TP Voix sur IP

Aurore Mathias, Arnaud Vasseur

7 septembre 2011

Table des matières

Chapitre 1

Comment fonctionne la voix sur IP ?

1.1 L'application voix

1.1.1 Contraintes

Les contraintes de la voix sur IP sont multiples :

- *Faible temps de latence* : la latence ne doit pas excéder 200ms. On considère qu'à 100ms, on dispose d'une bonne qualité de service.
- *Perte de paquets* : le protocole utilisé ne permet pas d'assurer la non perte de paquets.
- *Bande passante assurée* : sans compression, la bande passante consommée est énorme. La voix non compressée occupe 64Kbs par seconde, et il faut que le réseau puisse assurer son transit.
- *Gigue* : la voix sur IP transitant, comme son nom l'indique, via le protocole IP, les paquets IP sont susceptibles de parvenir à l'utilisateur final dans un ordre aléatoire. Les paquets IP transitent potentiellement par des routeurs différents - pour des raisons liées à la variation de charge du réseau, ils n'empruntent pas forcément tous la même route. Ce délai dans les arrivages de paquets peut entraîner des temps de latence suffisamment longs pour être gênants.

SIP pour limiter la latence. RTP pour limiter la gigue. RTCP pour la perte de paquets. L'ADSL est suffisamment fiable pour assurer un bon transit des paquets, sans trop de pertes.

1.1.2 Solutions sur IP

Sur un réseau IP, on distingue plusieurs protocoles permettant de résoudre ces contraintes. Ils possèdents tous en commun ces points :

- *Simplicité* : la simplicité de fabrication des paquets diminue la durée des traitements par les routeurs et les terminaux.
- *Compression* : tous les protocoles compressent les données, profitant de la puissance de calcul des derniers matériels, afin de réduire la bande passante et d'améliorer le débit.

Cependant, ces protocoles, qui passent sur IP, ne possèdent pas de mécanisme de contrôle de la qualité de service[1]. On distingue 2 principaux protocoles de voix sur IP :

- *H.323* : protocole normalisé et standardisé, conçu par l'ITU (International Telecommunication Union). Il dérive du protocole H.320, utilisé précedemment sur le réseau RNIS. Il repose sur RTP.
- *SIP* : protocole normalisé et standardisé, conçu par l'IETF (The Internet Engineering Task Force) en 2002.

1.2 Codecs

1.2.1 G.711

Le codage G.711 a été standardisé par l'ITU en 1988. Il s'agit de quantifier linéairement le signal sonore sur 8 bits, à intervalles de temps égaux à 1/8000s (soit 8000Hz, ou 1 quantification toutes les 125 microsecondes)[4]. Le signal résultant a un débit de 64kbs par secondes.

1.2.2 G.729

Le codage G.711 a été standardisé par l'ITU. Le signal sonore est compressé en utilisant une technique de prédiction linéaire. Le signal sortant a un débit de 8kbs par seconde. Chaque trame compressée représente 10ms de signal audio, soit 80bits par trame[5].

1.2.3 G.723

Le codage G.723.1 a été standardisé par l'ITU. Le signal sonore est compressé en utilisant une technique de prédiction linéaire. Le codage est utilisable à des débits de 5.3kbps et 6.3kbps. La taille d'une frame est de 30ms[6].

1.3 SIP

1.3.1 Qu'est-ce que SIP ?

SIP est un protocole de contrôle pour créer, modifier et terminer des sessions entre un ou plusieurs participants. Il opère sur la couche Application. Ces sessions peuvent être de natures différentes : telephone, multimedia, conférences. Le protocole SIP aide les utilisateurs à communiquer entre eux. Ses tâches sont les suivantes [2] :

- *Recherche* : SIP gère un annuaire d'utilisateurs. Il permet aux utilisateurs de s'enregistrer dans l'annuaire, et de rechercher des correspondants.
- *Accord* : SIP fourni le protocole de négociation des codecs entrant en jeu dans une correspondance.

1.3.2 Mise en place d'une session SIP

Pour mettre en place une session SIP, il faut ces différents éléments [2] :

- *Les utilisateurs (User Agents - UA)* : ils sont à la fois clients et serveurs.

- *Un ou plusieurs serveurs Proxy (Proxy Server)* : établissent la connexion entre plusieurs utilisateurs. Ils sont utiles pour établir le routage le plus efficace disponible entre les utilisateurs.
- *Des serveurs d'annuaire (Registrars)* : ils enregistrent les adresses des utilisateurs. Chaque Registrar gère un domaine particulier, et chaque utilisateur souhaitant utiliser le service s'enregistre auprès de ce Registrar.

1.3.3 Le protocole SIP

Les principales méthodes SIP et les réponses associées [2] :

- *INVITE* : sert à inviter un autre utilisateur. Usuellement, l'utilisateur souhaitant appeller envoie INVITE à un serveur Proxy, puis ce dernier transmettra jusqu'à/aux serveurs Proxy suivants, jusqu'à atteindre la cible. On distingue 3 réponses :
  - *100 = Trying* : lors de la première connexion à un proxy ; le proxy donnera cette réponse lorsqu'il transmettra la requête sans connaître d'autre information sur l'état final de la destination à atteindre.
  - *180 = Ringing* : l'utilisateur cible a reçu l'invitation, et son téléphone "sonne".
  - *200 = OK* : l'utilisateur cible a répondu à son téléphone.
- *ACK* : lorsque l'utilisateur cible décroche, il envoie un message "OK". L'appellant, qui reçoit ce message, répond pas un "ACK".
- *BYE* : le premier des correspondants qui met fin à la conversation envoie un message "BYE", qui est acquitté par le message "OK".

1.3.4 SIP-URI

Pour communiquer, un appellant doit être enregistré sur un Registrar. Il enregistre son URI (Unified Resource Identifier). Cet URI est formé de cette façon :

sip :alice@domain.com

Le nom de domaine est le Registrar gérant ce domaine.

1.3.5 Exemple de session SIP utilisant un proxy

Soient Alice et Bob. Alice décide d'appelle Bob. Voici ce que le téléphone IP d'Alice envoie vers son proxy, et ce que le téléphone IP de Bob répond.

1.3.6 Lien entre SIP et H.323

SIP et H.323 sont tous deux des protocoles de gestion de télécommunication, et de négociation de codecs de données. Ils sont concurrents.

1.4 RTP (Real Time Procol)

1.4.1 Qu'est-ce que RTP ?

RTP est un protocole de transport de messages en temps réel [1]. Les applications temps réel concernent les applications de transmission audio et video,

par exemple. RTP (couche session) est encapsulé dans un paquet UDP (couche transport).

1.4.2 Utilité

RTP est encapsulé par UDP. Il profite des fonctionnalités d'UDP : multicast, simplicité. RTP ajoute plusieurs fonctionnalités à UDP, ce qui le rend indispensable pour les applications de téléconférence et transmissions de flux multimedia :

- *Un numéro de séquence (Sequence Number)* : permet de définir un ordre aux paquets, afin de ne pas mélanger les données.
- *Un marquage temporel (Timestamp)* : donne une information sur l'instant où les données ont été prélevées (sampling).
- *Une source de synchronisation (SSRC)* : défini l'horloge de synchronisation.

TCP quant à lui ne peut pas prendre en charge le transport de flux multimedia, car il implémente le contrôle de validité de paquets, ce qui n'est pas compatible avec des applications temps réel.

1.4.3 RTCP et RTSP

RTCP sert à [1] :

- *Qualité de service* : permet de récupérer des données sur la qualité de service : congestion, encodages adaptatifs.
- *Garder des traces* : garde la trace de chaque participant, au cas où un problème arriverait dans la source emettrice.
- *Débit* : il permet l'adaptation du débit des données entre plusieurs sources.

RTSP utilise aussi RTP. Il permet de contrôler, à distance, l'émission de flux multimedia par un serveur de streaming sur demande (on-demand).

1.4.4 Description d'un dialogue RTP/RTCP

Après établissement d'une session SIP, les utilisateurs s'échangent des paquets RTP : c'est la communication. A intervalle régulier (quelques secondes), chaque utilisateur envoie un paquet RTCP aux autres utlisateurs, et en reçoit. Ces paquets sont de 2 types : Receiver Report, Sender Report. Ils servent à informer les sources et les récepteurs d'émissions sur la ligne de télécommunication : qualité de service, etc...

Chapitre 2

Analyse d'un trafic voix

2.1 Initiation à Wireshark

Fichier de capture utilisé : trace.pcap.
Wireshark possède plusieurs modes de présentation des données :

- *Statistics/Protocol Hierarchy* : permet d'afficher dans une vue hiérarchique, tel un arbre, les différents paquets capturés. On peut en déduire leur nombre et leur pourcentage, en fonction des protocoles qu'ils utilisent.
- *Telephony/RTP/Show all stream* : analyse les flux de paquets RTP et donne des statistiques : paquets perdus, gigue, latence, et moyennes de la gigue et de la latence. On peut analyser plus finement un flux en particulier : voir tous les paquets, leur numéro de séquence, leur latence et gigue, etc...
- *Telephony/VoIP Calls* : donne la liste des communications VoIP capturées. On peut connaître le début et la fin de la communication, l'adresse SIP de l'appellant et du correspondant, ainsi que le protocole utilisé. On peut ensuite utiliser la fonction "Player", qui décode et lit les paquets VoIP : on peut donc écouter le flux ! De plus, on peut utiliser la fonction "Graph", qui montre les échanges de paquets entre les 2 correspondants, en incluant les principaux paquets caractéristiques du protocole (par exemple, les paquets "Trying", "Ringing", "ACK").
- *Flow Graph* : permet de présenter graphiquement le trafic, en découpant en paquets, entre les différentes machines (adresses IP). On peut distinguer la direction des paquets, ainsi que les ports utilisés et le type de paquet.

Comment filtrer un trafic DHCP ou ping ?

Pour filtrer le trafic DHCP, dans la case "Filter", entrer : bootp.dhcp. Pour filtrer le trafic ICMP (ping) , dans la case "Filter", entrer : icmp.

Quelles sont les 2 méthodes pour filtrer un flux SIP, RTP ou RTCP ?

On peut entrer sip, rtp ou rtcp dans le champ "Filter" pour filtrer ses différents protocoles. De plus, on peut utiliser des caractéristiques propres aux protocoles pour filtrer les messages. Dans le cas de SIP, on peut filtrer tous les paquets UDP qui vont ou viennent sur le port 5060 : "udp.dstport==5060 or

udp.srcport==5060", ou bien "udp.port==5060". Concernant RTP et RTCP, on ne peut pas filtrer en utilisant le numéro de port, car il est différent à chaque fois.

Comment filtrer un flux SIP pour une adresse IP destination donnée (192.168.1.8) et un flux RTP pour un port donné (16318) ?

Protocole SIP et adresse distance connue : sip and ip.dst==192.168.1.8 Protocole RTP et port défini : rtp and udp.port==16318

Quel est le codec utilisé dans le fichier trace pour transporter la voix ?

On filtre les paquets RTP utilisés pour cette communication. Le champ PT (Payload Type) contient la référence du codec, ici G.711 (défini par l'ITU).

Comment pourrait-on mesurer la qualité de la voix de la trace ? Cette mesure est-elle empirique dans le cas de Wireshark ?

On analyse le trafic RTP : Telephony/RTP/Show all stream. On sélectionne un flux d'un utilisateur vers un autre. On peut estimer la qualité de la communication selon le nombre de paquets perdus, le temps de latence (Delta), la gigue (Jitter), et leurs moyennes. Pour la communication de 192.168.1.8 vers 192.168.1.5, on observe un temps de latence maximum de 115ms, ce qui est plus que moyen, ainsi qu'un taux de perte de plus de 1%. Wireshark analyse les paquets et déduit de manière empirique les mesures. Pour s'en convaincre, on sélectionne une communication avec fort taux de perte, et on "Analyze" : on se rend compte que, s'il manque des paquets, Wireshark déduit un temps de latence pour le paquet suivant.

2.2 Analyse d'une session SIP

2.2.1 Connexion au serveur SIP

Fichier de capture utilisé : sip1.pcap

Quelle est l'adresse de l'initiateur de la connexion ? Quelle est l'adresse du serveur ?

L'initiateur de la connexion est l'adresse 192.168.1.5. L'adresse du serveur est 192.168.1.8.

Quel est le protocole de transport utilisé par les messages SIP ? Pour quelle raison ce protocole est utilisé ?

Le protocole de transport utilisé par les messages SIP est UDP. Ce protocole est moins lourd que TCP, par conséquent il convient mieux pour réduire la latence.

Décrivez la structure d'une requête SIP et d'une réponse SIP en y indiquant les principaux champs

Une requête SIP contient des champs dans son en-tête. Certains sont obligatoires [2] :

- *To* : l'adresse SIP du destinataire.
- *From* : l'adresse SIP de l'appellant.
- *CSeq* : un numéro de séquence et une méthode. Il sert à ordonner les transactions.
- *Call-ID* : un identifiant unique correspondant à la série de messages qui va suivre.
- *Max-Forwards* : limite le nombre de sauts par lesquels une requête peut transiter jusqu'à sa destination.
- *Via* : indique le transport utilisé lors de la transaction (UDP), et le lieu où la réponse doit être envoyée (adresse IP + port).

Chaque requête possède une ligne de départ (Start Line).

- *Requête* : "Request-Line", qui contient : la méthode (REGISTER, INVITE, CANCEL, BYE, OPTIONS) ; "Request-URI" ; "SIP-Version".
- *Réponse* : "Status-Line", qui contient : "SIP-Version" ; "Status-Code" ; "Reason-Phrase", description textuelle du statut.

Détaillez les différents échanges SIP liés à la connexion à un serveur SIP

Connexion SIP client-serveur :

1. Le client envoie une requête REGISTER au serveur.
2. Le serveur renvoie une réponse TRYING au client, ce qui signifie que le serveur est en train d'essayer de traiter la requête. C'est une réponse provisoire.
3. Le serveur envoie sa réponse définitive, par exemple : OK, Unauthorized.

Quelle serait l'explication des 2 messages *unauthorized* ?

1. On reçoit "Unauthorized" quand on se connecte au serveur sans être authentifié. Il faut définir le champ "Authorization" dans l'en-tête du message.
2. Bug dans la conception du RFC : envoi simultané de 2 messages avec Register = 0 et 3600. Si celui de 0 arrive avant, ça bug : déconnexion.

2.2.2 Appel

Fichier de capture utilisé : sip2.pcap ("Tarantino Sound Track").

Quelles sont les adresses de l'appellant et de l'appelé ?

Appellant : Alice, adresse SIP : sip :6123@192.168.1.8 ; Appelé : 6456, adresse SIP : sip :6456@192.168.1.8.

Quels sont les ports associés aux différents protocoles du fichier de trace ?

- SIP : 5060 côté serveur.
- RTP : côté appellant, 63176 ; côté appellé, 14106.
- RTCP : côté appellant, 63177 ; côté appellé, 14107.

Décrivez les différentes étapes de l'initiation et la clôture d'un appel basé sur une signalisation SIP

Ouverture d'appel :

1. *INVITE* : le client envoie "INVITE" au proxy. Le proxy répond.
2. *OK* : le proxy répond "OK" au client lorsque le correspondant répond au téléphone.
3. *ACK* : le client envoie un "ACK", directement au correspondant.

Clôture d'appel :

1. *BYE* : le correspondant envoie "BYE" au client lorsqu'il raccroche.
2. *OK* : le client répond au correspondant avec "OK".

Quel est le rôle du message ACK ?

L'appellant signifie à l'appelé qu'il a reçu une réponse définitive.

Quels sont les principaux champs d'un message RTP ? Comment identifie-t'on le codec utilisé ?

- *Version* : la version.
- *Extension* : indique si un extension est présente.
- *CSRC count* : le nombre d'identifiants CSRC qui suivent l'en-tête fixe. Les utilisateurs qui ont contribué à intégrer des données dans le paquet.
- *Payload Type (1 octet)* : le type de contenu (image, audio, video, ...).
- *Sequence Number* : l'ordre d'émission des paquets.
- *Timestamp* : l'instant de l'échantillonnage du premier octet des données multimedia transmises.
- *SSRC* : identifie la source de synchronisation.

Le Payload Type détermine le type de codage employé. Ici, on remarque que la valeur 0 donne le codage "ITU-T G.711 PCMU".

Quels sont les champs que l'on retrouve dans les messages RTCP Receiver Report et Sender Report ?

Les paquets Sender Report contiennent des informations statistiques sur les participants émetteurs actifs. Les paquets Receiver Report contiennent des informations sur les participants non émetteurs[7]. Voici les champs partagés :

- *V - Version*
- *P - Padding* : indique si les données contiennent des données de padding (bourrage), et si oui, leur taille.
- *RC - Reception report Count* : indique le nombre de rapports de réception contenus dans le paquet.

- *PT - Packet Type* : indique le type du paquet. Pour un paquet de type Sender Report, PT=200 (Receiver Report, PT=201).
- *Length - Longueur* : Indique la longueur totale du paquet.
- *SSRC of sender* : indentifie la source émettrice.
- *SSRC_X - X désignant chaque source* : donne les infos suivantes sur chaque source :
  - *Fraction Lost* : indique la fraction de paquets perdus en provenance de cette source.
  - *Cumulative number of packets lost* : indique le nombre de paquets perdus en provenance de cette source.
  - *Last SR* : indique le timestamp enregistré dans le dernier paquet SR reçu de cette source.
  - *Delay since last SR (LSR)* : indique l'intervalle de temps entre la réception du dernier paquet SR en provenance de cette source, et l'émission de ce paquet RR.

Calculez pour le délai, la gigue et le débit, le min, le max, la moyenne et l'écart type associés. Que pouvez-vous en conclure ?

Il y a 2 flux :
De (192.168.1.) 5 vers (...) 8 (appellant vers appelé) :

	Délai (ms)	Gigue (ms)	Bande passante (kbps)
min	0	0	1.6
max	30.01	3.1	84.8
moyenne	20.00	0.51	80.30
écart-type	0.75	0.13	4.62

De (...) 8 vers (...) 5 (appelé vers appellant) :

	Délai (ms)	Gigue (ms)	Bande passante (kbps)
min	0	0	1.6
max	130.85	23.07	89.6
moyenne	20.22	3.27	79.43
écart-type	9.13	4.47	4.82

Que peut-on dire de la qualité de l'appel ?

On remarque que les paquets ont des caractéristiques de propagation (délai, gigue, ...) à peu près égales, peu importe le sens de transmission. Le délai moyen est d'environ 20ms, avec un écart type max de 9.13, ce qui est un très bon score. La gigue est faible. La bande passante est supérieure à 64kbps, en considérant la moyenne et l'écart type, ce qui est suffisant pour faire passer sans problème des trames VoIP non compressées. On peut donc dire que la communication est de bonne qualité.

2.2.3 Déconnexion du serveur

Fichier de capture utilisé : sip3.pcap.

Décrivez les échanges liés à la clôture d'une connexion au serveur SIP

Un utilisateur, pour se déconnecter d'un serveur SIP, a tout d'abord besoin de récupérer un élément du codage nécessaire à son authentification (un nonce). Pour celà, il envoie une requête REGISTER avec un champ 'expires' égal à 0, qui aboutit sur un "401 Unauthorized", ce dernier contenant le nouveau nonce à utiliser. Le nonce connu, l'utilisateur renvoie la même requête REGISTER, toujours avec la champ 'expires' à 0, mais cette fois, le serveur pourra l'authentifier. Le serveur répond donc avec un message "200 OK".

A quoi correspond un "binding" ?

Un "binding" est une liaison de flux de données entre un utilisateur et un serveur SIP.

2.3 Analyse de traces VoIP

2.3.1 voip1.pcap

Nombre de sessions SIP

Il y a 7 sessions SIP :

1. *158.863-158.872* : appel rejeté.
   - *Erreurs* : Bob, sip :6456@192.168.1.8, de la machine 192.168.1.6, appelle sip :1256@192.168.1.8. L'appel est rejeté car le proxy (192.168.1.8) ne connaît pas le correspondant sip :1256@192.168.1.8.
   - *Adresses des utilisateurs* :
     - *Appellant* : Bob, sip :6456@192.168.1.8, de la machine 192.168.1.6.
     - *Appelé* : sip :1256@192.168.1.8.
   - *Ports utilisés* :

	Appellant		Appelé	
	Port entrant	Port sortant	Port entrant	Port sortant
SIP	6836	6836	5060	5060
RTP	NA	NA	NA	NA
RTCP	NA	NA	NA	NA

   - *Statistiques* : non applicable.
   - *Qualité de la voix* : non applicable.
2. *177.092-177.092* : appel rejeté.
   - *Erreurs* : Bob, sip :6456@192.168.1.8, de la machine 192.168.1.6, appelle sip :000000000000@192.168.1.8. L'appel est rejeté car le proxy (192.168.1.8) ne connaît pas le correspondant sip :000000000000@192.168.1.8.
   - *Adresses des utilisateurs* :
     - *Appellant* : Bob, sip :6456@192.168.1.8, de la machine 192.168.1.6.
     - *Appelé* : sip :000000000000@192.168.1.8.
   - *Ports utilisés* :

	Appellant		Appelé	
	Port entrant	Port sortant	Port entrant	Port sortant
SIP	6836	6836	5060	5060
RTP	NA	NA	NA	NA
RTCP	NA	NA	NA	NA

- *Statistiques* : non applicable.
- *Qualité de la voix* : non applicable.

3. *208.383-208.935* : appel rejeté.
   - *Erreurs* : Bob, sip :6456@192.168.1.8, de la machine 192.168.1.6, appelle sip :0169159658@192.168.1.8. L'appel est rejeté car le proxy (192.168.1.8) ne connaît pas le correspondant sip :0169159658@192.168.1.8.
   - *Adresses des utilisateurs* :
     - *Appellant* : Bob, sip :6456@192.168.1.8, de la machine 192.168.1.6.
     - *Appelé* : sip :0169159658@192.168.1.8.
   - *Ports utilisés* :

	Appellant		Appelé	
	Port entrant	Port sortant	Port entrant	Port sortant
SIP	6836	6836	5060	5060
RTP	NA	NA	NA	NA
RTCP	NA	NA	NA	NA

   - *Statistiques* : non applicable.
   - *Qualité de la voix* : non applicable.

4. *222.881-243.935* : appel reçu et terminé.
   - *Erreurs* : aucune erreur.
   - *Adresses des utilisateurs* :
     - *Appellant* : Bob, sip :6456@192.168.1.8, de la machine 192.168.1.6.
     - *Appelé* : sip :6123@192.168.1.8.
   - *Ports utilisés* :

	Appellant		Appelé	
	Port entrant	Port sortant	Port entrant	Port sortant
SIP	6836	6836	5060	5060
RTP	18850	18850	16794	16794
RTCP	18851	18851	16795	16795

   - *Statistiques* :

	Délai (ms)	Gigue (ms)	Bande passante (kbps)
min	0	0	1.6
max	20.85	0.53	81.6
moyenne	19.97	0.46	77.94
écart-type	0.89	0.05	11.93

Sens Appellant (6) vers Appelé (8)

	Délai (ms)	Gigue (ms)	Bande passante (kbps)
min	0	0	1.6
max	26.39	1.29	81.6
moyenne	19.96	0.51	77.25
écart-type	1.07	0.11	13.27

Sens Appelé (8) vers Appellant (6)

   - *Qualité de la voix* : .

5. *254.200-273.834* : appel reçu et terminé.
   - *Erreurs* : aucune erreur.
   - *Adresses des utilisateurs* :
     - *Appellant* : Bob, sip :6456@192.168.1.8, de la machine 192.168.1.6.
     - *Appelé* : sip :6123@192.168.1.8.
   - *Ports utilisés* :

	Appellant		Appelé	
	Port entrant	Port sortant	Port entrant	Port sortant
SIP	6836	6836	5060	5060
RTP	57794	57794	19304	19304
RTCP	57795	57795	19305	19305

- *Statistiques* :

	Délai (ms)	Gigue (ms)	Bande passante (kbps)
min	0	0	1.6
max	20.99	1.09	81.6
moyenne	19.95	0.4812	78.35
écart-type	0.988	0.0725	11.25

Sens Appellant (6) vers Appelé (8)

	Délai (ms)	Gigue (ms)	Bande passante (kbps)
min	0	0	1.6
max	22.46	0.7	81.6
moyenne	19.96	0.4572	77.25
écart-type	0.9719	0.06497	13.27

Sens Appelé (8) vers Appellant (6)

- *Qualité de la voix* : .

6. *281.082-289.670* : appel reçu et terminé.
   - *Erreurs* : aucune erreur.
   - *Adresses des utilisateurs* :
     - *Appellant* : Bob, sip :6456@192.168.1.8, de la machine 192.168.1.6.
     - *Appelé* : sip :6123@192.168.1.8.
   - *Ports utilisés* :

	Appellant		Appelé	
	Port entrant	Port sortant	Port entrant	Port sortant
SIP	6836	6836	5060	5060
RTP	33004	33004	16590	16590
RTCP	?	33005	33005	?

   - *Statistiques* :

	Délai (ms)	Gigue (ms)	Bande passante (kbps)
min	0	0	1.6
max	30.21	2.46	81.6
moyenne	19.90	1.174	70.98
écart-type	3.164	0.6239	21.71

Sens Appellant (6) vers Appelé (8)

	Délai (ms)	Gigue (ms)	Bande passante (kbps)
min	0	0	1.6
max	20.53	0.49	81.6
moyenne	19.90	0.4498	71.18
écart-type	1.473	0.09004	20.62

Sens Appelé (8) vers Appellant (6)

   - *Qualité de la voix* : .

7. *332.287-337.631* : appel rejeté.
   - *Erreurs* : Alice, sip :6123@192.168.1.8, appelle Bob, sip :6456@192.168.1.8, sur la machine 192.168.1.6, port 6836. L'appel est rejeté par Bob, qui signale qu'il est temporairement injoignable (480 Temporarily Unavailable).

- *Adresses des utilisateurs* :
  - *Appellant* : Alice, sip :6123@192.168.1.8.
  - *Appelé* : Bob, sip :6456@192.168.1.8, sur la machine 192.168.1.6, port 6836.
- *Ports utilisés* :

	Appellant		Appelé	
	Port entrant	Port sortant	Port entrant	Port sortant
SIP	5060	5060	6836	6836
RTP	NA	NA	NA	NA
RTCP	NA	NA	NA	NA

- *Statistiques* : non applicable.
- *Qualité de la voix* : non applicable.

Bibliographie

[1] IETF, *RTP : A Transport Protocol for Real-Time Applications (RFC 3550)*, The Internet Society, July 2003

[2] IETF, *SIP : Session Initiation Protocol (RFC 3261)*, The Internet Society, June 2002

[3] IETF, *SIP-Specific Event Notification (RFC 3265)*, The Internet Society, June 2002

[4] VoipThink, *http ://www.en.voipforo.com/codec/codecs-g711-alaw.php*

[5] Data-compression.com, *Speech Compression*

[6] Union Internationale des Télécommunications, *G.723.1* http ://www.itu.int/rec/T-REC-G.723.1-200605-I/fr

[7] Etudes et FORmations en Télécommunications, *RTP_EFORT.pdf* http ://efort.com/