chaque nouvel enregistrement pour le même user agent, le numéro Cseq est incrémenté.
- Le header Contact indique les différentes adresses auxquelles l'utilisateur SIP est joignable.
- Le champ Expires associé au header Contact est optionnel et exprime une durée d'enregistrement en secondes. S'il est absent, alors l'URL SIP sera supprimée de la table d'enregistrement de la fonction Registrar au bout d'une heure après l'enregistrement pour l'environnement Internet et au bout d'une semaine pour l'environnement IMS.

La fonction Registrar retourne une réponse 200 OK en cas de succès de l'enregistrement.

Ses headers Via, From, To, Call-ID et Cseq ont les mêmes valeurs que dans la requête REGISTER.

Le champ Expires de la réponse 200 OK peut indiquer une valeur égale ou inférieure à celle spécifiée dans la requête REGISTER.

Dans l'exemple à la Figure 4, Mary s'est logguée sur la machine station1.tn.com. Une méthode REGISTER a alors été émise à la fonction Registrar locale. Le header Contact indique que Mary Taylor est joignable à l'adresse IP 192.190.132.20.

La requête et la réponse contiennent les headers présentés ci-dessus.



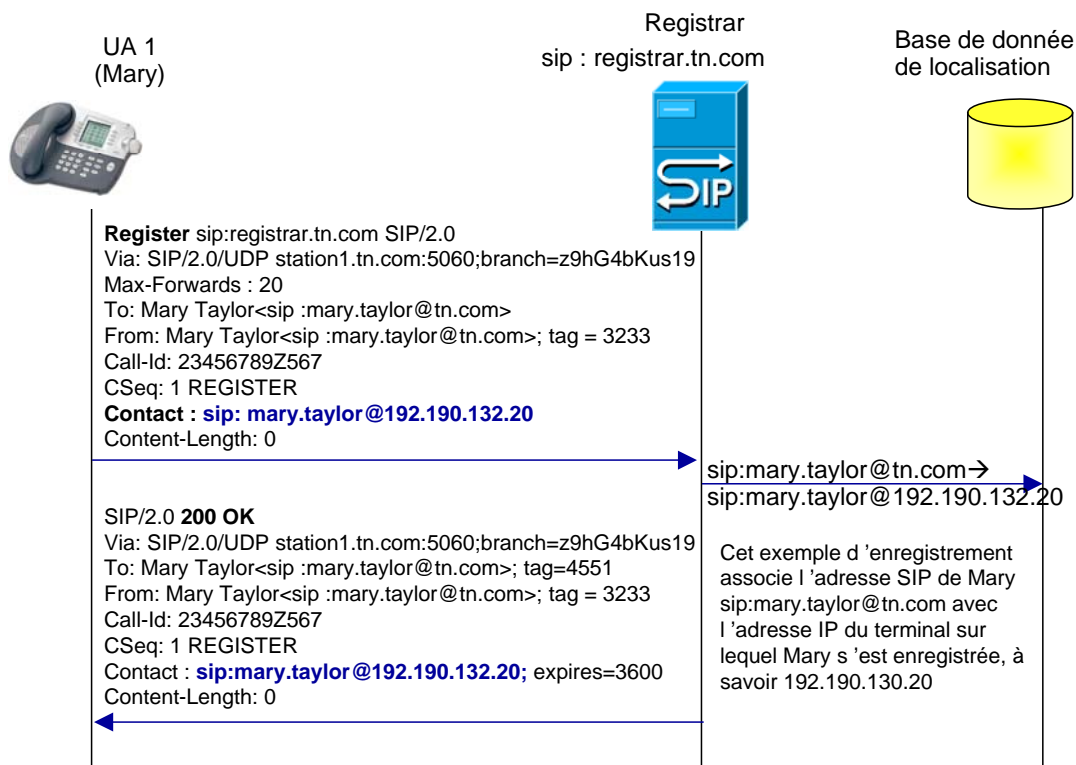


Figure 4 : Procédure d'enregistrement

Dans l'exemple de la Figure 5, Mary Taylor est enregistrée simultanément sur deux terminaux. Le header Contact de chaque méthode REGISTER précise l'adresse IP du terminal associé ainsi que la durée d'enregistrement souhaitée par Mary Taylor sur le terminal.

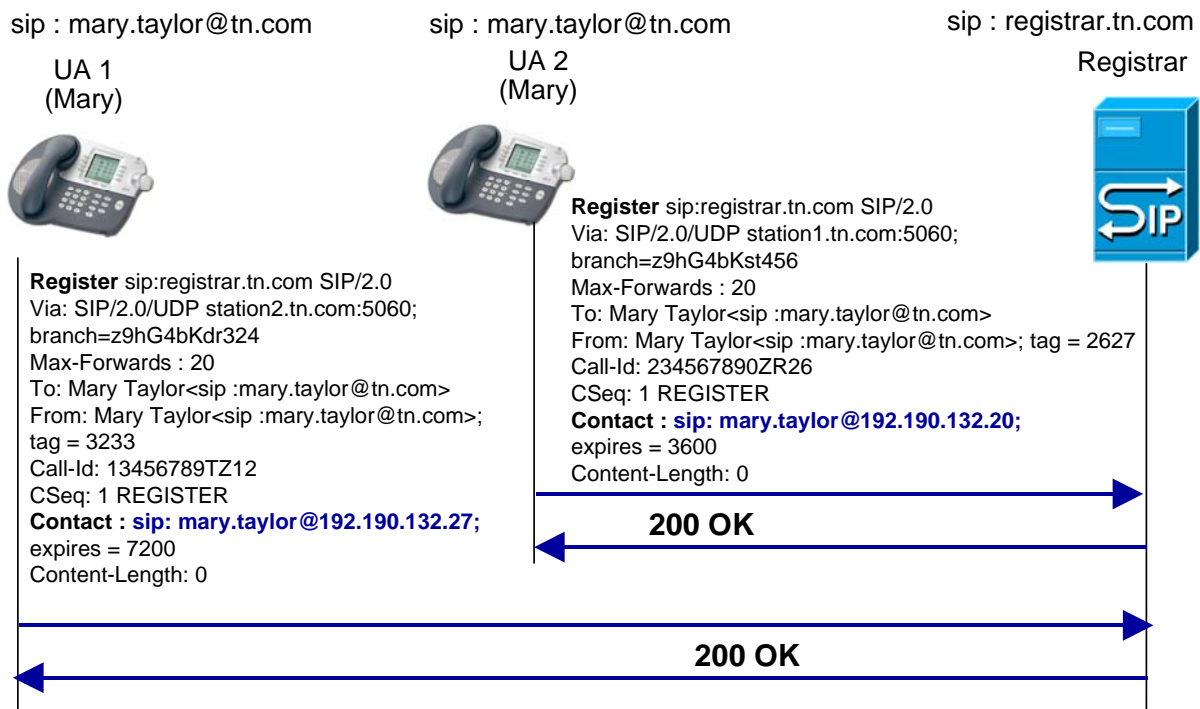


Figure 5 : Enregistrements multiples pour la même identité SIP

## 5 Méthodes SIP additionnelles

Huit méthodes SIP additionnelles ont été proposées dans divers RFCs :

- SUBSCRIBE : Demande de notification d'événement de session (RFC 3265)
- NOTIFY : Notifie un événement pour lequel une souscription a été effectuée (RFC 3265)
- PUBLISH : Publie un état (RFC 3903)
- MESSAGE : Transporte un message de données hors session (RFC 3428)
- REFER : Transfère l'utilisateur à une URL (RFC 3515)
- UPDATE : Modifie la session avant son acceptation (RFC 3311)
- INFO : Permet l'échange de signalisation comme les tonalités DTMF pendant la session (RFC 2976)
- PRACK : Acquiesce des réponses provisoires (RFC 3262)

Une entité SIP peut souscrire à un événement afin d'être notifiée de son occurrence. La requête **SUBSCRIBE** permet la souscription alors que la requête **NOTIFY** est utilisée afin de notifier l'événement souscrit. **PUBLISH** permet à un usager de mettre à jour son information d'état dans le contexte d'un service tel que la présence .

La méthode **REFER** renvoie un client SIP vers une ressource identifiée dans la méthode. REFER permet d'émuler différents services ou applications dont le transfert de session. Considérons UA1, l'entité à l'origine du transfert, UA2, l'entité transférée et UA3, le destinataire du transfert. Le transfert d'appel permet à UA1 de transformer un appel en cours entre UA1 et UA2 en un nouvel appel entre UA2 et un UA3 choisi par UA1. Si le transfert d'appel aboutit, UA2 et UA3 pourront communiquer tandis que UA1 ne pourra plus dialoguer avec UA2 ou UA3.

La méthode **MESSAGE** a été proposée comme extension au protocole SIP afin de permettre le transfert de messages instantanés. La messagerie instantanée (IM, Instant Messaging) consiste en l'échange de messages entre usagers en pseudo temps réel. Cette nouvelle méthode est routée entre usagers comme toute méthode SIP. La requête MESSAGE peut transporter plusieurs types de contenus en s'appuyant sur le codage MIME.

La méthode **INFO** permet de transférer des informations de signalisation durant l'appel. Parmi les exemples d'information figurent les digits DTMF, les informations relatives à la taxation d'un appel, des images, etc.

Les réponses finales 2XX, 3XX, 4XX, 5XX et 6XX à une requête INVITE sont acquittées par la requête ACK alors que les réponses provisoires de type 1XX ne sont pas acquittées. Or, certaines réponses provisoires telles que 180 Ringing sont critiques et leur réception est essentielle pour la détermination de l'état de la session, notamment lors de l'interconnexion avec le RTCP. La méthode **PRACK** a donc été définie afin d'acquitter la réception de réponses provisoires, de type 1XX sauf 100 Trying.

La méthode **UPDATE** permet à un user agent de mettre à jour les paramètres d'une session multimédia. (e.g., flux média et leurs codecs). La méthode UPDATE peut être envoyée avant que la session soit établie. UPDATE est donc particulièrement utile lorsqu'il s'agit de mettre à jour des paramètres de session avant son établissement, e.g. mise en attente du destinataire.

## 5.1 Méthode SUBSCRIBE

La méthode SUBSCRIBE est utilisée par un UA afin d'établir une souscription pour la réception de notifications (via la méthode NOTIFY) lorsqu'un événement souscrit survient. Une souscription établit un dialogue entre l'UAC et l'UAS. La requête de souscription contient un header Expires qui indique la durée souhaitée pour la souscription.

Après cette durée de souscription, la souscription est automatiquement terminée. La souscription peut être rafraîchie via un autre SUBSCRIBE dans le dialogue avant le moment d'expiration.

Un serveur qui accepte la souscription retourne une réponse 200 OK qui contient un header Expires égal ou plus petit que celui proposé dans le SUBSCRIBE.

Il n'existe pas de méthode UNSUBSCRIBE avec SIP. La requête SUBSCRIBE avec un header Expires positionné à 0 demande la fin de la souscription et donc du dialogue.

Une souscription terminée à cause de l'expiration de la souscription ou parce qu'une demande explicite de fin de souscription a été émise résulte en un NOTIFY final indiquant que la souscription est terminée.

## 5.2 Méthode NOTIFY

La méthode NOTIFY est utilisée par un UA pour transporter l'information sur l'occurrence d'un événement particulier. Une requête NOTIFY est toujours émise dans un dialogue lorsqu'une souscription existe entre le souscripteur et le notificateur.

Une requête NOTIFY est normalement acquittée par une réponse 200 OK pour indiquer qu'elle a été reçue.

Une requête NOTIFY contient un header Event indiquant le nom du package utilisé lors de la souscription et un header Subscription-State indiquant l'état courant de la souscription (active, pending ou terminated).

Une requête NOTIFY est toujours émise au début de la souscription et à la fin de la souscription comme le montre la Figure 6.

UA1 souscrit à un événement auprès d'UA2 (1-2). UA2 notifie que la demande est active (3-4). L'événement souscrit est observé; il est donc notifié par UA2 à UA1 (5-6). La durée de souscription a expiré. UA2 notifie cet événement à UA1 (7-8).

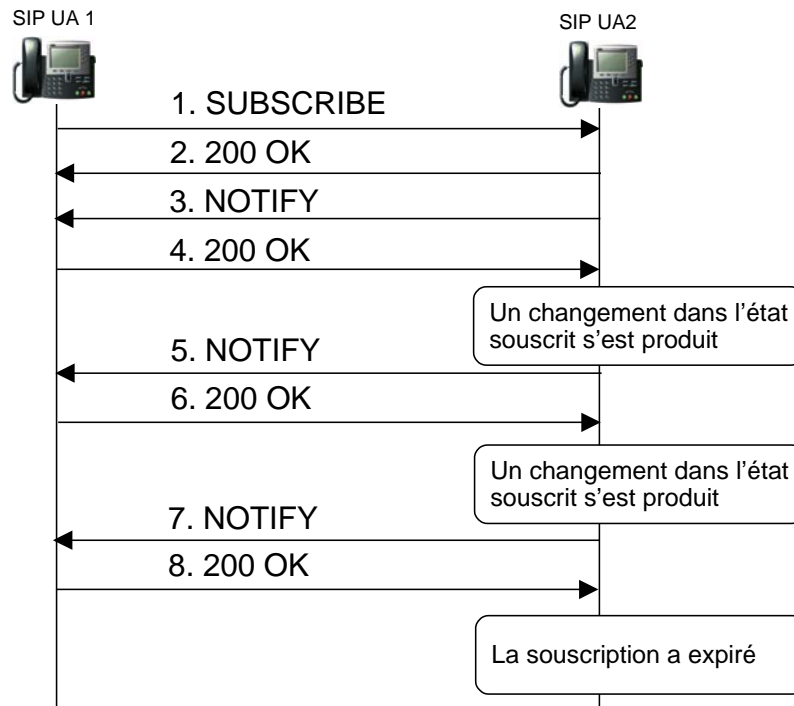


Figure 6 : Modèle de souscription/notification avec SIP

Différents événements ont été spécifiés dans différents RFCs:

- [RFC 3515]: event package for the REFER method. Cet événement permet de notifier le résultat d'une opération de transfert par la méthode REFER.
- [RFC 3680]: event package for SIP-registration events. Cet événement permet de notifier le changement d'état d'enregistrement d'un usager.
- [RFC 3842]: event package for voice-mail notification events. Cet événement permet de notifier le changement d'état de la boîte vocale d'un usager.
- [RFC 3856]: Event Package for SIP Presence events. Cet événement permet de notifier le changement d'état de présence d'un usager.
- [RFC 4235]: event package for SIP-dialog events. Cet événement permet de notifier le changement d'état d'un dialogue, utile notamment à un service comme le rappel automatique sur occupation.
- [RFC 4575]: event package for SIP-conferencing events. Cet événement permet de notifier le changement d'état d'un participant à une conférence (a joint, a quitté, a mis en garde la conférence)
- [RFC 4730]: event package for media server events. Cet événement permet à un serveur application de demander à un serveur de média de le notifier lorsque l'utilisateur a composé des touches DTMF particulières.

L'exemple de la Figure 7 concerne le modèle souscription / notification associé à l'événement voice-mail notification.

Mary Taylor doit participer à une réunion. Elle active le renvoi d'appel inconditionnel sur messagerie vocale. Elle souscrit par ailleurs pour une durée de 3 heures à la messagerie vocale afin d'être notifiée lorsqu'un nouveau message est enregistré dans sa boîte vocale (1-2).

Le système de messagerie vocale lui retourne une notification (3-4) pour lui indiquer que sa souscription est active.

Mary Taylor reçoit un appel renvoyé à la messagerie vocale. Comme un message vocal est déposé, l'état de sa boîte vocale a changé et une notification lui est envoyée (5-6).

La durée de souscription a expiré. Mary Taylor reçoit une nouvelle méthode NOTIFY qui l'informe de cet événement (7-8).

Comme la réunion n'est pas terminée, elle re-souscrit à l'événement de changement d'état de boîte vocale pour une durée d'une heure (9-10). Elle est notifiée du fait que la souscription est active (11-12). Comme la réunion se termine avant la fin de la souscription, Mary Taylor envoie une méthode SUBSCRIBE pour annuler cette souscription (13-14). Elle reçoit une notification qui lui indique la fin de la suscription.

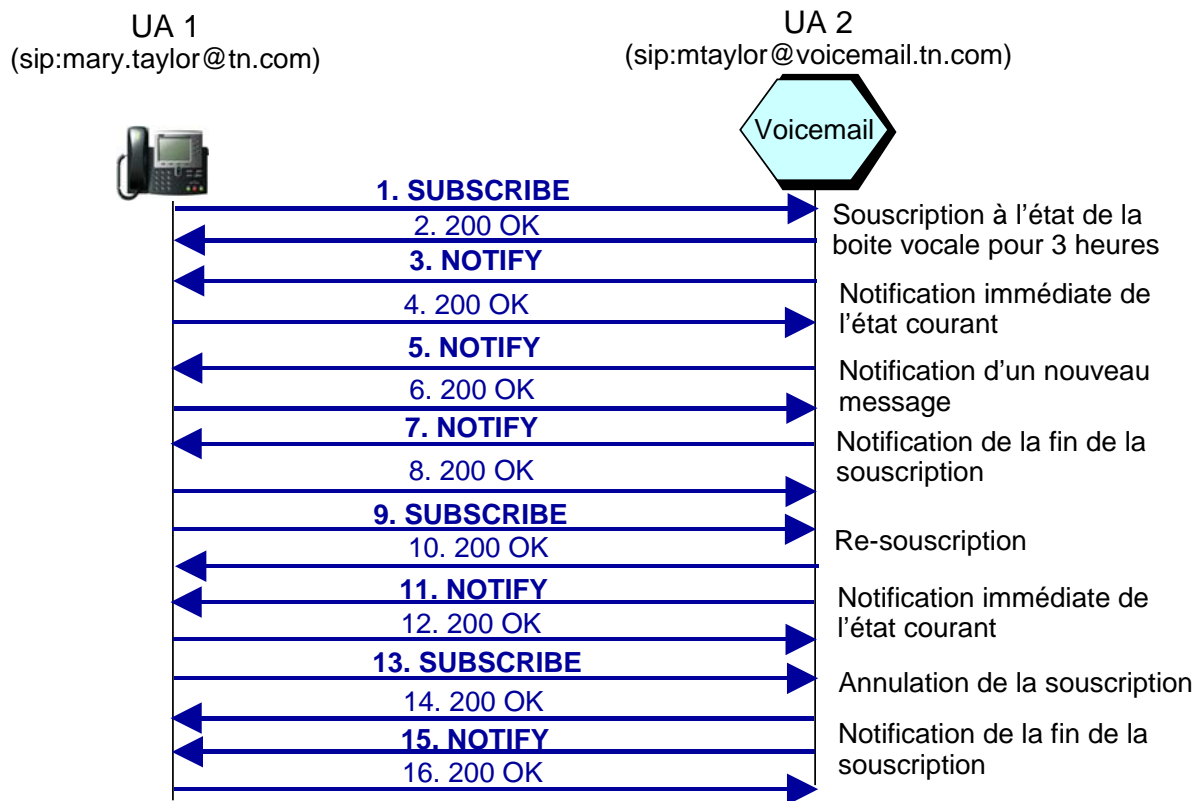


Figure 7 : Scénario de souscription/notification de l'événement voice-mail notification

```

SUBSCRIBE sip: mtaylor@voicemail.tn.com SIP/2.0
Via : SIP/2.0/UDP station1.tn.com:5060; branch=z9hG4bK776asdrqs
Max-Forwards : 20
To : <sip:mtaylor@voicemail.tn.com>
From : Mary Taylor <sip:mary.taylor@tn.com>; tag=1928301774
Call-Id: a84b4c76e66710
CSeq: 2342 SUBSCRIBE
Allow-Events : message-summary, presence
Contact: sip:mary.taylor@192.190.132.20
Event : message-summary
Expires : 10800
Accept : application/ message-summary
Content-Length : 0
  
```

```

NOTIFY sip: mary.taylor@tn.com SIP/2.0
Via : SIP/2.0/UDP station1.tn.com:5060; branch=z9hG4bK756asebgt
Max-Forwards : 20
To : Mary Taylor <sip:mary.taylor@tn.com>; tag= 1928301774
From : < sip:mtaylor@voicemail.tn.com >; tag= 456887766
Call-Id: a84b4c76e66710
CSeq: 755 NOTIFY
  
```

Allow-Events : message-summary  
Contact: sip:mary.taylor@192.190.132.29  
Event: message-summary  
Subscription-State: active  
Content-Type: application/simple-message-summary  
Content-Length: 99

Messages-Waiting: yes  
Message-Account: sip:mary.taylor@vmail.tn.com  
Voice-Message: 2/8 (0/2)

2/8 (0/2) signifie que la boîte vocale contient 2 messages non lus et 8 anciens messages lus mais conservés. Parmi les messages non lus, aucun n'est urgent. Parmi les messages lus et conservés, deux étaient urgents.

### 5.3 Méthode MESSAGE

La messagerie instantanée (IM, Instant Messaging) consiste en l'échange de messages entre usagers en pseudo temps réel. Ces messages sont généralement courts, mais pas nécessairement. La messagerie instantanée est utilisée en mode conversationnel, i.e., l'échange de messages entre les usagers est rapide afin de maintenir une conversation interactive. Les messages instantanés ne sont généralement pas stockés (à la différence des SMS dans le monde GSM) ; en effet, ce stockage n'est pas "forcément" nécessaire car des usagers s'échangent en principe des messages instantanés lorsqu'ils sont tous connectés. C'est la raison pour laquelle, le service de messagerie instantanée est souvent couplé avec le service de présence afin que les usagers s'informent de leur présence avant de s'échanger des messages instantanés. Toutefois, il est possible d'introduire un serveur de messagerie instantanée réalisant la fonction de store-and-forward similaire à un SMSC (Short Message Service Center) dans un réseau mobile. La méthode MESSAGE a été proposée comme extension au protocole SIP afin de permettre le transfert de messages instantanés. La méthode MESSAGE n'initie pas de session à la différence des méthodes INVITE/200 OK/ACK; il s'agit d'un message asynchrone (Figure 8). On peut noter la similitude avec le monde GSM. En effet, un usager GSM établit un appel téléphonique en utilisant le protocole Call Control (messages SETUP, ALERTING, CONNECT équivalents à INVITE/180 Ringing/200OK) et envoie des SMS séparément à travers le protocole Short Messaging (messages SUBMIT, DELIVER équivalents à MESSAGE).

Une réponse 200 OK est retournée par la destination si elle accepte la méthode MESSAGE mais cela ne signifie pas pour autant que la destination a lu son contenu.

Une réponse 4xx ou 5xx indique que la méthode MESSAGE n'a pas été délivrée correctement. Une réponse 6xx signifie que la méthode MESSAGE a pu être délivrée mais a été refusée par la destination.

La taille maximum d'un message instantané ne doit pas excéder 1300 octets. Pour des contenus volumineux, l'URL où le contenu est accessible pourrait être indiquée dans le corps de la méthode MESSAGE. Le contenu pourrait être téléchargé par le destinataire de la méthode MESSAGE en utilisant le protocole HTTP.

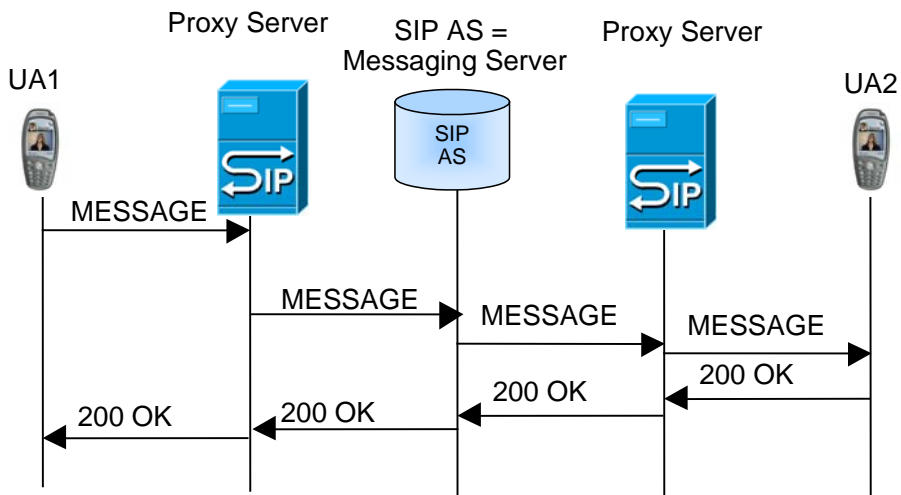


Figure 8 : Envoi d'un message instantané.

```
MESSAGE sip :mark.rich@tn.com SIP/2.0
Via: SIP/2.0/UDP 192.190.132.20:5060; branch=z9hG4bK3
Max-Forwards : 20
To: Mark Rich <sip :mark.rich@tn.com>
From: Mary Taylor<sip :mary.taylor@tn.com>; tag=1949
Call-Id: 123456782349
CSeq: 121 MESSAGE
Contact : sip :mary.taylor@192.190.132.20
Content-Type : text/plain
Content-Length : 24
Mark, How are you today?
```

```
SIP/2.0 200 OK
Via: SIP/2.0/UDP 192.190.132.20:5060; branch=z9hG4bK3
To: Mark Rich <sip :mark.rich@tn.com>;tag=2774
From: Mary Taylor<sip :mary.taylor@tn.com>;tag=1949
Call-Id: 123456782349
CSeq: 121 MESSAGE
Content-Length : 0
```

## 5.4 Méthode REFER

La méthode REFER renvoie le récepteur vers une ressource identifiée dans la méthode REFER. Cette méthode permet d'émuler différents services ou applications dont le transfert de session. Considérons UA1, l'entité à l'origine du transfert, UA2, l'entité transférée et UA3, le destinataire du transfert. Le transfert de session permet à UA1 de transformer une session en cours entre UA1 et UA2 en une nouvelle session entre UA2 et un UA3 choisi par UA1. Si le transfert de session aboutit, UA2 et UA3 pourront communiquer tandis que UA1 ne pourra plus dialoguer avec UA2 ou UA3.

Dans l'exemple de transfert de session à la Figure 9, UA2 établit une session téléphonique avec UA1 (1-4). UA1 émet une méthode REFER (5-6) à UA2 afin que ce dernier établisse une nouvelle session avec UA3. UA1 met en garde la session en cours avec UA2; il s'agit d'un transfert supervisé (7-9). UA2 établit une nouvelle session avec UA3 (10-13) et notifie UA1 (14-15) dès lors que la session est effective entre UA2 et UA3. Si le transfert aboutit, UA1 peut libérer la session avec UA2 (16-17).

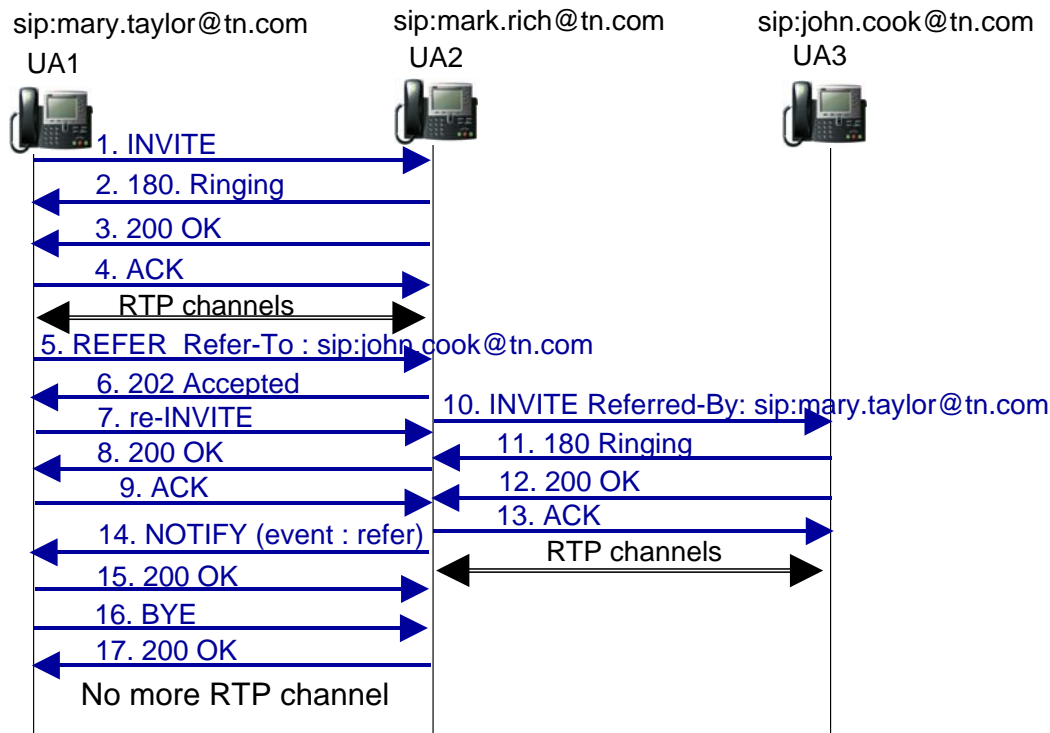


Figure 9: Méthode REFER

```
REFER sip:mark.rich@tn.com SIP/2.0 SIP/2.0
Via SIP/2.0/UDP 192.120.11.12:5060; branch=z9hG4bK932039
Max-Forwards: 20
To: Mark Rich <sip:mark.rich@tn.com>
From: Mary Taylor <sip:mary.taylor@tn.com>;tag=213424
Call-ID: 3419fak3kFD23s1A9dkl
CSeq: 5412 REFER
Refer-To: <sip:john.cook@tn.com>
Referred-By: <sip:mary.taylor@tn.com>
Contact: sip: < sip:mary.taylor@192.120.11.12 >
Content-Length: 0
```

## 5.5 Méthode INFO

Les tonalités Dual tone multifrequency (DTMF) sont utilisées pour l'échange de signalisation usager à usager pendant la session, par exemple envoi d'un code d'accès après avoir établi la communication avec un serveur vocal interactif. Comme son nom l'indique, DTMF génère deux fréquences pour émettre un digit (0–9, \*, #, ou moins communément, A–F). Dans le réseau RTC, DTMF est encodé comme la voix avec le codec G.711 (DTMF dans la bande). Par contre, les codecs bas débit utilisés pour la voix sur IP qui optimisent l'encodage de la voix, n'encodent pas généralement les tonalités DTMF de façon fiable. D'autres approches sont à considérer pour l'envoi des DTMFs dans ce contexte:

- Tonalités DTMF dans le flux RTP (RFC 4733).
- Tonalités DTMF dans la méthode SIP INFO

La méthode INFO (Figure 10) permet de transférer des informations de signalisation durant la session. Parmi les exemples d'information figurent les tonalités DTMF, les informations relatives à la taxation d'un appel, etc. La méthode INFO n'a pas pour but de changer l'état de la session en cours ou des paramètres de la session ; la méthode re-INVITE est utilisée à ces fins. La route de signalisation empruntée par la requête INFO est la même que celle



utilisée par les requêtes SIP qui ont permis l'établissement de la session correspondante (e.g., INVITE, ACK). Par ailleurs une méthode ne pas être émise si la session n'a pas été établie. L'information peut être communiquée dans un header ou dans le corps du message INFO. Une des réponses suivantes est retournée par le destinataire de la requête INFO :

- 200 OK si la requête INFO est applicable à une session,
- 481 Call Leg/Transaction Does Not Exist, si la requête INFO ne correspond à aucune session en cours,
  - 415 Unsupported Media Type message si la requête INFO contient un corps de message que le récepteur n'a pas su interpréter faute de disposer des règles de traitement correspondantes.
- 4xx, 5xx et 6xx comme pour les autres méthodes SIP.

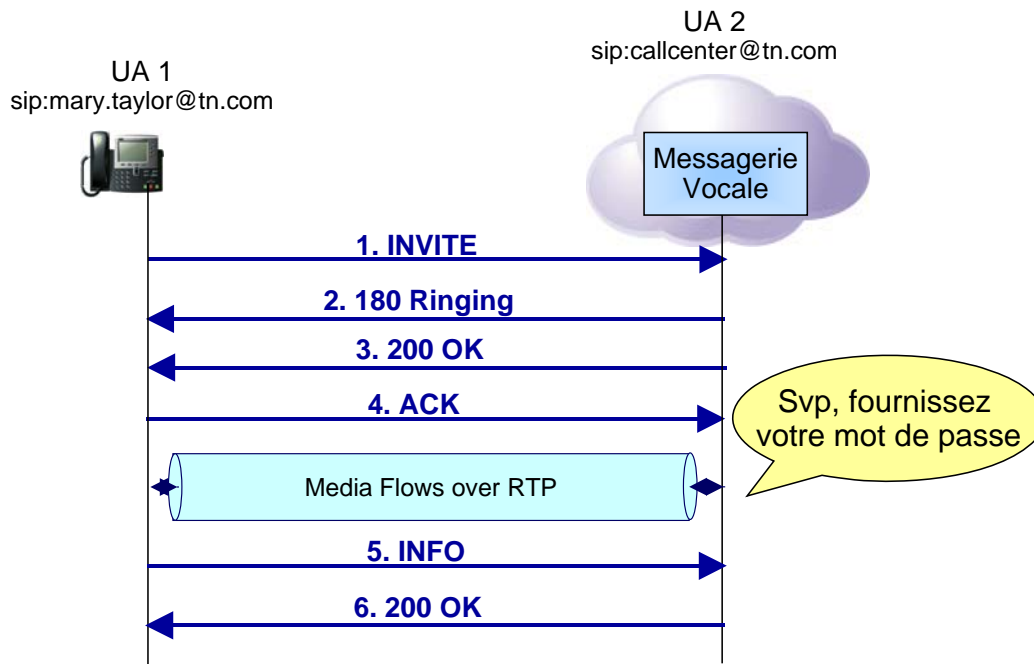


Figure 10 : Méthode INFO

L'entité émettrice ne peut émettre qu'un seul DTMF par méthode INFO, avec un header Content-Type positionné à "application/dtmf-relay". Si tel est le cas, alors l'entité réceptrice, recherche le DTMF sans la ligne "Signal=" line et la durée du DTMF en millisecondes dans la ligne "Duration=" . Si plus d'un DTMF est présent dans la méthode INFO, seul le premier sera accepté et traité.

```

INFO sip: callcenter@tn.com SIP/2.0
Via: SIP/2.0/UDP station1.tn.com:5060 ; branch=z9hG4bK932039
Max-Forwards : 20
From: Mary Taylor<sip : mary.taylor@tn.com>; tag=213424
To: < sip: callcenter@netphone.com>; tag=435642
Subject : Accès à la boite vocale
Call-Id: 214534zte4
CSeq: 5 INFO
Content-Type: application/dtmf-relay
Content-Length: 26

```

```

Signal= 6
Duration= 100

```

## 5.6 Méthode PRACK

Le protocole d'initiation de session (SIP) (RFC 3261) est un protocole de demande-réponse pour initier et gérer des sessions de communication. SIP définit deux types de réponses, provisoires et finales. Les réponses finales apportent le résultat du traitement de la demande, et leur envoi est fiable. Les réponses provisoires fournissent des informations sur la progression du traitement de la demande, mais leur envoi n'est pas fiable dans le RFC 3261.

Il a été observé par la suite que la fiabilité était importante dans plusieurs cas, y compris celui des scénarios d'interopérabilité avec le RTC. Il était donc nécessaire de disposer d'une capacité facultative à prendre en charge la transmission fiable des réponses provisoires. C'est le rôle de la requête PRACK. Alors que la méthode ACK acquitte une réponse finale à a requête INVITE, la méthode PRACK acquitte une réponse provisoire à la requête INVITE.

Si un UA reçoit une requête INVITE contenant le header "Require" dont la valeur est 100rel, il doit émettre toute réponse provisoire (e.g., 180 Ringing) de manière fiable. S'il ne le peut pas, il doit rejeter la demande en retournant une réponse "420 Bad Extension" contenant un header "Unsupported" dont la valeur est 100rel. L'émetteur de la méthode INVITE devra alors retransmettre cette méthode sans header "Require".

Si un UA reçoit une requête INVITE contenant le header "Supported" dont la valeur est 100rel, il peut émettre s'il le souhaite toute réponse provisoire de manière fiable.

Enfin, il ne doit pas émettre de réponse provisoire de façon fiable si la méthode INVITE reçue ne contient pas de header "Require" ou "Supported".

Supposons le cas d'un UA qui souhaite émettre une réponse provisoire de manière fiable (Figure 11).

La réponse provisoire (e.g., 180 Ringing) doit contenir un header "Require" dont la valeur est 100rel et un header "Rseq" qui correspond à un numéro de séquence fiable, dont la valeur est choisie initialement de manière aléatoire entre 1 et  $2^{31}-1$ . Cette valeur doit être incrémentée de 1 à chaque envoi d'une nouvelle réponse provisoire. Un temporisateur T1 dont la valeur par défaut est 500 ms est armé lors de l'envoi de la réponse 180 Ringing. Si le temporisateur expire avant qu'une requête PRACK soit reçue, la réponse 180 Ringing doit être retransmise. La réception d'une méthode PRACK indique que la réponse provisoire a bien été reçue, et annule toute retransmission de la réponse provisoire. La requête PRACK doit être acquittée par une réponse 200 OK qui aura pour effet d'annuler toute retransmission de la requête PRACK. En effet, l'émetteur de la requête PRACK arme un temporisateur T1 dès son envoi. Si aucune réponse n'est reçue avant l'expiration de ce temporisateur, la méthode PRACK doit être retransmise. Une méthode PRACK doit contenir un header "RACK" qui doit contenir les valeurs des headers "Cseq" et "Rseq" présents dans la réponse provisoire permettant de corréler la requête PRACK et la réponse provisoire qu'elle acquitte.

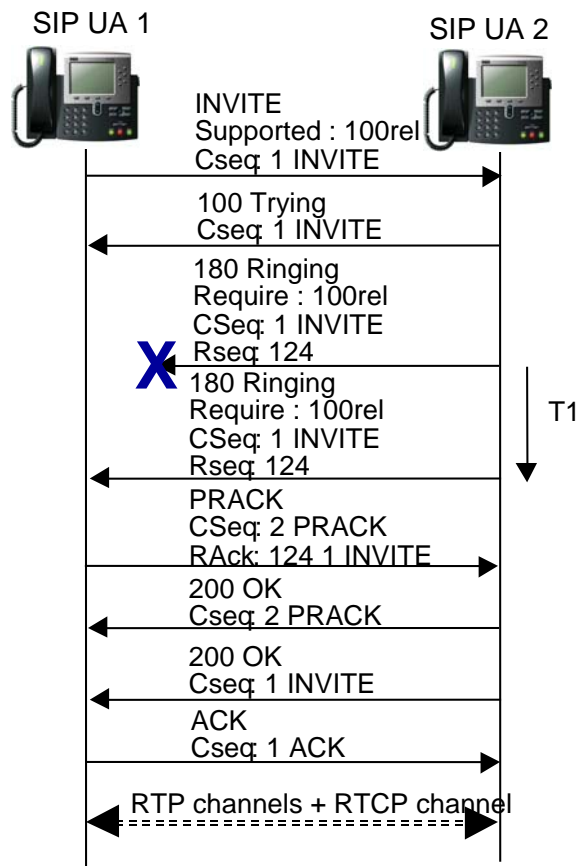


Figure 11 : Méthode PRACK

## 5.7 Méthode UPDATE

La méthode UPDATE permet à un user agent de mettre à jour les paramètres d'une session multimédia. (e.g., flux média et leurs codecs). Son rôle est similaire à celui de la méthode re-INVITE à la différence près que la méthode UPDATE peut être envoyée avant que la session soit établie. UPDATE est donc particulièrement utile lorsqu'il s'agit de mettre à jour des paramètres de session avant son établissement, e.g. mise en attente du destinataire.

Lorsqu'un UA émet une méthode INVITE, elle doit contenir un header "Allow" dont la valeur doit contenir "UPDATE" pour indiquer sa capacité à recevoir des méthodes UPDATE. Cela vaut aussi pour le récepteur lorsque ce dernier retourne une réponse provisoire fiable. Si la réponse provisoire n'est pas retournée de façon fiable, elle peut contenir le header "Allow" alors qu'une réponse finale 2XX doit le contenir.

Cette information permettra d'indiquer à l'émetteur que la destination a la capacité de supporter une procédure UPDATE.

Considérons l'exemple de la Figure 12 pour illustrer l'établissement d'un appel entre deux usagers IMS avec QoS :

UA1 émet une méthode INVITE (1) contenant une description SDP1 pour établir une session avec UA2. Cette description SDP indique des préconditions de QoS pour la session (e.g., session téléphonique). UA1 ne souhaite pas que UA2 soit alerté tant que les ressources n'ont pas été réservées dans les deux sens (sendrecv). UA2 accepte de réserver des ressources pour cette session avant d'alerter l'appelé. UA2 prend en charge la réservation des ressources dans le sens UA2?UA1 ; par ailleurs il est nécessaire que UA1 réserve des ressources dans le sens UA1?UA2. Afin de le lui indiquer, UA2 retourne une réponse provisoire 183 Session Progress (2) à UA1 lui demandant de commencer la réservation des ressources et de confirmer à UA2 dès que la session est prête dans le sens UA1?UA2. Cette réponse provisoire contient aussi une description SDP2. A la réception du 183 Session Progress, UA1 émet une méthode PRACK (3) acquittée par 200 OK (4).

UA1 et UA2 commencent la réservation de ressources (dans le cas de UE SIP, en créant des contextes PDP avec la QoS requise pour la session; cette création implique les entités UE, SGSN et GGSN). Même si UA2 a reçu une confirmation du réseau qui indique la réservation des ressources, l'appelé n'est pas alerté tant qu'une confirmation n'est pas envoyée par UA1. Lorsque UA1 a finalisé sa procédure de réservation, il émet un message UPDATE (5) à UA2 contenant une description SDP3 qui indique cette réservation dans le sens UA1?UA2. UA2 retourne une réponse 200 OK (6) de confirmation de la méthode UPDATE, indiquant que toutes les préconditions de Qos pour la session ont été remplies et que la QoS a bien été réservée dans les deux sens (La réponse 200 OK contient la description SDP4). A cet instant, UA2 alerte l'appelé. Une réponse provisoire 180 Ringing (7) est émise par l'UA2 à l'UA1. L'appelé décroche. UA2 retourne une réponse 200 OK (10) (réponse à la méthode INVITE) sans description SDP. UA1 émet une méthode ACK (11) afin de finaliser l'établissement d'appel. L'échange de média peut donc avoir lieu entre les deux participants à la session.

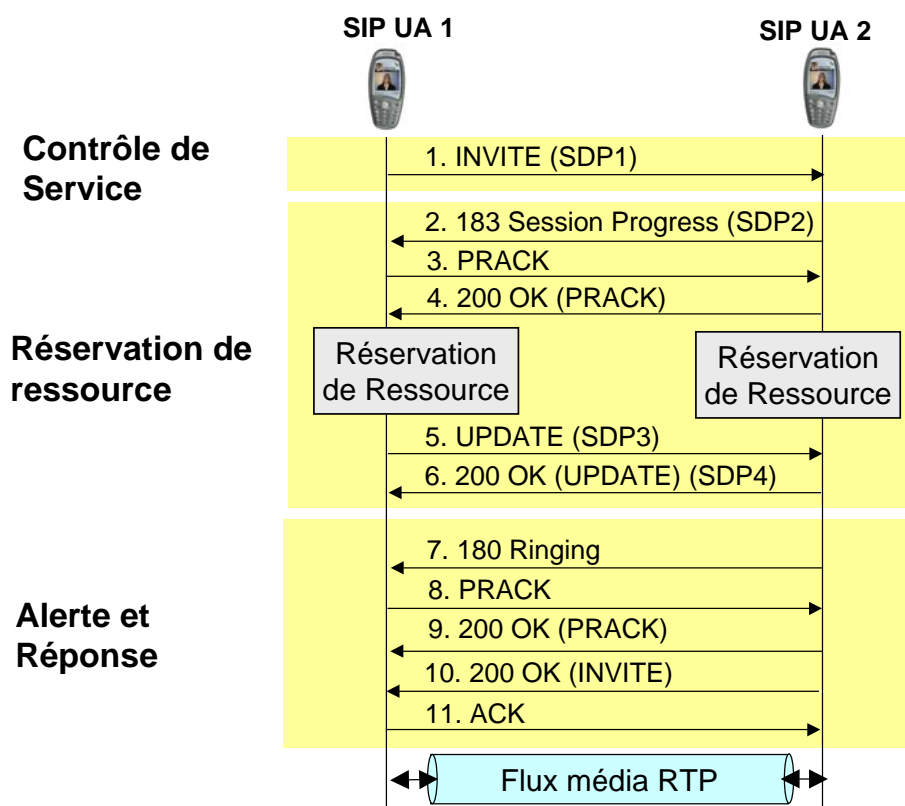


Figure 12 : Méthode UPDATE

```
SDP1
m=audio 20000 RTP/AVP 97
c=IN IP4 192.0.2.2
a=curr:qos local none
a=curr:qos remote none
a=des:qos mandatory local sendrecv
a=des:qos mandatory remote sendrecv
```

```
SDP2
m=audio 30000 RTP/AVP 97
c=IN IP4 192.0.2.4
a=curr:qos local none
```

a=curr:qos remote none  
a=des:qos mandatory local sendrecv  
a=des:qos mandatory remote sendrecv

#### SDP3

m=audio 20000 RTP/AVP 97  
c=IN IP4 192.0.2.2  
a=curr:qos local sendrecv  
a=curr:qos remote none  
a=des:qos mandatory local sendrecv  
a=des:qos mandatory remote sendrecv

#### SDP4

m=audio 30000 RTP/AVP 97  
c=IN IP4 192.0.2.4  
a=curr:qos local sendrecv  
a=curr:qos remote sendrecv  
a=des:qos mandatory local sendrecv  
a=des:qos mandatory remote sendrecv

## Références

- RFC 2778 A Model for Presence and Instant Messaging. M. Day, J. Rosenberg, H. Sugano. February 2000.
- RFC 2779 Instant Messaging/Presence Protocol Requirements. M. Day, S. Aggarwal, G. Mohr, J. Vincent. February 2000.
- RFC 2976 The SIP INFO Method. S. Donovan. October 2000.
- RFC 3261 SIP: Session Initiation Protocol. J. Rosenberg, H. Schulzrinne, G. Camarillo, A. Johnston, J. Peterson, R. Sparks, M. Handley, E. Schooler. June 2002.
- RFC 3262 Reliability of Provisional Responses in Session Initiation Protocol (SIP). J. Rosenberg, H. Schulzrinne. June 2002.
- RFC 3263 Session Initiation Protocol (SIP): Locating SIP Servers. J. Rosenberg, H. Schulzrinne. June 2002.
- RFC 3264 An Offer/Answer Model with Session Description Protocol (SDP). J. Rosenberg, H. Schulzrinne. June 2002.
- RFC 3265 Session Initiation Protocol (SIP)-Specific Event Notification. A. B. Roach. June 2002..
- RFC 3311 The Session Initiation Protocol (SIP) UPDATE Method. J. Rosenberg. October 2002.
- RFC 3515 The Session Initiation Protocol (SIP) Refer Method. R. Sparks. April 2003.
- RFC 3680 A Session Initiation Protocol (SIP) Event Package for Registrations. J. Rosenberg. March 2004.
- RFC 3842 A Message Summary and Message Waiting Indication Event Package for the Session Initiation Protocol (SIP). R. Mahy. August 2004.
- RFC 3856 A Presence Event Package for the Session Initiation Protocol (SIP). J. Rosenberg. August 2004.
- RFC 3903 Session Initiation Protocol (SIP) Extension for Event State Publication. A. Niemi, ed. October 2004.
- RFC 4566 SDP: Session Description Protocol. M. Handley, V. Jacobson, C. Perkins. July 2006.